POSITIVE RESEARCH

rus

positive technologies

TECHNICAL GUIDE | № 2 / 2023

NEW

POSITIVE RESEARCH

rus

ЖУРНАЛ О РЕЗУЛЬТАТИВНОЙ
КИБЕРБЕЗОПАСНОСТИ
ОТ POSITIVE TECHNOLOGIES

По следам PHD 12

Сентябрь 2023

СЛОВО РЕДАКТОРА

Всем привет!

Перед вами — первый номер обновленного Positive Research! Мы уже 10 лет выпускаем журнал для ИБ-экспертов, хакеров, бизнеса и всех, кто интересуется кибербезом. В этом году мы провели ребрендинг проекта: обновили печатную версию и запустили сайт, чтобы контент всегда был у вас под рукой.

Теперь Positive Research — это открытая площадка для обмена опытом. На ней сотрудники Positive Technologies, наши партнеры, клиенты и независимые эксперты разбирают реализованные кейсы, отвечают на сложные (и даже неудобные) вопросы и обсуждают ИТ- и ИБ-тренды.

Мы разделили все выпуски журнала на два больших блока — технический и бизнесовый. В первом — фокусируемся на технологиях, во-втором — обсуждаем бизнес-аспекты их внедрения.

Анастасия Дискина

Главный редактор Positive Research

Positive Research всегда открыт для сотрудничества!
Если у вас есть интересные идеи, пишите на journal@ptsecurity.com

СОДЕРЖАНИЕ

РЕДАКЦИЯ ЖУРНАЛА

ГЛАВНЫЙ РЕДАКТОР: АНАСТАСИЯ ДИСКИНА
РЕДАКТОР: ДМИТРИЙ АЛФУЦКИЙ
АРТ-ДИРЕКТОР: ВИКТОРИЯ ТАКТАШЕВА

ИЛЛЮСТРАЦИИ В НОМЕРЕ:
СТР. 14—23 — ОЛЬГА ТЕРЕХОВА
СТР. 80—87 — КАМИЛА ЛИГАЙ
СТР. 154—159 — АНДРЕЙ ГЛАЗКОВ

ФОТОГРАФ: РАФАЭЛЬ ЮСИПОВ

АДРЕС РЕДАКЦИИ: Г. МОСКВА, 105187, ПРЕОБРАЖЕНСКАЯ ПЛ., Д. 8
БИЗНЕС-ЦЕНТР «ПРЕО 8»

ДАТА ВЫХОДА В СВЕТ: 20.09.2023

ИЗДАТЕЛЬ: POSITIVE TECHNOLOGIES

РАСПРОСТРАНЯЕТСЯ БЕСПЛАТНО

СОТРУДНИЧЕСТВО: JOURNAL@PTSECURITY.COM

АВТОРЫ

ГЕОРГИЙ АЛЕКСАНДРИЯ, СТАНИСЛАВ АНТОНОВ, АЛЕКСАНДР БАРИНОВ, ДЕНИС БАТРАНКОВ, СЕРГЕЙ ГОРЛЕНКО, КОНСТАНТИН ГРИЩЕНКО, ИГОРЬ КАБАНОВ, ДЕНИС КОРАБЛЕВ, ПАВЕЛ КОРОСТЕЛЕВ, АНТОН КУТЕПОВ, ВЛАДИМИР КОЧЕТКОВ, АЛЕКСЕЙ ЛЕДНЕВ, РУСЛАН ЛОЖКИН, АЛЕКСЕЙ ЛУКАЦКИЙ, МИХАИЛ МАКСИМОВ, АНТОН НЕЧИПОРЕНКО, КОНСТАНТИН ПОЛИШИН, АЛЕКСАНДР РЕПИН, АНДРЕЙ СИКОРСКИЙ, ДМИТРИЙ СКЛЯРОВ, ДМИТРИЙ СТУРОВ, ДМИТРИЙ ФЕДОСОВ, ЮЛИЯ ФОМИНА, ИВАН ЧЕРНОВ, ЮРИЙ ШАБАЛИН, КИРИЛЛ ШИПУЛИН, ИГОРЬ ШМАКОВ

ТРЕНДЫ I И II КВАРТАЛА 2023 Г.

ТОП-5 РЕЗОНАНСНЫХ АТАК

I квартал 2023 г.

Royal Mail

Включи VPN

Окленд

ION Group

Dish Network

Ross Memorial

ВРЕДОНОСНОЕ ПО

- Похищенные данные размещаются на сайтах с адресом, похожим на домен жертвы, чтобы об утечке узнали клиенты и партнеры компании.
- Жертв убеждают раскрывать детали киберстрахования, чтобы скорректировать требования к выкупу и гарантированно получить средства от страховщика.
- Самым популярным семейством вредоносного ПО стали инфостилеры RedLine.

ФИШИНГ С ИСПОЛЬЗОВАНИЕМ ОБЛАКОВ

- В 10 раз за последние два года выросло количество фишинговых страниц, размещенных на облачных платформах.
- Dropbox и OneDrive активно используются для распространения вредоносов.

КРИПТОВАЛЮТНОЕ МОШЕННИЧЕСТВО

- Злоумышленники создают сайты с несуществующими криптовалютами и рекламируют их среди инвесторов.
- В магазинах появляются поддельные приложения для инвесторов.
- Во II квартале блокчейн-проекты становились жертвами атак в два раза чаще, чем в начале года.

OneDrive

Dropbox

СОЦИАЛЬНАЯ ИНЖЕНЕРИЯ

- Атакующие все чаще используют QR-коды, чтобы обойти средства защиты.
- Растет число писем на темы трудоустройства, увольнения, получения льгот и пособий.

ЗЛОУМЫШЛЕННИКИ ОТКАЗЫВАЮТСЯ ОТ ШИФРОВАНИЯ И ВСЕ ЧАЩЕ УГРОЖАЮТ ЖЕРТВАМ ПУБЛИКАЦИЕЙ УКРАДЕННЫХ ДАННЫХ

Discord

РОСТ СЛУЧАЕВ SEO POISONING

- Выросло количество вредоносной рекламы в поисковых системах.
- ВПО распространяется через Discord и Dropbox: злоумышленники маскируют инфостилеры RedLine и Raccoon под графическое ПО (Blender), программы для записи видео (OBS) и др.
- В поисковой выдаче стало больше сайтов с фишинговыми формами входа.

Полные версии отчетов PT за I и II кварталы 2023 г. ищите здесь

ТОП-5 РЕЗОНАНСНЫХ АТАК

II квартал 2023 г.

Bitmarck

Mountain View

Microsoft

TSMC

«Инфотел»

РЫНОК ГОВОРИТ

ЭКСПЕРТЫ КРУПНЫХ РОССИЙСКИХ КОМПАНИЙ ОТВЕЧАЮТ НА ДВА ВОПРОСА:

Руслан Ложкин

Руководитель службы информационной безопасности
АКБ «Абсолют Банк»

1. КАКИЕ УГРОЗЫ КАЖУТСЯ ВАМ НАИБОЛЕЕ АКТУАЛЬНЫМИ ДЛЯ РОССИЙСКОГО РЫНКА В ПЕРВОМ ПОЛУГОДИИ 2023 Г.?

Во-первых, DDoS. Основной заказчик — IT ARMY of Ukraine. Эти ребята публикуют в открытом доступе список доменов организаций с датами будущих атак и отрабатывают его. По какому принципу они составляют свой план, известно только им. В распоряжении IT ARMY of Ukraine имеется большой пул мощностей как за границей, так и в России. Нужно отметить, что злоумышленники стали чаще использовать DDoS L7, который схож с легитимными запросами и требует более детального анализа. В целом такие кейсы решаются довольно быстро (при условии подготовленности персонала), но могут возникать разного рода побочные эффекты, которые лучше отработать на киберучениях. К таким побочным эффектам можно отнести, к примеру, невозможность корректной удаленной работы в условиях фильтрации трафика, а также потерю связи с глобальным DNS при включении геофильтрации.

Во-вторых, фишинг, в том числе с вредоносными вложениями. Чаще всего это тот же заказчик. Сейчас распространяется масса писем с разными угрозами, в которых злоумышленники просят выкуп за отказ от реализации атаки. Обычно письма рассылаются массово, некоторые содержат вложения, вредоносное ПО и реквизиты для перевода средств. Здесь я бы предложил создать чат российских CISO для оперативной отработки таких кейсов.

В-третьих, социальная инженерия. Причем дело даже не в самой «социалке», а в степени обработки жертв. Участились кейсы с поколением 65+, когда в офис банка для перевода средств привозят человека, находящегося под сильным влиянием злоумышленников. Жертва не отрывается от телефона и доверяет удаленному собеседнику больше, чем сотрудникам банка и службы безопасности. В некоторых случаях приходится вызывать полицию, чтобы человека просто забрали, потому что иначе остановить злоумышленников просто не получается.

2. КАКИЕ СОБЫТИЯ ЯВЛЯЮТСЯ ДЛЯ ВАШЕЙ КОМПАНИИ НЕДОПУСТИМЫМИ И ПОЧЕМУ?

Я думаю, в этом плане «Абсолют Банк» не отличается от других финансово-кредитных организаций. На ситуацию нужно смотреть несколько шире. В начале XX века недопустимым событием для банка была кража средств из хранилища, 10 лет назад — с корсчета, а сейчас недопустимые события сильно завязаны на степень цифровизации компании. Процесс завязан на зависимость: от данных, технологий и людей, которые являются субъектами данных либо участниками технологий. Если все это является активами компании, ответ очевиден. Если злоумышленник повредит данные или процесс доступа к ним либо будет оказывать негативное воздействие на людей, которые могут повлиять на активы компании, это и будет недопустимым событием.

В случае неполной цифровизации можно рассматривать в качестве недопустимых событий нарушение работоспособности отдельных критически важных систем (например, АБС, КБР, ДБО) или кражу данных из них. Но когда цифровая трансформация завершена, люди, данные и технологии становятся единым целым и их нужно рассматривать в совокупности. Информационная безопасность становится неотъемлемой составляющей цифровизации, поэтому ее отсутствие можно считать недопустимым событием.

РЫНОК ГОВОРИТ

ЭКСПЕРТЫ КРУПНЫХ РОССИЙСКИХ КОМПАНИЙ ОТВЕЧАЮТ НА ДВА ВОПРОСА:

Юрий Шабалин

Генеральный директор «Стингрей Технолоджиз» (ГК Swordfish Security)

1. КАКИЕ УГРОЗЫ КАЖУТСЯ ВАМ НАИБОЛЕЕ АКТУАЛЬНЫМИ ДЛЯ РОССИЙСКОГО РЫНКА В ПЕРВОМ ПОЛУГОДИИ 2023 Г.?

Как мы и прогнозировали [1] осенью 2022 г., количество кибератак на российские компании значительно возросло. Согласно данным «РТК-Солар», во II квартале 2023 г. было выявлено 325 000 инцидентов в области ИБ. Это на 12% больше, чем в I квартале, и на 38% превышает показатель аналогичного периода прошлого года.

Мы заметили увеличение числа кейсов с применением шифровальщиков, причем не столько взломов, сколько уничтожения критически важной информации. В целом атаки хорошо организованы, в них участвует множество группировок и индивидуальных хакеров. Кроме этого, злоумышленники используют технологии искусственного интеллекта, машинного обучения и средства автоматизации атак.

Также мы столкнулись со впечатляющим ростом числа атак на цепочку поставок ПО через компоненты с открытым исходным кодом (не менее 100% в первом полугодии 2023 г. по сравнению с первым полугодием 2022 г.). Мы ожидали подобного в свете проблем с железом и переходом компаний в облака, но тренд, похоже, оказался мощнее, чем мы прогнозировали. Во втором полугодии он только усилится.

Особым интересом у злоумышленников, по нашим данным, пользуются веб-приложения компаний. Причем эксплуатируются известные уязвимости. Кроме того, потенциальную угрозу как для компаний, так и для частных лиц несут мессенджеры и бездумная установка любого рода приложений на смартфоны из надежных и не очень источников. Некоторые сервисы умеют в фоновом режиме осуществлять запросы на доступ к другим приложениям и похищать оттуда информацию. Злоумышленников интересует банковский и криптовалютный софт, а для доставки вредоносного кода используются игры. Это не новая информация, но, к сожалению, о ней крайне редко информируют россиян.

[1]

2. КАКИЕ СОБЫТИЯ ЯВЛЯЮТСЯ ДЛЯ ВАШЕЙ КОМПАНИИ НЕДОПУСТИМЫМИ И ПОЧЕМУ?

Для нас в первую очередь критичны те события, которые могут ударить по бизнесу клиента и, как следствие, по нашей репутации.

Наш бизнес делится на два направления: консалтинг и решения, обеспечивающие анализ защищенности цифровых сервисов и программных продуктов. Для консалтинга недопустимо, если злоумышленники взломают непосредственно нас и получат данные клиентов. К примеру, исходный код или информацию для удаленного доступа, а также результаты анализов защищенности с перечислением набора найденных уязвимостей, которые хакеры смогут эксплуатировать.

Для решений по анализу защищенности недопустимо, если осуществится утечка исходников (например, алгоритмов по анализу, данных фидов, результатов аудитов). А также, поскольку мы предоставляем сервисы в облаке, утечка данных об уязвимостях, найденных у наших клиентов.

В «Темной комнате» мы рассказываем о том, о чем обычно не принято говорить. Герои рубрики — технари, представители бизнеса и бэк-подразделений российских компаний. Мы обсуждаем подробности реальных взломов, озвучиваем непопулярные мнения о популярных решениях и делимся инсайтами. От первого лица и только анонимно.

Все совпадения с реальными лицами, компаниями, продуктами и событиями ~~случайны~~ не случайны.

«ВОЗМОЖНО, ВЫМОГАТЕЛИ ВСЕ ЕЩЕ КОНТРОЛИРУЮТ СКОМПРОМЕТИРОВАННУЮ ИНФРАСТРУКТУРУ»

О чем

История одного шифровальщика в инфраструктуре компании X. Злоумышленник потребовал выкуп за возврат доступа к зашифрованным данным

Герой материала

Руководитель ИТ-подразделения

Масштаб компании

Около 120 сотрудников

— С чего все началось?

— Я устроился в компанию в августе прошлого года. У нас достаточно большой серверный парк (порядка 200 виртуальных машин), но инфраструктура была в ужасном состоянии, а вопросами кибербезопасности никто всерьез не занимался. Ни систем ИБ-мониторинга, ни антивирусов... Зато была масса сомнительных VPN-соединений, за которые отвечали внешние подрядчики. При этом мы были подключены к сети и оборудованию организации-арендодателя, в здании которой снимали офисные помещения. Ни о каком защищенном контуре речи не шло.

Проблем было много, но не хватало рук, чтобы во всем разобраться. Предыдущая ИТ-команда ушла — у меня осталось всего два инженера поддержки, у которых отсутствовало понимание базовых вещей: нельзя постоянно работать под учетной записью доменного администратора, давать пользователям права локальных админов и др. Приведу пример. Сотрудник удаленно работает через RDP: когда сессия сбрасывается, он просто инициирует новое подключение и продолжает выполнять свои задачи. То тут, то там мелькали синие экраны — всё в лучших традициях.

В конце 2022 г. у меня появился достаточно сильный системный администратор, и мы начали разгребать завалы. Например, виртуализовали файловый сервер и настроили синхронизацию на базе Microsoft DFS (это наше «чеховское ружье»). Тем не менее закрыть все дыры мы не успели.

— Как вы поняли, что злоумышленники проникли в инфраструктуру компании?

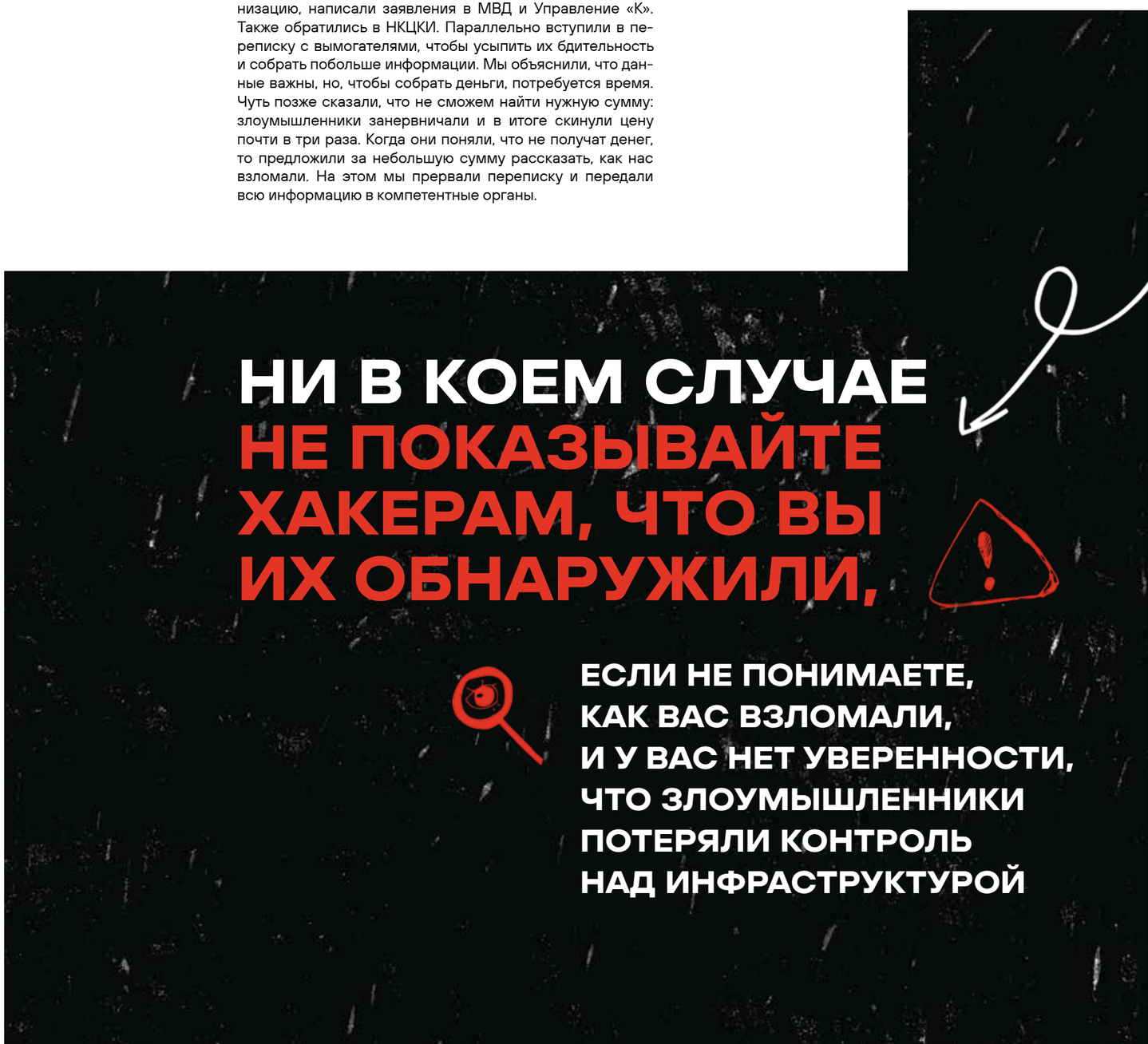
— В начале 2023-го я защищал перед руководством планы на год. Выступил с очередным докладом, рассказал о сомнительной активности и других проблемах в сети. Один из моих коллег предложил: «Давайте попробуем отключить сетевой интерфейс для машин, которые „шумят“ в сторону интернета». Попробовали — как раз перед 8 Марта. Вернулись на работу после праздников и увидели, что инфраструктура лежит :)

Из строя было выведено около 60 рабочих станций (примерно половина от общего числа) и 20 Windows-серверов. Пострадали только те устройства, которые оставались включенными на праздниках. Зачастую срочные задачи прилетают в нерабочее время, поэтому некоторые сотрудники не выключали свои машины, чтобы подключаться к ним дистанционно. При этом удаленный доступ был реализован через обычный проброс портов — без VPN. Мы знали об этом и начали переходить на более безопасную схему, но не успели. В результате злоумышленники смогли зашифровать данные на станциях, которые использовали незащищенное подключение. Цель предельно проста: получить выкуп — 6 тыс. долларов.

— Какой была ваша первая реакция? Были мысли заплатить вымогателям?

— Мы понимали, что идти на поводу у них нельзя. Некоторые коллеги предлагали заплатить, но всерьез этот вариант даже не рассматривался. Отмечу, что ранее компания уже неоднократно сталкивалась с подобными инцидентами, но не такими масштабными и сложными. Например, злоумышленники несколько раз шифровали отдельные рабочие станции, а в марте 2022 г. вывели из строя почтовый сервер Microsoft Exchange.

Мы сразу сообщили об инциденте в головную организацию, написали заявления в МВД и Управление «К». Также обратились в НКЦКИ. Параллельно вступили в переписку с вымогателями, чтобы усыпить их бдительность и собрать побольше информации. Мы объяснили, что данные важны, но, чтобы собрать деньги, потребуется время. Чуть позже сказали, что не сможем найти нужную сумму: злоумышленники занервничали и в итоге скинули цену почти в три раза. Когда они поняли, что не получат денег, то предложили за небольшую сумму рассказать, как нас взломали. На этом мы прервали переписку и передали всю информацию в компетентные органы.



— Как именно вас взломали?

— Расследовать инцидент нам помогали эксперты Positve Technologies. Вымогатели атаковали включенные рабочие станции, но среди них осталось несколько живых машин, на которых мы обнаружили декриптор, кейлоггер и другие вредоносные инструменты.

Партнеры запросили информацию о нашей инфраструктуре и сети, проанализировали данные с рабочих станций и за несколько дней подготовили отчет. Он состоял из двух частей: общее описание инцидента и более подробный анализ действий хакера.

Нас взломали через устаревший механизм удаленного доступа к рабочему столу по RDP (проброс портов на центральном маршрутизаторе). Схема была достаточно простой:

- Злоумышленники инициировали кратковременный сбой на рабочей станции.
- Пользователь обращался в техническую поддержку.
- Специалист техподдержки с правами администратора домена подключался к машине.
- Хакеры перехватывали его данные, получали возможность управлять всеми рабочими станциями и запускать механизмы шифрования.
- Шифрование было реализовано на уровне логических загрузочных записей, а не отдельных файлов. Это проще и быстрее, а восстановить данные в этом случае сложнее.

Самое интересное, что все закладки были заложены очень давно: хакеры проникли в сеть еще полтора (!) года назад — осенью 2021-го.

— Как вы устраняли последствия инцидента?

— Мы начали поднимать бэкапы и понемногу восстанавливать данные, чтобы обеспечить штатную работу сотрудников. К счастью, во время атаки пострадали не все участки сети: мы смогли оперативно изолировать финансовые и кадровые подразделения в отдельный сегмент. Затем попытались поднять новый домен в скомпрометированной сети и перевести людей туда. В ответ злоумышленники написали, что мы зря стараемся, и прислали список пользователей нового домена. Тогда мы перешли к более радикальным мерам:

- Децентрализовали и вывели из домена рабочие станции.
- Перенесли почтовый сервер на отдельный хостинг за пределами компании.
- Защитили файловые ресурсы с помощью Nextcloud (с дополнительной авторизацией).

› Установили антивирусное ПО на Windows-серверы и рабочие станции.

Проще говоря, мы децентрализовали инфраструктуру, защитили отдельные участки и начали изолироваться от сети арендодателя. Мы не уверены, что злоумышленники больше не контролируют скомпрометированную инфраструктуру бизнес-центра.

— Сколько данных удалось зашифровать вымогателям?

— Изначально мы думали, что потеряли порядка 60–70% данных, но в процессе расследования выяснили, что хакеры кое-что не учли. Пришло время «чеховского ружья»: я упоминал, что до инцидента мы успели виртуализовать файловый сервер — это помогло нам восстановить информацию. Скорее всего, вымогатели просто не предполагали, что зашифрованные файловые данные, размещенные на виртуальной инфраструктуре, могут быть проанализированы и восстановлены.

НИ ОДНА КОМПАНИЯ, В КОТОРУЮ МЫ ОБРАЩАЛИСЬ, НЕ СМОГЛА ПРЕДОСТАВИТЬ ГАРАНТИЙ ВОССТАНОВЛЕНИЯ ДАННЫХ. ХОТЯ СТОИМОСТЬ РАБОТ СОСТАВЛЯЛА ОТ 600 ТЫС. ДО 1,5 МЛН РУБ.

Мы взяли виртуальную машину с зашифрованными данными и «завели» ее с помощью альтернативного загрузчика. На выходе получили кучу мусора, проанализировали его с помощью инструментов LiveCD и в конечном счете восстановили порядка 80% информации. Зашифрованные данные с физических рабочих станций и серверов таким образом восстановить не удалось.

— Сколько времени вам потребовалось, чтобы вернуть компанию в штатный режим работы?

— На децентрализацию ушло около четырех дней, затем мы подняли облачный файловый сервис в Nextcloud и запустили процедуру восстановления данных — в сумме это заняло порядка двух-трех недель.

— Вы выяснили, кто вас взломал?

— Эксперты Positive Technologies определили, что атака шла из-за границы: из Нидерландов, потом из Украины и Германии. Скорее всего, это прокси-серверы, а злоумышленники находятся где-то в России. Других подробностей мы не знаем.

Было много предположений о том, кому это было выгодно, вплоть до кулуарных слухов из серии: «Как только пришли новые ИТ-шники, сразу стало что-то происходить. Почему у старых все было хорошо? Может быть, это они?»

— Вы предполагали, что такой инцидент возможен?

— Да, все предпосылки были. Я работал в достаточно крупных организациях, строил распределенные сети с тысячами хостов и прекрасно знаю, что ИБ нужно уделять много внимания. К сожалению, это понимают далеко не все. Простой пример: однажды я отвечал за распределенную сеть на три тысячи сотрудников в Москве. Она не была должным образом защищена, потому что руководство понимало, для чего им ИТ, а вот с ИБ ситуация была значительно сложнее. На мой взгляд, это достаточно распространенная проблема.

— Учитывая полученный опыт, что бы вы изменили в своих действиях, если бы могли переиграть ситуацию с самого начала?

В первую очередь заблаговременно внедрил бы систему мониторинга, усилил контроль сетевой активности, установил антивирусы и быстрее переходил с RDP на защищенную схему удаленного доступа. Хотя мы все равно не смогли бы полностью контролировать ситуацию — не хватило бы опыта и специалистов. Кроме того, все наши ресурсы уходили на тушение пожаров и текучку, а в таких условиях сложно заниматься развитием.

— Какие действия вы предпринимаете прямо сейчас?

— Мы перешли на независимого интернет-провайдера. В рамках выполнения работ по подключению интернет-канала он проложил независимую вертикальную кабельную систему между этажами и другим зданием и предоставил необходимое оборудование (коммутатор, трансиверы, оптические патч-корды), что позволило нам отключиться от локальной сети арендодателя. Также мы внедрили в сеть программные маршрутизаторы с функциями межсетевых экранов. Взяли за правило мониторить сетевую активность и выявлять попытки проникновения. Перенастраиваем беспроводную сеть, изолируем от доступа к построенной инфраструктуре. На некоторых серверах вообще обнаружили сервисы, очень похожие на майнинговые.

Кроме того, мы провели аудит существующей Wi-Fi-сети. Выявили и отключили взломанные точки, завели все точки на централизованное управление контроллером, изолировали от общей сети и отключили административный доступ из беспроводной сети.

— Поделитесь планами на будущее?

— В ближайшее время мы будем обсуждать с новым генеральным директором вопросы кибербезопасности. Нам нужны не только специалисты, но и современные инструменты: DLP, песочница, средства анализа сетевой активности и т. д. Кроме того, детали многих процессов лучше закрепить нормативными актами. Чтобы руководство выделило бюджет, все это нужно будет тщательно обосновывать — проводить пилоты и показывать наглядные результаты, вплоть до обнаружения конкретных уязвимостей. Пока наши запросы вызывали у топов только недоумение: «Почему так дорого?», «Зачем вообще в это вкладываться?». Ситуация усугубляется тем, что многие системы, которыми пользуется компания, были написаны почти 20 лет назад. Это огромные незадокументированные legacy-решения, которые всегда обслуживали подрядчики. Нормально разобраться в них практически невозможно. Единственный выход — переходить на новые системы, выполнять миграцию данных, бизнес-процессов и обеспечивать их безопасность. Это долгий и сложный путь, но другого выхода у нас нет.

P. S. Конец августа 2023 г.

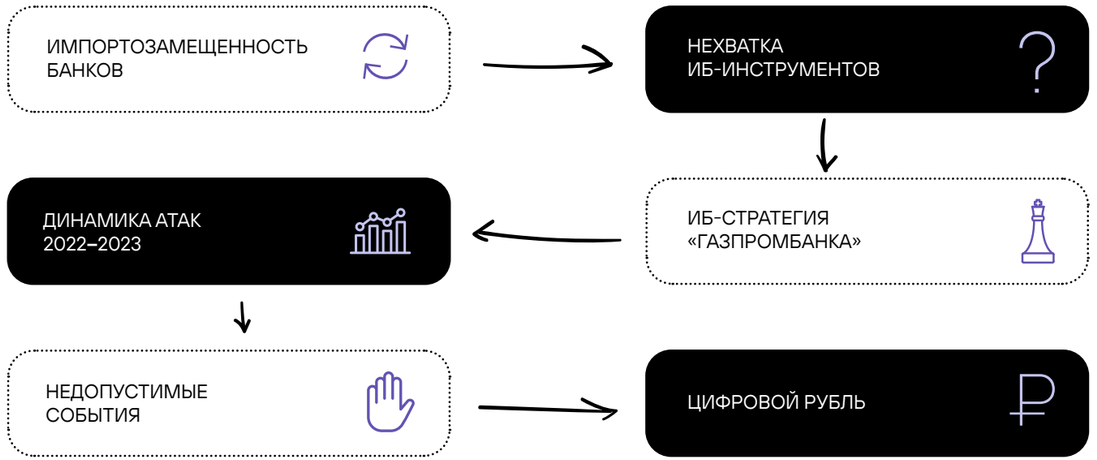
На текущий момент мы практически без привлечения дополнительных затрат смогли выстроить защиту новой базовой инфраструктуры на базе pfSense. Сконфигурировали новый почтовый сервер на основе Microsoft Exchange, настроили расширение SPF, политику DMARC и метод email-аутентификации DKIM. Сейчас идет миграция на новую инфраструктуру.

интервью

ФАКТИЧЕСКИ НАШУ ЭФФЕКТИВНОСТЬ ПРОВЕРЯЮТ АТАКУЮЩИЕ

Сергей Горленко

Вице-президент, начальник департамента развития технологий защиты информации «Газпромбанка»



— Насколько российские банки импортозаместились на сегодняшний день?

— За этот год понятие «импортозамещение» кардинально изменилось. В 2014-м, когда про переход на отечественные решения только начали говорить, на совещании в Министерстве цифрового развития, связи и массовых коммуникаций РФ было озвучено, что банковская сфера импортозамещена на 95%. В основном обсуждалась ситуация именно с прикладным программным обеспечением, а тот факт, что отечественные разработки собраны из импортных библиотек, функционируют на импортном общесистемном ПО и далеко не на отечественной технике, был вынесен на рассмотрение профильных рабочих групп.

Сегодня отраслевой комитет «Финансы» (профильный для кредитных организаций) рассматривает импортозамещение на всех уровнях — от железа до прикладного ПО. И здесь начинаются определенные проблемы. Катастрофически не хватает железа, особенно высокопроизводительного сетевого оборудования. Вызывает сомнение возможность оперативной организации полного цикла производства компьютеров, включая элементную базу. Определенные сложности возникают с использованием свободно распространяемого ПО, а ведь с его помощью создаются практически все коммерческие продукты. Есть большие проблемы с импортозамещением инфраструктурных систем и ряда высоконагруженных ИБ-решений. Резюмирую: у профильного ПО высокий уровень импортозамещения, в области ИБ дела обстоят чуть хуже, а ИТ-инфраструктура, как сейчас принято говорить, является зоной роста.

У ПРОФИЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВЫСОКИЙ УРОВЕНЬ ИМПОРТОЗАМЕЩЕНИЯ, В ОБЛАСТИ ИБ ДЕЛА ОБСТОЯТ ЧУТЬ ХУЖЕ, А ИТ-ИНФРАСТРУКТУРА, КАК СЕЙЧАС ПРИНЯТО ГОВОРИТЬ, ЯВЛЯЕТСЯ ЗОНОЙ РОСТА

— Каких российских ИБ-инструментов вам не хватает прямо сейчас? Вы принимаете участие в разработке или тестировании отечественных продуктов?

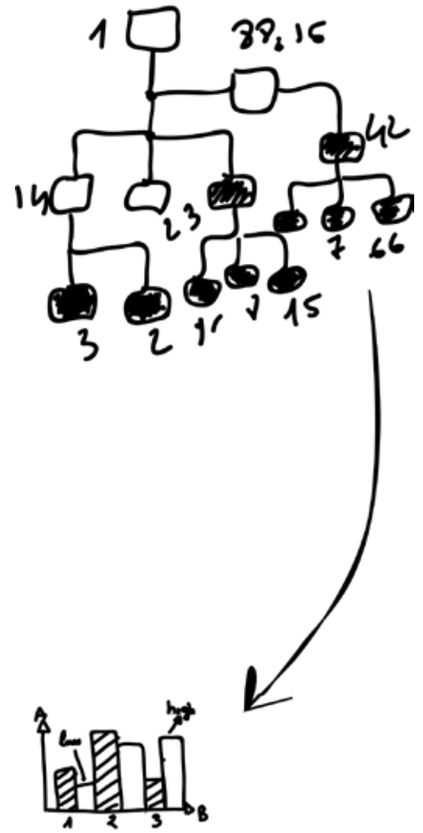
— Крайне не хватает производительных решений в области сетевой безопасности. Возможно, нам не везло, но найти межсетевые экраны производительностью больше 4 Гбит нам не удалось. На нашей инфраструктуре и с нашими правилами экранирования мы не смогли достичь даже паспортной производительности ни с одним из протестированных решений. Кроме того, при переходе на отечественные продукты мы лишаемся возможности микросегментации ЛВС. В совокупности с нехваткой сетевого железа проблема перерастает в необходимость комплексного перепроектирования защищенной ИТ-инфраструктуры и, возможно, пересмотра концепции использования корпоративного облака.

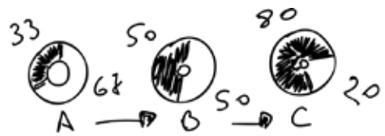
Отдельно отмечу отсутствие эффективных инструментов для реализации практик SecDevOps, а также отечественных баз по уязвимостям и рейтингам компонентов свободно распространяемого ПО. Похвастаться богатством решений в области безопасности контейнерных приложений тоже пока нельзя.

Конечно, наш банк принимает участие в тестировании российских продуктов. У нас большая программа пилотирования импортозамещенных ИТ- и ИБ-решений. Как члены отраслевого комитета «Финансы» и его подкомитетов мы анализируем поступающие по этой линии продукты. С разработкой несколько сложнее. Я разделяю мнение о нецелесообразности концепции «каждая организация разрабатывает свою импортозамещенную операционную систему или базу данных». Во-первых, крупные игроки вряд ли смогут договориться о базовых требованиях к продуктам. Во-вторых, таким образом мы впустую тратим ресурс компаний, которые должны, а главное, могут создавать реально работающие решения и постепенно повышать их эксплуатационные характеристики. Схожую позицию занимает Минцифры РФ: нужно определить ключевые продукты в разных областях и выделить гранты на их реализацию. Тем не менее в нашем банке есть ряд оптимизированных решений, которые сильно отличаются от коробочных версий. Мы активно взаимодействуем с их производителями в разных форматах.

— Сколько отечественных ИТ- и ИБ-решений было в банке до весны прошлого года?

— Около 95% прикладного ПО, 70–80% ИБ-продуктов и порядка 5–10% инфраструктурных решений.





«НАМ НЕ ПРИХОДИТСЯ ОБЪЯСНЯТЬ НА ПАЛЬЦАХ»

— Как изменилась ИБ-стратегия банка по сравнению с 2022 г.? От каких инициатив и проектов вам пришлось отказаться, а какие, наоборот, запустить?

— Наша ИБ-стратегия не изменилась. Это обусловлено тем, что система обеспечения информационной безопасности фактически доказала возможность противодействия внешним угрозам. При этом акценты практической работы, безусловно, сместились с обеспечения выживания еще не замещенных продуктов к их плановой замене. В целом работа ведется в соответствии с намеченным roadmap, без пересмотра проектного портфеля.

Отдельно выделю вопрос о необходимости безопасного использования текущих решений — как программных, так и инфраструктурных. Конечно, мы, как представители ИБ, ожидаем от коллег формирования обновленной ИТ-стратегии, но при этом нам нужно продумать, как безопасно использовать неимпортозамещенные компьютеры и ПО. Фактически перед нами стоит задача построения доверенной среды из недоверенных компонентов. Она особенно актуальна в части сетевой инфраструктуры.

— Переход на российские решения был болезненным?

— Только в части инфраструктурных решений, и он до сих пор не завершен. Коллеги из ИТ-блока проводят массу пилотов — они разделены на 22 направления, и по некоторым из них мы нашли достойные продукты. Поэтапный перевод ИТ-инфраструктуры банка на отечественные компоненты начнется в 3-м квартале 2023 г. Отмечу, что большой объем систем мы разрабатываем самостоятельно, при этом слово «Oracle» в течение следующих 4–5 лет будет считаться ругательным. Глобальных проблем нет, но вопрос перехода на российские ОС остается открытым. В части импортозамещения ИБ проблем меньше, например если мы говорим о высоконагруженных системах для ситуационного центра. Мы уже подобрали подходящее решение и проводим миграцию.

Если говорить о нашей экосистеме в целом, то «Газпромбанк Мобайл» решает вопросы импортозамещения самостоятельно. Gazprom Pay — наша собственная разработка, здесь сложностей возникнуть не должно. Скорее всего, мы столкнемся с трудностями при переводе на отечественный стек продуктов ЦФТ и «Диасофта». Они будут связаны с тем, что у производителей банально не хватит времени, чтобы обслужить весь российский банковский сектор к 2025 г.

— Как вы обосновывали ИБ-бюджет на этот год? Какие цели и задачи сейчас в приоритете?

— Обычно бюджет банка на ИБ состоит из двух больших блоков: поддержки и развития уже имеющихся инструментов и внедрения новых систем. В этом году добавилась третья категория — затраты на импортозамещение. Поскольку отчеты о состоянии киберграниц банка в 2022 г. были как никогда востребованы на всех уровнях руководства, проблем с обоснованием не возникло.

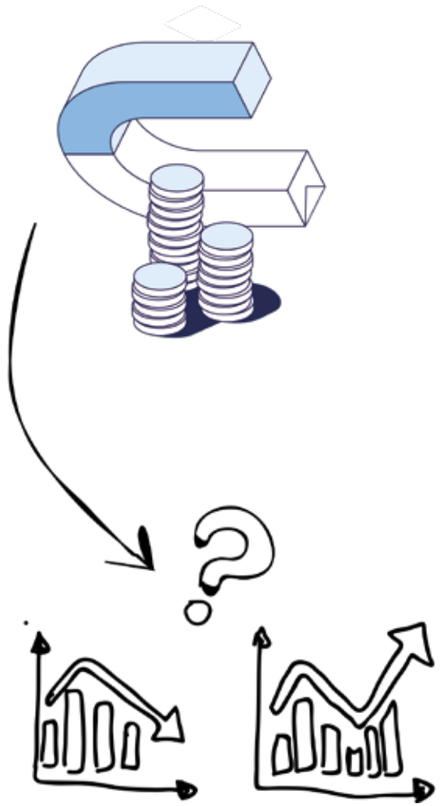
В приоритете, конечно же, импортозамещение, обеспечение непрерывности ИТ-производства (SecDevOps) и защита деятельности нашей группы во внешних облаках. Также банк уделяет большое внимание защищенности компаний группы. Помимо этого, нужно будет сделать упор на безопасность цепочек поставок. Сейчас много говорится о создании отечественных репозиториев для замены привычного GitHub. Однако определить, что именно привнесет эта замена с точки зрения повышения уровня ИБ, пока сложно. Кроме абстрактных заявлений о выполнении проверок по обеспечению ИБ, никакой конкретики нет. Нет и понимания роли государства в регулировании этого вопроса. Мы можем сколько угодно внедрять лучшие практики по проверке кода, но покупка отечественного ПО, разработанного без использования этих подходов, перечеркнет все наши усилия. А повлиять, например, на ЦФТ, кроме государства или регулятора, не сможет никто.

— Как много атак вы фиксируете по сравнению с прошлым годом?

— Мы достаточно крупный банк, поэтому злоумышленники постоянно нас «проверяют» — не отключилась ли случайно одна из защитных функций. Мы фиксируем регулярные, но несильные DDoS-атаки, эффект от которых практически незаметен. Кроме того, хакеры проводят целевые атаки, которые проверяют нашу стойкость на прикладном уровне, пытаются взломать наши протоколы.

Само собой, в 2022-м активность злоумышленников была выше — весной и осенью тех же DDoS-атак было заметно больше. Мы успешно отразили все атаки, но нужно заметить, что нам противостояли организованные преступные группировки, а не регулярные кибервойска. Остается вопрос: ждет ли нас усиление злоумышленников, вовлеченных в атаки на банк? Пока ситуация стабильна — все перешло в фоновый режим, но мы продолжаем повышать уровень защищенности. Например, анализируем возможности для более раннего предотвращения атак.

НАША ИБ-СТРАТЕГИЯ НЕ ИЗМЕНИЛАСЬ



— Какие инновационные ИБ- и ИТ-решения и подходы вы внедряете сегодня?

— Импортозамещение сложно назвать инновацией, хотя кейс по обеспечению деятельности процессингового центра на «Эльбрусах» можно отнести к этой категории. Также хочу отметить проекты по повышению защищенности каналов ДБО и обеспечению безопасности во внешних облаках. В рамках первого направления мы занимаемся темой API security. Это стало возможным благодаря переводу значительного количества систем на собственную разработку и созданию DevOps-конвейера (совместно с ИТ-блоком). Фактически мы разрабатываем инструмент, который позволяет выявлять уязвимости в выставленных наружу сервисах банка, автоматизировать проверки, которые выполняют наши специалисты, и преобразовывать выявленные недостатки в задания для разработчиков.

Второе направление стало актуальным в связи с появлением новой законотворческой инициативы ЦБ — о возможности обработки данных, содержащих банковскую тайну, во внешних облаках. Кроме того, из-за потенциальных проблем с железом нам, вполне возможно, придется перенести часть процессов на сторонние ресурсы. Пока мы концентрируемся на нормализации управления доступом работников банка к внешним сервисам. В дальнейшем планируем внедрение полноценного CASB-решения.

— ИБ-специалисты все чаще говорят о недопустимых событиях. Какие события являются недопустимыми для «Газпромбанка»?

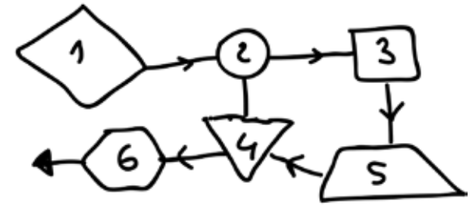
— Да, действительно, такой термин звучит все чаще, но я в каком-то смысле считаю это подменой понятий. Обычно ИБ сравнивают с гигиеной: выполнение простых правил позволяет избежать необходимости бороться с серьезными проблемами. Продолжая аналогию, можно сказать: недопустимое событие — это болезнь, которая случается с теми, кто не выполнял на регулярной основе гигиенические процедуры. Я считаю, что недопустимые события во всех кредитных организациях одинаковы: это серьезные денежные и репутационные потери. Существует масса путей, которые могут к ним привести, но все они сводятся к возможности эксплуатации тех или иных уязвимостей в ИТ-ландшафте организации.

Наш бизнес постоянно развивается, поэтому мы регулярно проводим переоценку рисков ИБ с учетом появления новых и изменения уже знакомых угроз. Те угрозы, которые признаны существенными, минимизируются внедрением новых ИБ-решений или изменением процессов. Я — за процессный подход к обеспечению ИБ, а не за создание некоторого количества табу. Возможно, нашему банку повезло, потому что нам не приходится на пальцах объяснять руководству значимость ИБ.

— Как вы оцениваете эффективность ИБ-направления в банке? Какие KPI перед вами стоят?

— Было бы очень удобно ответить, что эффективность ИБ оценивается по отсутствию недопустимых событий, но мы за процессы. Я считаю, что важными показателями являются качество оценки актуальных для организации угроз, полнота охвата / покрытия угроз ИБ-процессами и регулярность выполнения этих процессов. По большому счету, нашу эффективность оценивают атакующие. По итогам прошлого года можно сказать, что созданная в банке система ИБ работает хорошо.

По поводу KPI у нас с одним из коллег была шутка: на выставках предлагать самым активным менеджерам по продажам разработку системы показателей эффективности инфобеза. Как правило, это была последняя фраза в разговоре, после которой нам предлагали кофе навынос. Поэтому позволю себе не конкретизировать, что непосредственно у нас является KPI.



— «Газпромбанк» принимал участие в предварительных испытаниях цифрового рубля. Как будет реализована защита цифровых кошельков ваших клиентов?

— О защите цифровых кошельков уже подумал сам владелец платформы цифрового рубля — ЦБ РФ. Фактически банкам предоставляется программный модуль, на борту которого будет полный набор специфических для цифрового рубля средств защиты. Наша главная задача в данном случае — правильно донести электронные документы до платформы цифрового рубля.

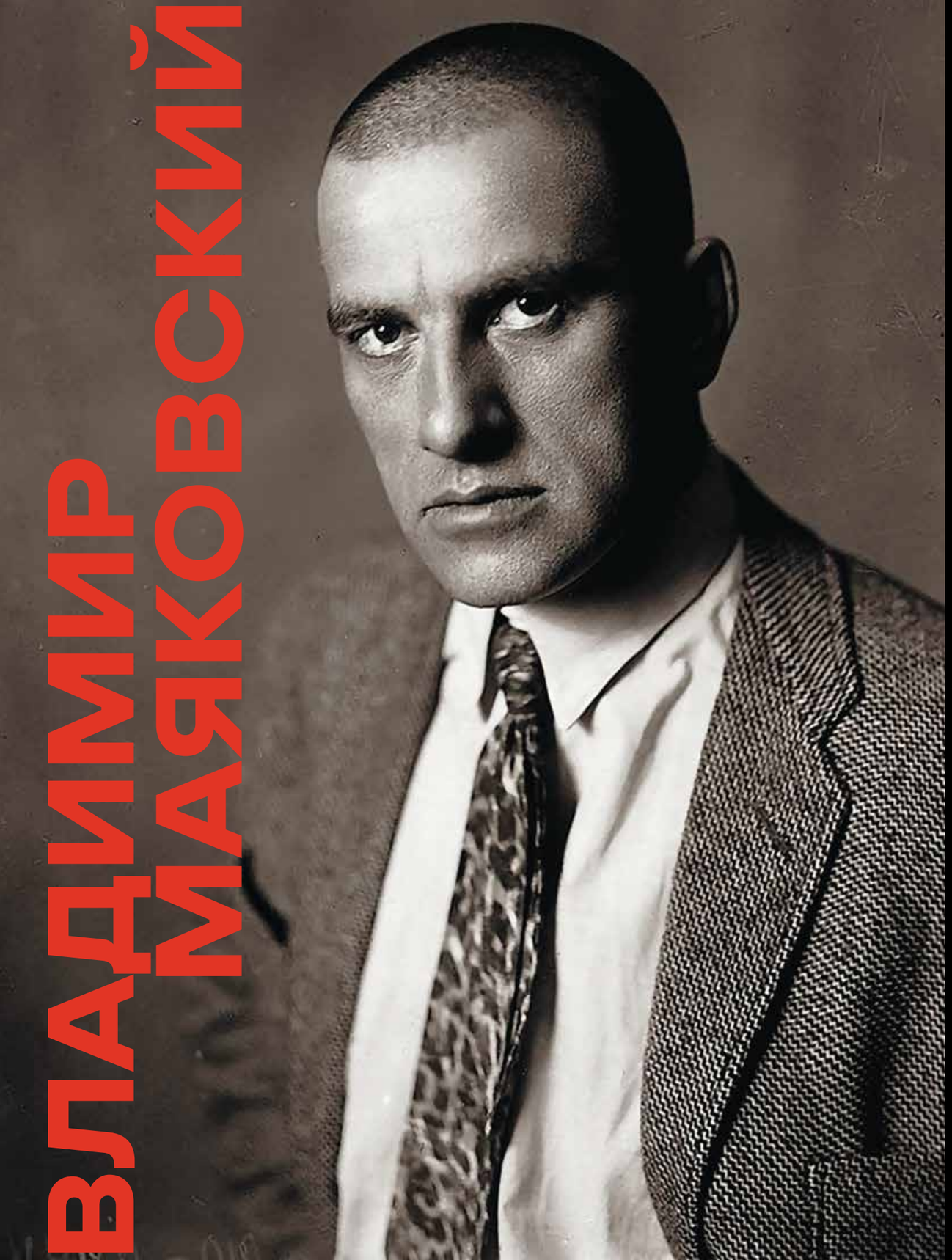
ЧТО ЕЩЕ ПОЧИТАТЬ:

Ого, какая ИБ!

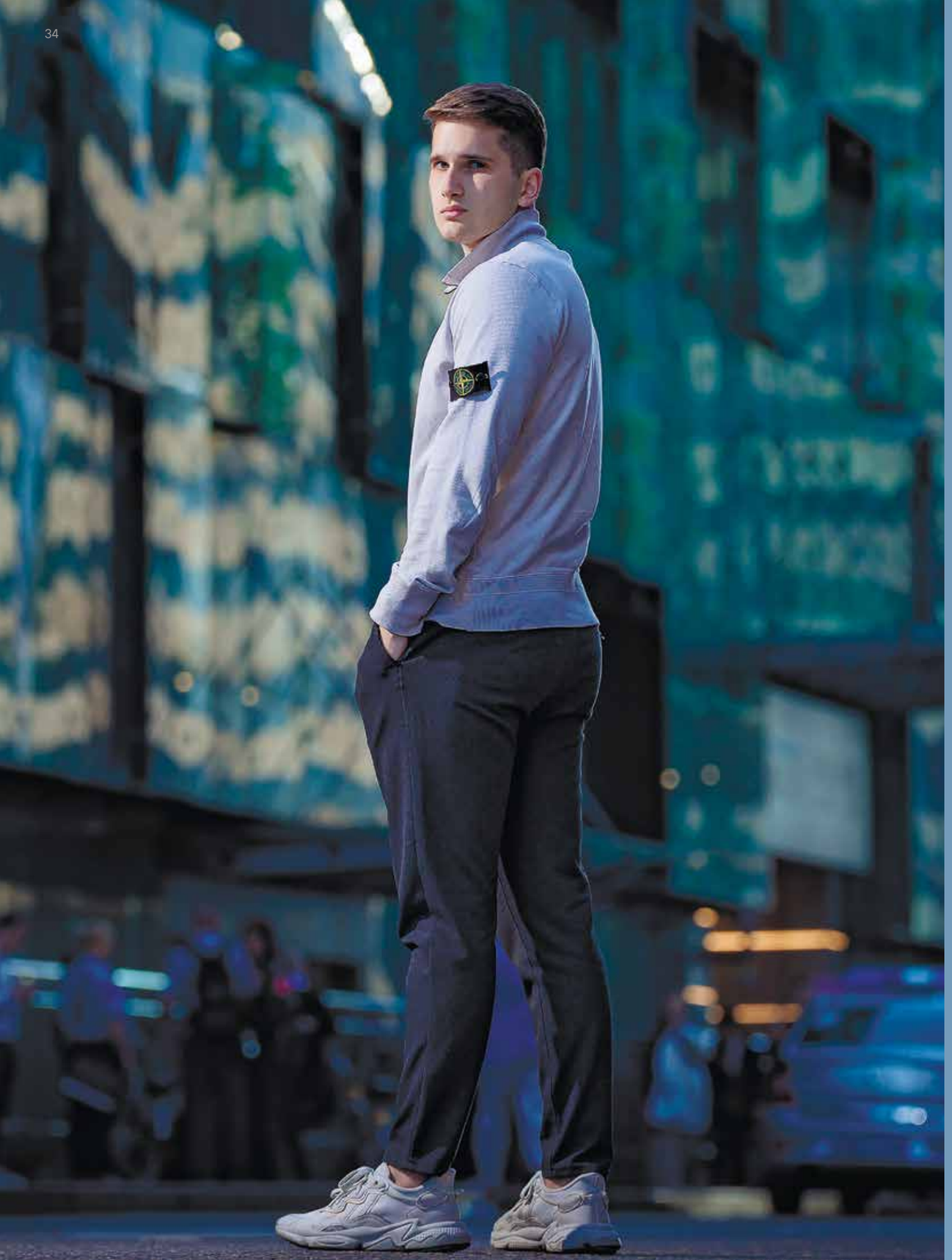
Защищенность финансовой отрасли — промежуточные итоги 2022 года

Импортозамещение в банках затрагивает весь технологический стек — от оборудования до софта

Уральский форум «Кибербезопасность в финансах». Итоги. Тренды. Ожидания



ВИКТОРИЯ АЛЕКСЕЕВА ЕЛЕНА ЛАККАЙ ДМИ
СКЛЯРОВ АЛЕКСАНДРА МУРЗИНА АЛЕКСЕЙ
ЛУКАЦКИЙ ВЛАДИМИР КОЧЕТКОВ ЭЛЬМАР
НАБИГАЕВ ЕВГЕНИЙ КАТУША ЮЛИЯ СОРОК
НИКИТА ПОПОВ ВИКТОРИЯ КЛОЧКОВА СЕМ
КРУТИЦКИЙ БОРИС ЧУБИН ГРИГОРИЙ АЛЕК
ОЛЬГА БЕЛЕЦКАЯ ОЛЬГА МОСКВИЧЕВА СВЕ
ИСАЕВА ВАЛЕРИЯ МИТЯЕВА ЯРОСЛАВ БАБ
ВЛАДИМИР НАЗАРОВ МАКСИМ КОСТИКОВ
АНДРЕЙ БАЧУРИН АЛЕКСАНДР МОРОЗОВ Н
ПОПОВ АЛЕКСЕЙ ЛУКАЦКИЙ ВЛАДИМИР КО
СВЕТЛАНА ИСАЕВА АЛЕКСАНДР МОРОЗОВ
АЛЕКСЕЕВА **КОМАНДА PHDAYS** ЕЛЕНА ЛАКК
ДМИТРИЙ СКЛЯРОВ АЛЕКСАНДРА МУРЗИН
АЛЕКСЕЙ ЛУКАЦКИЙ ВЛАДИМИР КОЧЕТКОВ
ЭЛЬМАР НАБИГАЕВ ЕВГЕНИЙ КАТУША ЮЛИ
СОРОКИНА НИКИТА ПОПОВ ВИКТОРИЯ КЛО
СЕМЕН КРУТИЦКИЙ БОРИС ЧУБИН ГРИГОР
АЛЕКСЕЕВ ОЛЬГА БЕЛЕЦКАЯ ОЛЬГА МОСКВ
СВЕТЛАНА ИСАЕВА ВАЛЕРИЯ МИТЯЕВА ЯРО
БАБИН ВЛАДИМИР НАЗАРОВ МАКСИМ КОС
АНДРЕЙ БАЧУРИН АЛЕКСАНДР МОРОЗОВ Н
ПОПОВ АЛЕКСЕЙ ЛУКАЦКИЙ ВЛАДИМИР КО
СВЕТЛАНА ИСАЕВА АЛЕКСАНДР МОРОЗОВ
ВИКТОРИЯ АЛЕКСЕЕВА ЕЛЕНА ЛАККАЙ ДМИ
СКЛЯРОВ АЛЕКСАНДРА МУРЗИНА АЛЕКСЕЙ
ЛУКАЦКИЙ ВЛАДИМИР КОЧЕТКОВ ЭЛЬМАР



СЛОЖНЕЕ, ЧЕМ КАЖЕТСЯ: СОЦИАЛЬНАЯ ИНЖЕНЕРИЯ 2023

Константин Полишин

Старший специалист отдела тестирования
на проникновение Positive Technologies

Время прочтения:	20 минут
Для кого:	ИБ-специалисты, эксперты по SE, разработчики
Прокачиваем знания:	полезная нагрузка, начальный доступ, общая эффективность kill chain

В ИБ-комьюнити сформировалось представление о том, что социальная инженерия (SE) — это что-то гуманитарное и легкореализуемое: «Пару часов изучаем статьи из Google, ищем репозитории на GitHub — и готово!» На самом деле целевой фишинг — это симбиоз самых разных hard- и soft навыков, технологий и масса сценариев реализации целевой фишинговой атаки. Чтобы развеять сложившийся миф, нужно погрузиться в тренды социальной инженерии. Само собой, в рамках одной статьи мы не сможем подробно описать все возможные механизмы SE, поэтому кратко пройдемся по самым интересным.

АНАЛИТИКА RED TEAM: ЦИФРЫ, МЕТРИКИ, НЮАНСЫ

В последние годы мы фокусировались на реализации проектов red team и проведении атак с использованием SE в условиях, максимально приближенных к реальности. На сегодняшний день мы фиксируем следующие тенденции red team:

Таргетность рассылок «общих» сценариев. Чем больше фокус-группа, тем выше шанс обнаружения.

Тщательное формирование и изучение фокус-групп. Выявление наиболее уязвимых категорий сотрудников, исключение из выборки технических специалистов и др.

Тщательная подготовка перед отправкой полезной нагрузки. Анализ стека СЗИ, изучение архитектуры защиты, проведение предварительных вишинг-атак для определения эффективных сценариев и т. п.

Многоступенчатость фишинговых сценариев.

Рост влияния социальной инженерии на результат проекта.

Тип SE \ Метрика	Письма	Звонки	Креды	Пробивы	Конверсия
Spearphishing	53	X	8	3	20,7%
Internal phishing	44	X	4	5	20,45%
Internal phishing + vishing	X	8	6	X	75%
Smishing via TG	8	X	2	4	75%
Vishing + smishing via TG	X	7	5	X	71,4%

Рис. 1. Статистика разных типов SE на red team проектах

ПОЧЕМУ ВЛОЖЕНИЯ — МОВЕТОН

Крупные российские компании обладают высоким уровнем зрелости ИБ и имеют большой портфель средств защиты информации, в том числе почтовых. В таких условиях отправлять полезную нагрузку в виде классического аттачмента — банально неэффективно.

Во-первых, большинство вложений с опасными расширениями блокируются на уровне почтового шлюза. Письмо, отправленное с внешнего адреса на корпоративную почту сотрудника с аттачментом в виде файла с расширением .zip, .html, .lnk или .js, будет заблокировано и не дойдет до адресата.

Во-вторых, если blacklist не настроен или его удалось обойти, почтовый антивирус может обнаружить полезную нагрузку в статике или выдать false positive.

В-третьих, если вложение из нашего письма прошло первый эшелон проверок и базовых средств защиты, то стоящий в разрезе sandbox может детектировать полезную нагрузку во время динамического анализа, заблокировать доставку письма сотруднику и создать инцидент ИБ.

Так или иначе, исход один: мы целенаправленно раньше времени отдаем полезную нагрузку на анализ всем имеющимся у компании СЗИ, в ходе которого она либо будет ими задетектирована, либо может попасть на ручной анализ.

Как результат — количество фишинговых писем с аттачментами стремительно уменьшается, так как существует риск потерять полезную нагрузку на этапе ее доставки до сотрудника.

По данным ProofPoint, использование макросов VBA и XL4 с октября 2021 г. по июнь 2022 г. сократилось примерно на

66%

Также снижается эффективность фишинга с документами Microsoft Office. Это происходит на фоне следующих действий Microsoft:

1. Октябрь 2021 г. [1]: блокировка макросов Excel 4.0 (XML-макросов) по умолчанию.

2. Февраль 2022 г. [2]: начало блокировок VBA-макросов в полученных из интернета документах Microsoft Office , также по умолчанию.

В результате этих блокировок использование офисных документов с макросами в фишинговых рассылках значительно сократилось — в силу снижения их эффективности.

Кроме того, в марте 2023 г. в Microsoft [3] уловили тренд на OneNote-векторы и собираются заблокировать возможность взаимодействия со 120 опасными файловыми расширениями. В итоге из интересной опции контейнера для хранения и доставки полезной нагрузки OneNote постепенно превращается в обычный фиолетовый блокнот.

Раз аттачменты становятся неэффективными, как доставлять полезную нагрузку? На текущий момент и в будущем ссылки будут самым эффективным способом: они с трудом поддаются анализу и стандартными почтовыми средствами защиты, и песочницами. На последних остановимся немного подробнее.

Задачу анализа URL-адресов в принципе нельзя назвать простой. Даже лучшим Sandbox-решениям на мировом рынке сложно вынести однозначный вердикт — четко определить сомнительность прикрепленной к письму ссылки и намерения целевой страницы, на которую собирается перейти пользователь. Тем не менее песочницы обладают богатым функционалом для динамического анализа прикрепленных URL-адресов, благодаря которому создают головную боль атакующим и заставляют их придумывать сложные методы обхода динамических проверок.

Ситуация осложняется тем, что в России защита корпоративной почты резко перешла в руки локальных вендоров. К сожалению, песочницы, которые остались на нашем рынке, плохо справляются с анализом ссылок. Чаще всего механизм проверки реализован через контейнер, в котором содержимое HTML-страницы забирается с помощью Wget и отдается на сканирование YARA-правилам. Как результат — снова становятся актуальными старые проблемы и риски. Приведу несколько примеров:

- Оставшиеся на рынке sandbox-решения не умеют работать с векторами Steal Credentials: детектировать фишинговые формы, вводить тестовые учетные данные и блокировать доставку писем с такими ссылками.

- Если ссылка ведет на облачный сервис, на котором хостится полезная нагрузка, песочница просто скачает HTML-страницу.

- Оставшиеся на рынке sandbox-решения не умеют получать доступ к целевой странице с нагрузкой, на которую пользователя перенаправляют после ввода учетных данных, или в тех случаях, где требуется явная пользовательская активность, чтобы получить доступ к нагрузке.

- Сложности в анализе содержимого обфусцированных версий загруженных HTML-страниц. При сильной обфускации кода (например, использовании HTML Smuggling, BitB, BitM) возникают трудности в обнаружении атак, основанных на JavaScript.

АКТУАЛЬНЫЕ СПОСОБЫ ДОСТАВКИ И ХРАНЕНИЯ ПОЛЕЗНОЙ НАГРУЗКИ

HTML Smuggling

Это техника скрытной доставки вредоносного ПО, в которой используются легитимные функции HTML5 и JavaScript. Закодированный JavaScript Blob-объект (например, методом Base64) помещается в специально созданный HTML-документ или веб-страницу. Когда атакуемый пользователь открывает HTML-страницу в своем браузере, тот, в свою очередь, декодирует хранимые в ней данные, в результате чего Blob-объект передает полезную нагрузку на узел пользователя.

Таким образом, полезная нагрузка компилируется локально на рабочей станции пользователя в обход стандартных средств защиты сетевого периметра, таких как прокси, IDS/IPS и анализаторы трафика. Они проверяют только подозрительные вложения (например, файлы с расширениями .exe, .zip или .docx) или только трафик. Поскольку вредоносные файлы создаются после того, как HTML-файл загружается на конечной точке через браузер, данные, которые фиксируются такими СЗИ, представляют собой легитимный HTML- и JavaScript-трафик. При этом вариант отключения работы JavaScript на хостах сотрудников является не лучшей идеей: это может остановить бизнес-процессы и работу других легитимных веб-приложений.

Распространенные варианты использования HTML Smuggling:

- Доставка HTML-файла на хост в каком-либо контейнере (например, .zip или .zip + .iso), приложенном аттачментом к письму.

- Отправка ссылки на веб-приложение. Жертва переходит по ссылке, и нагрузка компилируется на стороне атакуемого хоста.

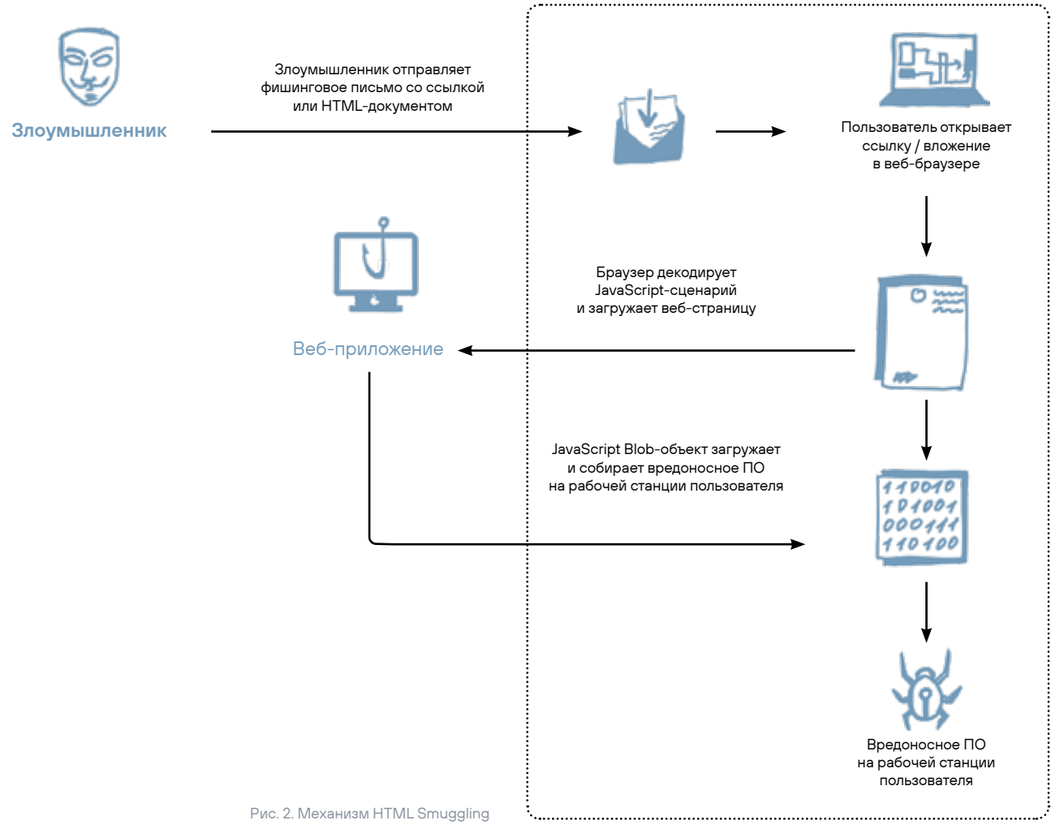


Рис. 2. Механизм HTML Smuggling

Облачные сервисы

Второй актуальный способ хранения и доставки полезной нагрузки связан с облачными сервисами: Yandex.Cloud, Облаком Mail.Ru, Google Диском, Dropbox и др. Схема проста: загружаем нужный файл, получаем ссылку и отправляем пользователю. Такой способ, к примеру, использует группировка OldGremlin.

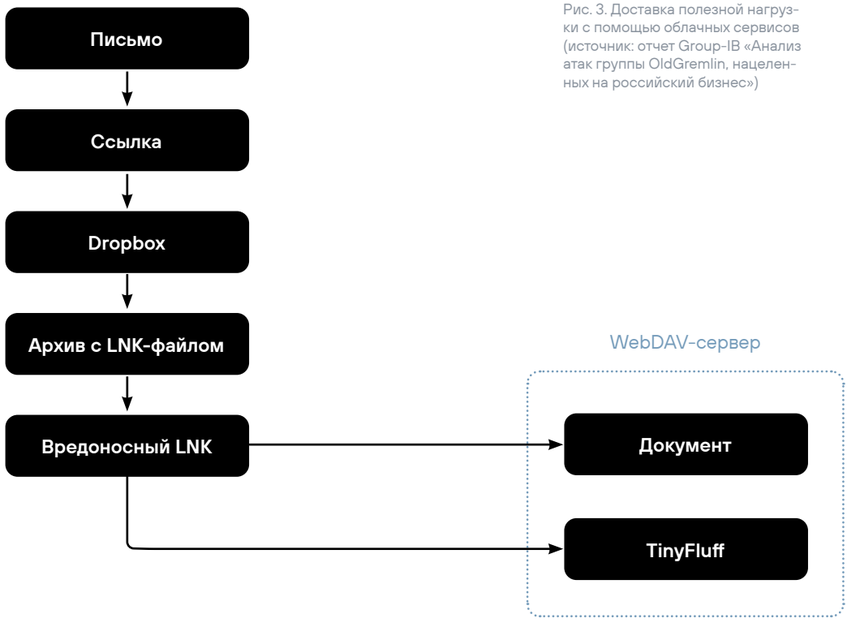


Рис. 3. Доставка полезной нагрузки с помощью облачных сервисов (источник: отчет Group-IB «Анализ атак группы OldGremlin, нацеленных на российский бизнес»)

Преимущества подхода:

› Пользователи доверяют ссылкам на знакомые облачные сервисы.

› Заблокировать домены mail.ru, yandex.ru и др. проблематично, а разделегировать невозможно.

› Доступность и простота размещения полезной нагрузки.

› Возможность проходить мимо правил корреляции и радаров средств защиты. Даже если СЗИ зафиксируют условное событие «Сотрудник перешел по ссылке из письма на Mail/Yandex/Google-диск», никто не придаст этому должного значения, пока не произойдет инцидент.

Тем не менее у данного способа существуют и недостатки:

› **Недостаточная операционная безопасность (OPSEC).** Хранение полезной нагрузки не на своих мощностях и хостингах, а на сторонних сервисах несет риски для ее «здоровья». Потенциально сами сервисы могут получить доступ к хранимой нагрузке и отправить на анализ антивирусными решениями.

› **Оперативное удаление полезной нагрузки из облачного сервиса.** Как-то мы использовали этот способ размещения полезной нагрузки на проекте: при расследовании инцидента специалистами SOC заказчика она была быстро удалена. Из-за этого пользователи, которые получили от нас письма со ссылками на облако с нагрузкой, не могли скачать файлы. В случае хостинга полезной нагрузки на своем сервере такое развитие событий, естественно, невозможно.

› **Предварительное отображение содержимого** для пользователей.

WebDAV

Концептуально способы доставки удаленной нагрузки с сервера WebDAV сводятся к необходимости выполнения консольных команд на хосте (User Execution этап). Есть масса User Execution техник — от экзотичных до более распространенных и эффективных.

Из экзотики: в феврале 2023 г. хакеры, распространяющие вредоносное ПО IcedID, использовали URL-файлы для скачивания пользователем .bat-стейджеров, выполняющих CMD-команды по запуску удаленной нагрузки, размещенной на WebDAV-сервере.

Более распространенный и эффективный вариант — использование LNK-файлов для выполнения команды net use и запуска нагрузки, размещенной на сервере WebDAV. Данным способом пользуются в своих атаках все те же OldGremlin.

Преимущества подхода:

› Готовая реализация сетевого взаимодействия на основе расширенного набора HTTP-команд и автоматическое распознавание корпоративного прокси-сервера (включая аутентификацию) для выхода в интернет.

› Взаимодействие с WebDAV-сервером выглядит так, будто ОС просто выполняет сетевой запрос.

Недостатки подхода:

› Возможен долгий процесс подключения виртуального сетевого диска и выполнения консольных команд (в зависимости от ресурсов атакуемого хоста), и все это время пользователь будет видеть CMD-окно, которое он может закрыть.

› Вся полезная нагрузка, доступная или загружаемая по пути UNC, локально копируется в кэш клиента WebDAV (C:\Windows\ServiceProfiles\LocalService\AppData\Local\Temp\TfsStore\Tfs_DAV\), что может покрываться мониторингом SOC.

› Недостаточная операционная безопасность (OPSEC): курсирование трафика через все СЗИ, прокси-серверы, анализаторы внешнего и внутреннего трафика, firewall и NGFW-решения.

ДОСТАВКА И ХРАНЕНИЕ НАГРУЗКИ: КЛЮЧЕВЫЕ ТРЕНДЫ

Низкая эффективность противодействия link-векторам у оставшихся на рынке продуктов защиты корпоративной почты

HTML Smuggling позволяет компилировать полезную нагрузку на хосте атакуемого пользователя в обход средств защиты

Популярность использования готовых решений для хостинга полезных нагрузок: облачные сервисы и WebDAV-серверы

АКТУАЛЬНЫЕ СПОСОБЫ ПОЛУЧЕНИЯ НАЧАЛЬНОГО ДОСТУПА

Контейнеризация

Kill chain состоит как минимум из двух компонентов — полезной нагрузки и User Execution техники. Все это нужно доставить без лишних браузерных предупреждений в интуитивно понятном виде для атакуемого пользователя. Для этого можно использовать разные виды контейнеризации полезной нагрузки:

1) архивы (RAR, ZIP, Gzip и т. п.),

2) образы дисков (ISO, IMG, VHD, VHDX),

3) CAB-файлы.

Кроме удобной доставки необходимых компонентов, контейнеры могут служить способом обхода технологии Mark-of-the-Web. Она создает альтернативный поток данных (ADS) с именем Zone.Identifier и добавляет к нему правильный ZoneId, чтобы указать, из какой зоны был получен файл. А также отображает предупреждение для пользователя, который хочет осуществить взаимодействие с полученным из интернета файлом.

Примеры обхода технологии Mark-of-the-Web с помощью контейнеров:

[4]

› CVE-2022-41049 [4]: когда неправильно генерировался альтернативный поток данных, без добавления в поток Zone.Identifier правильного ZoneId из-за выставленного атрибута для упакованного файла Read-Only, что при извлечении вызывало ошибку Access Denied при попытке Windows выставить ZoneId для файла, полученного из интернета. Это приводило к тому, что разархивированные файлы не помечались как файлы, полученные из интернета, и пользователь мог запускать их без каких-либо предупреждений и в обход технологии SmartScreen.

[5]

[6]

› CVE-2022-41091: проблема с проставлением правильного ZoneId файлам в образах диска публично известна давно [5] и эксплуатируется еще со времен Windows 8 [6]. Она была связана с тем, что альтернативные потоки данных (ADS) — это функция NTFS, и идентификатор зоны не распространялся на другие типы файловых систем, например FAT32 и UDF. Поэтому MOTW не распространялся на файлы внутри ISO/IMG и других образов дисков.

Техники обхода MOTW будут появляться и дальше. Например, уже существует целое семейство уязвимостей, используемых для обхода проверки цифровой подписи SmartScreen (CVE-2022-44698 и CVE-2023-24880). Microsoft, конечно, выпустили патчи, но здесь остается огромное пространство для исследований, поскольку необходимую ошибку для проверки цифровой подписи можно вызвать и передать в SmartScreen разными способами.

LNK

Если nmap — «швейцарский нож» пентестера, то LNK — «швейцарский нож» социального инженера. В основном я сталкиваюсь с кейсами применения LNK-файлов в качестве техники выполнения команд по запуску полезной нагрузки, поэтому отношу их к User Execution техникам. Тем не менее они универсальны и в зависимости от контекста могут использоваться по-разному:

1. **User Execution.** При взаимодействии запускается нагрузка, доставленная вместе с файлом.
2. **User Execution Dropper.** При взаимодействии доставляется и запускается нагрузка с удаленного сервера. Либо с помощью EmbedExeLnk и аналогичных техник нагрузка встраивается внутрь LNK, затем извлекается из него и помещается в отдельный файл с последующим запуском.
3. **Shortcut Modification.** При взаимодействии модифицируются легитимные ярлыки. Например, на запуск браузера с вредоносными расширениями для модификации и перехвата трафика.
4. **Persistence.** При взаимодействии происходит закрепление в системе с помощью создания LNK в автозагрузке.

На мой взгляд, это самый эффективный и универсальный вид User Execution техники. Используя LNK в своих TTP, можно рассчитывать на единичный запуск файла для выполнения всей логики полезной нагрузки и получения результата, так как защита от LNK-файлов сильно недостаточна в сравнении с угрозой. При этом для конечного пользователя LNK выглядят максимально легитимно, потому что могут мимикрировать под PDF, любые документы Microsoft Office или бинарные файлы.

С октября 2021 г. количество кампаний с использованием LNK-файлов выросло на

1675%

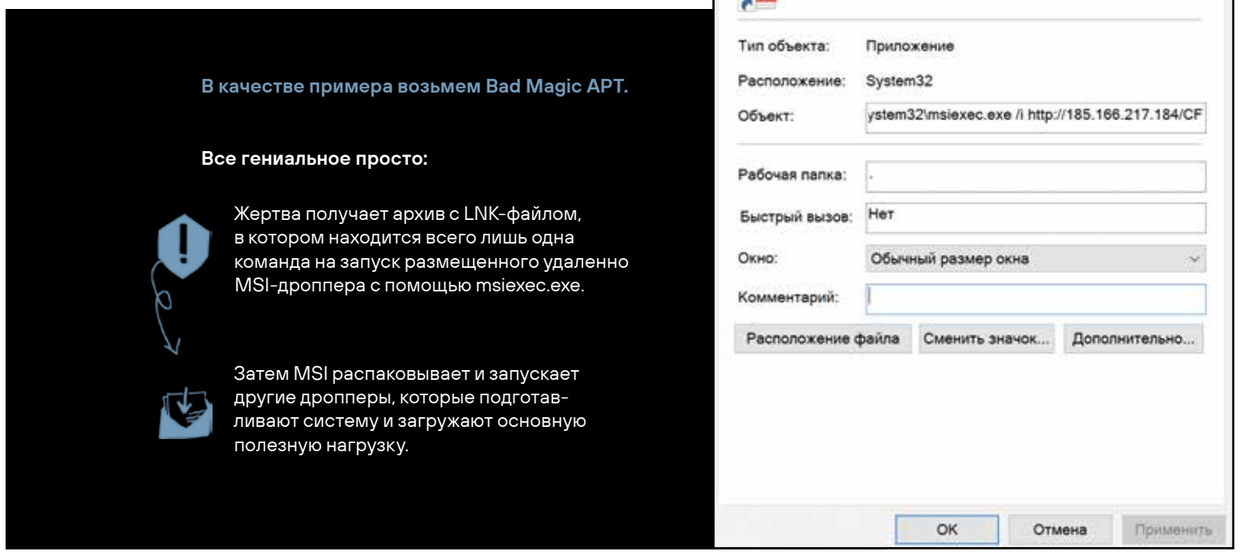
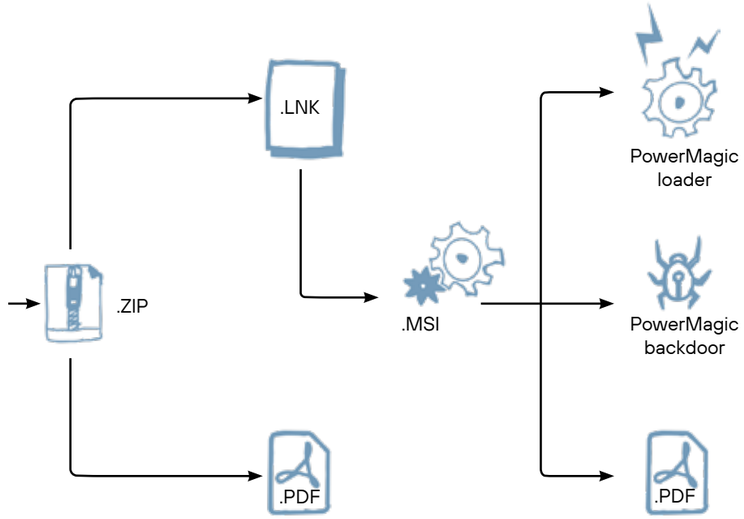


Рис. 4. Команды, зашитые в LNK-файл

Рис. 5. Схема применения LNK



Преимущества этого kill chain:

- Возможность эксплуатации LNK на всех Windows-целях. С помощью PS/CMD-команд можно реализовать сложную логику вектора kill chain на User Execution стадии.
- Возможность удаленного и локального запуска MSI.
- LNK позволяет использовать множество LOLBin-техник на User Execution стадии.
- MSI позволяет упаковать в себя все файлы, необходимые для установления обратного соединения с командным сервером.
- У MSI есть функция упорядочивания запуска извлекаемых файлов и возможность выполнения CMD/PS-команд для правильной отработки логики, заложенной в kill chain.
- Большая адаптивность под сценарий письма за счет мимикрирования LNK под легитимные документы и приложения.

Массовая миграция злоумышленников на LNK и рост их популярности начались относительно недавно. Хотя APT29 еще с 2016 г. использует в своих TTP связку HTML Smuggling + ZIP + LNK, модифицируя основную полезную нагрузку. А FIN7 начала миграцию с макросов Microsoft Word на LNK-файлы в 2017-м.

MSI

MSI в контексте фишинговых TTP представляет собой универсальный многофункциональный контейнер с возможностью извлечения файлов в нужную директорию и выполнения различных консольных CMD- и PowerShell-команд.

MSI-файлы используют технологию Microsoft COM Structured Storage, которая позволяет им иметь структуру, аналогичную файловой системе компьютера. По сути, это контейнеры со взаимосвязанными таблицами, образующими реляционную базу данных. Таблицы позволяют управлять процессом установки и задавать логику с помощью отдельных действий и их последовательностей.

MSI содержит Main Stream (основной поток), который также называется потоком базы данных (Database Stream). Последний состоит из таблиц, которые образуют реляционную базу данных (то есть связаны между собой для выполнения заложенной логики установки).

File Table. Важная часть базы данных MSI, определяет список файлов, которые должны быть установлены или удалены в процессе установки и удаления ПО.

Binary Table. Отвечает за хранение бинарных данных (например, исполняемых файлов, скриптов, изображений и др.), которые могут быть использованы Custom Actions или другими компонентами ПО.

CustomAction Table. Отвечает за определение пользовательских действий (Custom Actions) — дополнительных операций, которые можно выполнять во время установки или удаления ПО и которые не предусмотрены стандартными действиями Windows Installer.

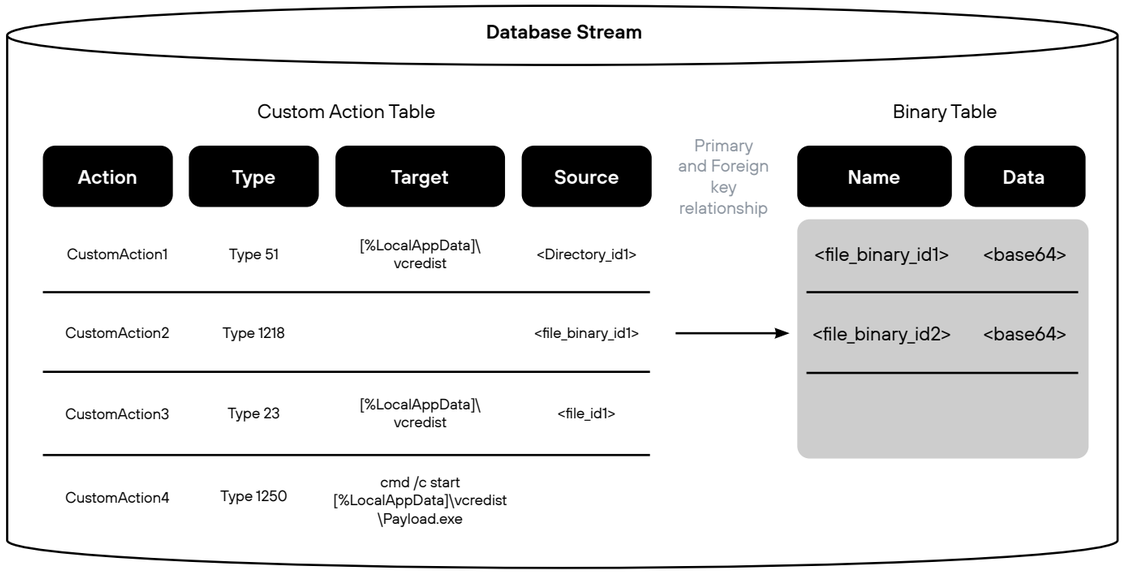


Рис. 6. Database Stream

Например, в таблицах может храниться следующая информация:

- **File Table** [7]**:** содержит полный список исходных файлов с их атрибутами.
- **Binary Table** [8]**:** содержит двоичные данные для таких элементов, как растровые изображения, анимации и значки. Двоичная таблица также используется для хранения данных пользовательских действий.
- **CustomAction Table** [9]**:** предоставляет средства интеграции пользовательского кода и данных в установку. Источником выполняемого кода может быть поток, содержащийся в базе данных, недавно установленный файл или существующий исполняемый файл.

[7]
[8]
[9]

Другие таблицы 10 содержат информацию о выполняемых пакетах последовательностей установки и выполнения.

Действия делятся на два типа:

Стандартные (Standard Action 11). Встроенные действия, которые позволяют разработчикам выполнять установку. Например:

1. InstallFiles 12 — извлечь файлы, указанные в таблице File 13, в целевой каталог установки.
2. InstallFinalize 14 — отмена всех установочных действий и удаление извлеченных файлов с диска.
3. WriteRegistryValues 15 — добавить значение реестра для устанавливаемого ПО.

Пользовательские действия (Custom Action) 16. Позволяют внедрять логику за пределами стандартных действий, запуская во время установки определенные двоичные файлы или скрипты. Например:

1. Исполняемые файлы, хранящиеся в файле MSI.
2. Экспортированные функции из DLL.
3. Скрипты JavaScript или VBS.

Как атакующих нас интересуют оба типа действий.

Custom Action Type 23. Используется для извлечения файлов из .cab-контейнеров (архивов) во время установки или удаления ПО. При выполнении Custom Action Type 23, Windows Installer извлекает файлы из указанных .cab-контейнеров и помещает их в указанную директорию на диске. Этот тип Custom Action обычно используется для установки компонентов, файлов или ресурсов, которые были сжаты в .cab-контейнеры, для уменьшения размера установочного пакета.

Custom Action Type 1218. Используется для запуска исполняемых файлов, которые хранятся в таблице Binary Table. Во время установки Custom Action Type 1218 извлекает бинарные данные из Binary Table и сохраняет во временном каталоге (C:\Windows\Installer\MSIXXXX.tmp) на целевой системе. Затем исполняемый файл запускается для выполнения определенной задачи или действия.

Media Table. Хранит сведения об исходных носителях, на которых содержатся установочные файлы ПО. Исходные носители могут быть представлены в виде cab-архивов, содержащих сжатые файлы. Благодаря этому обеспечивается эффективность установки и уменьшается размер установщика.

InstallExecuteSequence Table. Одна из основных таблиц в базе данных MSI (Windows Installer Database). Отвечает за определение последовательности выполнения действий в процессе установки или удаления ПО. В этой таблице определяется, какие Custom Actions и Standard Actions должны быть выполнены и в каком порядке.

Рассмотрим функциональные возможности Custom Action и MSI на примере использования фреймворка WiX 17 для генерации установочных пакетов. Он раскрывает весь потенциал пользовательских действий 18 и позволяет выполнять следующее:

- Сжимать и упаковывать файлы с полезной нагрузкой в .cab-контейнер с помощью Media Element 19. Контейнер с нашими файлами помещается в Media Table 20.
- Устанавливать целевой каталог для извлечения файлов из .cab-контейнера с помощью SetDirectory 21 (в терминологии Microsoft — Custom Action Type 51 22).
- Выполнять консольные CMD/PowerShell-команды и запускать извлекаемые файлы в процессе установки с помощью атрибута ExeCommand пользовательских действий.
- Задавать права для выполнения логики MSI с помощью атрибута InstallPrivileges в Package Element 23, чтобы при наличии прав администратора все установочные действия выполнялись от привилегированного пользователя.
- Запускать JS/VBS-скрипты с помощью атрибутов Script, VBScriptCall и JScriptCall пользовательских действий.

Используя WiX-фреймворк для комбинации лучших возможностей Standard Action и Custom Action, мы получаем достаточно широкий набор возможностей:

- Упаковка и хранение файлов в MSI разными способами: в Base64-формате внутри Binary-таблицы и в сжатом виде в .cab-контейнере (с помощью Media Element в Media Table).
- Извлечение файлов в директории с помощью InstallFiles и SetDirectory вместе с Custom Action Type 23.
- Выполнение произвольных команд, в том числе по запуску файлов с помощью атрибута ExeCommand.
- Запуск скриптов для изучения системы атакуемого хоста с помощью атрибутов Script, VBScriptCall и JScriptCall.

Также для управления логикой и последовательностью действий при установке существует таблица последовательностей 24. В ней задаются условия и порядок выполнения стандартных и пользовательских действий.

Приведу пример таблицы последовательностей, задающей логику работы MSI, и кратко объясню взаимодействие таблиц между собой.

В начале установки Windows Installer изучит последовательности действий в InstallExecuteSequence Table. CustomAction1 с последовательностью 2 в InstallExecuteSequence Table перенаправит Windows Installer за инструкциями к **CustomAction Table**. Движение по заложенной в MSI последовательности действий будет выглядеть следующим образом:

CustomAction Table:

1. **CustomAction1** с помощью пользовательского действия **Custom Action Type 51** инициализирует директорию *[%LocalAppData]\vcredist* в качестве каталога, в который будут извлечены файлы.

2. **CustomAction2** с помощью пользовательского действия **Custom Action Type 1218** запустит ранее упакованную бинарную нагрузку, которая хранится в **Binary Table**.

Связь между двумя этими таблицами основана на указанном ключе отношений **<file_binary_1>** в Source-столбце **CustomAction Table**, на основе этого ключа будет найден нужный бинарный файл в **Binary Table** для его запуска из временного .tmp-файла C:\Windows\Installer\MSIXXXX.tmp.

3. **CustomAction3** с помощью пользовательского действия **Custom Action Type 23** извлечет файл Payload.exe из .cab-архива в директорию, инициализированную **CustomAction1.**

Для идентификации того, какие именно файлы необходимо извлечь на данном этапе последовательности, CustomAction3 на основе указанного ключа отношений <file_id1> в Source-столбце **CustomAction Table** обратится за информацией к **File Table**, в которой хранятся имя файла, размер и номер последовательности извлечения этого файла. В свою очередь, **File Table** на основе номера последовательности извлечения файла в столбце Sequence обратиться к таблице **Media Table**, в которой хранятся .cab-архивы, для идентификации нужного .cab-архива, хранящего файл.

4. **CustomAction4** с помощью пользовательского действия **Custom Action Type 1250** выполнит консольную команду по запуску извлеченного файла Payload.exe при выполнении логики **CustomAction3** в директорию, инициализированную **CustomAction1.**

Custom Action Type 1250. Используется для выполнения команд Command-Line во время установки или удаления ПО. Это позволяет разработчикам настраивать процесс установки и удаления и выполнять разные пользовательские действия в зависимости от конкретных условий.

File Table и Media Table:

На основе ключа отношений <file_id1>, указанного в столбце Source, **CustomAction Table** обратится за информацией к **File Table**. В ней хранятся имя файла, размер и номер последовательности извлечения этого файла. В свою очередь, **File Table** на основе номера последовательности извлечения файла в столбце Sequence обратится к таблице **Media Table** для идентификации нужного .cab-архива, хранящего файл (на базе столбца **LastSequence**).

Как это происходит?

Каждая запись в **File Table** представляет собой файл для установки и содержит информацию о его порядке — столбец **Sequence**. Чем меньше значение **Sequence**, тем раньше будет установлен файл.

В **Media Table** содержатся записи о носителях (дисках или .cab-контейнерах) и информация о последовательности файлов (столбец **LastSequence**) на каждом носителе. Также здесь имеется столбец **DiskId**, который является идентификатором носителя.

Теперь разберем пример.

Файл **Payload.exe** имеет **Sequence = 40**. Это означает, что он должен быть установлен 40-м по порядку. Чтобы определить, на каком диске (или в каком .cab-контейнере) находится этот файл, установщик проверяет запись в таблице **Media Table** с наименьшим значением **LastSequence**, которое больше или равно **40**. В нашем случае это **DiskId = 1** с **LastSequence = 80**. Это означает, что на DiskId = 1 (в первом cab1.cab-контейнере) находится файл с **Sequence = 40**, а также все файлы с более ранними **Sequence**-значениями.

Файл **Decoy.pdf** имеет **Sequence = 60**. Установщик снова проверяет записи в таблице **Media Table**. Наименьшее значение **LastSequence**, которое больше или равно 60, — это 80. Значит, **Decoy.pdf** тоже находится на DiskId = 1, потому что все файлы с Sequence = 80 и меньше содержатся на этом носителе.

Предположим, у нас был бы файл **Stager.exe** с **Sequence = 20**. Установщик проверил бы таблицу **Media Table** и увидел, что **Stager.exe** находится на DiskId = 2, потому что там содержатся файлы с **Sequence = 30** и меньше.

Таким образом, Windows Installer использует поля **Sequence** в **File Table** и **LastSequence** в **Media Table** для определения порядка установки файлов и их распределения по .cab-контейнерам на носителях. Это позволяет эффективно организовать процесс установки и в правильном порядке извлекать файлы с носителей.

Custom Action Type 51. Используется для указания директории, в которую будут устанавливаться файлы и другие ресурсы. Во время установки Custom Action Type 51 получает путь к директории из таблицы Directory Table. Данное действие полезно, когда нужно установить файлы в специфичное место на компьютере пользователя (например, в предопределенную системную папку).

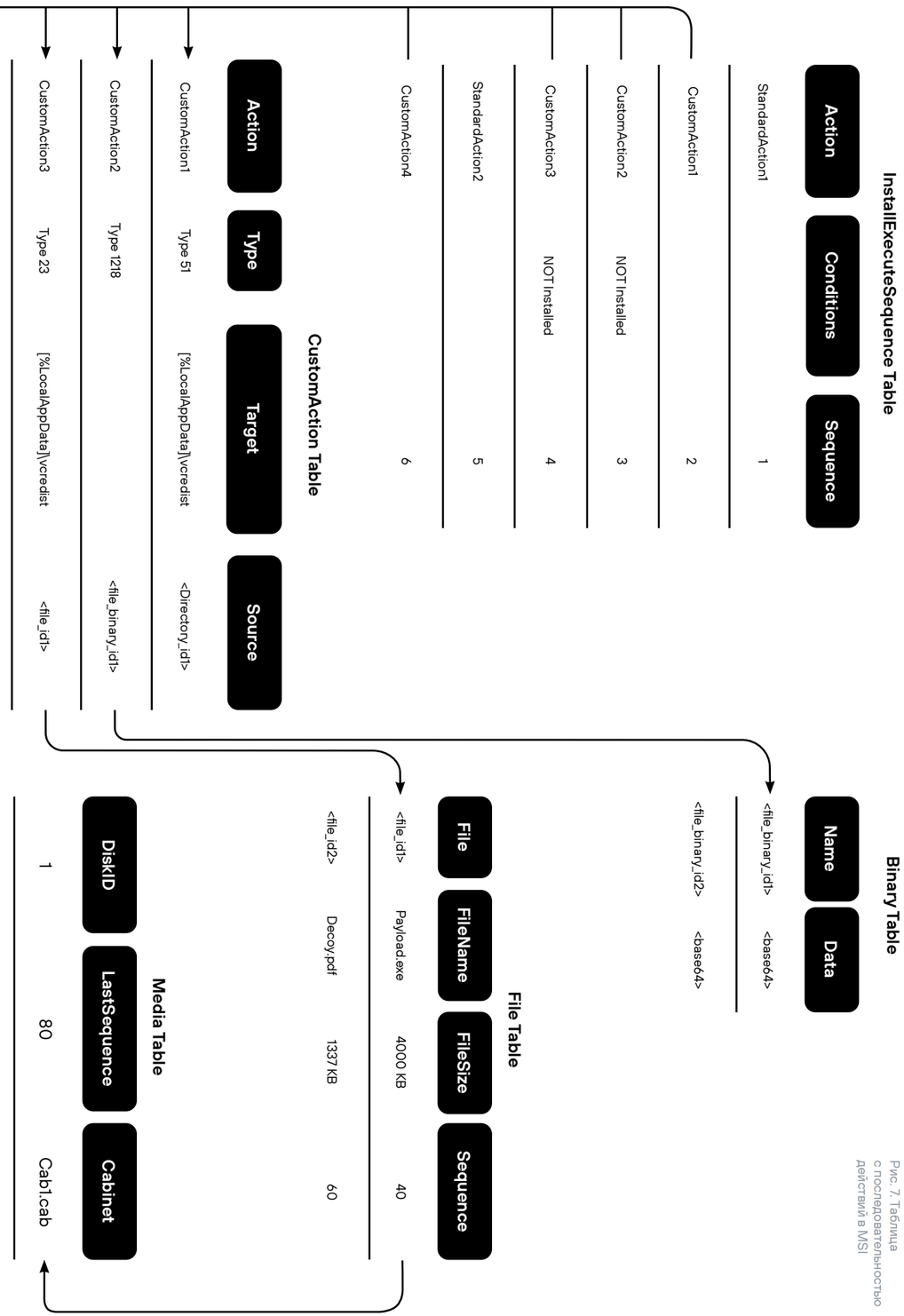
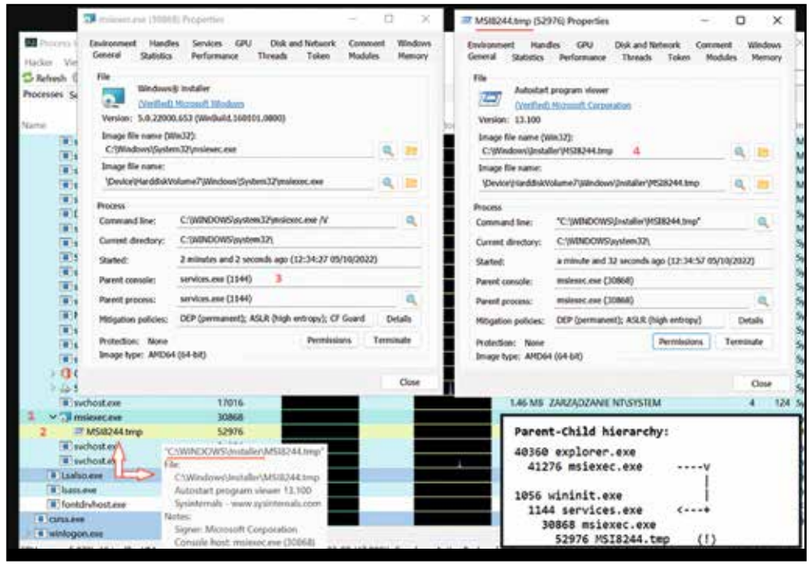


Рис. 7. Таблица с последовательностью действий в MSI

Мариуш Банах хорошо продемонстрировал [25] в своей статье разные сценарии использования MSI на основе Binary-таблиц. Подход подразумевает, что исходный файл кодируется в Base64 и хранится внутри MSI в Binary-таблице. При установке MSI он декодируется обратно и извлекается в исходном виде в указанную директорию. У этого способа есть существенный минус: из-за того, что исходные файлы запускаются сабпроцессом из временного .tmp-файла C:\Windows\Installer\MSIXXXX.tmp, мы теряем к ним доступ.

Рис. 8. Механизм использования MSI
Источник:



Отмечу, что в ходе исследований наша команда обнаружила более универсальный, уже упомянутый способ хранения и извлечения файлов из MSI. Он основан на использовании Media Element и SetDirectory возможностей фреймворка WiX, с помощью которых будут осуществлены сжатие и упаковка файлов с полезной нагрузкой в .cab-контейнер и хранением его в Media Table [26], а с помощью SetDirectory [27] (в терминологии Microsoft называется Custom Action Type 51 [28]) будет задан целевой каталог, в который Custom Action Type 23 извлечет наши файлы из .cab-архива.

Расшифровка опций написанного нашей командой инструмента для генерации установочных файлов MSI с помощью фреймворка WiX:

1. SetDirectory — инициализация директории, в которую будут извлечены упакованные в MSI файлы.
2. DropFiles — список файлов, которые будут сохранены на целевой системе во время работы MSI.
3. ExecCommand — выполнение консольных CMD-команд для реализации логики kill chain.
4. StartupShortcutSrc — указание, для какого файла необходимо создать LNK в папке автозагрузки пользователя (закрепление на хосте).
5. StartupShortcutName — указание имени LNK-файла.
6. ProductName, ProductVersion, Manufacturer — заполнение метаинформации об установщике MSI.

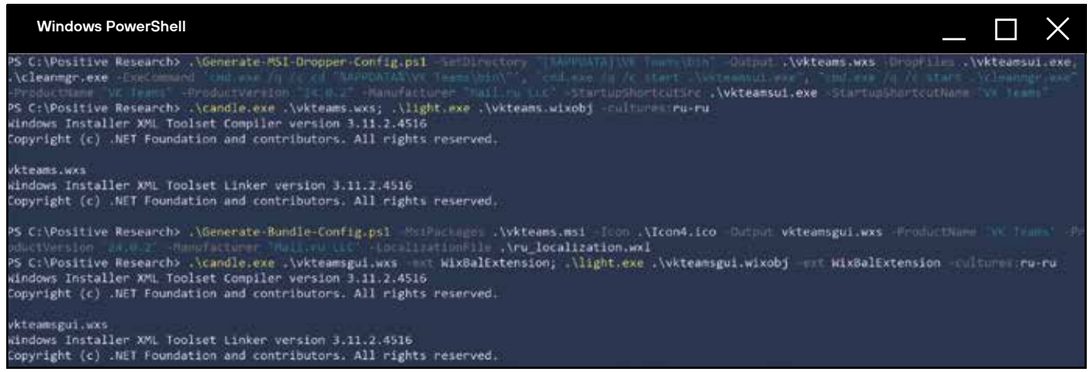


Рис. 9. Использование DropFiles для хранения и извлечения файлов из MSI: компилируем бандл, упаковываем в него MSI и другие необходимые файлы

Рис. 10. Скомпилированный бандл с указанной метаинформацией и знакомой иконкой ПО

На выходе получаем возможность использовать как MSI-установщик, так и графический UI-бандл для имитации установки легитимного ПО. В ходе установки будет реализована заложенная логика созданного нами MSI.

Рис. 11. При запуске файла пользователь увидит знакомый интерфейс установщика

[29]

[30]

В апреле 2021 г., за год до публикации материалов Mrd0x, на Springer появилась исследовательская статья, которая осталась не замеченной в комьюнити. Рекомендую ознакомиться: авторы подробно разбирают концепт BitM-атак и объясняют их отличия от MitB и MitM.

ПОЛУЧЕНИЕ НАЧАЛЬНОГО ДОСТУПА: КЛЮЧЕВЫЕ ТРЕНДЫ

- Внимание к OPSEC: основная полезная нагрузка не будет загружена, если хост не подходит для ее эксплуатации.
- Контейнеризация полезной нагрузки и поиск различных способов обхода MOTW.
- LNK — главная User Execution техника.
- MSI-дропперы универсальны и обладают огромным потенциалом.

НОВЫЕ ВЕКТОРЫ STEAL CREDENTIALS

Атака Browser-in-the-Browser

Исследователь Mrd0x задался вопросом: можно ли снизить эффективность проверки URL-адресов со стороны пользователей? В марте 2022 г. он описал [29] метод атаки Browser-in-the-Browser с имитацией всплывающего окна входа в систему с легитимным URL.

В динамике атака выглядит следующим образом:

1. Отправляем пользователю ссылку на веб-страницу с реализацией BitB-атаки, используя фишинговый домен.
2. Жертва переходит по ссылке и видит кнопку.
3. После нажатия кнопки перед пользователем появляется привычная форма аутентификации в имитируемом браузерном окне. При этом мы можем редактировать отображаемый контент (URL-адрес и др.).
4. Пользователь вводит учетные данные, которые мы логируем на своем сервере.

Реализация:

1. Находим понравившуюся форму компании, клонируем и размещаем на своем веб-сервере.
2. Создаем еще одну аналогичную форму, но убираем из нее поля ввода логина и пароля — оставляем только кнопку «Войти».
3. С помощью JavaScript и iframe создаем функцию — обработчик кнопки, которая загрузит HTML-содержимое формы в пределах текущего окна браузера с возможностью перемещать это «псевдоокно». То есть выполняем загрузку удаленно размещенной страницы в текущей странице и отображаем ее пользователю.

Минус этой атаки в том, что среднестатистический российский пользователь не сталкивался с подобными окнами авторизации и это, скорее всего, его отпугнет. К тому же характерный паттерн JavaScript-функций для вызова таких окон легко детектится статическим анализом HTML-содержимого страницы в песочнице.

Атака Browser-in-the-Middle

Новая, более интересная техника по обходу 2FA, которую также описал [30] Mrd0x. Концептуально BitM-атака не отличается от MitM и MitB: суть в том, чтобы максимально легитимно и незаметно для пользователя вклиниться между его браузером и сервисом.

Основная идея BitM в том, чтобы предоставить жертве инкапсулирующий страницу браузер, который выглядит и ведет себя так же, как легитимный сайт. С помощью своего браузера жертва будет перемещаться по веб-приложению, невольно используя прозрачный вредоносный браузер. Как это реализовать? Механика похожа на облачные сервисы, которые предоставляют удаленный графический доступ к виртуальным машинам в браузере — с помощью VNC.

Архитектура атаки Browser-in-the-Middle:

1. Виртуальная машина на базе одной из NIX ОС.
2. VNC-сервер на виртуальной машине (TightVNC или TigerVNC).
3. ПО для предоставления графического доступа к VNC-серверу в браузере. Например, noVNC, Apache Guacamole, TeamViewer или Chrome Remote Desktop.

Так как жертва попадает во вредоносный браузер в режиме киоска с открытой целевой страницей на хосте злоумышленника, у него есть масса вариантов апгрейда атаки и способов логирования на стороне сервера.

Возможности атакующего:

1) предустановка браузерных расширений для перехвата, модификации и логирования запросов,

2) предустановка любого другого ПО,

3) экспорт профиля браузера,

4) Keylogger,

5) доступ к активной сессии атакуемого.

Рис. 12. Варианты развития BItM-атаки

Однако у этой концепции есть много сырых моментов:

1. **Ужасная операционная безопасность.** Во время атаки мы предоставляем доступ к своему хосту, поэтому нужно настроить отправку залогированных данных на удаленный хост, чтобы не хранить их на BitM-платформе.
2. **Kiosk bypass (Citrix style)**: пользователь может выйти из режима киоска и получить полноценный доступ к хосту. Нужно настраивать пользователей без привилегированных прав, из-под которых браузер будет запускаться в режиме киоска.
3. **Не работает ПКМ, и есть проблемы с буфером обмена.** Это может затруднить ввод сложных паролей и вызвать подозрение у пользователей. Например, из-за невозможности вставить пароль из KeePass.
4. **URL всегда статичен** и не меняется.
5. **Кнопка «Назад» изменит навигацию** в реальном браузере, а не в браузере на BitM-платформе.
6. Концепция пока **не работает на мобильных устройствах.**

КЛЮЧЕВЫЕ ТРЕНДЫ

1.

Классический фишинг эффективен и важен для развития атаки, перемещения внутри сети и получения привилегированных доступов без лишнего шума.

2.

Прогресс в части frontend и JavaScript технологий порождает прогресс фишинговых техник.

3. Новые мощные концепции по обходу 2FA и фишингу.

КАК ХАКЕРЫ ПОВЫШАЮТ ЭФФЕКТИВНОСТЬ KILL CHAIN

› **Служебные заголовки писем.** Они могут многое рассказать об используемом стеке почтовых СЗИ и архитектуре компании. Эта информация вкупе со знанием типичных болячек конкретного почтового решения помогает в подготовке и выборе правильных TTP для реализации kill chain.

› **Спуфинг почтового домена.** Благодаря наличию эффективных зарубежных решений на почтовых шлюзах, многие забыли, что спуфинг почтового корпоративного домена в принципе существует и от него нужно уметь защищаться. С уходом зарубежных вендоров стало понятно, что необходимы дополнительные комплексные меры для противодействия подобным угрозам.

› **Корпоративно-житейский сценарий.** В целевых фишинговых атаках используются сценарии, которые мимикрируют под обычную корпоративную активность. К ней у пользователей по умолчанию должно быть больше доверия, нежели к сценариям про корпоративные скидки или каким-то ультимативно резким требованиям.

› **Telegram.** Использование мессенджеров и прямой контакт с атакуемым пользователем заметно увеличивают результативность kill chain.

› **Вишинг в эпоху корпоративных мессенджеров.** Наша практика показывает, что если позвонить и объяснить сотруднику, что сейчас ему в Telegram или на почту придет сообщение, он будет считать его более легитимным.

› **Хостинг полезной нагрузки на взломанных внешних ресурсах.** Не во всех организациях у сотрудников есть свободный доступ в интернет. Некоторые используют белый список для явного предоставления доступа к внешним корпоративным ресурсам, а черные списки — для блокировки потенциально опасных ресурсов. В случае компрометации одного из сервисов, входящих в whitelist, или при выборе ресурса, не попадающего в blacklist, можно хранить полезную нагрузку там и рассылать пользователям соответствующие ссылки, обходя установленные ограничения

› **Spearphishing via Service.** Если атакующие скомпрометировали корпоративный ресурс, который посещают много пользователей, с помощью встраивания своего JavaScript-кода в исходный код приложения они могут разместить там сценарий-плашку. Например, о необходимости обновиться перед использованием ресурса. Пользователь обновляется, плашка исчезает, а атакующие получают пробив, то есть доступ к машине пользователя.

Самые интересные kill chains 2023 г.

О BadMagic APT я уже упоминал. Он гарантирует, что LNK будет эксплуатироваться на всех Windows-системах, а нужный набор консольных команд будет выполнен. При этом LNK позволяет использовать множество LOLBin-техник на стадии User Execution, а MSI — упаковывать в себя необходимые файлы для установления обратного соединения с командным сервером. Атаку можно адаптировать под разные сценарии, поскольку LNK-файлы могут мимикрировать под легитимные документы и приложения. Это базовый и достаточно универсальный kill chain с высоким уровнем OPSEC для полезной нагрузки.

Второй эффективный сценарий, который все чаще реализуют APT-группировки, — kill chain с использованием ссылок, техники HTML Smuggling, LNK-файлов и техники DLL Side-Loading. Хакеры используют PDF-файлы со ссылками на Dropbox, Google Drive или ресурсы вроде WordPress. На них хостятся файлы для HTML Smuggling, которые дропают ISO-контейнер, содержащий LNK-файл. Он выполняет всего одну команду — запускает бинарный файл, который реализует логику для отработки техники DLL Side-Loading.

ВЫВОД

Во время подготовки к выступлению 31 на Positive Hack Days я получил сообщение от коллеги: «Мне кажется, накатать фишинг и сделать рассылку — особых навыков не требует...» Он искренне интересовался, зачем мы ищем экспертов и собираем целую SE-команду. Думаю, после этой статьи все стало немного понятнее :)

Мы ожидаем, что в 2025 г. фишинговые атаки будут гораздо более успешными, поскольку организации в массовом порядке перейдут с запрещенных ИБ-продуктов на доступные корпоративные решения для защиты электронной почты. У российских поставщиков средств защиты электронной почты есть около полутора лет, чтобы перейти от концепции «наш продукт делает электронную почту на N% безопаснее и пропускает на N% меньше вредоносных программ» к утверждению «наш продукт обеспечивает безопасность корпоративной электронной почты, точка». Необходимо изменить подход к разработке продуктов таких классов, как Sandbox, Email Security Gateway и Web Security Gateway.

31

БИЗНЕС-ТРЕК PHDAYS

НАША ДРУЖБА С НЕЙРОСЕТЯМИ НЕ ЗАКАНЧИВАЕТСЯ НА МАЯКОВСКОМ. ЧИТАЙТЕ СТЕНОГРАММЫ БИЗНЕС-ТРЕКА PHDAYS, РАСШИФРОВАННЫЕ И ОБРАБОТАННЫЕ НЕЙРОСЕТЬЮ.

«Мы часто читаем в недружественных телеграм-каналах о том, что взломана компания X, украдено все, что можно украсть. Звоним ее руководству: «Ребята, что у вас там происходит?» — «Ничего. Нам ничего не сообщали». Начинаешь вникать, и выясняется, что взломали не головную организацию, где здоровая атмосфера с точки зрения кибербезопасности, а дочернюю. А там такая культура, что людей много лет «драли» за каждый залет. Поэтому наверх сообщают только хорошие новости и скрывают все остальное.

Да, наши специалисты или коллеги из других кибербез-компаний могут расследовать такие инциденты и поделиться деталями в закрытом комьюнити. Но что происходит на самом деле: на место для расследования никто не выезжает (не зовут), информацию напрямую не получает, сисадмины в компании трут все доказательства того, что инцидент был. Прямо стирают, стирают, стирают... Для чего? Чтобы не прилетело.

Я думаю о необходимости своеобразной индульгенции для безопасников: ребята, давайте создавать такую культуру, в которой мы не будем наказывать людей за то, что они где-то что-то недосмотрели. Они воспитывались и учились в условиях мирного времени, но мир поменялся. Давайте уже все простим и начнем стимулировать диалог между бизнесом и кибербезом.

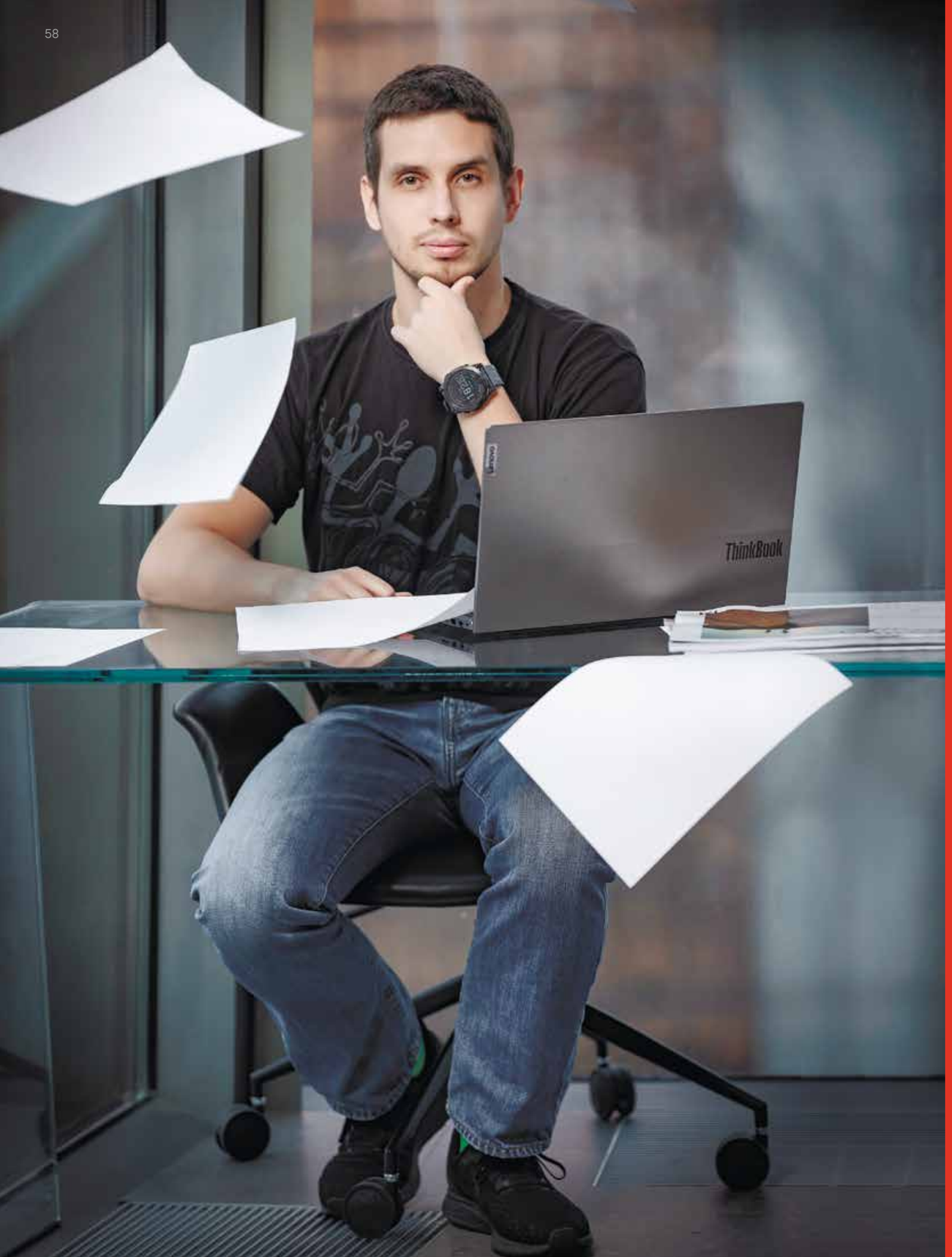
Денис Баранов

Генеральный директор Positive Technologies

«При подготовке Указа № 250 мы думали, что, когда назначим ответственного за кибербез, это будет очень крутая история. У нас возникнет центр ответственности! А на самом деле случилась парадоксальная ситуация: когда мы создали этот институт, прозрачности стало меньше — как раз потому, что теперь появился ответственный за безопасность. Если возникает инцидент, все сходится на нем, а он же не может признаться, что плохо справляется со своей работой. Раньше ИТ-шники бежали и говорили: «Все сломалось». А сейчас все упирается в офицера по кибербезу. Он говорит: «Слушайте, да это не такой масштаб, все нормально, мы переживем, надо еще посмотреть». Нужно решать эту проблему. Это вопрос культуры: нужен ответственный за безопасность, который сможет признаться, что мы все потеряли.

Максут Шадаев

Министр цифрового развития, связи и массовых коммуникаций РФ



INCIDENT RESPONSE НА УДАЛЕНКЕ:

КАК ВЫЗОВЫ COVID-19 ПРЕВРАТИЛИСЬ В НОВЫЕ ПРАКТИКИ

Александр Репин

Старший специалист отдела расследования
и реагирования на угрозы ИБ Positive Technologies

Время прочтения:	10 минут
Для кого:	CISO, эксперты SOC
Прокачиваем знания:	реагирование на инциденты

Мы — отдел реагирования на угрозы информационной безопасности, и наша работа — помогать заказчикам расследовать компьютерные инциденты и реагировать на них. К нам за помощью обращаются самые разные компании, и уровень зрелости ИБ у них может быть абсолютно любой: от крупной инфраструктуры с собственным SOC и штатом обученных специалистов по ИБ до обычной локалки без файервола или антивируса, где роль безопасников в свободное время исполняют сисадмины.

ONCE UPON A TIME

В те далекие времена, когда COVID-19 еще не отправил всех на удаленку, процесс реагирования на инцидент у заказчика мог выглядеть так:

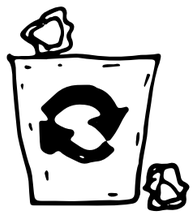
1. Веселая толпа выезжает на совещание, а порой — и в командировку.
2. Общается с заказчиком, качает головой и говорит: «разберемся».
3. Наши эксперты в компании с технарями заказчика собирают релевантную информацию об инциденте, данные со средств защиты и затронутых систем.
4. Изучают данные на месте или везут их в офис и разбираются там.

Процесс сбора и анализа непосредственно в инфраструктуре заказчика продолжался до тех пор, пока инцидент не считался полностью исчерпанным. В некоторых случаях, особенно если инцидент был сложным, а расследование — долгим, заказчик соглашался дать удаленный доступ.

НО ВОТ НАЧИНАЕТСЯ ПАНДЕМИЯ...

Инциденты продолжают происходить, их даже становится больше: компании поспешно переходят на удаленную работу, но не у всех получается безопасно организовать удаленный доступ. А еще хакеры начинают активно использовать тему COVID-19 для рассылки фишинга.

Мы теперь не можем просто так взять и поехать на площадку заказчика для реагирования на инцидент. В моменты жесткого локдауна для этого нужно было получать разрешение на перемещение по городу, а у самих заказчиков могли быть строгие правила ПЦР-тестирования для тех, кто находится на объектах. Понятно, что такие ограничения осложняют работу с инцидентами, которые сами по себе плохо прогнозируются

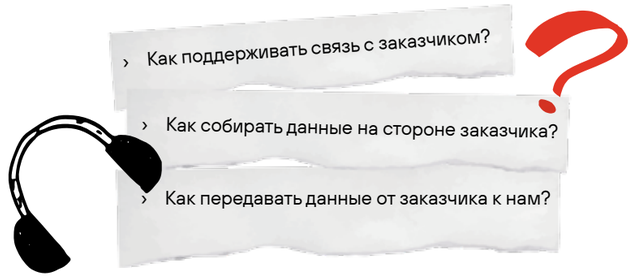


и носят случайный характер. Даже тот факт, что компании стали массово внедрять удаленный доступ, нам помогал слабо. Для некоторых заказчиков открыть его для подрядчика — это целый квест, который требует множества согласований и может занимать неделю или две.

Но были в этой ситуации и плюсы для нас. В новой реальности заказчики в принципе более охотно начали соглашаться на удаленное реагирование. Потому что других вариантов у них, собственно, не было.

ДАНО: ЗАКАЗЧИК, УДАЛЕНКА, ИНЦИДЕНТ

Мы столкнулись с тремя основными проблемами:



Первая проблема оказалась не такой уж проблемой. Довольно быстро компании выбрали себе те или иные решения для видео-конференц-связи, никуда не делась электронная почта, а оперативные вопросы можно было решать в мессенджерах.

Для решения второй и третьей проблем мы рассмотрели несколько вариантов и выбрали подходящие исходя из нашего понимания того, как можно было организовать наиболее эффективное взаимодействие с заказчиком на удаленке.

По опыту мы знали, что технические специалисты у заказчика бывают самого разного уровня: от толковых ребят, которые могут разобрать и собрать любое ПО, до таких, которые не могут утилиту командной строки запустить без ошибок. Также было понятно, что в инфраструктуре могут присутствовать разные платформы, однако в первую очередь мы ориентировались на разные версии Windows, в том числе устаревшие.

Мы сформулировали для себя минимальный набор требований к используемым инструментам. Они должны были:

- запускаться без сложного конфигурирования и длинных командных строк;
- оставлять минимальный след в операционной системе;
- работать на максимально широком спектре операционных систем.

Отмечу, что определенные наработки у нас были и до пандемии, но именно она заставила обратить на эти проблемы больше внимания и форсировать разработку.

СБОР ДАННЫХ ТОЧЕЧНО

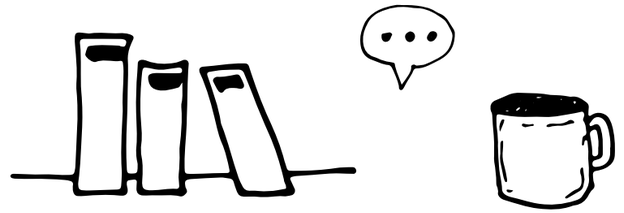
Итак, какие у нас были варианты сбора данных?

Для начала можно выгрузить данные из средств защиты, таких как антивирусы, файерволы, IDS, EDR и прочие. Для этого никакие специальные средства обычно не требуются, все делается через консоли соответствующих инструментов.

Если нужно собрать данные непосредственно с хостов, то есть несколько способов. Первый — классический: снять образ диска. Плюсы такого подхода: возможность получить все данные, которые были на жестком диске, в том числе удаленные или находящиеся в неразмеченных областях. Минусы: образы диска занимают много места и не содержат волатильных данных (то есть тех, которые есть в системе во время ее работы: информация о сетевых соединениях, запущенных процессах и т. п.).

Второй похож на первый: можно выгрузить виртуальный диск соответствующего хоста из системы виртуализации (конечно, если это виртуальный хост). Плюсы и минусы в целом такие же, как у первого способа.

Третий способ набрал популярность в последние годы; он называется live response. Суть такова: специальная утилита собирает с работающей системы ограниченный набор файлов, необходимый для проведения



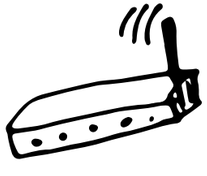
расследования, а также волатильную информацию. Плюсы такого подхода: относительно маленький объем данных и наличие информации с работающей системы (зачастую это более показательно, чем «мертвые» данные: например, можно сразу увидеть подозрительные процессы или сетевые соединения). Минусы: ограниченность собираемого набора может привести к тому, что какие-то интересные файлы собраны не будут и их придется запрашивать дополнительно. Кроме того, полученные таким образом данные сложно будет использовать в качестве доказательной базы, если проводится расследование в рамках уголовного дела.

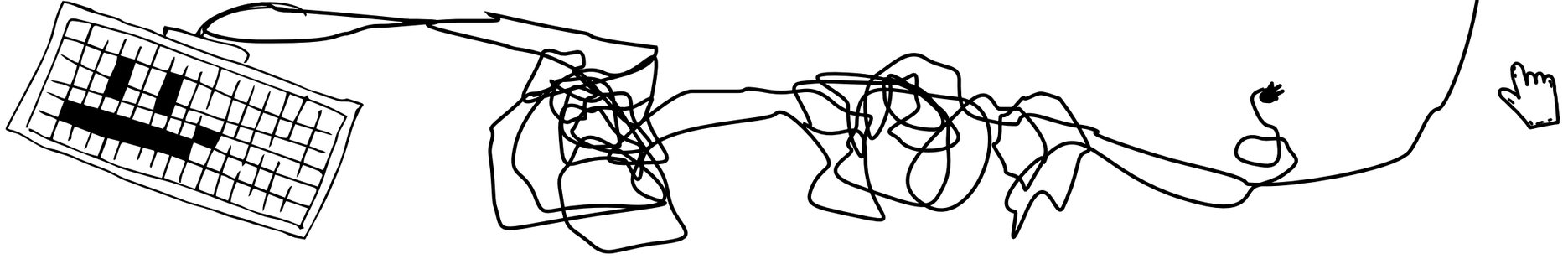
Мы остановились на третьем варианте. Изначально мы проанализировали доступные на рынке инструменты и поняли, что они не удовлетворяют нашим требованиям. Решили, что нужно разрабатывать свое средство.

В качестве языка программирования выбрали Go:

- он довольно прост для изучения и позволяет эффективно писать программы;
- позволяет статически компилировать все зависимости в один файл;
- позволяет компилировать программу под разные платформы и операционные системы;
- поддерживает Windows аж в версии XP (и такое в кейсах случается).

«КОСТЫЛИ», ПРИЗВАННЫЕ МИНИМИЗИРОВАТЬ БОЛЬ ОТ ОГРАНИЧЕНИЙ COVID-19, СТАЛИ НЕОТЪЕМЛЕМОЙ ЧАСТЬЮ НАШИХ ПРАКТИК





Так возникла утилита pt-dumper. Что же она делает? В самом простом случае она запускается двойным кликом и собирает с системы информацию, которую мы считаем необходимой для расследования. Набор данных не удивит тех, кто работает в этой области: системные файлы, такие как $MFT и файлы реестра, файлы различных forensic-артефактов, системные журналы и т. д. Кроме того, мы добавили возможность сбора любых файлов по регулярным выражениям. Это полезно: в процессе расследования мы можем обнаружить, что хакер обычно хранит свои файлы в какой-то директории или использует утилиты с какими-то конкретными именами. Конечно, собирается и волатильная информация: список запущенных процессов, список сетевых соединений, содержимое кэша DNS и т. д.

Конфигурирование осуществляется на нашей стороне на этапе сборки утилиты. Заказчику остается только запустить ее и передать нам на анализ получившийся архив с результатами.

Со временем мы добавили в утилиту новые фишки. Например, поддержку снапшотов Shadow Copy. Часть артефактов теперь парсится непосредственно на этапе сбора и попадает в собранный архив уже в текстовом виде, что облегчает жизнь эксперту. Мы также внедрили ряд эвристик, которые позволяют выявлять в системе подозрительные файлы и копировать их для дальнейшего анализа. Например, если есть признаки того, что файлу меняли временные метки, или если вдруг у файла нет валидной цифровой подписи.

Еще мы разработали версии pt-dumper для Linux и macOS. Правда, с версией для Windows их объединяет в основном название, так как по своей архитектуре эти операционные системы сильно отличаются и набор данных там совершенно другой. Впрочем, концепция осталась той же самой.

СБОР ДАННЫХ МАССОВО

Сбор данных с отдельных хостов — это хорошо, но иногда нужно просканировать все хосты в сети заказчика. Например, если у него в инфраструктуре нет никаких средств защиты, то один из немногих вариантов — поиск информации непосредственно на хостах. Или, допустим, в процессе расследования мы уже собрали список индикаторов компрометации и хотим поискать эти индикаторы на всех машинах, чтобы выяснить масштаб атаки.

Также это полезно в том случае, если атака на инфраструктуру продолжается длительное время и различные журналы, по которым можно было бы восстановить последовательность компрометации систем, уже ротировались. Тогда собранные данные позволяют построить таймлайн атаки и сделать выводы об исходном векторе.

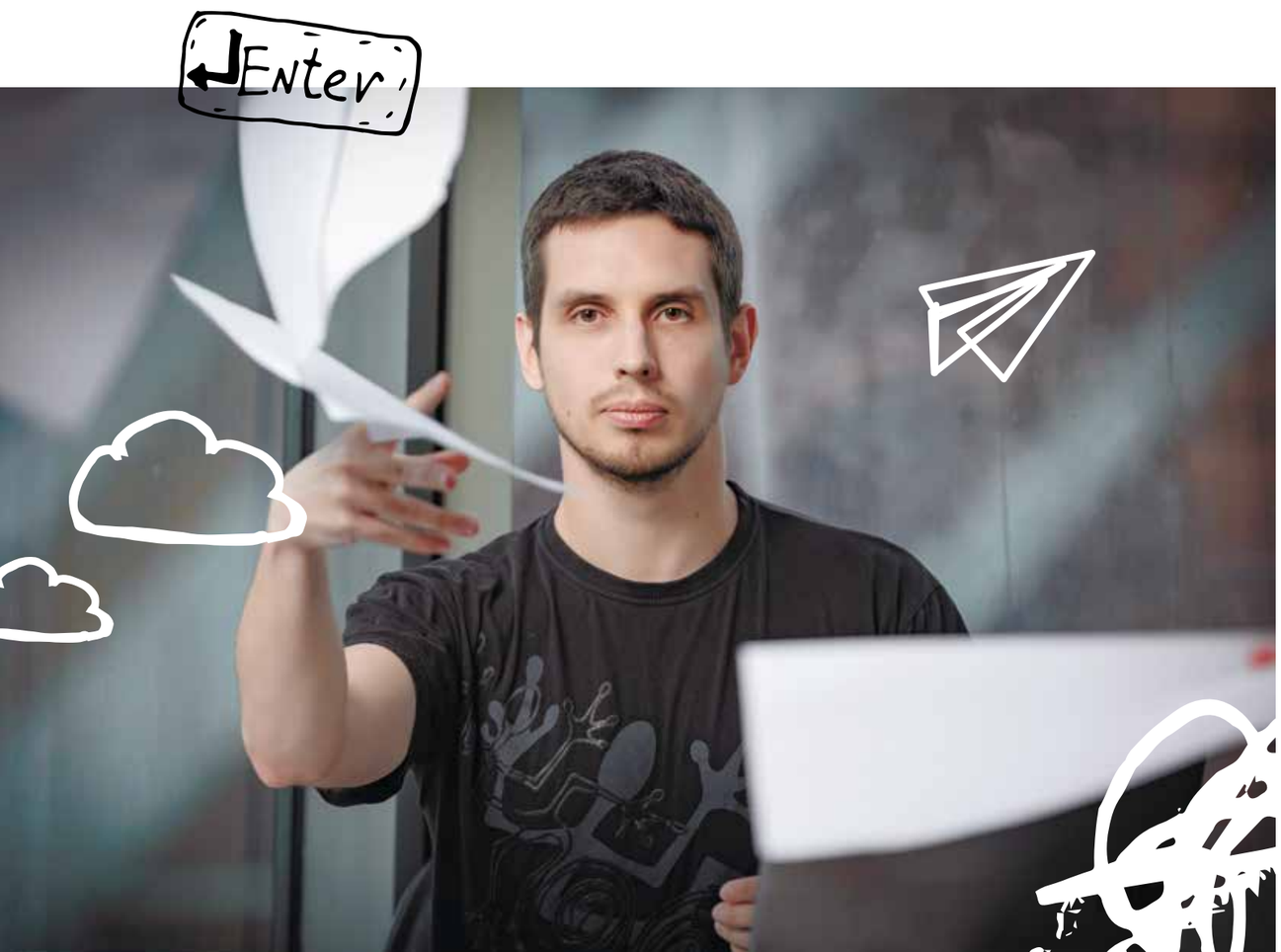
Отдельно стоит задача так называемого ретроспективного анализа. Она больше похожа не на реагирование, а на threat hunting, когда у тебя нет никаких исходных данных и нужно найти подозрительную активность или аномалии. Наличие данных с большого количества машин помогает решать эту задачу.

Для массового сканирования инфраструктуры мы создали утилиту pt-scanner. Она состоит из двух компонентов — сервера и клиента. Сервер запускается на специально выделенной машине в инфраструктуре заказчика и ждет информацию от клиентов. Клиенты запускаются на сканируемых машинах и собирают необходимую хостовую информацию, затем отправляют ее на сервер. Для взаимодействия необходимо обеспечить связность между клиентами и сервером по 80-му порту. В остальном и клиент, и сервер запускаются так же просто, как и pt-dumper, — даблкликом и без параметров. При этом вопрос распространения клиентской части заказчик решает самостоятельно и использует те средства удаленного запуска, которые ему удобны. Если заказчик не знает, как это сделать, мы можем предложить ему проверенные варианты (PsExec, доменные политики, SCCM, KSC и т. д.).

Набор собираемых данных в целом похож на тот, который собирается с помощью pt-dumper, но не включает в себя тяжеловесные данные. При этом в клиенте есть функциональность проверки процессов и файлов по хешам, путям и с помощью YARA-правил.

Сервер просто получает данные от клиентов и складывает их в архивы. Фактически никакого другого взаимодействия между клиентом и сервером не происходит. Эта схема эффективно и надежно работает в том случае, когда все задачи по запуску сканирования выполняет сам заказчик. Но когда в инфраструктуре работаем мы, функциональность иногда оказывается недостаточной. Еще до пандемии COVID-19 мы начали разработку интерактивного аналога pt-scanner и назвали его pt-responder.

В клиентскую часть добавили возможность закрепления на сканируемых системах (на случай перезагрузок), а в сервер встроили веб-интерфейс для управления работой агентов. Но самые большие изменения произошли в логике работы агента: теперь сырые данные обрабатывались прямо на хосте и отправлялись на сервер в формате JSON. Для этого мы написали парсеры для практически всех интересующих нас forensic-артефактов, а также сборщики различных волатильных данных. А чтобы можно было использовать pt-responder как pt-scanner, то есть для неинтерактивного сбора информации, мы реализовали механизм пресетов.



Впрочем, практика показала, что pt-scanner мы все равно используем гораздо чаще, чем pt-responder. Но наработки pt-responder не пропали даром: разработанные парсеры мы внедрили как в pt-scanner, так и в pt-dumper. За счет этого удалось ускорить процесс анализа: часть данных в собранном архиве теперь приходят уже в разобранном виде, и можно не тратить время на запуск утилит для парсинга артефактов.

ПЕРЕДАЧА ДАННЫХ

Итак, заказчик собрал данные, теперь ему нужно как-то передать их нам на анализ.

До эпидемии COVID-19 можно было съездить к заказчику и забрать все на жестком диске. Недостатки очевидны: надо тратить время на дорогу и копирование данных. В реалиях локдаунов такой метод вообще малоприменим и остается на крайний случай.

Еще можно копировать данные непосредственно по VPN/RDP, если заказчик предоставляет такую возможность. Часто такие каналы не отличаются стабильностью и скоростью работы, поэтому передавать большие объемы данных проблематично.

Можно использовать облака или собственные файлообменники. Такой файлообменник есть и у нас, иногда мы используем его для работы с заказчиками, но неудобное управление доступами, нестабильная закачка и постоянные превышения квоты также не позволяют назвать такое решение идеальным.

Поэтому мы решили изобрести свой велосипед и назвали его pt-cloud. На самом деле это не облачное решение, как можно было бы решить по названию. Заказчику мы отдаем клиентскую часть приложения, она содержит всего две кнопки: «Выбрать файл» и «Отправить файл». Ошибиться невозможно.

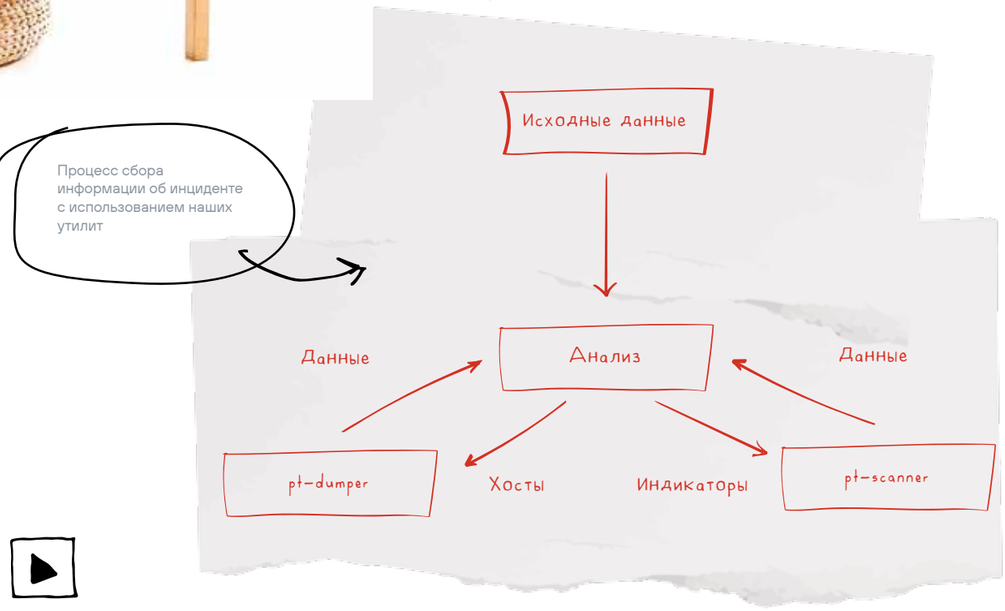
Серверная часть находится у нас в инфраструктуре. Это позволяет аналитикам быстрее получать доступ к данным заказчика. И если изначально скачивать данные с сервера приходилось с помощью чего-то вроде scp, то теперь у нас есть веб-интерфейс, который позволяет не только скачивать файлы по ссылке, но и управлять доступами, а также видеть, какой заказчик и какие файлы загружает, в режиме реального времени.

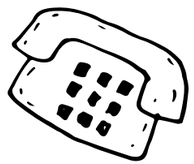
Мы периодически реализуем в инструменте новые фишки. Например, сейчас интегрируем его в пайплайн для автоматизированной обработки данных, поступающих от заказчика.

НОВЫЕ ПРАКТИКИ

Теперь процесс реагирования стал выглядеть так.

Заказчик передает нам исходную информацию об инциденте: как обнаружили, когда, какие системы затронуты и т. д. На основании этих данных мы просим заказчика снять дампы с тех или иных хостов. В процессе анализа мы выявляем новые интересные хосты и запрашиваем с них дампы, а также смотрим, какие техники и инструменты использует злоумышленник, с какими хостами в интернете он взаимодействует. Таким образом накапливаем индикаторы компрометации. В какой-то момент мы понимаем, что тактика злоумышленника в целом ясна и индикаторов уже достаточно для того, чтобы провести сканирование всей инфраструктуры и, возможно, выявить новые затронутые хосты, новые инструменты, а также пополнить коллекцию индикаторов. Этот процесс продолжается итеративно до тех пор, пока инцидент не будет полностью расследован либо пока не будут исчерпаны все возможности для его расследования.



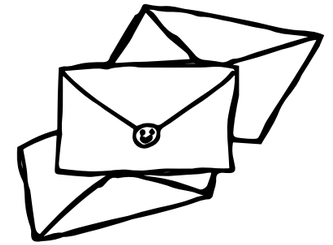


Почему же «костыли», призванные минимизировать боль от ограничений, которые наложила пандемия COVID-19, стали неотъемлемой частью наших практик?

Оказалось, что это просто удобно. Если к нам обращается заказчик с просьбой помочь расследовать инцидент, мы передаем ему наши утилиты для сбора и передачи данных и говорим, с каких систем необходимо собрать информацию. При этом ему не нужно думать, как запускать эти утилиты, потому что мы специально сделали их простыми, а экспертная часть закладывается с нашей стороны. Заказчик отправляет нам данные по мере сбора, а мы их анализируем.

Нам такая схема вполне подходит, потому что не нужно никуда ехать, а значит, специалисты могут работать в комфортных условиях и с максимальной отдачей, а не тратить время на поездки в офис к заказчику. К тому же можно параллельно выполнять несколько задач.

Наши утилиты особенно удобны, когда инцидент происходит в нерабочее время. Эксперту гораздо проще подключиться к расследованию удаленно, чем ехать куда-то среди ночи или в выходные.



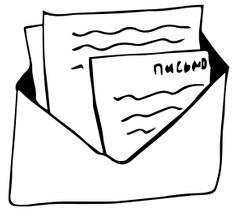
Плюсы для заказчиков:

Не нужно пускать посторонних людей в свою инфраструктуру. Можем проиллюстрировать градус недоверия на собственном примере. Positive Technologies является известным вендором в области ИБ, то есть заказчики вроде бы должны доверять нам, но тем не менее мы регулярно обнаруживаем, что они загружают наши утилиты на VirusTotal.

Возможность быстро получить значимые результаты. Если заказчик готов быстро собирать и передавать данные, то мы так же быстро можем их анализировать и давать результат. Это означает, что в течение пары часов после передачи наших утилит заказчик уже может получать от нас информацию, как именно развивается инцидент, каким мог быть исходный вектор, а также различные индикаторы компрометации, которые можно загрузить в средства защиты информации для улучшения видимости и локализации инцидента.

ВЫВОДЫ

Иногда сложные условия стимулируют поиск новых решений. То, что начиналось как борьба с ковидными ограничениями в работе, в результате итеративных улучшений превратилось в эффективные подходы к реагированию на инциденты. Мы продолжаем пользоваться ими и сейчас. Так что, видимо, нет худа без добра :)



ЧТО ЕЩЕ ПОЧИТАТЬ:

Включи VPN ;)

Security White Papers

Awesome Forensics

Incident response overview

Awesome Incident Response

СИЛА В СООБЩЕСТВЕ: ФРЕЙМВОРК ERM&СК

Антон Кутепов

Руководитель направления развития инициатив сообществ ИБ Positive Technologies

Андрей Сикорский

Руководитель направления развития экспертизы CyberOK

Время прочтения:	10 минут
Для кого:	ИБ-специалисты, ИТ-специалисты, сотрудники SOC
Прокачиваем знания:	реагирование на инциденты

На Positive Hack Days 12 мы вместе с коллегами из CyberOK представили фреймворк реагирования на ИБ-инциденты ERM&CK (читается «Ермак»). Главная особенность ERM&CK в том, он дает пользователям возможность не только описывать абстрактные рекомендации, но и создавать конкретные инструкции по реагированию. Стоит отметить, что это не только фреймворк, но и база знаний, которая наполняется сообществом. Мы опубликовали проект в Security Experts Community [1], он является практическим примером реализации подхода экспертной открытости. В рамках сообщества разрабатывается открытая база знаний по ИБ, и ERM&CK станет одной из ее составляющих. Сегодня мы подробнее расскажем об истории проекта и поделимся планами по его развитию.

РАЗНИЦА МЕЖДУ РЕАГИРОВАНИЕМ И РАССЛЕДОВАНИЕМ

Многие считают, что формализовать процесс реагирования на инциденты невозможно: «Ситуаций много, они непредсказуемы и разнообразны, поэтому их нельзя заранее описать и заложить в модель». Проблема в том, что даже эксперты зачастую путают оперативное реагирование и техническое расследование.

Расследование подразумевает тщательный сбор цифровых улик, глубокий анализ всех вовлеченных систем, восстановление полной последовательности действий атакующих, определение точек закрепления в инфраструктуре и анализ данных киберразведки. Это сложный творческий процесс, который может длиться месяцами, и на данный момент его действительно трудно полноценно описать в виде модели.

Переходим к первичному реагированию. В данном случае цель — локализовать текущий инцидент и не позволить хакеру продвинуться дальше. Для этого не нужно строить большие отчеты и долго восстанавливать все детали атаки — достаточно понять, где сейчас обитает злоумышленник, и оперативно принять соответствующие меры защиты. После принятия первичных мер по нейтрализации угрозы желательно провести комплексное расследование и полный ретроспективный анализ инфраструктуры.

Верхнеуровнево процесс реагирования может включать следующие действия:

- введение превентивных защитных мер и улучшение подхода к менеджменту ИБ-инцидентов;
- оценка инцидентов;
- активация защитных мер для локализации инцидента и смягчения его последствий;
- анализ и извлечение уроков из инцидентов.

С 2022 г. вопрос эффективного реагирования на инциденты стал особенно актуален для отечественных компаний. Существует масса общепринятых стандартов (например, ГОСТ Р ИСО/МЭК ТО 18044-2007, NIST Cybersecurity Framework или Computer Security Incident Handling Guide) и абстрактных рекомендаций, но на практике, когда инцидент уже произошел и нужно что-то делать, этого недостаточно. В открытом доступе нет четких инструкций для конкретных кейсов: какие действия и в каком порядке предпринимать, какие инструменты использовать и т. д. Проще говоря, рынку не хватает открытого инструмента для реагирования на инциденты. Точнее, не хватало...

ИСТОРИЯ ERM&CK

Идея создания ERM&CK [2] (Enterprise Response Model &Common Knowledge) родилась еще в 2021 г. Тогда злоумышленники пробили периметр одной российской компании с помощью популярной в то время связки уязвимостей под общим названием ProxyLogon (CVE-2021-26855, CVE-2021-26858, CVE-2021-27065). Она позволяла обойти механизм аутентификации Microsoft Exchange и загрузить веб-шелл на сервер.

Выполнением действий реагирования занималась ИТ-служба атакованной компании, а не ИБ-специалисты (как правило, в небольших организациях это именно так). У них не было понимания того, что нужно делать и в каком порядке: каких пользователей блокировать, когда отключать сеть, как в целом устранять уязвимость. Мы осознали, что когда реагированием занимаются ИТ-шники, им нужно предоставлять четкую инструкцию — фреймворк. Мы помогли коллегам разобраться с проблемой, а в 2022 г. стартовала разработка ERM&CK.

Основные цели проекта звучали так:

- разработать удобный инструмент для подготовки инфраструктуры к процессам реагирования на ИБ-инциденты;
- предоставить пользователям четкие инструкции для описанных в базе знаний кейсов;
- автоматизировать построение сценариев реагирования и реализовать возможности для анализа полученных данных.

За основу решения мы взяли ATC RE&CT [3] — открытый проект по описанию домена реагирования. При этом мы изменили его архитектуру и функционал, а также полностью переписали кодовую базу.

Мы расширили исходную модель данных (рис. 1) новыми сущностями (рис. 2) и добавили к каждому абстрактному действию его реализацию. К примеру, в RE&CT есть абстрактное действие «Блокировка учетной записи». В ERM&CK же прописано, как реализовать его в конкретной системе с помощью определенных действий.

ERM&CK состоит из следующих сущностей:

- профиль инфраструктуры;
- угроза (процедура);
- сценарий реагирования;
- действие реагирования;
- ПО;
- реализация действия реагирования;
- ресурс.

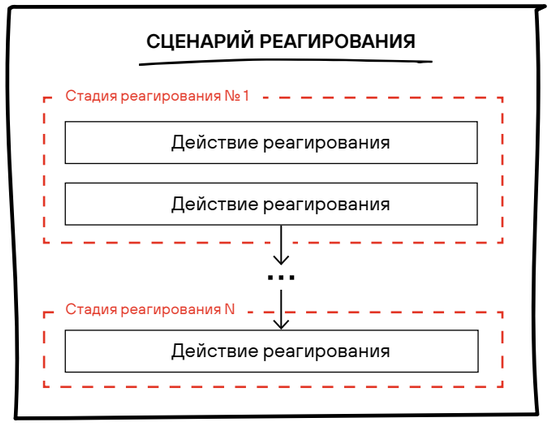


Рис. 1. Схема сущностей RE&CT

Рис. 2. Схема сущностей ERM&CK

Другой важный момент: действия в RE&CT представлены в виде плоского списка и не объединены причинно-следственными связями. Когда аналитик смотрит в сценарий, он не всегда понимает, почему их нужно выполнять именно в таком порядке. Согласно концепции ERM&CK, у всех действий должны быть пререквизиты и результаты — благодаря этому они складываются в логичные цепочки (результат одного становится пререквизитом другого).

Наш проект содержит перечень исследованных угроз, и для каждой (например, см. рис. 3) предусмотрен отдельный сценарий (рис. 4) с инструкцией по устранению: набор абстрактных действий реагирования и их конкретные реализации со списком подходящего ПО.

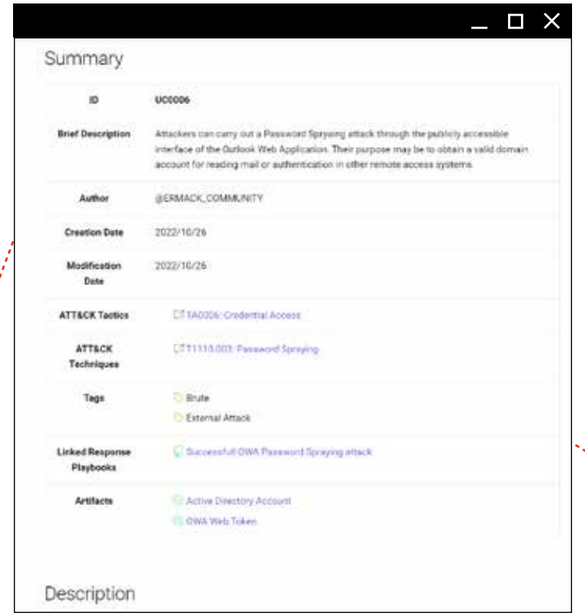


Рис. 3. Описание угрозы

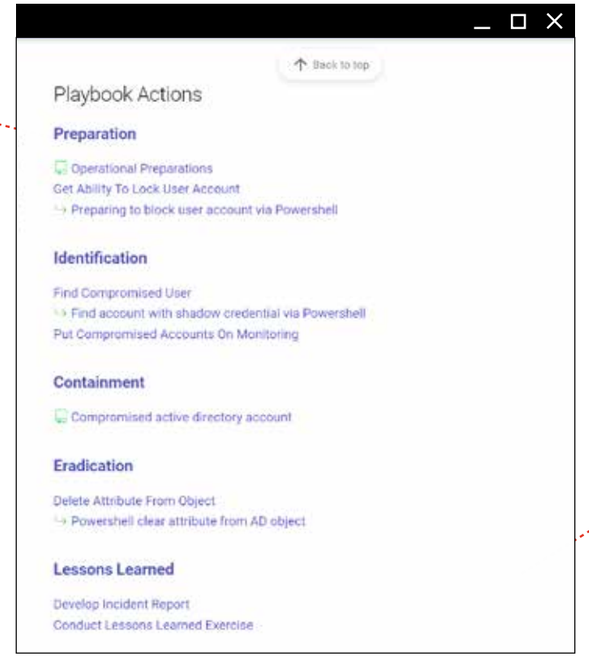
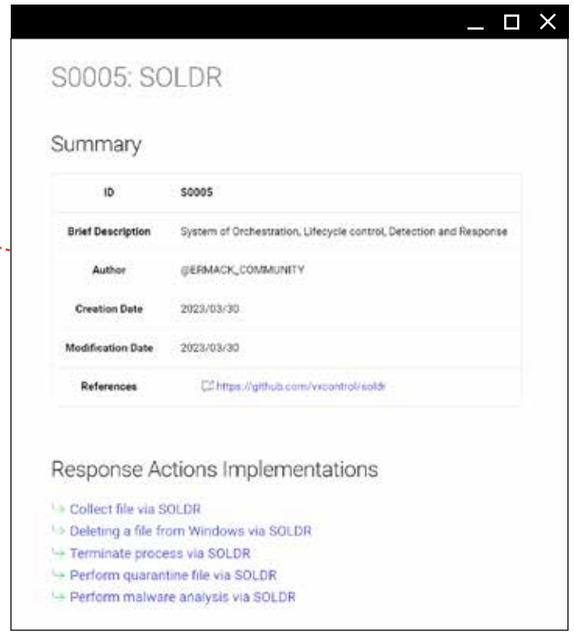


Рис. 4. Сценарий реагирования

Рис. 5. Список действий, которые можно выполнить с помощью конкретного продукта

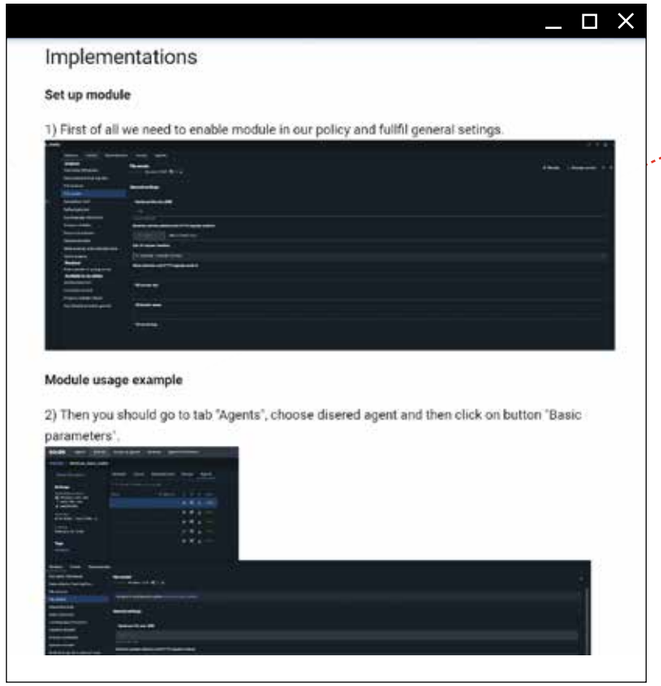


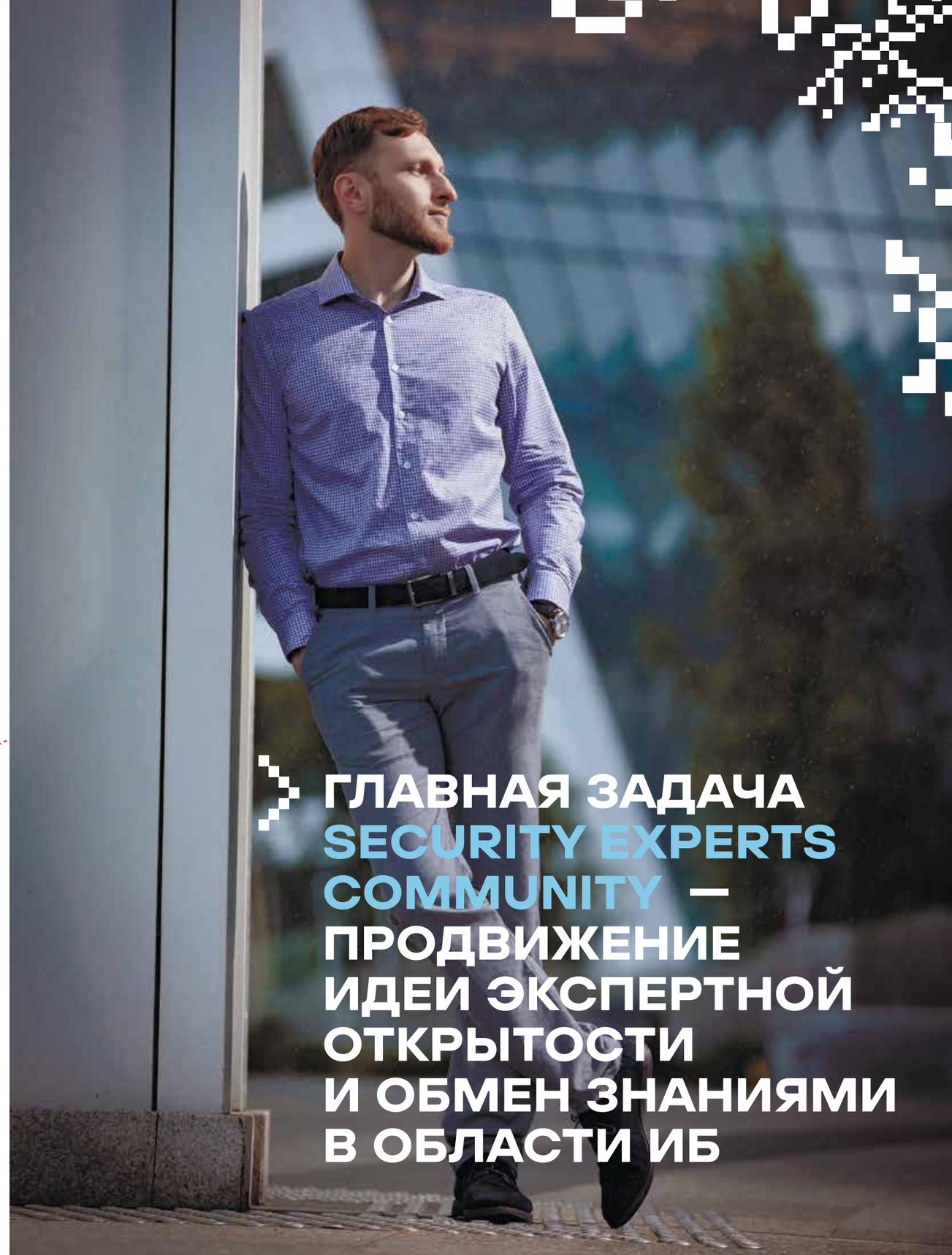
У ERM&CK большой потенциал, потому что он будет интересен не только ИБ-специалистам, но и компаниям, в которых реагированием занимаются ИТ-шники.

Плюс сами вендоры заинтересованы в описании того, как их продукты могут выполнять то или иное абстрактное действие.

Для пользователей такая информация от первоисточника будет очень ценной.

Рис. 6. Реализация действия





ГЛАВНАЯ ЗАДАЧА SECURITY EXPERTS COMMUNITY — ПРОДВИЖЕНИЕ ИДЕИ ЭКСПЕРТНОЙ ОТКРЫТОСТИ И ОБМЕН ЗНАНИЯМИ В ОБЛАСТИ ИБ

Развитием проекта занимается Security Experts Community [4], главная задача сообщества — продвижение идеи экспертной открытости и обмен знаниями в области кибербезопасности:

- вендоры могут добавлять в базу свои продукты и подробно описывать, в каких кейсах их решения будут полезны пользователям;
- ИТ-администраторы и ИБ-специалисты могут делиться экспертизой. Описывать реализации действий можно с помощью скриптов или просто текстом с картинками;
- валидировать предложенные реализации будут эксперты по реагированию.

ГЛАВНАЯ ЦЕННОСТЬ — ОТКРЫТАЯ БАЗА ЗНАНИЙ

Фактически фреймворк ERM&CK — это модель организации данных по реагированию плюс знания, оформленные в рамках этой модели. Благодаря открытости проекта знания могут быть провалидированы участниками сообщества. Причем домен реагирования — лишь одна из частей итоговой базы знаний. Также в ней будет собрана информация о детектировании, смягчении последствий, симуляции вредоносных действий и т. д.

При разработке модели мы вдохновлялись проектом Atomic Threat Coverage [5], который ставил перед собой схожую задачу — объединить разные типы экспертного контента по защите инфраструктуры. Однако в текущей версии проекта не все сущности связаны друг с другом. В целом технология хранения связей и данных в Atomic Threat Coverage для наших задач оказалась неподходящей, поэтому мы придумали собственную модель данных, позволяющую описывать и связывать экспертизу из различных доменов SecOps.

Одной из задач Security Experts Community является создание площадки для обсуждения экспертных тем и публикации своих проектов. Для поддержания комфорта и прозрачности в сообществе действует свод обязательных правил. Например:

1. Сообщество открытое, нетоксичное и строится в соответствии с Кодексом поведения [7].
2. Деятельность сообщества прозрачна и полностью доступна на открытых площадках — есть проекты на GitHub и два зеркала.
3. Общение ведется в GitHub Discussions и Telegram-группе.
4. Вклад можно внести, создав PR в репозиторий, предложив свой полезный проект или даже приняв участие в обсуждении в чате.
5. Присоединяйтесь! [8]

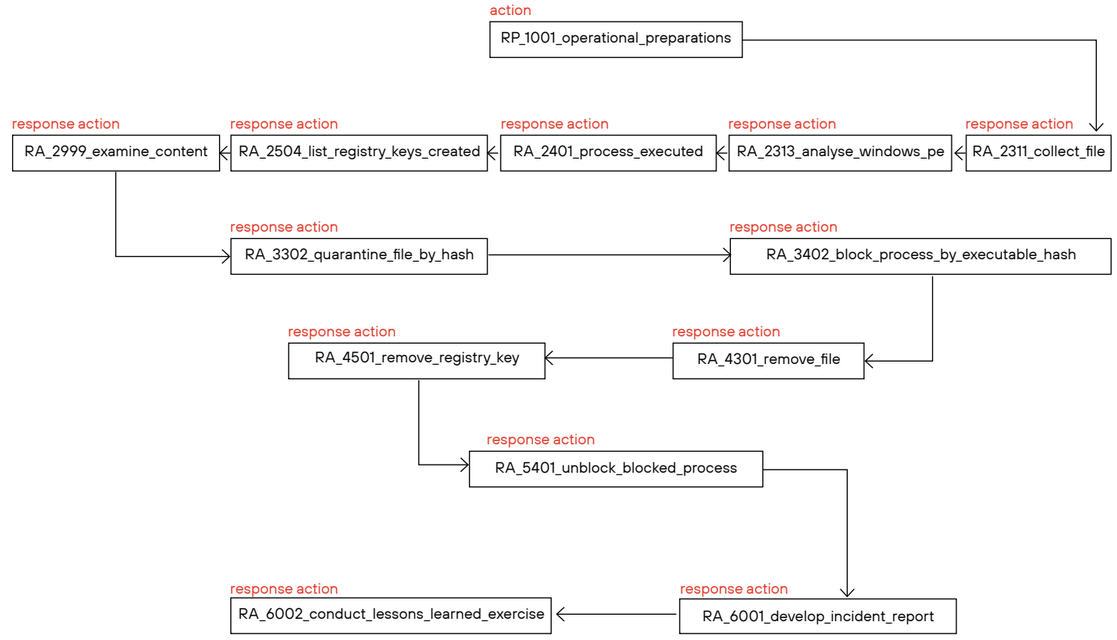


Рис. 7. Схема сценария в Response Flow

ПЛАНЫ НА БУДУЩЕЕ

Сегодня ERM&CK в тестовом режиме используется в нескольких российских компаниях. Мы получаем обратную связь от пользователей и повышаем эффективность фреймворка. На данный момент свой вклад в проект внесли специалисты CyberOK, Positive Technologies, «Когорты» и ряд независимых экспертов.

В первую очередь мы работаем над расширением базы знаний. Планируем наращивать объем полезного контента и добавлять новые артефакты. В будущем это позволит автоматически генерировать плейбуки и связывать вместе разные кейсы.

Кроме того, мы развиваем редактор Response Flow, который призван упростить работу с ERM&CK. Он уже позволяет создавать новые действия реагирования и схемы сценариев (рис. 7). В дальнейшем планируем реализовать полную поддержку всех аспектов работы с фреймворком из единого графического интерфейса. Проект уже доступен на GitHub [9].

БИЗНЕС-ТРЕК CISO ONDAY

«В этом году мы отразили порядка 6,5 млн кибератак на сотрудников, инфраструктуру и, самое главное, на клиентов. Мы также предотвратили около 300 внутренних инцидентов, которые могли бы привести к серьезным последствиям. Но каждый раз, когда возникает ситуация, в которой лично я считаю, что необходимо идти с докладом к руководству, первое, что возникает у меня в душе, — это внутреннее расстройство. Почему мы не смогли обнаружить это раньше и предотвратить? И я, прежде чем что-то докладывать, собираю соответствующую информацию с полным раскладом: что произошло и почему; что мы не сделали, чтобы этого избежать; что мы сделали, чтобы не допустить распространения; что мы будем делать дальше, чтобы кардинальным образом улучшить ситуацию. Нужно честно, открыто и прямо докладывать о происходящем руководству. И это трудно.

6,5 млн кибератак

Алексей Волков

Вице-президент, директор по информационной безопасности «ВКонтакте»

«Насчет общечеловеческих качеств, которыми должен обладать безопасник. У меня получилось два: цинизм и эмпатия. С одной стороны, он должен вставать на место руководства и понимать, в чем его цели, в чем цели организации, наблюдательного совета, чего хотят акционеры, в конце концов. А цинизм — это про стрессоустойчивость. Ну сломали — окей, у меня есть механизм, как с этим работать, и я буду с этим работать. Хватит на меня давить, я не должен бояться ходить к руководству с рассказом о том, что у меня нашли какую-то уязвимость. Да, бывает, но я умею реагировать. Цинизм здесь — это попытка сказать руководителю: «Стоп, без паники, работаем как надо».

Сергей Демидов

Директор департамента операционных рисков, информационной безопасности и непрерывности бизнеса ПАО «Московская Биржа»

«Считаю, что грамотность наших сотрудников — это, наверное, 99% успеха внутренней безопасности. Потому что именно неграмотность и внутренний нарушитель как раз-таки генерируют огромное количество проблем. Вот с этим нужно работать. И, по крайней мере, мы в нашей организации проводим киберучения, разные проверки, аудиты. И сейчас — я вижу это по своим пользователям, а у нас более 60 000 человек в компании — у них смирение наступило, есть такая стадия в психологии. Да, действительно отрубили его, да, виноват, да, сейчас придется писать объяснительную.

99%

Александр Чариков

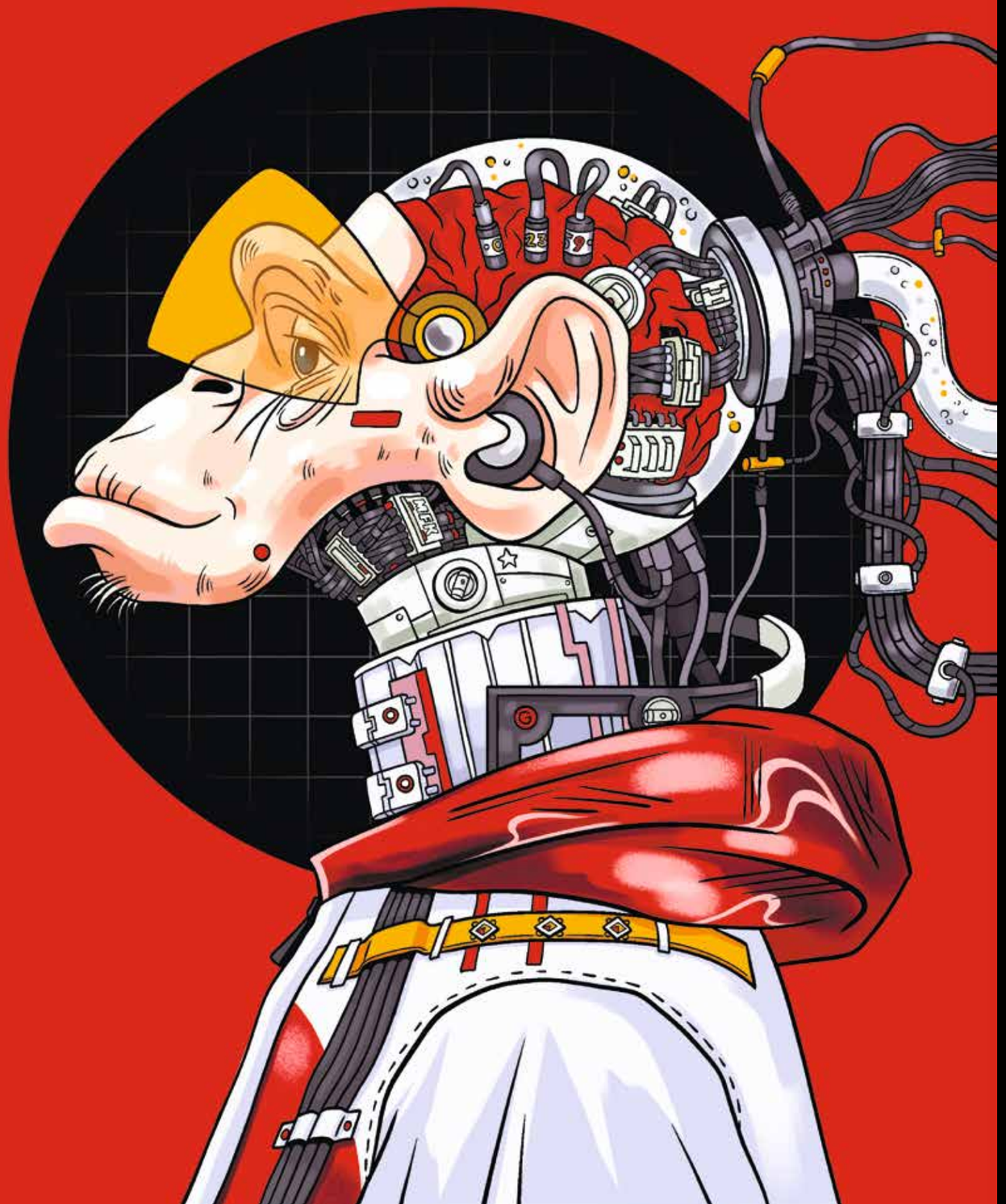
Заместитель генерального директора ПАО «РусГидро»

«Есть компания, у которой можно украсть 300 млн руб. Это можно сделать двумя способами. Первый — украсть ликвидные ценности на 300 млн. Второй — показать, как можно причинить ущерб, за который в ходе шантажа компания заплатит 300 млн. Ты говоришь: «Ребят, смотрите. Кто может это сделать?» Предлагаешь 10, 20 или 30 млн, и, о чудо, находятся те, кто может все реализовать. Платишь хакерам 30 млн, они тебе приносят 300. Это экономика, физика и химия киберпреступления.

10, 20, 30 ... 300

Юрий Максимов

Сооснователь, мажоритарный акционер и председатель совета директоров Positive Technologies



SIEMMONKEY: ПЛАГИН ПРОТИВ РУТИНЫ В SOC

Константин Грищенко

Руководитель отдела мониторинга информационной безопасности Positive Technologies

Время прочтения: 10 минут

Для кого: специалисты SOC

Прокачиваем знания: автоматизация работы с SIEM

Для решения задач security operations center существует масса разных подходов, технологий и ПО. Сегодня мы остановимся на работе с SIEM и расскажем о решении, которое позволяет автоматизировать скучные рутинные операции.

SiemMonkey [1] — это браузерный плагин, который упрощает работу специалистов SOC. Многим из них, особенно специалистам первой линии, часто приходится выполнять однообразную, монотонную работу, а это прямой путь к выгоранию.

Аналитикам, которые пользуются SIEM, зачастую требуются дополнительные данные, которые приходится собирать и проверять вручную. Нужно дорабатывать систему так, чтобы упрощать подобные операции, но это не всегда можно сделать в сжатые сроки силами команды разработки продукта. Можно, конечно, подождать, но упростить себе жизнь хочется прямо сейчас...

В качестве основы для UI многих современных ИБ-продуктов, в том числе SIEM, выступает обычный браузер. Почему бы не сделать небольшой плагин, который будет выполнять роль тех самых инструментов, упрощающих работу аналитика? Звучит вполне реально. Помимо прочего, у этого подхода есть несколько важных преимуществ:

- **Пользователям не придется устанавливать дополнительное ПО.** Браузер есть у всех.
- **Автоматическая аутентификация.** Аналитик выполнит ее, когда откроет веб-интерфейс SIEM. Если встроить плагин в интерфейс системы, запросы к бэкенду будут выполняться автоматически.
- **Не нужно думать, как отображать результаты работы системы.** Если вместо плагина использовать, к примеру, скрипт на Python, придется придумывать что-то с выводом результатов. В консоль? Формировать отдельный отчет? Создавать окна с помощью WinForms? Разобраться с HTML и CSS и вывести данные в интерфейс SIEM гораздо проще.
- **Браузерные плагины работают на любых ОС.**

Для реализации решения мне понадобились базовые навыки разработки, доступ к исходному коду SIEM (той части, которая выполняется в браузере на стороне клиента) и немного свободного времени. С технической точки зрения в SiemMonkey нет ничего необычного, это стандартное расширение для Chrome. Внутри — классический JavaScript с примесью jQuery и ряда других библиотек, самописные обработчики, отслеживание изменений в DOM, а также простые манипуляции с CSS.

[1]

КЕЙСЫ: ПЯТЬ ПРИМЕРОВ АВТОМАТИЗАЦИИ

1. Проверка хеша

Что может сделать аналитик, чтобы проверить, является ли активность вредоносной? Например, посмотреть вердикты по хешу на VirusTotal. Для этого нужно выделить хеш, скопировать его, открыть новую вкладку в браузере, перейти на virustotal.com, затем — на страницу поиска, вставить значение и нажать Search. Только после всей этой рутины можно будет изучать результаты.

Что умеет SiemMonkey? Нажимаете на название поля с хешем и автоматически открываете новую вкладку и заполненную строку поиска — даже Enter нажимать не нужно. Никакой рутины: один клик — и можно переходить к интеллектуальному труду.

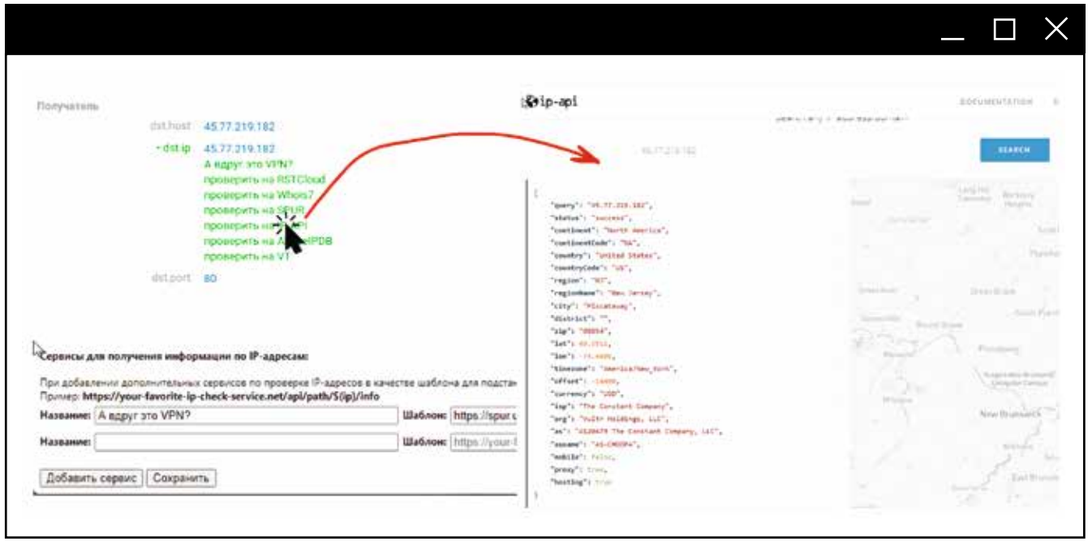
Рис. 1. Проверка хеша

2. Проверка IP-адресов

В событиях зачастую встречаются подозрительные IP-адреса: источников, получателей, промежуточных узлов и др. Чтобы их проверить, аналитику придется выполнить те же самые скучные действия: выделить IP-адрес, скопировать, открыть новую вкладку с VirusTotal (с WHOIS, AbuseIPDB) и т. д.

SiemMonkey снова сводит рутину к минимуму. Плагин добавляет в интерфейс SIEM раскрывающееся меню со ссылками на ресурсы, позволяющие проверить репутацию IP-адреса. Нажимаете на ссылку, и в соседней вкладке открывается нужный сервис. Отмечу, что помимо поддержки популярных ресурсов мы реализовали возможность добавления пользовательских ссылок.

Рис. 2. Проверка IP-адреса



3. «Интеграция»

Почему в кавычках? Потому что интеграция в привычном смысле слова уже реализована в большинстве ИБ-продуктов. Даже решения разных вендоров зачастую можно подружить с помощью стандартных API. Тем не менее это не всегда избавляет нас от рутины.

Предположим, SIEM зафиксировал у процесса подозрительную сетевую активность. Как проверить, что находится внутри трафика? Нужно воспользоваться NTA-решением. В нашем случае это PT NAD. Аналитику придется дважды копировать из SIEM IP-адреса и порты, переключаться на NTA и вбивать данные в фильтр. Как SiemMonkey упрощает этот процесс? Открываете всплывающее окно и нажимаете на ссылку для поиска сетевой сессии. В соседнем окне сразу откроется NTA-система с развернутым анализом — данными о сигнатурах и трафике. Схема работает и наоборот: если аналитик заподозрит что-то, глядя в NTA, он сможет найти соответствующие трафику события в SIEM.

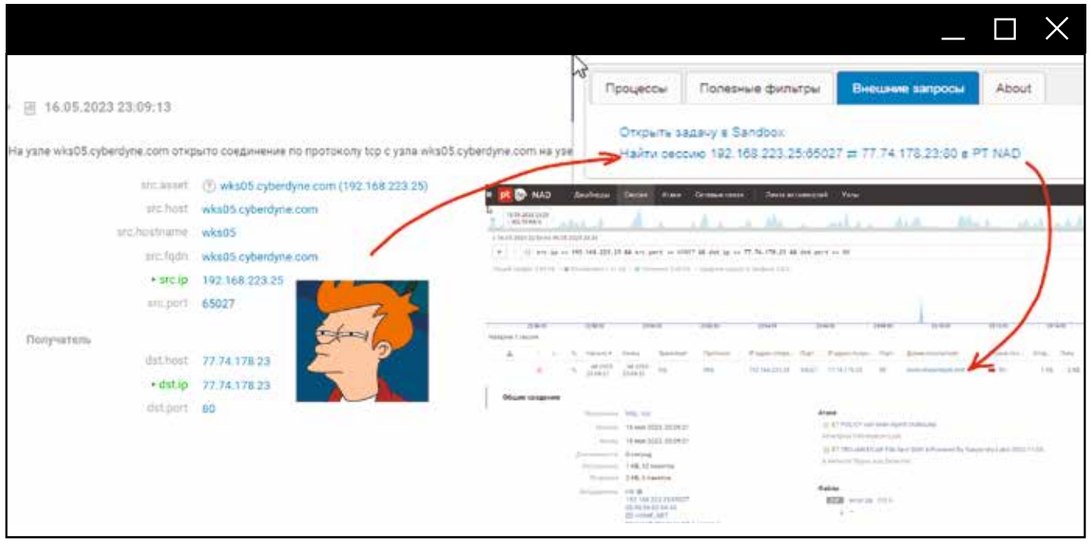


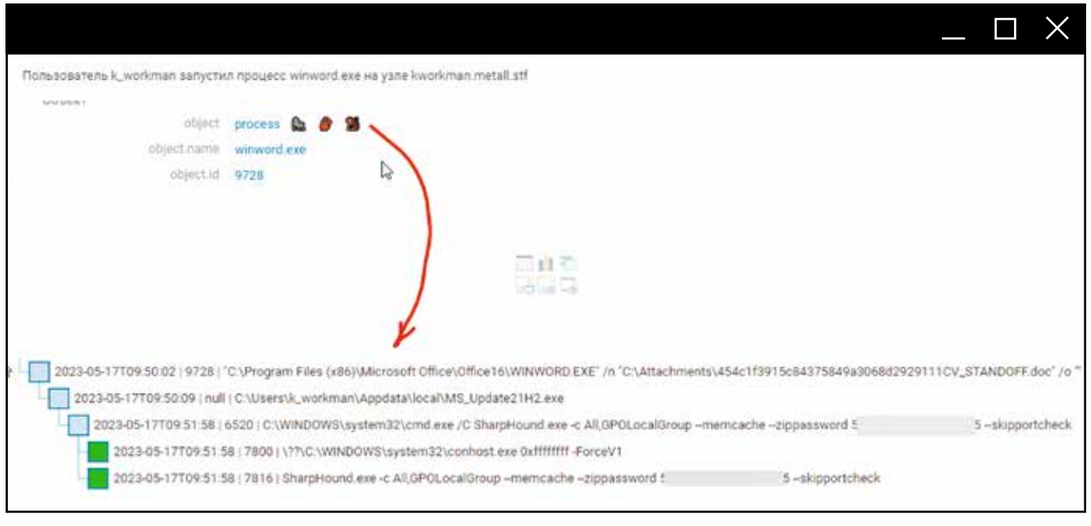
Рис. 3. «Интеграция» SIEM с NTA

4. Дерево

Во многих SIEM есть табличные интерфейсы. Это один из важнейших инструментов аналитика, но не все данные удобно анализировать в таком виде. Допустим, специалист смотрит список запущенных на хосте процессов и находит среди них подозрительные. Дальше ему нужно понять, в каком порядке они были запущены: кто родитель, какие у него потомки и т. д. В этом случае работать с деревом гораздо удобнее, чем с таблицей.

Для решения задачи в SiemMonkey есть три специальные кнопки. Одна позволяет получить всех потомков текущего процесса. Вторая — всех прямых предков. Третья строит дерево всех процессов в рамках одной пользовательской сессии. Для получения информации по процессам плагин фоном отправляет запросы в бэкенд SIEM (так же, как если бы аналитик выполнял их вручную).

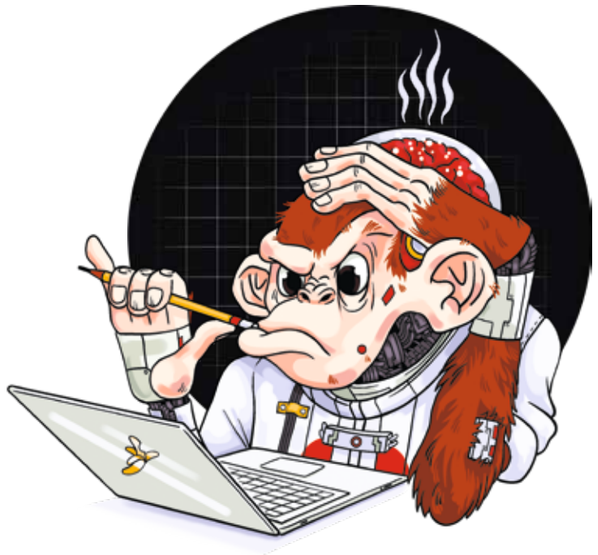
Рис. 4. Дерево процессов



5. Параметризованные фильтры

Фильтры, которые использует аналитик для поиска определенных событий, можно сохранять во многих SIEM-системах, но, если они не зависят от контекста (конкретного события), это не всегда удобно. Зачастую, изучая одно событие, аналитик понимает, как найти другое, связанное с ним. Для этого нужно подставить определенные значения из первого события в поисковый фильтр второго. Если передавать такие данные как параметр, получится параметризованный фильтр.

При нажатии на соответствующую кнопку SiemMonkey найдет на странице все параметры текущего события. В этот момент во всплывающем окне автоматически появятся подходящие под контекст фильтры. Аналитику останется лишь нажать на подходящий фильтр, чтобы открыть в соседней вкладке окно с нужными событиями. При этом набор фильтров задается заранее простым JSON-файлом, в котором можно указать все необходимые параметры.



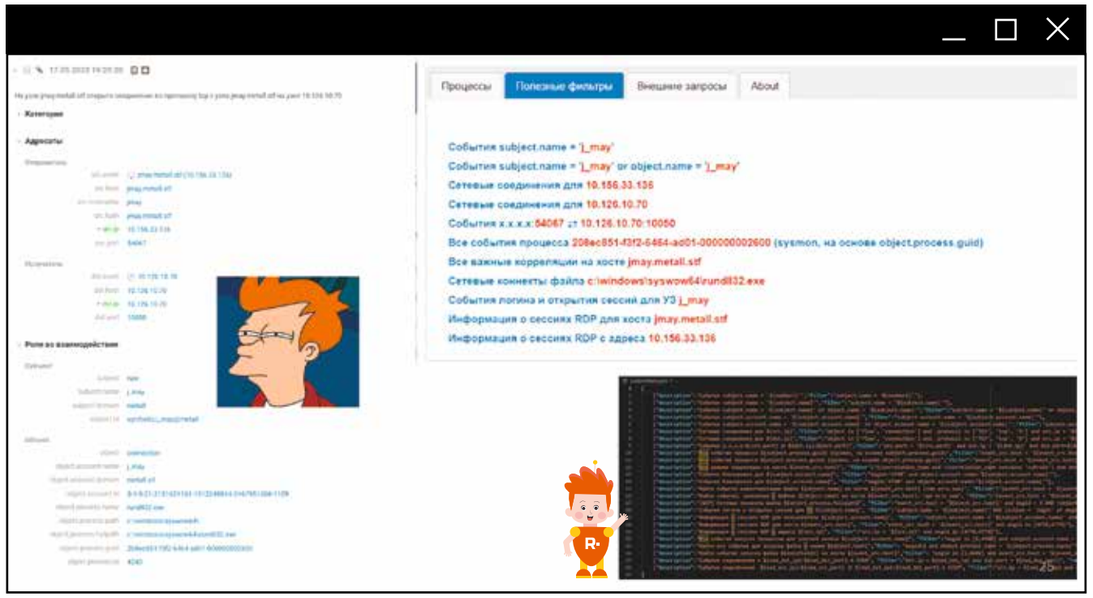


Рис. 5. Параметризованные фильтры

СПЕЦИАЛИСТАМ ПЕРВОЙ ЛИНИИ SOC ЧАСТО ПРИХОДИТСЯ ВЫПОЛНЯТЬ МОНОТОННУЮ РАБОТУ, А ЭТО ПРЯМОЙ ПУТЬ К ВЫГОРАНИЮ

ЗАЧЕМ НУЖЕН ПЛАГИН, КОГДА ЕСТЬ SIEM...

Функционал SiemMonkey не ограничивается перечисленными кейсами. Да, для решения подобных задач не обязательно писать отдельный плагин, ведь хороший SIEM во время сбора событий может сразу их обогатить, подцепит все нужные данные и предоставит аналитику. Это возможно, но не очень разумно.

Главная задача SIEM — собирать множество исходных событий, приводить их к единому виду, обогащать и коррелировать, выявляя признаки подозрительной активности и компьютерных атак. Как правило, этих событий действительно много: наш внутренний SOC за день собирает как минимум 1 000 000 000, среди которых 15–20% — сетевые. Если каждое из них автоматически проверять на сторонних ресурсах, скорее всего, на корреляцию мощности SIEM уже не хватит. Таким образом, в попытке облегчить работу аналитикам, можно больно ударить по работоспособности всего SOC. Куда разумнее вынести этот функционал в отдельное решение.

Наконец, довольно простой вывод: если приложить немного усилий, можно заметно упростить себе работу. А еще лучше — поделиться результатами с сообществом и вместе развивать решение.

Пользователи SiemMonkey оставляют нам фидбэк и делятся идеями на GitHub. Мы постепенно улучшаем решение с учетом их запросов. Например, теперь в плагине можно быстро генерировать ссылки на конкретные события (раньше пользователи просто обменивались фильтрами). Другой пример: по запросу сообщества мы реализовали возможность сохранять нормализованные события в формате JSON-файлов.

ЯЗЫК EXTRACTION AND PROCESSING: РАЗРАБОТКА КОРРЕЛЯЦИЙ БЕЗ БОЛИ И СТРАДАНИЙ

Дмитрий Федосов

Старший специалист отдела обнаружения атак
Positive Technologies

Юлия Фомина

Ведущий специалист отдела обнаружения атак
Positive Technologies

Время прочтения:	20 минут
Для кого:	ИБ-специалисты, разработчики, аналитики SOC
Прокачиваем знания:	корреляция событий, разработка контента, обнаружение атак

ЧЕМ ХОРОШ EXTRACTION AND PROCESSING

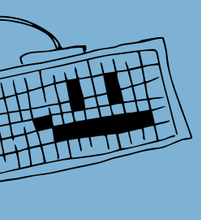
Начнем с терминологии. eXtraction and Processing (XP) [1] — созданный в Positive Technologies язык разработки правил нормализации, корреляции и обогащения. Они используются в MaxPatrol SIEM, PT XDR и SOLDR для обнаружения атак на основе анализа потока событий с конечных точек. VS Code — это IDE (integrated development environment) для разработки программ на классических языках: JavaScript, C#, Python и др. Инструмент позволяет добавить поддержку произвольного языка с помощью языковых расширений. Реализовав расширение VSCode XP [2], мы получили возможность использовать VS Code для разработки контента на XP.

Почему именно eXtraction and Processing:

1. Запрос со стороны сообщества на экспертную открытость вокруг языка.
2. Простой текстовый формат и подробная документация.
3. Наличие опыта и знаний, которыми мы готовы делиться.
4. Позволяет описать сложную логику обнаружения.
5. Публичные инструменты, позволяющие вести разработку контента даже без соответствующих ИБ-продуктов.
6. Базовые правила нормализаций для Windows и Linux (auditd) и общедоступные правила [3], которые будут пополняться.

Подчеркнем, что XP можно использовать, даже если у вас нет соответствующих ИБ-решений. Прежде всего это универсальный способ описания экспертизы по обнаружению атак. А если хотите проверить, сработает ли правило на определенные события, достаточно вручную извлечь их из журналов (например, Windows Event Log), скопировать в расширение, запустить корреляцию и получить набор сработавших правил.

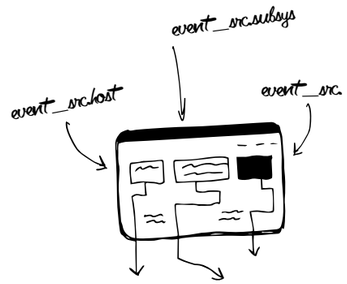
Начнем с нормализации событий в XP.



```
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385f-…}" />
    <EventID>17</EventID>
    <Version>1</Version>
    <Level>4</Level>
    <Task>17</Task>
    <Opcode>0</Opcode>
    <Keywords>0x8000000000000000</Keywords>
    <TimeCreated SystemTime="2023-05-17T10:24:33.2700764Z" />
    <EventRecordID>90359678</EventRecordID>
    <Correlation />
    <Execution ProcessID="9264" ThreadID="4564" />
    <Channel>Microsoft-Windows-Sysmon/Operational</Channel>
    <Computer>Win10x64-133.testlab.com</Computer>
    <Security UserID="S-1-5-18" />
  </System>
  <EventData>
    <Data Name="RuleName">-</Data>
    <Data Name="EventType">CreatePipe</Data>
    <Data Name="UtcTime">2023-05-17 10:24:33.267</Data>
    <Data Name="ProcessGuid">{b56fc2d9-ab37-6464-5711-…}</Data>
    <Data Name="ProcessId">5852</Data>
    <Data Name="PipeName">\fkkgxm</Data>
    <Data
Name="Image">C:\Users\i_ivanov\Downloads\installer.exe</Data>
    <Data Name="User">TESTLAB\i_ivanov</Data>
  </EventData>
</Event>
```

```
{
  "action": "create",
  "event_src.category": "Other",
  "event_src.fqdn": "win10x64-133.testlab.com",
  "event_src.host": "win10x64-133.testlab.com ",
  "event_src.hostname": "win10x64-133",
  "event_src.rule": "-",
  "event_src.subsys": "Microsoft-Windows-Sysmon/Operational",
  "event_src.title": "sysmon",
  "event_src.vendor": "microsoft",
  "generator.type": "logcollector",
  "generator.version": "N26.0.2936",
  "id": "PT_Microsoft_Windows_eventlog_Sysmon_17_Pipe_created",
  "importance": "info",
  "input_id": "00000000-0000-0000-0000-000000000000",
  "mime": "application/x-pt-eventlog",
  "msgid": "17",
  "normalized": true,
  "object": "resource",
  "object.name": "fkkgxm",
  "object.type": "pipe",
  "recv_ipv4": "127.0.0.1",
  "recv_time": "2023-05-18T07:51:00.747Z",
  "status": "success",
  "subject": "process",
  "subject.account.domain": "testlab",
  "subject.account.id": "synthetic:i_ivanov@testlab",
  "subject.account.name": "i_ivanov",
  "subject.process.fullpath": "c:\\users\\i_ivanov\\downloads\\installer.exe",
  "subject.process.guid": "b56fc2d9-ab37-6464-5711-000000005e00",
  "subject.process.id": "5852",
  "subject.process.name": "installer.exe",
  "subject.process.path": "c:\\users\\i_ivanov\\downloads\\",
  …
}
```

Рис. 1. Нормализация события в XP



В левом столбце на рис. 1 показана запись в журнале Windows в формате XML (17 Sysmon). Справа — нормализованное событие в формате нашей таксономии: все поля унифицированы, приведены к единому стандарту.

Унификация — это всегда полезно. Она позволяет осуществлять поиск событий без привязки к формату журналов конкретного вендора или продукта. Например, по фильтру *subject.account.name = "pushkin" and action = "login"* можно найти все приложения, к которым осуществлял доступ данный пользователь.

Поля разделены на группы. Например, *event_src*, описывающая источник события. В ней есть *event_src.host* — имя хоста, *event_src.subsys* — источник события (журнал или подсистема), *event_src.vendor* — вендор источника событий. Ниже находятся поля группы *subject.account*, содержащие информацию о пользователе. Еще один важный блок — *subject.process*, включающий идентификатор процесса, его имя и директорию, откуда он был запущен.

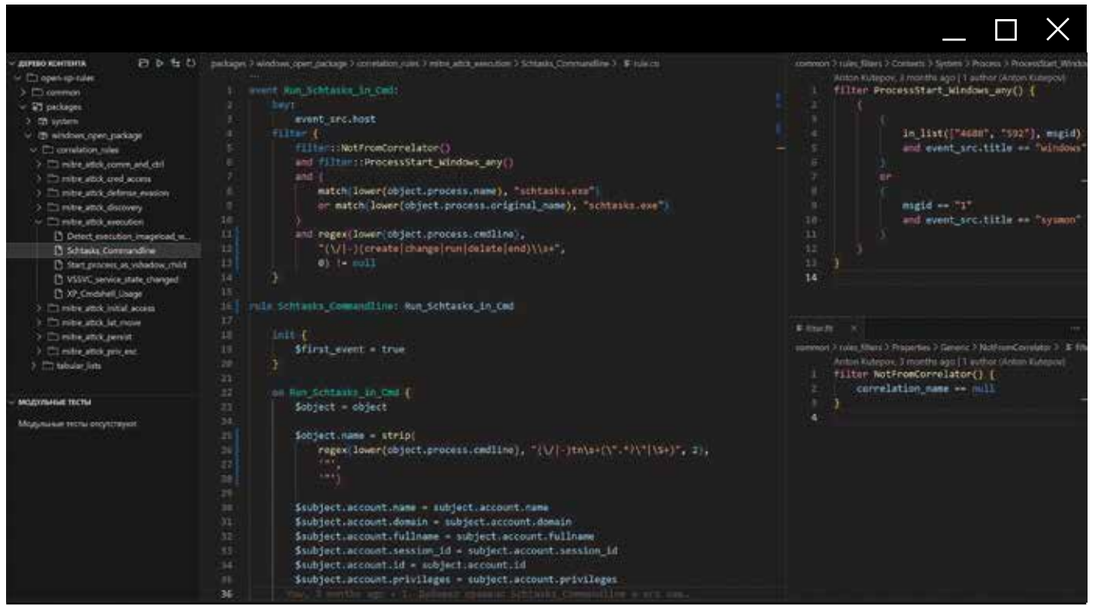
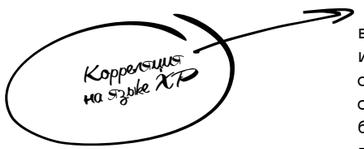


Рис. 2. Пример простейшей корреляции на языке XP



Рассмотрим пример простейшей корреляции для выявления манипуляции задачами в Windows. На рис. 2 инструкция *event* определяет событие, которое нужно отфильтровать из потока. Вложенная инструкция *filter* определяет условие, по которому будут отбираться события, а *key* показывает, как будут разделяться различные потоки, подходящие под фильтр и имеющие разный набор значений полей, указанных в *key*.

Также здесь можно использовать макросы [4]. Они напоминают функции в классических языках программирования, определяются ключевым словом *filter* и хранятся в отдельных файлах. В них выносят повторяющиеся в фильтрах различных корреляций условия для повторного использования, что позволяет уменьшить дублирование кода фильтра и упростить его. Например, макрос *NotFromCorrelator* определяет, что среди событий будут отобраны только нормализованные, а корреляционные — пропущены. *ProcessStart_Windows_any()* показывает, что нам интересны именно события запуска процессов в Windows, события с Event ID: 1 (sysmon), 4688, 592. Также в макросы можно передавать параметры, что позволяет сделать код фильтра еще проще и понятнее.

Фильтр осуществляет логические проверки, которые помогают найти запущенный процесс — в нашем случае schtasks.exe (поле *object.process.name*). Поле *object.process.cmdline* содержит параметры процесса schtasks.exe, среди которых мы отбираем *create, change, run, delete, end* с помощью регулярного выражения. В результате мы получаем простейший детект: когда прилетит соответствующее событие, оно будет нормализовано и успешно пройдет фильтр, а затем коррелятор породит корреляционное событие, которое визуализируется в продукте.

4

Подробнее о работе с eXtraction and Processing мы рассказали в докладе «VS Code XP: корреляции без боли и страданий»

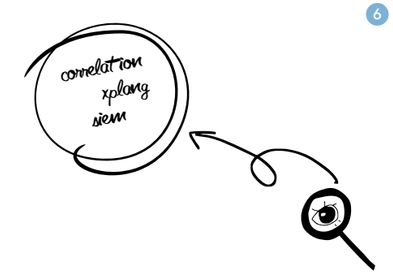
и на воркшопе «Лучшие практики по созданию правил на языке XP».

После директивы *rule* также описывается условие корреляции, определяющее в том числе порядок и количество событий, а также временные отрезки, в которые они должны быть получены для срабатывания правила корреляции. В примере на рис. 2 ожидается появление только одного события. Узнать больше о синтаксисе условий корреляции можно здесь [5].

Еще одним преимуществом языка XP является простота настройки окружения по работе с ним через расширение VSCode-XP. Нужно скачать релиз из репозитория XP Knowledge Base Toolkit [6], сделать клон репозитория с публичными правилами — и вуаля. Расширение лежит в магазине VS Code — ищите по ключевым словам «correlation, xplang, siem». Единственное, что потребуется сделать после установки, — указать в настройках путь до Knowledge Base Toolkit. Буквально пара минут, и можно разрабатывать контент, изучать атаки и вносить свой вклад в сообществе.

5

6



ПОЧЕМУ МЫ ВЫБРАЛИ VS CODE

Самый простой ответ: потому что большинство наших экспертов уже использовали эту IDE и имели соответствующий опыт. Кроме того, если погрузиться в функционал VS Code, можно выделить следующие преимущества этого инструмента:

1. Поддержка Language server protocol (LSP), возможность легко расширить поддержку для редакторов Eclipse, Emacs, Notepad++, Sublime Text и других.
2. Расширения поддерживают произвольные WebView, что позволяет реализовывать практически любые функции нативных приложений.
3. Большой магазин существующих расширений: GitLens, Spell Checker и т. д.
4. Кросс-платформенность (Windows, Linux, MacOS).
5. Доступность веб-версии через Docker [7].
6. Множество цветовых тем на любой вкус и цвет.

7

Кроме того, при написании экспертизы мы хотели использовать все плюшки, которые уже давно есть у разработчиков: подсветка синтаксиса, автодополнение, сниппеты и многое другое. VS Code позволил нативно их реализовать.

В левой части интерфейса VS Code XP (см. рис. 2) представлено дерево контента. Open-xp-rules — ключевая директория системы, своего рода база знаний. В common находятся макросы, а ниже в packages — пакеты экспертизы. Обычно в директориях пакетов есть поддиректории, которые определяют тип содержащегося в них контента, — например, correlation_rules, enrichment_rules, normalization_formulas и др. Для упрощения поиска правил корреляции мы раскладываем их по техникам согласно классификации MITRE ATT&CK. На рис. 2 показаны несколько правил: ProxyNotShell (Initial Access), Schtasks_Commandline (Execution) и всеми любимый Mimikatz (Credential Access).

ПЯТЬ ПРАВИЛ РАБОТЫ С КОНТЕНТОМ В POSITIVE TECHNOLOGIES

› **Пишем тесты. Всегда пишем тесты.** Любое правило не существует, если у него нет тестов; это значит, что перед вами какой-то код, который, возможно, работает, но, скорее всего, нет.

› **Применяем test-driven development (TDD)**. Проводим разные варианты атак, собираем нужные события, на основе которых делаем тесты. После этого разрабатываем правило и запускаем тесты для проверки. Если все тесты прошли — правило работает, если нет — нужно исправить ошибки и доработать правило.

› **Используем систему контроля версий.** В нашем случае Git, но это не принципиально. Главное — иметь удобный инструмент для хранения и управления контентом.

› **Используем feature branch workflow.** Для создания правила от основной ветки разработки отводится feature branch, в которой создаются и исправляются правила. Затем автор заводит merge request в основную ветку, и его проверяют два других эксперта. После ревью ветка вливается в основную ветку разработки.

› **Continuous integration для запуска тестов.** CI поможет запустить все тесты (нормализаций, корреляций и обогащений) и через определенное время выдаст вердикт: все работает, можно собирать пакет и отдавать клиентам.

КАК ПИСАТЬ ПРАВИЛА: РАЗБИРАЕМ НА ПРОСТЕЙШЕМ ПРИМЕРЕ MIMIKATZ

Работа начинается со сбора событий. Предположим, у вас есть тестовая инфраструктура с настроенным аудитом. Если события заведены в MaxPatrol SIEM, можно скопировать нужные через интерфейс. Также можно копировать события в формате XML из Event Viewer или брать их из журналов Syslog и Auditd (скоро).

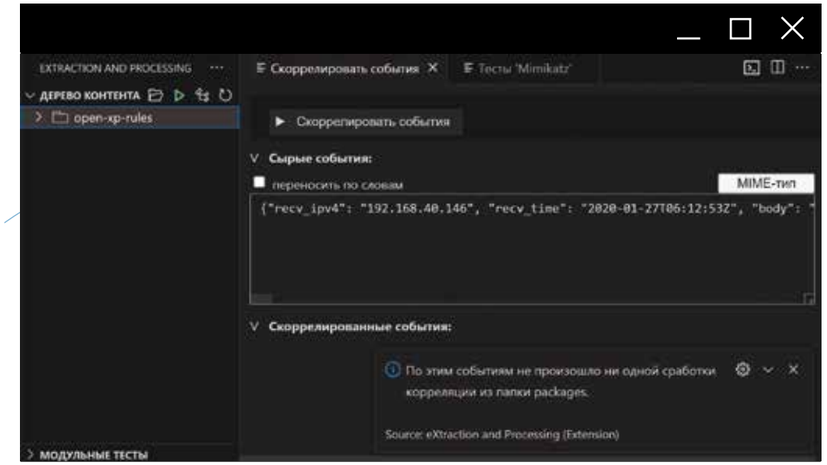
Следующий важный шаг — предварительная корреляция собранных событий, которая позволит понять, имеются ли какие-то сработки на данную последовательность. Здесь может быть несколько вариантов развития событий.

› **Сработок нет.** Если вы не можете задетектить хакерскую утилиту или атаку, пора разработать правило для выявления данной активности.

› **Сработки есть, и по ним имеются адекватные данные.** Поздравляем: кейс покрыт, дополнительные правила не потребуются.

› **Есть ложные сработки (False Positive).** Если видите, что сработка не релевантна вашим событиям, необходимо скорректировать правило.

Рис. 3. Корреляция событий



VSCode XP может вернуть сообщение: «По этим событиям не произошло ни одной сработки корреляции», значит, пора разрабатывать правило. Для удобства пользователей мы реализовали механизм создания правил из шаблонов: открываете раздел, вводите название правила и выбираете наиболее близкий из списка.

Например, шаблон *For_Profiling* подходит, если вы хотите разработать правила, профилирующие доступ к какой-либо системе — например, GitLab, Teamcity или 1C_Enterprise. Следом идет группа шаблонов для Unix-систем. Все они привязаны к основному событию, на основании которого вы планируете писать детект. Например, сетевое подключение, запуск процесса и др. Далее следует группа шаблонов корреляций для Windows, содержащие в качестве основных событий загрузку DLL или EXE в адресное пространство процесса, создание потока в другом процессе, выполнение команд Powershell и т. д. Кроме того, существуют универсальные шаблоны, которые можно использовать, если не подошли другие. В случае с *Mimikatz* выбираем *Windows_Process_Start*.

```
event Process_Start:
    key:
        event_src.host
    filter {
        filter::NotFromCorrelator()
        and filter::ProcessStart_Windows_any()
    }
rule Mimikatz: Process_Start

    on Process_Start {
    }
emit {
    $correlation_type = ""

    $subject = "account"
    $action = "start"
    $object = "process"
    $status = "success"
    $importance = ""

}
```

Рис. 4. Шаблон правила Mimikatz

У корреляционного события есть несколько обязательных полей, их можно задавать в *on*, но хорошая практика — делать это в блоке emit. Остановимся на блоке *emit* нашего шаблона (см. рис. 4). *$correlation_type* отвечает за типы корреляционных событий. Основные — *event* и *incident*, также есть *subrule* и *draft*. Субъект активности — это *account*; действие, которое он производит, — *start*; то, что он запускает, — *object;* статус действия — *success*. Поле *$importance* отвечает за важность срабатывания — она может варьироваться для разных типов корреляционных событий. Например, если обнаруженная атака действительно угрожает инфраструктуре, нужно использовать «high». В поля группы *$category* вносится информация о категории детекта, тактике и технике MITRE ATT&CK, которые соответствуют данной сработке.

```
emit {
    $correlation_type = "incident"

    $subject = "account"
    $action = "start"
    $object = "process"
    $status = "success"

    $importance = "medium"

    $category.generic = "Attack"
    $category.high = "Credential Access"
    $category.low = "OS Credential Dumping"

    ...
}
```

Рис. 5. Директива emit, адаптированная для детекта Mimikatz

Далее добавляем smoke-тесты, которые покажут, что мы подготовили полноценную заготовку для правила. Они нужны на этапе разработки правила для проверки отсутствия синтаксических ошибок, заполнения всех обязательных полей и наличия на выходе корреляционных событий (см. рис. 6).

Рис. 6. Пример smoke-теста

Для этого:

1. Запускаем с разными параметрами утилиту, активность которой хотим детектировать, и собираем группы сырых событий (raw events) из журналов аудита или ИБ-продуктов.

2. Группы событий добавляем в поле *Сырые события* у разных тестов и оборачиваем в нужный конверт (технические метаданные события).

3. В поле *Условия прохождения теста* автоматически добавляется нужный код:

```
expect 1 {«correlation_name»: «Mimikatz»}
```

Если кратко, код означает следующее: после поступления собранных сырых событий с узла, их нормализации, корреляции и обогащения на выходе будет получено одно событие с полем *correlation_name*, равным Mimikatz. То есть будет сработка корреляционного правила с именем *Mimikatz*.

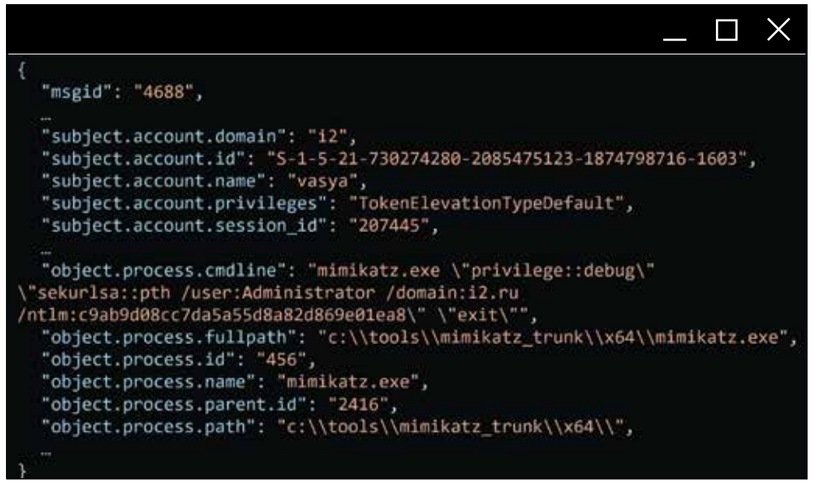
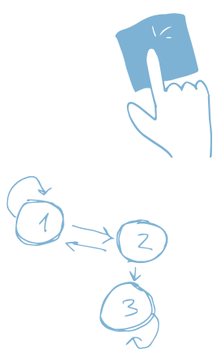


Рис. 7. Нормализованное событие Event ID 4688

Также в окне интеграционных тестов есть кнопка *Нормализовать событие*. После ее нажатия исходное сырое событие превратится в нормализованное (внезапно). Теперь видно, какие поля и значения нужно использовать в фильтре правила. В нашем случае Windows-событие Event ID 4688 будет выглядеть как msgid: 4688.

Начнем с самого простого: в значении поля *object.process.fullpath* попробуем найти «Mimikatz». Берем шаблон, добавляем в него вызов внутренней функции языка regex [8]. Передаем ей приведенное к lower-кейсу значение *object.process.fullpath* для избежания неоднозначности пути в ОС Windows. Вторым параметром передаем регулярное выражение, третьим — номер группы, которая извлекается из регулярного выражения и является возвращаемым значением функции regex.



[8]

Если забыли, как правильно вызвать функцию в XP, расширение подскажет, какие параметры она принимает, что возвращает, а также покажет пример правильного использования (см. рис. 8).

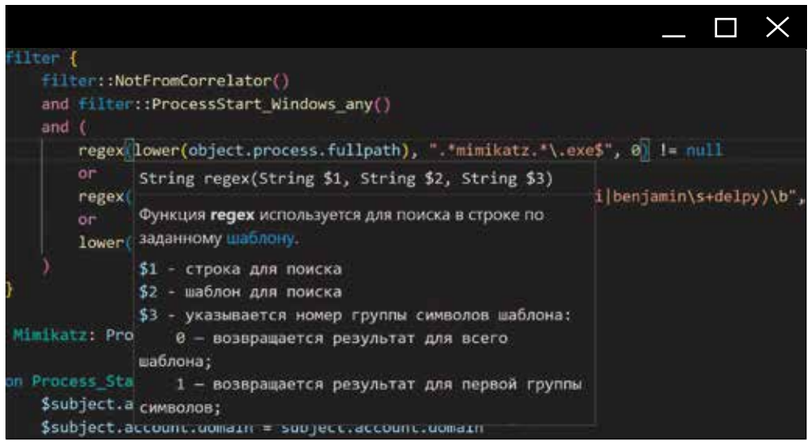


Рис. 8. Подсказка по параметрам функции regex

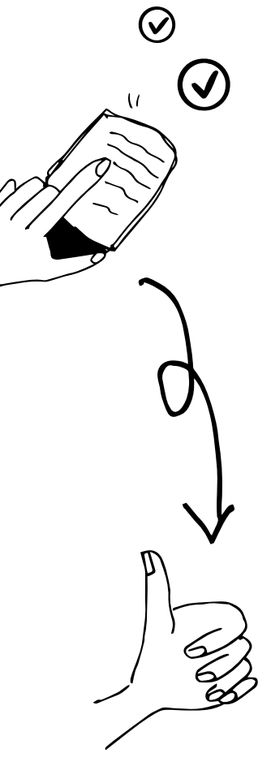
Следующее событие, которое мы нормализуем,— EventID 1 (Sysmon). Может получиться так, что Script kiddie, который раньше просто качал Mimikatz с GitHub, переименует его. Беспокоиться не о чем. Версия с GitHub все равно содержит метаданные *(object.process.meta)*, в которых есть *description*, где черным по белому написано: это Mimikatz. Для решения этой задачи достаточно расширить условия детекта.

Если хакер не только переименовал Mimikatz, но и заменил метаданные (в результате чего Mimikatz превратился, к примеру, в Bibikatz), заданными ранее условиями мы его уже не задетектим. Однако остаются параметры: их реже прячут или переименовывают, потому что для этого нужно обладать опытом разработки на C. Снова расширяем условия детекта.

Фактически мы проверяем *object.process.cmdline* (см. рис. 9) на предмет ключевых слов, которые чаще всего используются Mimikatz. Если они есть в *cmdline*, это зловредная активность.

Рис. 9. Еще больше расширяем условия детекта

Если считаете, что детектировать Mimikatz в 2023 г. — это моветон, посмотрите, как написать с помощью VSCode XP чуть более сложное и актуальное правило на уязвимость ProxyNotShell. Это пример того, как два события мало о чем говорят по отдельности, но вместе позволяют обнаружить эксплуатацию опасной уязвимости.



Прохождение smoke-тестов означает, что мы почти сделали простейшее правило по выявлению Mimikatz. Заметим, что, если бы мы писали правило с нуля, а не пользовались шаблоном, нам пришлось бы сделать еще один шаг: пробросить в корреляционное событие (присвоить корреляционному событию) значения наиболее важных и репрезентативных полей из исходного события (событий) в инструкции *on*. Обычно это достаточно большой кусок однообразного кода.

Пора переходить от smoke-тестов к полноценным интеграционным. Хорошая практика — помимо *correlation_name* проверять и другие результирующие поля правила, которые мы ранее пробросили из событий в блоках *on*. Для их формирования необходимо в каждом тесте нормализовать сырые события и получить ожидаемое с помощью одноименной кнопки.

После получения ожидаемого события автору нужно проверить правильность заполнения полей корреляции и добавить комментарий, описывающий данный тест. Если заполнение полей не соответствует ожиданиям, необходимо исправить код правила и повторно сгенерировать событие. И так до получения нужного результата.

Еще раз запускаем все тесты и проверяем, что они проходят. Отлично — мы готовы финишировать. Осталось несколько мелочей: добавить локализации (одну или несколько), заполнить метаданные (автор правила, описание выявляемых сценариев, необходимые события и т. д.), закоммитить правило и завести merge request с привлечением опытных ревьюверов.

Все готово, вы великолепны! ;)

Условия прохождения теста:

Очистить

```
# Детект на событие 4688
expect 1 {
    "_rule": "Mimikatz",
    "action": "start",
    "category.generic": "Attack",
    "category.high": "Credential Access",
    "category.low": "OS Credential Dumping",
    "correlation_name": "Mimikatz",
    "correlation_type": "incident",
    "count": 1,
    "event_src.fqdn": "win10x64-133.testlab.org",
    "event_src.host": "win10x64-133.testlab.org",
    "event_src.hostname": "win10x64-133",
    "event_src.subsys": "Security",
    "event_src.title": "windows",
    "event_src.vendor": "microsoft",
    "generator.type": "correlationengine",
    "importance": "medium",
    "incident.aggregation.key": "Mimikatz|win10x64-133.testlab.org|s-1-5-21-3389064948-2957360831-125328159-1105",
    "incident.aggregation.timeout": 7200,
    "incident.category": "Undefined",
    "incident.severity": "medium",
    "normalized": true,
    "object": "process",
    "object.account.domain": "testlab",
    "object.account.id": "S-1-5-21-3389064948-2957360831-125328159-1105",
    "object.account.name": "test-admin",
    "object.account.session_id": "1849449",
    "object.process.cmdline": "\"C:\\Users\\test-admin\\Documents\\Tools for raw events\\mimikatz\\x64\\mimikatz.exe\" privile
    "object.process.fullpath": "c:\\users\\test-admin\\documents\\tools for raw events\\mimikatz\\x64\\mimikatz.exe",
```

Рис. 10. От smoke-тестов к интеграционным

РЫНОК ГОВОРИТ

ЭКСПЕРТЫ КРУПНЫХ РОССИЙСКИХ КОМПАНИЙ ОТВЕЧАЮТ НА ДВА ВОПРОСА:

Дмитрий Стуров

Исполнительный директор, начальник управления информационной безопасности «Ренессанс Банка»

1. КАКИЕ УГРОЗЫ КАЖУТСЯ ВАМ НАИБОЛЕЕ АКТУАЛЬНЫМИ ДЛЯ РОССИЙСКОГО РЫНКА В ПЕРВОМ ПОЛУГОДИИ 2023 Г.?

Нельзя не обратить внимание внимания на то, что DDoS-атаки, присутствовавшие в новостной повестке почти весь 2022 г., в 2023-м пошли на убыль. На мой взгляд, это связано с несколькими вещами: естественным падением интереса злоумышленников-энтузиастов, отсутствием каких-либо значимых результатов и в целом неплохой готовностью отрасли к отражению угрозы.

К сожалению, прослеживается тенденция, связанная с ростом количества утечек информации у клиентов. В том числе хешированных паролей, которые потом могут применяться для взлома аккаунтов на других ресурсах. На эту тенденцию обращают внимание и сами организации, и регулирующие органы. Причем нельзя сказать, что это сугубо российская история: утечки происходят по всему миру, и пока эффективные способы переломить ситуацию неочевидны.

В результате утечек и консолидации персональных данных в руках злоумышленников усиливается фишинг, в том числе взлом соцсетей крупных организаций. Мы ожидаем применения нейросетей для создания дипфейков — изображений, видео, голоса.

Помимо прочего, отмечаем увеличение числа атак на периметр организаций и доступные из интернета информационные системы, а также усиление давления со стороны операторов шифровальщиков.

Сохраняется высокий риск атак через цепочки поставок — на ИТ-поставщиков вместо самих организаций. Это вызывает некоторое беспокойство, поскольку вся работа по защите ИТ-инфраструктуры может быть обнулена атакой через поставщика, обладающего удаленным доступом к ресурсам компании.

В заключение можно отметить усиление позиций IoT (автомобили с автопилотом, дроны). Пересечений физического и виртуального мира становится все больше, и интернет вещей заходит во все более критичные области: физическую безопасность, здоровье, частную жизнь.

2. КАКИЕ СОБЫТИЯ ЯВЛЯЮТСЯ ДЛЯ ВАШЕЙ КОМПАНИИ НЕДОПУСТИМЫМИ И ПОЧЕМУ?

Если считать недопустимыми события, наступление которых несет катастрофические последствия для бизнеса (его остановку либо существенное снижение прибыли), то к их перечню можно отнести недоступность ИТ-систем и данных, несоблюдение существенных требований со стороны регуляторов, а также репутационные потери.

Если говорить о применимости официального термина «недопустимые события», мы ждем выработки критериев, по которым их можно будет однозначно определить. Пока можно сказать, что хотя это и перспективная методология, но она еще не апробирована в реальной жизни с учетом наличия смежных процессов — моделирования угроз и риск-менеджмента.

РЫНОК ГОВОРИТ

ЭКСПЕРТЫ КРУПНЫХ РОССИЙСКИХ КОМПАНИЙ ОТВЕЧАЮТ НА ДВА ВОПРОСА:

Антон Нечипоренко

Начальник отдела информационной безопасности департамента анализа и защиты информации СПАО «Ингосстрах»

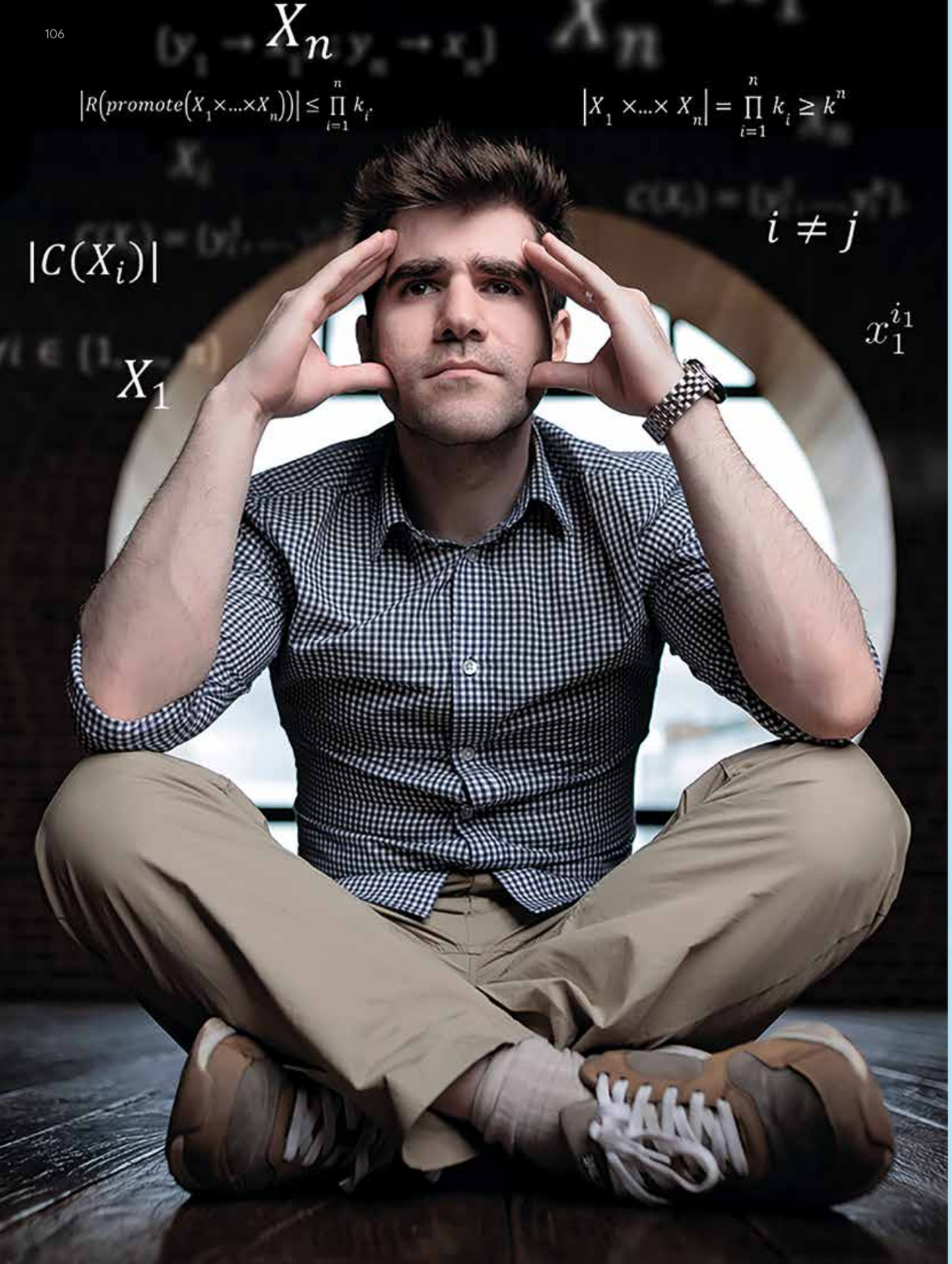
1. КАКИЕ УГРОЗЫ КАЖУТСЯ ВАМ НАИБОЛЕЕ АКТУАЛЬНЫМИ ДЛЯ РОССИЙСКОГО РЫНКА В ПЕРВОМ ПОЛУГОДИИ 2023 Г.?

В первом полугодии 2023 г. изменился вектор воздействия управляемых из-за рубежа и технически оснащенных организаций на информационную безопасность российских компаний, в том числе на «Ингосстрах». Злоумышленники преследуют конкретные цели, направленные на массовую дестабилизацию деятельности, — подобные кейсы автоматически получают серьезные оценки в карте рисков любой организации. ИБ-специалисты и ранее сталкивались с подобными случаями, однако не в таком объеме, масштабе и количестве. Мы фиксируем абсолютно новый уровень воздействия: у злоумышленников достаточно много ресурсов и средств для проведения длительных информационных атак с постоянно меняющимся вектором. Их цель — компании, обеспечивающие функционирование критической информационной инфраструктуры.

2. КАКИЕ СОБЫТИЯ ЯВЛЯЮТСЯ ДЛЯ ВАШЕЙ КОМПАНИИ НЕДОПУСТИМЫМИ И ПОЧЕМУ?

Это перечень из нескольких событий. Приведу примеры, но специально расположу их в случайном порядке.

1. **Недоступность внешнего сайта компании.** В современном мире сайт — уже не визитка, а рабочее пространство для клиентов и партнеров, которым необходимо управлять своими договорами. Например, включать дополнительных водителей в поездку, добавлять актуальные риски при страховании имущества или покупать страховой полис для путешествующих. Клиентоцентричность — одна из основных ценностей нашей компании, поэтому инструменты, которые мы предоставляем, должны работать безотказно.
2. **Потеря базы данных клиентов / договоров.** Это основная ценность для любой страховой компании. Исторические данные для расчета оптимальных тарифов, все контакты и конфиденциальная информация о клиентах выводят эту БД на первое место по значимости. Фактически достоверность информации, которая содержится в базе, является основой нашего бизнеса.
3. **Призывы к неправомерной деятельности.** Мы работаем в строгом соответствии с законодательством Российской Федерации. Для нас недопустимы какие-либо формы воздействия на население для дестабилизации обстановки. Мы проводим целый комплекс мер по контролю передачи информации на всех этапах (данные на внешних сайтах, исходящие email-рассылки и др.). Ситуация осложняется большим количеством вирусов, дефейсов и лозунгов, заложенных в свободно распространяемых библиотеках и ПО. Для недопущения нарушений мы отклоняем использование почти 35% запрошенных сотрудниками библиотек.



КАК РАЗРАБОТЧИКИ АНАЛИЗАТОРА ИСХОДНОГО КОДА С ОДНОЙ ЭКСПОНЕНТОЙ БОРОЛИСЬ

Георгий Александрия

Ведущий программист группы разработки средств статического анализа Positive Technologies

Время прочтения:	15 минут
Для кого:	разработчики
Прокачиваем знания:	статический анализ кода (SAST)

ВВЕДЕНИЕ

Тема SAST (static application security testing) довольно актуальна: многие хотят выявлять проблемные и уязвимые места в приложении еще на этапе разработки, когда стоимость их правок и риски относительно скромны.

Для анализа исходного кода применяются разные технологии и алгоритмы: от менее точных, но быстрых —до всеобъемлющих, но длительных. Ниже мы будем рассматривать один из них — абстрактную интерпретацию, а если точнее — символьное выполнение. Их подробное описание можно найти по ссылкам:

Symbolic Execution 17–355/17–665/17–8190: Program Analysis (Spring 2018)

Ищем уязвимости в коде: теория, практика и перспективы SAST

Abstract interpretation, Wikipedia

Если вкратце, то под абстрактной интерпретацией кода подразумевают интерпретацию без его конкретного выполнения для сбора информации о семантике кода (потоки передачи данных, потоки передачи управления и т. д.). Если же при абстрактной интерпретации все входные данные — аргументы функции точек входа, обращение к файловой системе, к внешним сервисам и т. д. — считать неизвестными, помечая их как символьные значения, то полученная надстройка будет называться символьным выполнением. В качестве примера см. рис. 1, где в восьмой строчке у переменной есть два возможных значения в зависимости от символьного значения входного аргумента.

Рис. 1. Пример символьного выполнения

```
1: procedure COMPUTE1(arg : integer)
2:     x ← 0
3:     if arg ≤ 10 then
4:         x ← 5
5:     else
6:         x ← 32
7:     end if
8:     y ← x        ▷ y будет равен 5, когда arg ≤ 10, и 32, когда arg > 10
9: end procedure
```


Чтобы говорить на одном языке, введем следующие термины:

- Вариантом мы будем называть пару (y, x), где x — некое значение, а y — условие достижимости этого значения. Обозначать будем как $y \to x$.

$$y_1 \to x_1 = y_2 \to x_2 \Leftrightarrow x_1 = x_2 \wedge y_1 = y_2$$

- Вариативным множеством будем называть множество вариантов и обозначать как

$$\{y_1 \to x_1; \ldots; y_n \to x_n\}.$$

Применение этих терминов см. на рис. 2.

Рис. 2. Пример символьного выполнения с вариантами

```
1: procedure COMPUTE2(arg : integer, path : string)
2:     x ← "init"
3:     if arg ≤ 10 then
4:         x ← "foo"
5:     else
6:         x ← path
7:     end if
8:     pvo(x)                    ▷ x = {arg ≤ 10 → "foo"; arg > 10 → path}
9: end procedure
```

СКОРОСТЬ РОСТА

В тот момент, когда формируется некоторое выражение языка, состоящее из подвыражений/переменных (например, конкатенация), которым соответствуют какие-то вариативные множества (рис. 3), количество вариантов итогового выражения растет с огромной скоростью. Если быть точнее, то скорость роста будет не чем иным, как экспоненциальной функцией:

$\exists X_1, \ldots, X_n$ — вариативные множества:

$\forall i \in \{1, \ldots, n\} : X_i = \{y_i^1 \to x_i^1; \ldots; y_i^{k_i} \to x_i^{k_i}\}$, где $|X_i| = k_i$

$\exists g - n$ — мерная функция языка (например, конкатенация, вызов функции от n аргументов и т. д.), в частности определенно $g(X_1, \ldots, X_n)$.

Введем отображение *promote*, которое элемент декартова произведения вариативных множеств переводит в вариант, как показано на рис. 4. Тогда все образы этого отображения формируют вариативное множество $g(X_1, \ldots, X_n)$. По построению *promote* следует, что количество элементов во множестве образов равно числу элементов во множестве декартова произведения вариативных множеств, коих экспоненциальное число:

$\exists \forall i \in \{1, \ldots, n\} : |X_i| = k_i \exists k = k_i$, тогда $|X_1 \times \ldots \times X_n| = \prod_{i=1}^{n} k_i \geq k^n$.

Рис. 3. Пример с подвыражениями, которым соответствуют вариативные множества

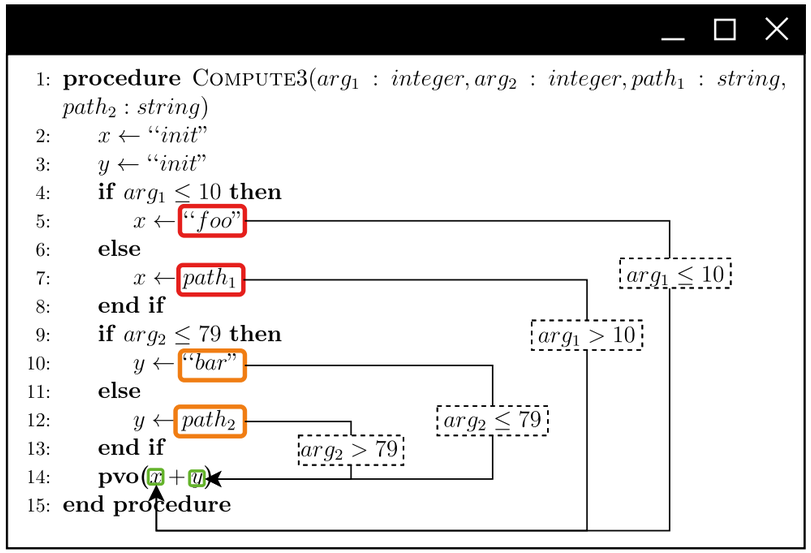
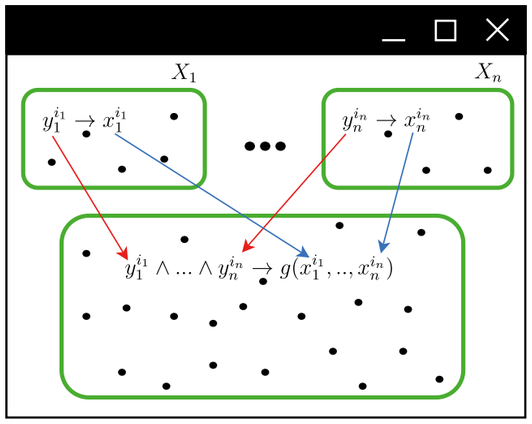


Рис. 4. Отображение promote



Экспоненциальное количество вариантов сильно сказывается на времени выполнения алгоритма анализа, которое вместо желаемых минут может достигать нескольких часов, дней, недель или лет. В реальности мы не готовы так долго ждать завершения анализа и хотим получить результат как можно быстрее. Но, к сожалению, экспоненциальный рост — фундаментальная проблема данного алгоритма, и в общем случае с этим ничего нельзя поделать. Но это не значит, что мы совсем бессильны.

ПУТИ РЕШЕНИЯ

Давайте введем R — функцию, фильтрующую ложные варианты. Тогда

$$\left|R\left(promote\left(X_1\times\ldots\times X_n\right)\right)\right| \le \prod_{i=1}^{n} k_i.$$

В дальнейшем для упрощения записи будем опускать написание *promote*:

$$R\left(promote\left(X_1\times\ldots\times X_n\right)\right) \equiv R\left(X_1\times\ldots\times X_n\right).$$

Предположим, что все рассматриваемые вариативные множества совпали, как показано на рис. 5, то есть

$$g\left(X_1\times\ldots\times X_n\right)\ \forall i, j\in\{1, \ldots, n\}: X_i = X_j = \\ = X \Rightarrow g\left(X_1\times\ldots\times X_n\right) = g(X\times\ldots\times X).$$

Такие случаи будем называть «совпадениями».

Рис. 5. Пример «совпадения»

```
1: procedure COMPUTE4(arg : integer, path : string)
2:     x ← "init"
3:     if arg ≤ 10 then
4:         x ← "foo"
5:     else
6:         x ← path
7:     end if
8:     pvo(x + x)            ▷ x = {arg ≤ 10 → "foo"; arg > 10 → path}
9: end procedure
```

ЭКСПОНЕНЦИАЛЬНОЕ КОЛИЧЕСТВО ВАРИАНТОВ СИЛЬНО СКАЗЫВАЕТСЯ НА ВРЕМЕНИ ВЫПОЛНЕНИЯ АЛГОРИТМА АНАЛИЗА

Тогда мы утверждаем, что количество выводимых вариантов, когда вариативные множества удовлетворяют случаю «совпадение», линейное, а не экспоненциальное:

$$|R(X\times\ldots\times X)| \le k, \text{ где } k = |X|.$$

Далее предположим, что вариативные множества не равны, не совпали, но условные множества этих вариантов (множество условий всех вариантов в вариативном множестве $C(X)$) равны, см. рис. 6 и 7. То есть

$$g(X_1, \ldots, X_n)\ \forall i, j \in \{1, \ldots n\} : i \neq j \mid X_i \neq X_j,$$
$$|X_i| = |X_j| = k,\ C(X_i) = C(X_j),\ |C(X_i)| = |C(X_j)| = k,$$
$$\text{где } C(X_i) = \{y_i^1, \ldots, y_i^k\}.$$

Такие случаи будем называть «условностями».

Тогда утверждается, что количество выводимых вариантов, когда вариативные множества удовлетворяют случаю «условность», тоже линейное, а не экспоненциальное:

$$|R(X_1 \times\ldots\times X_n)| \le k, \text{ где } \forall i \in \{1, \ldots, n\} : k = |X_i| = |C(X_i)|.$$

Рис. 6. «Условность»

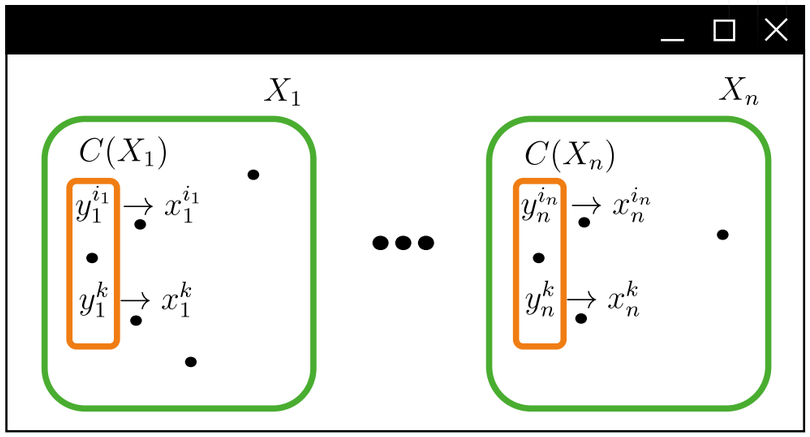


Рис. 7. Пример «условности»

```
1:  procedure COMPUTE5(arg : integer, path : string)
2:      x ← "init"
3:      y ← "init"
4:      if arg ≤ 10 then
5:          x ← "foo"
6:          y ← path
7:      else
8:          x ← path
9:          y ← "bar"
10:     end if
11:     pvo(x + y)          ▷ x = {arg ≤ 10 → "foo"; arg > 10 → path}
                              y = {arg ≤ 10 → path; arg > 10 → "bar"}
12: end procedure
```


РЕАЛИЗАЦИЯ ОПИСАННЫХ СЛУЧАЕВ УСКОРИЛА РАБОТУ АЛГОРИТМА АНАЛИЗА В 2,5 РАЗА

Теперь же предположим, что вариативные множества не равны, их условные множества тоже не равны, у них разное количество вариантов, но все условные множества имеют непустое пересечение, как показано на рис. 8 и 9. То есть

$$g(X_1 \times \ldots \times X_n)\ \forall i, j \in \{1, .., n\} : i \neq j \mid ,$$
$$X_i \neq X_j, |X_i| = k_i,\ C(X_1) \cap\ldots\cap C(X_n) = C, C(X_i) \neq C, |C| =$$
$$= k > 0.$$

Такие случаи будем называть «черпаками».

Тогда мы утверждаем, что количество выводимых вариантов, когда вариативные множества удовлетворяют случаю «черпак», тоже не экспоненциальное, но меньшее значение, что есть сумма:

$$|R(X_1\times\ldots\times X_n)| \le k + \prod_{i=1}^{n}(k_i - k), \text{ где } \forall i \in \{1, .., n\}\ |X_i| = k_i.$$

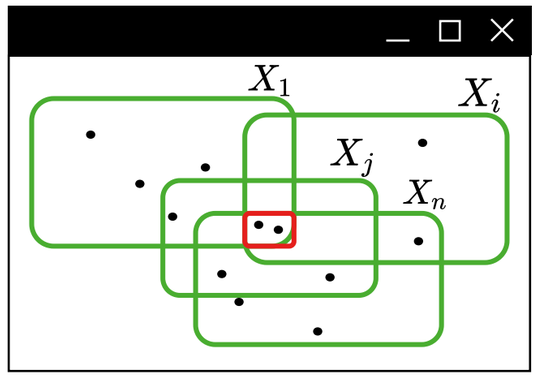


Рис. 8. «Черпак»

Рис. 9. Пример «черпака»

```
1: procedure Compute6(arg1 : integer, arg2 : integer, arg3 : integer,
   path1 : string, path2 : string)
2:     x ← "init"
3:     y ← "init"
4:     if arg1 ≤ 5 then
5:         x ← "foo"
6:     else
7:         x ← path1
8:     end if
9:     if arg2 ≤ 7 then
10:        x ← "bar"
11:        y ← path2
12:    end if
13:    if arg3 ≤ 9 then
14:        x ← path2
15:        y ← "put"
16:    end if
17:    pvo(x + y)                ▷ x = {
                                   arg1 ≤ 5 ∧ arg2 > 7 ∧ arg3 > 9 → "foo";
                                   arg1 > 5 ∧ arg2 > 7 ∧ arg3 > 9 → path1;
                                   arg2 ≤ 7 ∧ arg3 > 9            → "bar";
                                   arg3 ≤ 9                       → path2}

                                   y = {
                                   arg2 ≤ 7 ∧ arg3 > 9            → path2;
                                   arg2 > 7 ∧ arg3 > 9            → "init";
                                   arg3 ≤ 9                       → "put"}
18: end procedure
```

РЕЗУЛЬТАТЫ

Описанные выше утверждения были применены в нашем анализаторе исходного кода PT Application Inspector. В таблице ниже время сканирования исходной версией PT Application Inspector без применения описанных случаев считается за t.

Версия анализатора	Среднее время сканирования проектов
Исходный	t
«Совпадение»	(1/1,7)t
«Условность»	(1/2,2)t
«Черпак»	(1/2,5)t

То есть реализация описанных случаев ускорила работу алгоритма анализа в среднем в 2,5 раза за счет уменьшения количества перебираемых вариантов: мы исключили заведомо недостижимые и ложные. Соответственно, не тратили время и на их постобработку. Стоит отметить, что по ряду технических и исторических причин на момент реализации эти случаи не могли быть полностью покрыты в PT Application Inspector. Из-за этого продемонстрированные результаты не в полной мере показывают, насколько можно ускорить алгоритм анализа за счет наших утверждений.

По определенным причинам мы не можем показать какие-то другие случаи, если они возможны, но скажем, что да, они возможны.

Более подробную версию статьи, включающую доказательства корректности описанных утверждений, вы можете найти на сайте [1].

[1]

КОРРЕЛЯТОР И ~~НАША~~ ЕГО ЭКСПЕРТИЗА

это очень важно!

АВТОРЫ

Станислав Антонов

Руководитель департамента развития технологий
Positive Technologies

Михаил Максимов

Ведущий эксперт департамента развития технологий
Positive Technologies

Время прочтения:	20 минут
Для кого:	ИБ-специалисты, сотрудники SOC, специалисты по разработке средств защиты информации
Прокачиваем знания:	описание событий безопасности и их корреляция, обнаружение вредоносного воздействия на ИТ-инфраструктуру, устройство коррелятора в продуктах Positive Technologies

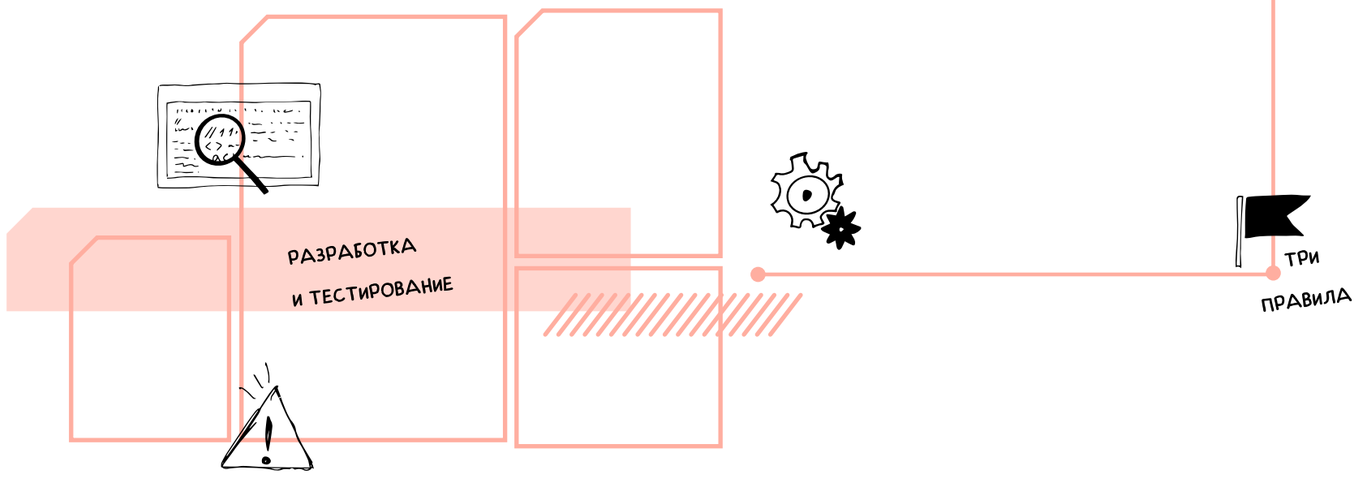
В этой статье мы расскажем о коррелляторе, который используем в продуктах Positive Technologies. В качестве примера возьмем MaxPatrol SIEM. Начнем с того, что именно мы подразумеваем под термином «коррелятор».

Большинство наших ИБ-решений получают на вход поток событий безопасности. Задача коррелятора — выявить в нем определенную последовательность (цепочку) событий в заданном временном окне. Типичный пример — brute force. Мы видим последовательность из нескольких попыток авторизации с некорректным паролем, которая завершается вводом правильных данных. Это пользователь вспомнил пароль или его подобрал злоумышленник? Инцидент нужно зафиксировать и проанализировать, а иногда лучше сразу заблокировать учетную запись.

Чтобы эффективно решать задачу поиска цепочек, входящие события нужно подготавливать. Как минимум, данные из разных источников важно нормализовать — привести к единому виду. Кроме того, события нужно обогащать по заранее подготовленным справочникам. Например, чтобы определять пользователей с правами администраторов и автоматически повышать уровень важности инцидентов. Вернемся к brute force: если свести все события авторизации к общей схеме, можно написать корреляционное правило для определения атаки в любой информационной системе, поставляющей события в коррелятор.

УСТАНАВЛИВАЕМ ПРАВИЛА

Любому продукту ИБ, в основе которого лежит экспертиза, нужны методы доставки экспертных знаний. В нашем случае это язык правил. Мы разработали DSL, с помощью которых специалисты, в том числе сотрудники наших клиентов и интеграторов, могут создавать правила для ряда продуктов Positive Technologies. Соответственно, важным элементом процесса становится SDK. К примеру, вместе с MaxPatrol SIEM поставляется набор консольных инструментов для разработки и тестирования правил.



В идеальном мире экспертиза работает из коробки, но в реальности для корректной реализации правил зачастую требуется дополнительная информация. Например, список контроллеров домена. Соответственно, правилам нужно предоставлять доступ к внешним источникам информации. Во время разработки MaxPatrol SIEM мы пробовали разные варианты реализации этой функциональности:

ТРИ ВАРИАНТА

1. Активные списки — именованные коллекции в Redis. Это простой, дешевый и вполне рабочий подход. Тем не менее в этом случае трудно работать с большими коллекциями и сценариями со сложной структурой.
2. Второй вариант — проекции активов — появился благодаря MaxPatrol 10. Мы выгружали часть данных об активах в Mongo, куда мог ходить коррелятор.
3. Финальным вариантом стала интеграция коррелятора с libfpta — реализация настраиваемых табличных списков. При этом коррелятор использует их не только на чтение, но и на запись. Тем самым мы даем экспертам возможность сохранять определенные данные событий и обращаться к ним в дальнейшем.

Аналогичным образом в коррелятор можно вносить репутационные списки, индикаторы компрометации и другую внешнюю информацию. С точки зрения эксперта работа с данными в правилах будет выглядеть как обращение к табличному списку.

Из чего состоят правила:

Описание событий. У каждого события должен быть фильтр: с его помощью коррелятор понимает, что именно нужно анализировать. Ключ события нужен для идентификации инстанса корреляции. Например, в случае того же brute force в роли ключа может выступать имя пользователя или кортеж «имя пользователя плюс адрес сервера», если нужно ловить атаки на отдельных серверах.

Rule. Описание последовательности, которую нужно найти: события, их количество, порядок и временной интервал.

Результат. Для всех событий в цепочке выполняется on-обработчик, который позволяет совершать разные операции с табличными списками. Сюда же входит emit — результат работы коррелятора.

Теперь о том, как все это реализовано:

1. Фильтры событий и ключи рассчитываются на роутере. Рассчитанный ключ позволяет отправить событие, предназначенное конкретному инстансу корреляции, на нужный шард. На выходе из роутера событие должно получить полный список инстансов и может попасть в разные воркеры коррелятора.

2. При попадании события в коррелятор его содержимое складывается в кэш, а все данные (инстанс, тип, время и идентификатор) обрабатываются воркером. Именно здесь при каждом входящем событии состояние инстанса корреляции пересчитывается на предмет того, было ли правило выполнено. Проверка соответствия каждый раз выполняется заново, потому что у нас нет гарантий того, что события идут на вход последовательно.

3. При выполнении правила корреляции система отправляет сообщение в рендерер. Его задача — получить данные о событии из кэша, выполнить on-обработчики и другие действия в рамках правила. Затем он выдает коррелированное событие наружу.

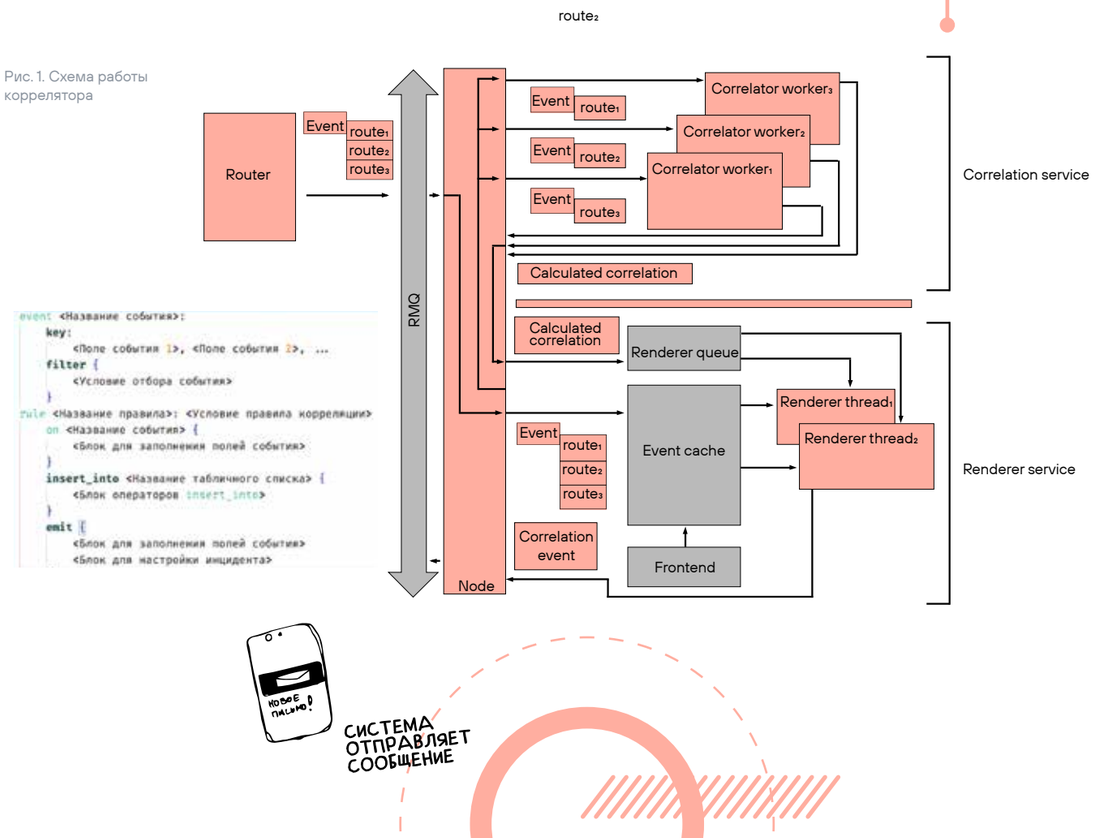


Рис. 1. Схема работы коррелятора

КОРРЕЛЯТОР В MAXPATROL SIEM

Правила нормализации

Приведение всех событий к единому виду для последующей обработки

Правила обогащения

Изменение событий и содержимого табличных списков по заданным условиям

Правила корреляции

Обнаружение интересующей активности путем анализа потока нормализованных событий в соответствии с заданными правилами

Правила агрегации

Объединение одинаковых событий. Фильтрация нормализованных событий по условию

Табличные списки

Массив данных, хранящийся в памяти и доступный для использования в правилах корреляции и правилах обогащения

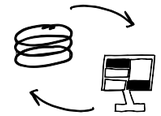


Рис. 2. Экспертный контент в MaxPatrol SIEM

Переходим непосредственно к MaxPatrol SIEM. Система содержит четыре типа правил для соответствующих сервисов, а также табличные списки с данными. Сегодня мы подробно поговорим о правилах нормализации и корреляции. Задача первых — приводить разношерстные события из разных источников к единому виду для последующей обработки. Вторые позволяют выявлять определенные активности в инфраструктуре на основе потока нормализованных событий. Но прежде чем переходить непосредственно к правилам, поговорим о схеме описания нормализованных событий.

Субъектно-объектное взаимодействие в событии

Subject fields
- + **subject:** Enum
- + **subject.name:** String
- + **subject.domain:** String
- + **subject.id:** String
- + **subject.priviliges:** String
- + **subject.group:** String
- + **subject.type:** String
- + **subject.version:** String
- + ...

Action with object fields
- + **action:** Enum
- + **status:** Enum

Object fields
- + **object:** Enum
- + **object.name:** String
- + **object.domain:** String
- + **object.id:** String
- + **object.group:** String
- + **object.type:** String
- + **object.name:** String
- + **object.state:** String
- + **object.path:** String
- + ...

Стороны взаимодействия

Address fields — From
- + **src.fqdn:** String
- + **src.hostname:** String
- + **src.ip:** IPAddress
- + **src.mac:** MACAddress
- + **src.port:** Number
- + **src.geo.city:** String
- + **src.geo.country:** String
- + **src.geo.org:** String
- + ...

Address fields — To
- + **dst.fqdn:** String
- + **dst.hostname:** String
- + **dst.ip:** IPAddress
- + **dst.mac:** MACAddress
- + **dst.port:** Number
- + **dst.geo.city:** String
- + **dst.geo.country:** String
- + **dst.geo.org:** String
- + ...

Информация о самом событии и его источнике

Event source fields
- + **event_src.vendor:** String
- + **event_src.title:** String
- + **event_src.subsys:** String
- + **event_src.category:** Enum
- + ...

Event fields
- + **start_time:** Datetime
- + **time:** Datetime
- + **duration:** Number
- + **importance:** Enum
- + ...

Дополнительная информация о взаимодействии в событии

Count fields
- + **count.bytes:** Number
- + **count.packets:** Number
- + ...

Info fields
- + **protocol:** String
- + **msgid:** String
- + **reason:** String
- + ...

Data fields
- + **datafield1:** String
- + ...
- + **numfield1:** String
- + ...

Рис. 3. Схема описания нормализованных событий в MaxPatrol SIEM

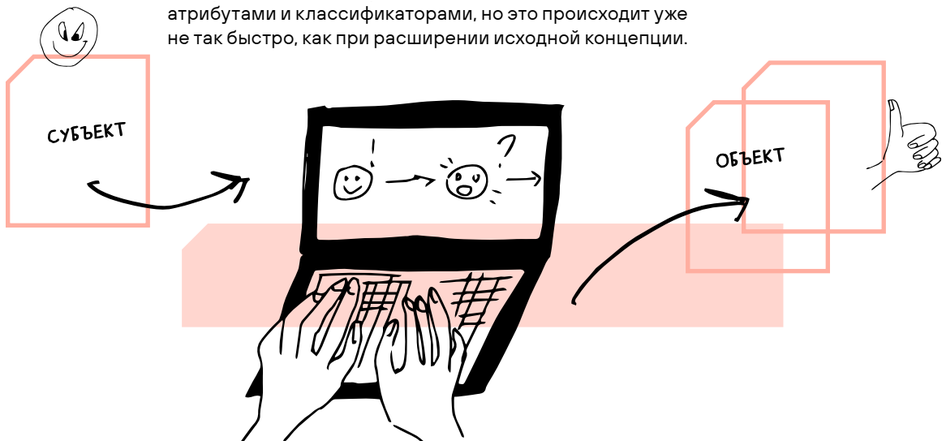
На старте разработки продукта мы использовали концепцию Common Event Expression и постепенно расширяли ее. В первых версиях это был набор сущностей с уникальными атрибутами, в том числе в части именования. По мере увеличения числа поддерживаемых событий у нас неоднократно возникала необходимость в расширении списка сущностей, их атрибутов и более подробной классификации объектов.

Со временем мы наработали существенную базу событий и правил их нормализации, провели анализ схемы и пришли к нескольким выводам:

1. Схема быстро разрастается, и с ней становится все труднее работать.
2. При описании конкретного события одновременно используется лишь малая часть доступных полей и сущностей.
3. В описаниях сущностей много дублирующихся по смыслу атрибутов.

Тогда мы решили по-другому взглянуть на описание события: отказались от уникальных атрибутов для каждой сущности и перешли к двум базовым сущностям — «субъект» и «объект». У каждой из них есть набор атрибутов для описания смыслового взаимодействия в событии, где назначение атрибутов определяется классификацией объекта и субъекта. Кроме того, мы выделили отдельные сущности для описания сетевого взаимодействия и источника события (он не всегда участвует в наблюдаемом взаимодействии).

Новый подход помог сократить количество дублирующихся по назначению полей и упростил процесс работы со схемой. Кроме того, он позволяет расширять список классифицирующих значений для объекта и субъекта без добавления новых атрибутов. Само собой, по мере роста экспертизы и усложнения рассматриваемых сценариев схема пополняется новыми атрибутами и классификаторами, но это происходит уже не так быстро, как при расширении исходной концепции.



ПРАВИЛА НОРМАЛИЗАЦИИ

В основе правил нормализации лежит язык eXtraction and Processing (XP). Он был разработан для создания правил преобразования данных в процессе обработки событий.

Структурно правило нормализации можно представить в виде двух блоков (см. рис. 4).

Первичный парсинг события и проверка выполнения условий отбора

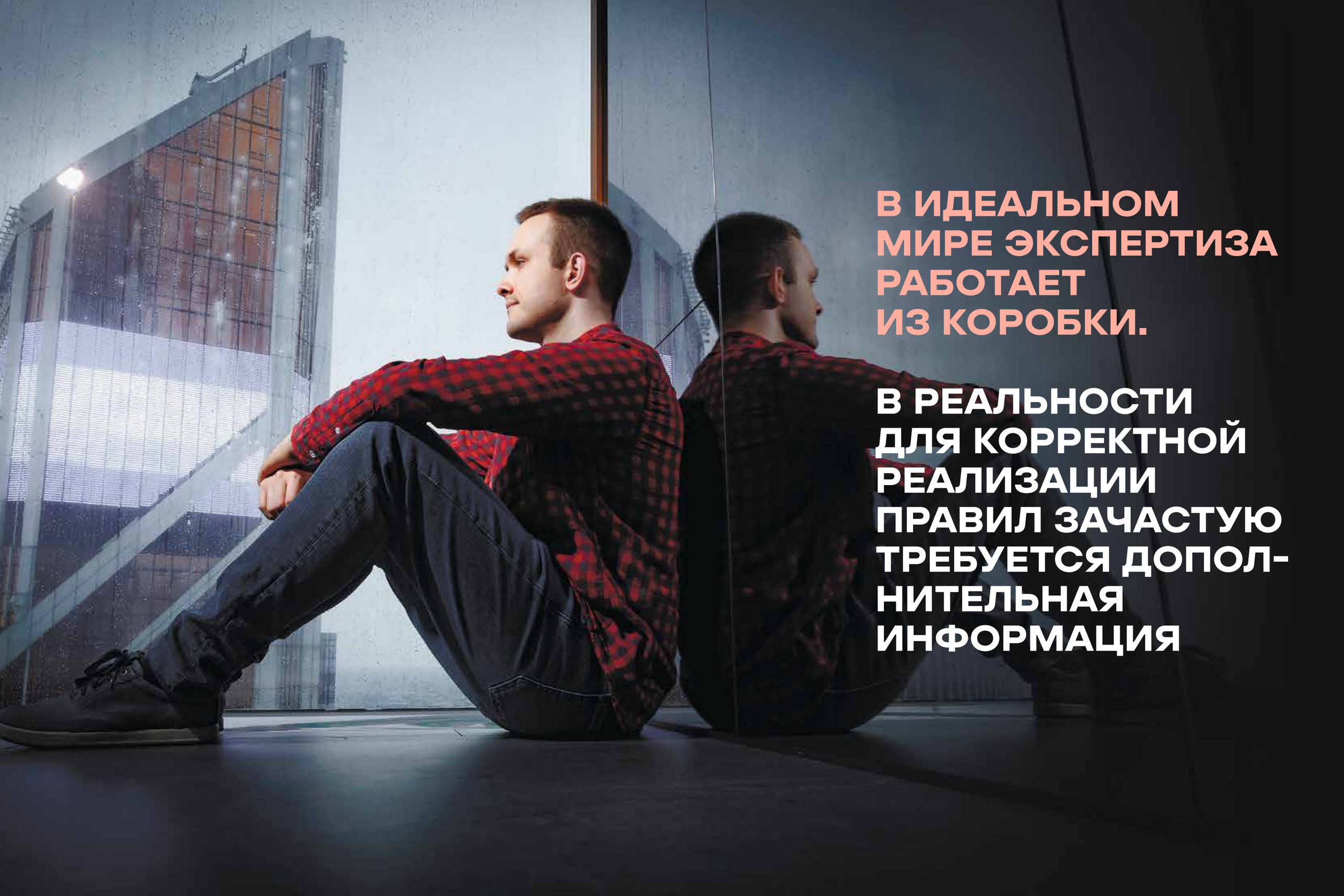
Необходимые преобразования исходных данных и сохранение их в поля схемы, формирование финального вида нормализованного события

Рис. 4. Структура правил нормализации

Если первый выполняется успешно и событие подходит под заданные условия, начинается выполнение второго. Переходим к примерам.

Рис. 5.1. Правила нормализации

```
# <142>Aug 30 20:10:53 stand nginx: 2016/08/30 20:10:53 [info] 29328#0: *16
client 127.0.0.1 closed keepalive connection (104: Connection reset by peer)

TEXT = '{"<"NUMBER">"?}{DATETIME}
        {event_src.ip=IPV4|event_src.ip=IPV6|event_src.hostname=HOSTNAME}
        {"("?}nginx{")"?}{"["NUMBER"]"?}:{time=DATETIME}
        [{WORD}] {NUMBER"#"NUMBER?}: {STRING?}
        client {src.ip=IPV4|src.ip=IPV6|src.hostname=HOSTNAME}
        closed keepalive connection {$reason_raw=REST}'

action = "close"
object = "connection"
status = "success"

object.type = "keepalive"

dst.ip = event_src.ip
dst.hostname = event_src.hostname

if $reason_raw != null and $reason_raw != "" then
    reason = strip($reason_raw, "(", ")")
endif

direction = "egress"
importance = "info"

category.generic = "Connection"
category.high = "Network Interaction Management"
category.low = "Communication"

event_src.vendor = "opensource"
event_src.title = "nginx"
event_src.category = "Web server"

id = "PT_Opensource_Nginx_syslog_client_closed_keepalive_connection"
```



В ИДЕАЛЬНОМ МИРЕ ЭКСПЕРТИЗА РАБОТАЕТ ИЗ КОРОБКИ.

В РЕАЛЬНОСТИ ДЛЯ КОРРЕКТНОЙ РЕАЛИЗАЦИИ ПРАВИЛ ЗАЧАСТУЮ ТРЕБУЕТСЯ ДОПОЛ-НИТЕЛЬНАЯ ИНФОРМАЦИЯ

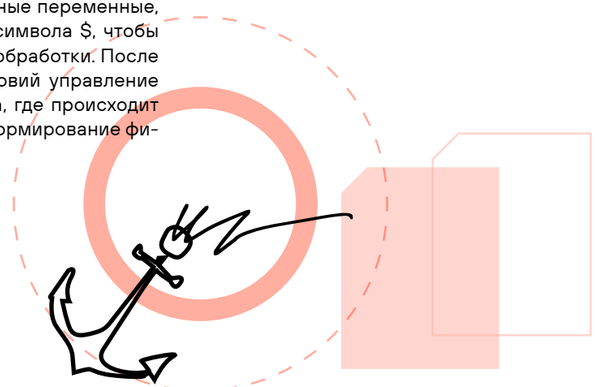
Рис. 5.2. Пример нормализованного события

```
{
  "action": "close",
  "object": "connection",
  "status": "success",
  "category.generic": "Connection",
  "category.high": "Network Interaction Management",
  "category.low": "Communication",
  "direction": "egress",
  "dst.hostname": "stand",
  "event_src.category": "Web server",
  "event_src.hostname": "stand",
  "event_src.title": "nginx",
  "event_src.vendor": "opensource",
  "id": "PT_Opensource_Nginx_syslog_client_closed_keepalive_connection",
  "importance": "info",
  "object.type": "keepalive",
  "reason": "104: Connection reset by peer",
  "src.ip": "127.0.0.1",
  "time": "2016-08-30T20:10:53Z"
}
```

На рис. 5.1 и 5.2 показан разбор текстового неструктурированного лога от nginx, сообщающего о закрытии соединения. Первичный разбор события и проверка условий отбора выполняются с помощью форматной строки TEXT. Ее основными элементами являются типизированные токены, которые позволяют разбирать меняющиеся от события к событию участки без использования сложных регулярных выражений. Другой важный элемент TEXT — якоря в виде текстовых данных, которые постоянны для рассматриваемого события (подсвечены зеленым цветом).

У событий могут быть опциональные участки. Задавать опциональность и ожидание альтернативных данных можно в форматной строке с помощью специальных конструкций. В представленном примере знаки вопроса используются для обозначения опциональных участков, а вертикальные черты — альтернативных методов разбора.

Если данные, извлеченные в форматной строке, не требуют дополнительной обработки, их можно сразу поместить в поля нормализованного события. В противном случае их нужно сохранить во временные переменные, наименование которых начинается с символа $, чтобы иметь к ним доступ для последующей обработки. После первичного парсинга и проверки условий управление передается в основную часть правила, где происходит дополнительная обработка данных и формирование финального нормализованного события.



Периодически встречаются события, содержащие данные сразу в нескольких форматах. Например, это может быть текстовый лог, часть которого представляет собой структурированные данные JSON. Или же в структурированном событии табличного вида (плоский JSON) в одном из полей может находиться полноценная XML-структура, из которой нужно извлечь определенные значения.

Для подобных случаев предусмотрен механизм, позволяющий обрабатывать части событий как самостоятельные целевые события. Фактически это вложенное правило нормализации, которое описывается внутри конструкции из ключевых слов subformula и endsubformula. Вызов обработки осуществляется с помощью функции submessage.

Рис. 6.1. Нормализация события, содержащего несколько форматов данных

```
TABULAR =  "{time=timegenerated},
            {subject.name=objectpath},
            {subject.group=publishername},
            number,
            {event_src.subsys=channel},
            {event_src.hostname=loggingcomputer},
            eventdata,
            {datafield2=rulename}
            "

COND = $publishername == "Health Service Modules" and $number == 10353

submessage("XML","Event_data",$eventdata)
subformula "Event_data"
    XML = 'DataItem'
    object.name = $DataItem / eventdata / Data [1]   # имя рабочего процесса
    object.property = "namespace"
    object.value = $DataItem / eventdata / Data [4]  # подпространство имён процесса
endsubformula
```

На рис. 6.2 и 6.3 показана обработка данных разными функциями. Таким образом можно привести информацию к подходящему для дальнейшей работы формату или представить в удобном для безопасника виде.

Рис. 6.3. Приведение данных к удобному виду

```
$cp_severity = $kv["cp_severity"]
if ($cp_severity != null) then
    importance = switch lower($cp_severity)
        case "informational" "info"
        case "low" "low"
        case "medium" "medium"
        case "high" "high"
        case "critical" "high"
    endswitch
else
    importance = "medium"
endif
```

Рис. 6.2. Сопоставление данных с помощью условных операторов

```
time = epoch_ms_to_datetime(number($kv["rt"]))

if ($kv["fileType"] != null) then
    $tmp_reason_description = "File type: " +
    $kv["fileType"]
endif
if ($kv["flexString2"] != null) then
    $tmp_reason_malware= "Malware action: " +
    $kv["flexString2"]
endif
if ($kv["portal_message"] != null) then
    $tmp_reason_message = "Portal message: " +
    $kv["portal_message"]
endif
reason = join(remove([$tmp_reason_description, $tmp_reason_attack,
$tmp_reason_message], null), ". ")
```

Рис. 7. Структура правила корреляции

ПРАВИЛА КОРРЕЛЯЦИИ

Кейс № 1

Начнем с простого кейса по обнаружению попытки создания и удаления учетной записи в течение короткого интервала времени. Для этого нужно фиксировать два события. Поскольку мы рассматриваем в качестве источника Windows, их можно поймать с помощью поля msgid, в котором идентификатор события сохраняется в терминологии источника. Обратите внимание на ключевое слово key: с его помощью задаются значения, по которым события будут группироваться в инстанс. Отметим, что при формировании ключа инстанса названия

Рис. 8. Фильтры событий

```
event User_add_in_local_system:
    key:
        event_src.host, object.account.name
    filter {
        filter::NotFromCorrelator()
        and (
                msgid == "4720"
                or msgid == "624"
            )
        and event_src.title == "windows"
        and filter::CheckWL_Specific_Only( ... )
    }
```

```
event User_delete_in_local_system:
    key:
        event_src.host, object.account.name
    filter {
        filter::NotFromCorrelator()
        and (
                msgid == "4726"
                or msgid == "630"
            )
        and event_src.title == "windows"
    }
```

полей неважны — роль играют только их значения. Таким образом, в инстанс можно собирать события из разных полей, главное, чтобы они имели одни и те же значения и порядок.

В коде на рис. 8 используется ключевое слово «filter::». Оно вызывает макросы — участки фильтров, вынесенные в отдельные объекты для переиспользования в правилах корреляции и обогащения. В данном кейсе применяются два макроса. NotFromCorrelator отсеивает корреляционные события. CheckWL_Specific_Only является составной частью большого экспертного механизма исключений и скрывает в себе логику обращений к нескольким табличным спискам. Этап фильтрации событий происходит в сервисе роутера.

Рис. 9. Описание цепочки событий и временного интервала

```
rule Fast_Create_and_Delete_Account: (User_add_in_local_system -> User_delete_in_local_system) within 1h
```

Нам нужно обнаружить строгую последовательность событий — сначала создание, затем удаление пользователя, поэтому при описании последовательности мы используем оператор строгого следования «->». Также мы задаем временной интервал в 1 час — для этого существует несколько операторов. В данном случае применяем within: он используется с цепочками, в которых однозначно определено количество ожидаемых событий (без опциональных).

Цепочка собирается в сервисе коррелятора.

Рис. 10. Обработка исходных событий и конструирование коррелированного события

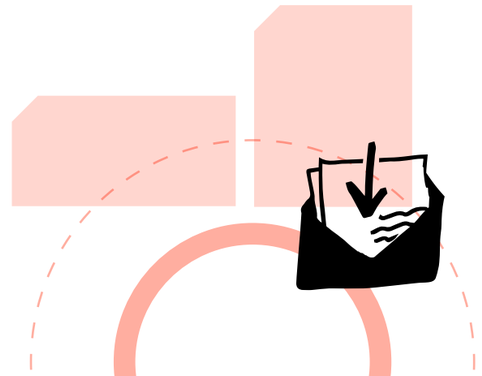
```
on User_add_in_local_system {
    $subject = subject
    $object = object
    $action = action
    $status = status
    $subject.account.id = subject.account.id
    $subject.account.name = subject.account.name
    $subject.account.domain = subject.account.domain
    ...
    $object.account.id = object.account.id
    $object.account.name = object.account.name
    $object.account.domain = object.account.domain
    ...
    $event_src.host = event_src.host
    ...
}
```

```
on User_delete_in_local_system {
    # Пользователь, удаливший учетную запись
    $datafield2 = subject.account.name
    ...
}
```

```
emit {
    $correlation_type = "event"

    $category.generic = "Attack"
    $category.high = "Persistence"
    $category.low = "Local Account"

    $importance = "high"
}
```

После сбора цепочки начинается обработка исходных событий и формирование корреляционного. Для этого используются on-обработчики, в которых может выполняться произвольный XP-код (в том числе для получения доступа к данным табличных списков). Пользователь может обратиться к полям исходного события, временным переменным в рамках инстанса корреляции, а также к полям результирующего события, выделенным знаком $. В данном случае код обработчика максимально прост: мы сохраняем нужные данные исходных событий в результирующее.

Правило завершает блок emit: если цепочка собрана успешно, он всегда выполняется последним. С его помощью мы получаем доступ к полям корреляционного события и временным переменным.



ОПИСАНИЕ КЕЙСА

Кейс № 2

Второй кейс: если пользователь обладает правами локального администратора на сервере сертификации, он может экспортировать сертификат CA и выписывать сертификаты для других пользователей домена. Как это отследить?

Рис. 11. Фильтры событий

```
event Key_Migration:                                  Основное событие,
    key:                                              факт экспорта
        event_src.host                                сертификата
    filter {
        filter::NotFromCorrelator()
        and event_src.title == "windows"
        and msgid == "5059"
        and lower(object.type) == "machine key"
        and match(lower(object.name), "*.ca")
        and filter::CheckAL_Specific_Only( ... )
    }

                                                      Опциональное событие,
event Key_Share_Access:                               доступ к объекту общего
    key:                                              сетевого ресурса
        event_src.host
    filter {
        filter::NotFromCorrelator()
        and event_src.title == "windows"
        and msgid == "5145"
        and match(lower(object.name), "*.pfx")
    }

event Certutil_BackUpKey:                             Опциональное событие,
    key:                                              запуск утилиты certutil для
        event_src.host                                экспорта сертификата
    filter {
        filter::NotFromCorrelator()
        and filter::ProcessStart_Windows("certutil.exe")
        and find_substr(lower(object.process.cmdline), "-backupkey") != null
    }
```

Первое и главное событие (обязательное) — факт экспорта сертификата. В поле msgid оно будет иметь значение 5059. В условиях отбора задается дополнительная фильтрация по полям object.type и object.name. Второе событие (опциональное) — факт запуска утилиты certutil с параметром backupkey для экспорта сертификата. В рамках фильтра используем еще один экспертный макрос — для отлова событий запуска процессов в Windows. В качестве параметра передаем ему наименование исполняемого файла утилиты. Еще одно опциональное событие — факт доступа к объекту сетевого ресурса. Оно нужно для случаев, когда экспорт сертификата выполняется удаленно. В рамках этого фильтра нас интересуют файлы с расширением .pfx. Ключ для группировки событий максимально прост: это узел, на котором они произошли.

Рис. 12. Описание цепочки событий и временного интервала

```
rule CA_Cert_Export: (Certutil_BackUpKey? and Key_Migration and Key_Share_Access?) timer 2s
```

Рис. 13. Обработка исходных и конструирование результирующего события

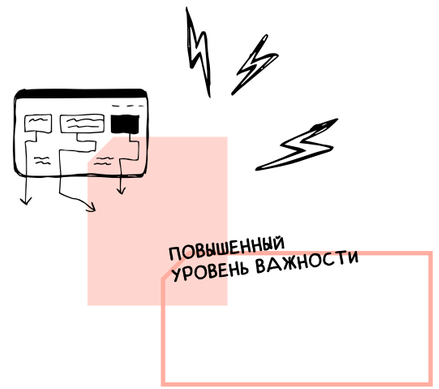
```
on Certutil_BackUpKey { # Забираем из события информацию об аккаунте, процессе
    ...
    $importance = "high"
    # Смотрим по характерным признакам, был ли использован определенный
    # инструмент, запускающий certutil
    if find_substr(lower(object.process.cmdline), "certipy") != null then
        $correlation_type = "incident"
        $alert.context = $alert.context + " | known tool: certsync"
    endif
}

on Key_Migration {
# Забираем из события имя сертификата
    ...
    if $importance == null then
        $importance = "medium"
    endif
    if $correlation_type == null then
        $correlation_type = "event"
    endif
}

on Key_Share_Access {
# Забираем информацию об атакующем при удаленном экспорте
# Событие характерно при использовании определенных инструментов
    $src.ip = src.ip
    ...
    $subject.account.name = subject.account.name
    ...
}

emit {
    $subject = "account"
    $action = "extract"
    $object = "certificate"
    ...
    $category.generic = "Attack"
    $category.high = "Credential Access"
    $category.low = "Steal or Forge Authentication Certificates"
}
```

Переходим к описанию цепочки. В данном случае неважно, в каком порядке придут события, поэтому вместо оператора строгого следования применяем and. Поскольку два из трех событий опциональны, помечаем их фильтры квантификатором «?». Для задания временного интервала используем ключевое слово timer. Timer отличается от within тем, что после взведения он будет ожидать заданное количество секунд (даже после прихода всех обязательных событий), чтобы отловить в потоке опциональные события.

Далее идут обработчики. Из обязательного события Key_Migration мы можем забрать только имя сертификата. Также заполняем поля важности и типа корреляции, если событие попало в цепочку первым и им еще не были присвоены значения.

Поймали событие запуска процесса утилиты certutil? Повышаем уровень важности до высокого в принудительном порядке, даже если обязательное событие Key_migration уже заполнило это поле. Также анализируем поле object.process.cmdline на наличие признаков, по которым можно определить использованные инструменты. Значение certipy будет признаком утилиты certsync — при экспорте она задает сертификату такое имя. При наличии этого признака устанавливаем тип корреляционного события в incident и записываем информацию об использованном инструменте в alert.context.

Таким образом, опциональное событие в цепочке может изменить важность корреляционного. В этом же обработчике сохраняем в результирующее событие информацию об аккаунте и о процессе. Остается последнее опциональное событие, возникающее при удаленном экспорте сертификата. Из него мы забираем информацию об атакующем.



Кейс № 3

Еще один кейс: обнаружение факта использования ВПО для выгрузки информации из контроллера домена. В этом случае рассмотрим только цепочку. Нас интересует комбинация квантификатора с верхней открытой границей «+» и оператора with_different, позволяющего задать поле, значение которого для всех пойманных событий должно различаться.

В инстанс попадут все подходящие под фильтр DNSZone_Query события, у которых различаются значения поля object.query. Если в потоке встретится несколько событий с одинаковым значением, в инстанс попадет только одно.

```
rule Domain_Dump_Tools_via_LDAP: (Schema_Query and DNSZone_Query+ with different object.query) timer 10s

key:
    event_src.host, src.ip, src.port, subject.account.name
```

Рис. 14. Обнаружение использования ВПО для выгрузки данных из контроллера домена

ВОЗМОЖНОСТИ ПРАВИЛ КОРРЕЛЯЦИИ

Реализация цепочек разной сложности: от одного события с заданной фильтрацией до сложных комбинаций фильтров с указанием последовательности, квантификаторов и дополнительных условий.

Использование подхода каскадных корреляций. Сложные и длительные кейсы можно разделить на этапы в виде подправил, результаты работы которых отлавливаются вышестоящим правилом. Таким образом можно осуществлять промежуточную фиксацию.

Выделение тематических участков фильтров правил корреляции в отдельные макросы для удобного переиспользования в неограниченном количестве правил. Функциональность избавляет от дублирования кода, помогает быстрее разрабатывать новые правила и позволяет централизованно вносить корректировки в несколько правил сразу.

Реализация в обработчиках логики любого уровня сложности (в рамках языка XP).

Возможность составления сложных запросов к табличным спискам, включая агрегатные функции. Запросы можно вызывать в фильтрах или обработчиках — чтобы извлечь из табличных списков нужную информацию и сохранить в поля корреляционного события.

ЧТО ЕЩЕ ПОЧИТАТЬ:

Схема полей событий

Язык eXtraction and Processing

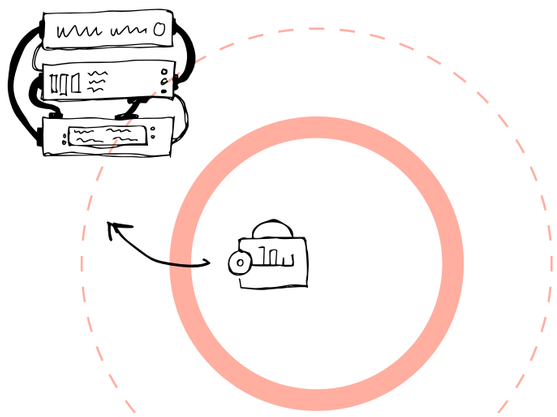
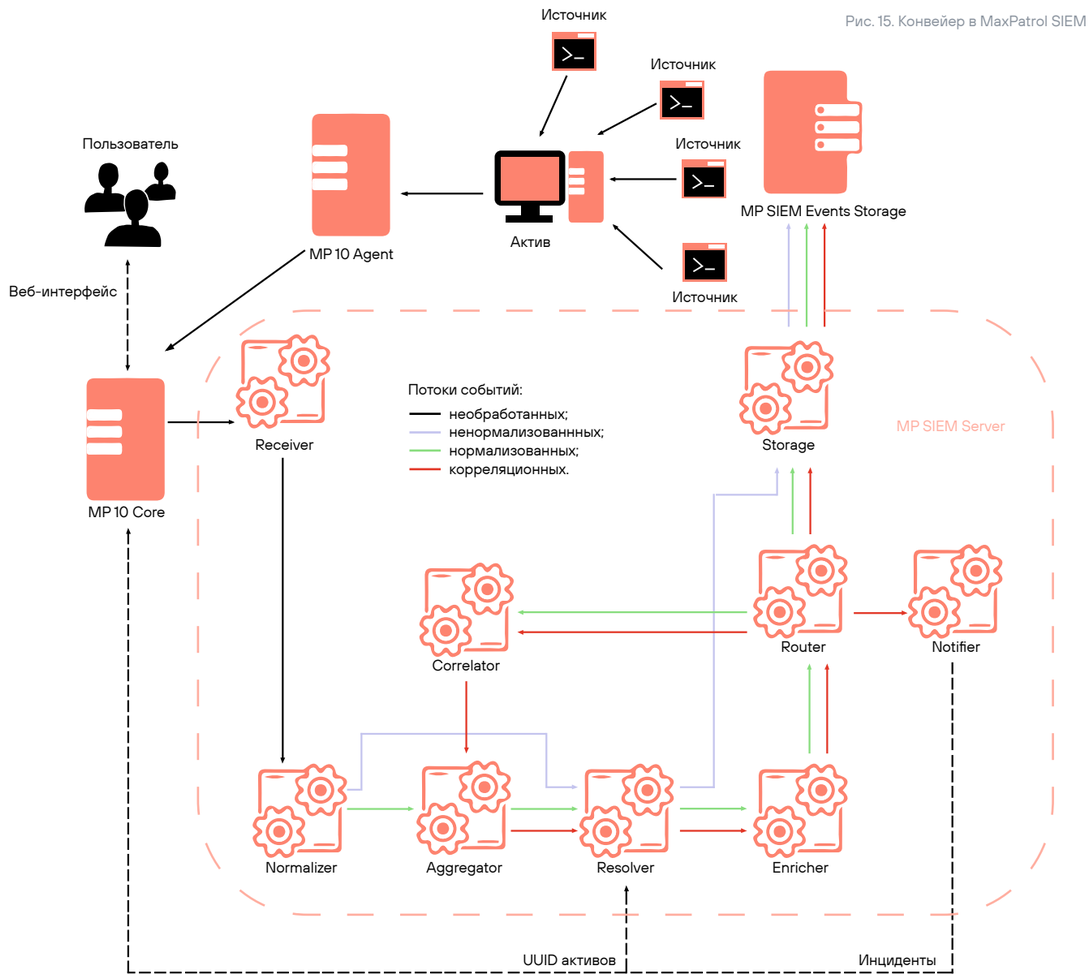


Рис. 15. Конвейер в MaxPatrol SIEM



Конвейер в MaxPatrol SIEM

Коррелятор является частью большого конвейера, который мы интегрируем в наши решения. Рассмотрим MaxPatrol PT SIEM.

Помимо нормализации, агрегации и обогащения, события также проходят этап привязки к активам (resolver). События из коррелятора идут обратно в агрегатор и могут использоваться в других корреляциях. Таким образом, эксперты строят цепочки из нескольких корреляций, выделяя общие части (например, авторизацию в каком-либо сервисе или создание пользовательской сессии), которые можно оформить как отдельные коррелированные события. Отдельно вынесен сервис нотификаций, который может сформировать в продукте инцидент из специальным образом созданного события.

При написании собственной экспертизы для MaxPatrol SIEM нужно помнить, что одним-единственным правилом корреляции можно легко устроить цикл. Защищаемся мы от подобных ситуаций мониторингом. Если правило генерирует слишком много событий, оно может быть выключено автоматически. При этом графы коррелятора будут физически пересобраны без него, а пользователь увидит предупреждение в интерфейсе.

Элементы нашего конвейера используются в разных продуктах Positive Technologies. Самый полный комплект — в PT SIEM и PT XDR. Нормализатор и коррелятор применяются в PT ISIM и PT Sandbox, агрегатор — в PT AF. В открытый проект SOLDR конвейер поставляется в виде freeware-модуля.

БИЗНЕС-ТРЕК

BIZNES-TREK

Zero Trust — это не только отношение к сетям, к АС-кам и т. д. Это про то, что ты не доверяешь полупроводнику, не доверяешь пользователю, не доверяешь сети, не доверяешь программе, не доверяешь устройству, не доверяешь микрофону в своем телефоне. Вот что такое Zero Trust. Есть у тебя что-то свое, нет у тебя чего-то своего — все равно ты ему не веришь. Это о том, что, даже если ты сам собрал компьютер в гараже и выпустил его в интернет, ты уже ему не доверяешь — с учетом того, что он просто попал не в твою среду. Какая разница, на каком чипе он был создан? Здесь все равно, какой полупроводник использовался.

Zero Trust

Сергей Голованов

Главный эксперт «Лаборатории Касперского»

Подсказка всем — и заказчикам, и исполнителям: кладите в контракт отдельной строкой прямую ответственность в случае инцидента (20%, 30% и т. д.) и при этом повышайте цену контракта. Это новый уровень обязательств, и это очень правильная история. Вам гораздо проще будет объяснить руководителю, почему теперь контракт на ИБ стоит больше денег: потому что исполнитель отвечает за совсем другой результат. Если мы начнем закладывать в контракты такие компенсации, у нас постепенно появится страховой рынок. А страховой рынок — это уже про уровень зрелости всей нашей отрасли.

Владимир Бенгин

Директор департамента по ИБ Минцифры РФ

4 трлн

Мы пытаемся построить систему безопасности, не зависящую от человека. Если завтра меня уволят вследствие утраты доверия, система не должна порушиться. При этом уровень моей ответственности довольно серьезный: чтобы вы понимали, у нас 4 трлн руб. оборота в день. Я шел, «упал – очнулся – гипс». И что, мы эти 4 трлн теперь потеряем?

Сергей Демидов

Директор департамента операционных рисков, информационной безопасности и непрерывности бизнеса ПАО «Московская Биржа»

Мне известны случаи, когда злоумышленники полтора года сидели в инфраструктуре и наблюдали. Самое главное, что причиной подавляющего большинства инцидентов становится не использование каких-то серьезных инструментов, zero day, супертехник и тактик, а банальный ИТ-шный бардак. Антивирус есть, но не обновляется. На операционную систему не ставятся патчи, второй фактор не прикручен, админки торчат наружу, никакого управления доступом в помине нет. Я уже не говорю про SSDLC.

Алексей Волков

Вице-президент, директор по информационной безопасности «ВКонтакте»

ГОНКА ЗА ЭФФЕКТИВНОСТЬЮ,

ИЛИ

ОТКУДА БЕРУТСЯ ОШИБКИ В КОДЕ

Дмитрий Скляров

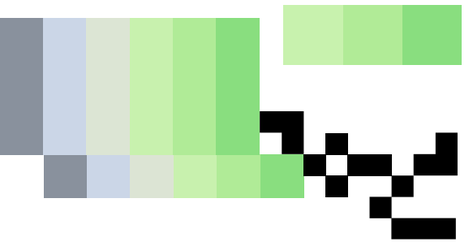
Руководитель отдела анализа приложений
Positive Technologies

Время прочтения: 10 минут

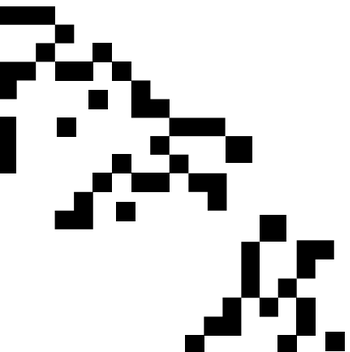
Для кого: разработчики, реверсеры

Прокачиваем знания: безопасная разработка, устранение ошибок в коде

Сегодня мы поговорим о причинах появления ошибок в коде. Ниже я попытаюсь объяснить, что видит и думает реверс-инженер, когда анализирует чужую программу. Спойлер: вы смиритесь с тем, что ошибки неизбежны.



НЕДОРОГОЙ РАЗРАБОТЧИК, СПОСОБНЫЙ ПИСАТЬ ХОРОШИЙ КОД, — ЭТО МИФ



НЕОПРАВДАННЫЕ ОЖИДАНИЯ

Ошибки в коде — прямое следствие гонки за эффективностью. В коммерческой разработке код пишут ради денег. Само собой, чем «эффективнее» программисты это делают, тем больше зарабатывает компания. При этом 95% кода, который создается сейчас, уже было кем-то и где-то написано. Отсюда возникает еще одна проблема — копипаст. На Stack Overflow можно найти массу кода и использовать его в своих проектах. Да, это удобно и быстро, но вы же не станете брать материал для научной работы из «Википедии»... Кроме того, разработчики зачастую копируют куски своего же кода из одной части программы в другую, после чего в ней возникают всевозможные сайд-эффекты.

На все эти нюансы накладываются неоправданные ожидания бизнеса:

- Разработчик должен быть недорогим.
- Он должен быстро писать код.
- Код должен быть легким в плане чтения и тюнинга. Все должно работать правильно и требовать минимума ресурсов.

Прежде чем вы начнете смеяться и позовете посмеяться коллег, поясню: само собой, все работает не совсем так. В большинстве случаев приходится выбирать: либо качественно, либо быстро, либо дешево. В первую очередь бизнес готов поступиться ресурсами, потому что, если у пользователя все тормозит, это, как известно, проблема пользователя :) Идем дальше: писать юнит-тесты — дорого и медленно, поэтому многие стремятся снизить их количество. Также код можно сделать не очень модифицируемым: «Почему бы и нет, все равно ведь работает». Наконец, все мы понимаем, что недорогой разработчик, способный писать хороший код, — это миф. Если он не допускает ошибок, скорее всего, вы просто их не нашли.

От трех ожиданий обычно остается одно: программист должен быстро писать код, который решает поставленную задачу. В нем есть ошибки, его неудобно читать и модифицировать? Это вторично, главное — скорость. Как ни странно, в этом есть логика: выводя на рынок новое решение, компания в первую очередь стремится занять нишу, а уже потом начинает думать о качестве и безопасности продукта.

КАК БОРОТЬСЯ С ОШИБКАМИ

Mitigation, или снижение возможности ошибки. Возьмем, к примеру, метод stack canaries. Суть в том, чтобы обнаружить разрушение стека и не дать неправильно написанной программе выполнить недопустимые действия. «Канарейка», конечно, не спасет от падений ПО, но помешает атакующему сделать что-то критичное. Или другой пример — аппаратные решения Data Execution Prevention и Control-Flow Enforcement.

С одной стороны, эти подходы не сокращают количество ошибок в коде. С другой — они реализуются в компиляторе или на аппаратном уровне, поэтому, с точки зрения разработчиков, идут for free. Даже если люди будут допускать столько же ошибок, как и раньше, ПО станет безопаснее.

Автоматизированный поиск ошибок. Разработчики давно поняли, что поиск ошибок в коде нужно автоматизировать, и начали создавать подобные инструменты. Уже в 1978 г. появился Lint — статический анализатор для C, который сообщал о подозрительных или непереносимых на другие платформы выражениях. Сегодня на рынке полно технологий, которые помогают выявлять дыры в коде на ранних этапах разработки. Например, SAST (static application security testing), DAST (dynamic application security testing) или IAST (interactive application security testing).

Правильный выбор языка. На мой взгляд, не бывает хороших и плохих языков. Главное — правильно подбирать решение под конкретную задачу. Можно начать небольшой проект на Haskell, но если бизнес захочет быстро нарастить объемы и нанять еще десять разработчиков, пищущих на этом языке, вы их просто не найдете.

Другой пример: C заточен на быстродействие, а Java фокусируется на устойчивости и безопасности. Первый позволяет не писать свой обработчик исключения, а для второго это обязательно. Java подразумевает, что разработчик должен прописывать весь возможный контроль — это заложено в саму концепцию языка, поэтому он хорошо подходит для разработки устойчивого корпоративного софта.

Изучение ООП до изучения ООП. Звучит странно, пока не расшифруем аббревиатуры. ООП — это не только объектно-ориентированное программирование, но и объектно-ориентированное проектирование. Разработчикам дали мощный инструмент в виде объектно-ориентированного программирования, но не научили правильно строить структуры классов. Некоторых посещают светлые мысли: «Все функции, у которых разные наборы аргументов, я назову одинаково! У меня всегда будет вызываться функция с одним и тем же именем, и программа будет работать». Идея полиморфизма в том, что имплементация скрыта в момент использования, но две функции с идентичным названием должны делать одно и то же. Если при этом первая принимает один аргумент, а вторая два, это странное решение. Люди часто допускают ошибки, потому что не знают или забывают, как нужно использовать логику языка.

Перестраховка. Не буду подробно останавливаться на defensive programming и secure coding (подходах, в которых вы пытаетесь предусмотреть вообще все), лучше расскажу про secure execution. В своей практике я встречал два вида его реализации. Первый достаточно прост: программа одновременно выполняется на трех вычислителях, и результаты их работы сравниваются между собой. Если совпадут хотя бы два, считается, что мы получили решение на очередном шаге.

Второй подход мне нравится больше: я встречал его в проектах, связанных с безопасностью железнодорожной автоматики. Существует специализированный язык Sternol, заточенный на проверку отсутствия конфликтов при взаимодействии группы дискретных устройств. В частности, на нем описывают топологию железнодорожных элементов: путей, семафоров, стрелок и др. Компилятор Sternol генерирует два набора правил, которые проверяют, не возникнет ли вероятность столкновения поездов после изменения состояния автоматики. Далее берутся две разные аппаратные платформы, например PowerPC и ARM. Разные компиляторы используются для сборки кода от препроцессора Sternol под эти платформы. Получившиеся программы независимо проверяют состояния и выдают ответ: безопасно решение о переключении автоматики или нет. Если да, стрелка или семафор переключаются. В противном случае ничего не происходит — поезда могут остановиться (но точно не столкнутся). В привычной нам корпоративной среде настолько параноидальные подходы используются редко, потому что это слишком дорого.

SSDLC (secure software development lifecycle). Это концепция разработки, еще один набор правил и техник, следуя которым можно снизить вероятность попадания ошибок в релиз.

«КЛИЕНТ НИКУДА НЕ ДЕНЕТСЯ»

Почему совсем избавиться от ошибок невозможно? Приведу простой пример. Еще в 1996 г. вышла книга на 368 страницах, посвященная одной операционной системе и одной задаче — разработке многопоточных приложений. Сегодня такие предложения пишут все, но, чтобы правильно сделать это хотя бы на одной ОС, нужно изучить почти 400 страниц текста. Само собой, проверить статическим анализом факт неправильного использования потока невозможно, поскольку это динамика. Но и динамический анализ не гарантирует, что программа не содержит логических ошибок, а все потоки правильно синхронизированы. Подобные проверки нельзя автоматизировать: пока программист не будет на 146% понимать, что он делает, ошибок не избежать.

Аналогичные сложности возникают и с криптографией. Чем они отличаются от других ошибок в коде? Тем, что криптографию не видно. Если отпустить неправильно работающую программу, она упадет. Если плохо написать криптографию, зашифрованные данные кто-то сможет расшифровать, но программа при этом будет работать.

Я ЗНАЮ ТОЛЬКО ДВЕ КОМПАНИИ, КОТОРЫЕ БЫСТРО ЛАТАЮТ ДЫРЫ, — MICROSOFT И APPLE

Еще одна причина связана со спецификой крупного бизнеса, а точнее, с реакцией больших компаний на обнаруженные уязвимости. На одной из зарубежных конференций ИБ-эксперт Siemens сказал мне: «Ваша компания присылает много репортов. Они классные, и мы все понимаем. Но и вы ведь понимаете, что исправления выходят только через год...» Я знаю только две компании, которые быстро латают дыры: Microsoft и Apple. В остальных случаях схема выглядит примерно так:

1. Компания получает репорт об ошибке, разбирает его и сообщает в разработку, что нужно исправить.
2. Примерно через месяц разработка отвечает, что сейчас они загружены релизом.
3. Когда заканчивается очередной цикл, разработка говорит, что в приоритете составленный бизнесом бэклог, поэтому другие задачи на следующий цикл они не возьмут.
4. Спустя еще один цикл специалисты наконец-то берут кейс в разработку, выпускают исправленную версию и вендор публикует патч. Как правило, примерно через год после первого сообщения об уязвимости.

Также отмечу, что в большинстве лицензионных соглашений прописано что-то из серии: «Если хотите сделать реверс-инжиниринг нашего продукта, вам нельзя». Особенно хитрые вендоры пользуются нашей политикой responsible disclosure, согласно которой мы не раскрываем информацию об уязвимостях до их устранения. Если мы найдем дыру в коде и в тот же день опубликуем детали, злоумышленники сразу начнут ее эксплуатировать. В итоге пострадают клиенты, поэтому мы всегда ждем патчей. Но, с точки зрения производителя, ситуация выглядит примерно так: «Пока мы не выпустим фикс, об уязвимости никто не узнает, значит, мы защищены». В итоге патчи просто не выходят.

Более того, однажды я услышал от вендора следующее: «Ну да, уязвимость. И что? Клиенты не сбегут, потому что перестройка бизнеса на другой продукт обойдется дороже, чем ущерб от уязвимости. Клиент посчитает деньги и никуда не денется, а значит, причин волноваться нет».

ЧТОБЫ ПОЛУЧИТЬ ХОРОШЕЕ ПО, МЕНЕДЖЕРАМ НУЖНО НАУЧИТЬСЯ ЖЕРТВОВАТЬ ЭФФЕКТИВНОСТЬЮ

1

В заключение, так уж заведено, нужно ответить на вопрос «Как жить дальше?». Скажу честно: меня как реверсера все устраивает: чем больше ошибок, тем интереснее моя работа :) А если серьезно, чтобы получить хорошее ПО, менеджерам нужно научиться жертвовать эффективностью. Одними технологиями проблему ошибок в коде не решить, нужно инвестировать в обучение специалистов и культуру разработки в целом. Программисты же, в свою очередь, должны понимать, зачем им писать хороший код и как это делать. И наконец, рекомендую присоединиться к нашему сообществу 1 — это существенно облегчит вам жизнь.

ПОВЕДЕНЧЕСКИЙ АНАЛИЗ + ML В ДЕЛЕ ОБНАРУЖЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ

Игорь Кабанов

Специалист отдела перспективных технологий Positive Technologies

Время прочтения:	10 минут
Для кого:	сотрудники ИБ-отделов, эксперты по ML, разработчики ИБ-продуктов, аналитики
Прокачиваем знания:	ВПО, ML, статический и динамический анализ, обучение моделей

Вредоносное ПО — одна из самых распространенных угроз ИБ. Существуют разные методы защиты от атак с применением ВПО, но чаще всего выделяют два подхода — статический и динамический (поведенческий) анализ.

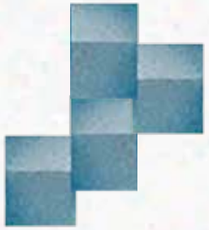
В первом случае задача сводится к поиску шаблонов вредоносного содержимого в файле или памяти процесса. Это могут быть строки кода, фрагменты закодированных или сжатых данных, а также последовательности скомпилированного кода. При этом можно искать не только отдельные шаблоны, но и их комбинации с дополнительными условиями: с привязкой к месту нахождения сигнатуры, с проверкой расстояния между ними и т. д.

Во втором случае речь идет об анализе поведения программы. Можно запустить ее в режиме эмуляции, который предполагает безопасное интерпретирование действий без причинения вреда операционной системе. Либо в виртуализированном окружении (песочнице). Этот подход подразумевает честное выполнение действий в системе с последующей фиксацией вызовов. Степень подробности логирования — это своего рода баланс между глубиной наблюдения и производительностью анализирующей системы. На выходе мы получаем журнал действий программы в ОС — трассу поведения, которую можно анализировать.

В сравнении со статическим анализом у динамического подхода есть важное преимущество: несмотря на все попытки злоумышленника запутать программный код и скрыть свои намерения, вредоносное воздействие будет зафиксировано. Сведение задачи обнаружения вредоносного ПО к анализу действий программы позволяет выдвинуть гипотезу об устойчивости продвинутого алгоритма обнаружения вредоноса. А воспроизводимость поведения благодаря одному и тому же изначальному состоянию среды (слепок состояния виртуальной машины) упрощает задачу классификации легитимных и вредоносных действий.

ОБОБЩАЮЩАЯ СПОСОБНОСТЬ МОДЕЛИ ХОРОША В СЛУЧАЯХ, КОГДА НУЖНО РАСШИРИТЬ РЕШЕНИЕ ДЕТЕКТОМ СХОЖИХ ИНЦИДЕНТОВ

В основе поведенческого анализа зачастую лежат наборы правил: экспертные заключения переносятся в сигнатуры, с помощью которых инструмент детекта делает выводы о вредоносности ПО. Этот подход хорошо себя показывает, но не исключает следующую проблему: решение будет фиксировать только те атаки, которые строго соответствуют заданным правилам. Аналогичные сложности возникают и в случае изменения или модернизации ВПО. Решить проблему можно двумя способами: задать более мягкие критерии срабатывания (написать более общее правило) либо сформировать множество правил под каждое вредоносное ПО. В первом сценарии есть риск получить массу ложных срабатываний, а для реализации второго требуется много времени, что может негативно повлиять на скорость выхода обновлений.



Как результат, появляется потребность в распространении имеющихся знаний на новые случаи, которые мы еще не покрыли правилами, но можем предположить, что это ВПО, на основе известных признаков. Здесь нам на помощь приходят ML-алгоритмы. При корректном обучении ML-модели имеют обобщающую способность. Проще говоря, модель не просто запоминает предложенные примеры, а может обрабатывать новые кейсы с учетом закономерностей из обучающей выборки.

Чтобы ML-модель работала корректно, во время ее обучения важно учитывать два фактора:

- Набор признаков должен быть как можно более полным (чтобы модель увидела как можно больше закономерностей и лучше распространяла знания на новые примеры), но не избыточным (чтобы не хранить и не обрабатывать бесполезные признаки).

- Набор данных должен быть репрезентативным, сбалансированным и регулярно обновляемым.

Мы решили провести исследование: сформировать набор признаков, обучить модель и проверить, можно ли доверять выводам ML-решения о вредоносном характере файлов и программ.

ЭКСПЕРТНЫЕ ЗНАНИЯ В ML-МОДЕЛИ

В контексте анализа вредоносного ПО исходные данные — это сами файлы (.exe, .pdf, .msi, .zip и др.), а промежуточные — созданные ими вспомогательные процессы. Эти процессы производят системные вызовы, последовательности которых (трассы) нужно преобразовать в набор признаков и проанализировать. При этом выводы о вредоносности с экспертной точки зрения можно делать на основе отдельных вызовов, их последовательностей, а также сочетаний первых и вторых.

Мы начали составление датасета с выбора признаков, которые, по мнению специалистов, являются значимыми в контексте обнаружения ВПО. Все эти признаки можно свести к виду n-грамм по системным вызовам. Затем мы оценили, какие из них вносят наибольший вклад в процесс обнаружения, отбросили лишние и получили итоговую версию датасета.

Исходные данные:

{«count»:1,»PID»:»764»,»Method»:»NtQuerySystemInformation»,»unixtime»:»1639557419.628073»,»TID»:»788»,»plugin»:»syscall»,»PPID»:»416»,»Others»:»REST: ,Module=\»nt\»,vCPU=1,CR3=0x174DB000,Syscall=51,NArgs=4,SystemInformationClass=0x53,SystemInformation=0x23BAD0,SystemInformationLength=0x10,ReturnLength=0x0»,»ProcessName»:»windows\\system32\\svchost.exe»}

{«Key»:»\\registry\\machine»,»GraphKey»:»\\REGISTRY\\MACHINE»,»count»:1,»plugin»:»regmon»,»Method»:»NtQueryKey»,»unixtime»:»1639557419.752278»,»TID»:»3420»,»ProcessName»:»users\\john\\desktop\\e95b20e76110cb9e3ecf0410441e40fd.exe»,»PPID»:»1324»,»PID»:»616»}

{«count»:1,»PID»:»616»,»Method»:»NtQueryKey»,»unixtime»:»1639557419.752278»,»TID»:»3420»,»plugin»:»syscall»,»PPID»:»1324»,»Others»:»REST: ,Module=\»nt\»,vCPU=0,CR3=0x4B7BF000,Syscall=19,NArgs=5,KeyHandle=0x1F8,KeyInformationClass=0x7,KeyInformation=0x20CD88,Length=0x4,ResultLength=0x20CD98»,»ProcessName»:»users\\john\\desktop\\e95b20e76110cb9e3ecf0410441e40fd.exe»}

Промежуточные данные (последовательности):

syscall_NtQuerySystemInformation*regmon_NtQueryKey*syscall_NtQueryKey

Вектор признаков:

	syscall_NtQuery SystemInformation* regmon_NtQueryKey	regmon_NtQueryKey* syscall_NtQueryKey	syscall_NtQuery SystemInformation* syscall_NtQueryKey	
...	1	1	0	...

НАКАПЛИВАЕМ ДАННЫЕ

Мы уже упоминали, что основные требования к данным для обучения ML-модели — это репрезентативность, сбалансированность и регулярная обновляемость. Рассмотрим эти характеристики в контексте поведенческого анализа вредоносных файлов:

- **Репрезентативность.** Данные должны иметь распределение по признакам, близкое к распределению в реальности.
- **Сбалансированность.** Исходные данные для тренировки модели поступают в нее с разметкой «легитимные» или «вредоносные». Количество примеров обоих классов должно быть примерно равным.
- **Регулярная обновляемость.** Во многом этот пункт связан с репрезентативностью данных. Поскольку тренды в области вредоносных файлов постоянно меняются, необходимо регулярно актуализировать решение.

С учетом перечисленных требований мы выстроили следующий процесс накопления данных:

1. Данные делятся на два типа — основной поток и эталонные примеры. Последние проверяются экспертами вручную, поэтому корректность их разметки гарантирована. Они нужны для валидации модели и управления тренировочной выборкой. Основной поток размечается правилами и автоматизированными проверками. Это необходимо для обогащения выборки всевозможными реальными примерами.
2. Все эталоны сразу добавляются в обучающую выборку.
3. Также в обучающую выборку добавляется некоторый изначальный набор данных из потока (чтобы собрать необходимый для обучения объем информации). Под необходимым объемом в данном случае подразумевается такое количество данных, с которым тренировочная выборка становится достаточно полной (разнообразной) и репрезентативной. Поскольку эталонные примеры проверяются вручную, собрать выборку из нескольких десятков тысяч таких кейсов невозможно — добирать разнообразие приходится с помощью потока.
4. Периодически модель тестируется на новых данных из потока.
5. Точность модели в первую очередь гарантируется для эталонных примеров. Соответственно, в случае противоречий предпочтение отдается эталонным данным.

Со временем мы собрали достаточно много данных из потока, настал момент отказаться от автоматизированного накопления на основе ошибок в пользу более контролируемой обучающей выборки:

1. Фиксируем накопленную обучающую выборку.
2. Данные из потока используем только для тестирования модели (ни один экземпляр не добавляется в обучающую выборку).
3. Обновление обучающей выборки возможно только в случае обновления набора эталонных примеров.

Каких результатов мы добились:

› Убедились, что обученная зафиксированная модель достаточно устойчива к «дрифту» данных.

› Контролируем каждый новый пример, который добавляем в обучающую выборку (эталонные примеры проверяются вручную).

› Можем отслеживать изменения и гарантировать точность работы модели на эталонном наборе данных.

С процессом накопления данных разобрались, но дальше возникает закономерный вопрос: почему мы уверены, что каждое обновление модели ее улучшает? Ответом на него является все та же эталонная выборка. Примеры из нее всегда проверяются и размечаются вручную, поэтому мы считаем ее корректной. Соответственно, при каждом обновлении мы в первую очередь проверяем, что модель показывает 100-процентную точность именно на эталонной выборке. А затем смотрим на результаты тестирования in the wild, которые также должны показывать, что эффективность решения повышается.

Положительный результат достигается за счет очистки обучающей выборки от противоречащих эталонных данных. Это примеры из потока, которые достаточно близки по векторному расстоянию к трассам из эталонной выборки, но при этом имеют противоположную метку. Наши эксперименты показали, что такие примеры являются выбросами даже с точки зрения данных из потока, так как после удаления их из обучающей выборки с целью повышения точности на эталонной выборке возрастала и точность на поток.

В ЧЕМ ХОРОША ML-МОДЕЛЬ

ML-модель отлично проявила себя в сочетании с поведенческими детектами в виде корреляций. Подчеркнем, что именно в сочетании: обобщающая способность модели хороша в случаях, когда необходимо расширить решение детектом схожих инцидентов. Однако она не так эффективна при детектах в рамках четкого понимания правил и критериев вредоносности ПО.

Примеры, где ML-подход показал себя с хорошей стороны:

› **Аномальные цепочки подпроцессов.** Само по себе большое количество ветвистых цепочек — вещь легитимная. При этом модель замечает аномалии в количестве узлов, степени вложенности, повторяемости/неповторяемости конкретных имен процессов и др. Человек вряд ли до такого додумается.

› **Нестандартные значения параметров вызовов по умолчанию.** В большинстве случаев аналитиков интересуют значимые параметры функций, в которых они ищут что-либо вредоносное. На остальные параметры (грубо говоря, значения по умолчанию) они не особо обращают внимание. Но в определенный момент вместо пяти значений по умолчанию их может стать шесть. Не факт, что аналитик это заметит, а модель сразу зафиксирует.

› **Нетипичные последовательности вызовов функций.** Тот случай, когда функции не делают ничего вредоносного ни по отдельности, ни в совокупности, но их последовательность не встречается в легитимном ПО. Аналитику потребуется гигантский опыт, чтобы заметить подобную закономерность. А модель заметит, причем не одну, нестандартно решая задачу классификации по признаку, который даже не закладывался как показатель вредоносности.

И напоследок — несколько примеров, где сигнатурный поведенческий анализ оказался более эффективным:

› **Использование одного конкретного компонента одним вызовом для вредоносного действия.** Система по-разному использует сотни объектов. Уловить использование одного объекта на фоне миллиона других едва ли удастся: гранулярность аномалии все же низковата.

› **Проактивный детект по модели угроз.** Предположим, мы решили, что определенное действие с определенным объектом в системе в принципе недопустимо — даже если это единичный случай. С первого раза модель может не понять, что это значимое явление. Соответственно, возникает вероятность ошибки или неуверенного решения на этапе классификации.

› **Обфускация последовательности действий.** Например, нам известно, что программа должна выполнить три-четыре действия в определенном порядке, чтобы модель классифицировала ее поведение как вредоносное. Что будет между ними — неважно. Если накидать между тремя-четырьмя ключевыми действиями несколько «мусорных», это собьет модель и она примет неверное решение. При этом размерность числа признаков не позволяет учитывать такие запутывания с помощью хранения всех комбинаций последовательностей вызовов.

ЧТО ЕЩЕ ПОЧИТАТЬ:

О машинном обучении с точки зрения ИБ: реальная обстановка

ПОДКАСТ:

Как взломать ИИ? Биометрия, DeepFake и умный дом

БИЗНЕС-ТРЕК BUSINESS TRACK

Я попробую поразмышлять из тапок некоего CEO. Окей, инфобез… В моем понимании CEO все-таки должен размышлять про устойчивость бизнеса и устойчивое развитие. И здесь инфобез, скорее, просто один из кирпичиков, который может повлиять на ту самую устойчивость. По большому счету, если размышлять из тапок топ-менеджера, в идеале я, наверное, не должен об этом думать. Если я CEO и понимаю, что у меня, например, процессы пожарной безопасности нормально построены, я об этом сильно-то и не задумываюсь и по ночам сплю спокойно. Просто вот уже полтора года у нас действительно все происходит по-другому. И наверное, прямо здесь и сейчас любой CEO должен потратить время и ресурс на то, чтобы обратить на это внимание. Я не вижу много примеров того, чтобы в календаре какого-то CEO стояла, например, встреча раз в месяц по вопросам инфобеза. Вроде бы все переживают, но вроде бы ничего не делают. Неплохо бы действительно выделять свой ресурс и уделять внимание, брать эту задачу прямо в фокус. Мне кажется, как раз сейчас CEO может уже не только подразделение информационной безопасности «нагибать» на какие-то KPI, но и в целом думать об изменении процессов. И самое главное, наверное, влиять на ИТ-подразделение, ведь точка приложений атак хакеров — это именно ИТ.

Айдар Гузаиров

Генеральный директор Innostage

Если интегратор внедрял системы, именно он и должен нести ответственность за безопасность. Понятно, что может начаться пинг-понг: это не наше, у нас этого не было в контрактах и т. д.

Но если случается реальный инцидент, никто, как правило, не спрашивает, было это в контракте или нет. Хочешь не хочешь, и у нас тоже бывают такие истории: ты просто подрываешься, и день сейчас или ночь — это никого не волнует.

Сергей Шерстобитов

Основатель и генеральный директор группы компаний Angara

По поводу суверенного интернета: это вещь хорошая, но от всех угроз и проблем, очевидно, не защитит. Я сомневаюсь, что у Китая нет проблем с компьютерными атаками. Или у американцев, которые обладают всеми возможными технологиями. Безусловно, это решение определенных проблем, но не панацея.

Виталий Лютиков

Заместитель директора федеральной службы по техническому и экспортному контролю России

Роль CISO иногда расслабляющая для топа: есть какие-то странные ребятки, где-то сбоку нас защищают… А это значит что? Это значит, что топ-менеджмент компаний и министерств по факту снимает с себя ответственность. И вот к чему мы пытаемся их склонить: ребята, это ваша ответственность, а не вот того парня из безопасности, которого вы плохо понимаете. В первую очередь — ваша.

Денис Баранов

Генеральный директор Positive Technologies

9 СТРАТЕГИЙ ЗАЩИТЫ.

НЕ МОЖЕШЬ УСТРАНИТЬ ХАКЕРА? УДЛИНИ ЕГО ПУТЬ К ЦЕЛИ

Алексей Лукацкий

Бизнес-консультант по информационной безопасности Positive Technologies

Время прочтения:	7 минут
Для кого:	CISO, blue team
Прокачиваем знания:	стратегии защиты от хакерских атак

Занимаясь защитой, мы обычно фокусируемся на том, чтобы не допустить плохих парней до их цели, до реализации недопустимого события, которое может быть осуществлено через одну или несколько целевых систем. Часто это фокус на том, чтобы не подпустить хакера к этим системам. Но единственная ли это стратегия?

Давайте посмотрим на простую модель реализации абсолютно любой атаки. Она состоит всего из трех элементов: хакера, актива/цели и канала реализации атаки. Устранение любого из этих элементов приводит к тому, что атака будет неуспешной.

Хакер

Канал реализации атаки

Актив

Давайте попробуем порассуждать и понять: какие стратегии защиты у нас могут быть с точки зрения этой модели? Я бы выделил девять:

1. Устранить хакера.
2. Сделать цель недоступной.
3. Устранить все уязвимости, через которые действуют злоумышленники.
4. Сделать канал реализации атаки невозможным.
5. Усложнить атаку (сделать ее долгой).
6. Сделать атаку очень дорогой.
7. Обнаруживать атаки быстрее.
8. Уменьшить площадь атаки.
9. Привлечь белых хакеров.

УСТРАНЕНИЕ ХАКЕРА

Эта стратегия направлена не на физическое устранение злоумышленника или отправку его в места не столь отдаленные (это все-таки находится вне сферы деятельности специалистов по ИБ), а на создание условий, при которых его деятельность становится невозможной — именно его, а не используемого им инструментария.

Навскидку можно предложить 5 вариантов реализации этой стратегии, требующих достаточно высокого уровня технической и юридической зрелости:

› **Контратаковать ресурсы атакующих.** Этот нередко рекомендуемый и даже иногда реализуемый сценарий находится в серой зоне законодательства, а может быть, даже и в явно запрещенной, квалифицируемой по статье 272 УК РФ.

› **Блокировать ресурсы атакующих.** Это гораздо проще, чем их контратаковать: достаточно просто внести соответствующие IP-адреса, домены или автономные системы в черный список в NGFW или иных средствах защиты или обеспечения доступа в интернет.

› **Разделегировать домены.**

› **Обратиться в правоохранительные органы.**

› **Провести threat actor intelligence**, то есть сработать на опережение, отслеживая в даркнете потенциальные площадки, где встречаются хакеры, размещаются и берутся заказы, обсуждают возможные цели.

НЕДОСТУПНОСТЬ ЦЕЛИ

Вместо создания условий, которые делают невозможной деятельность хакера, можно сделать недоступной цель атаки. Это может быть достигнуто:

› за счет блокирования доступа к цели с помощью межсетевых экранов, ACL, NAT;

› динамической смены адресации у цели (есть даже патенты, описывающие ежеминутную смену адреса сервера для защиты от хакеров);

› динамической смены сетевой топологии за счет применения виртуализации и облаков.

СКОРЕЙШЕЕ ОБНАРУЖЕНИЕ АТАКИ

Эта стратегия достаточно традиционна для любого специалиста по ИБ, и именно поэтому она требует отдельных пояснений. Стоит помнить, что для обнаружения атак бессмысленно использовать какое-то одно решение или технологию. Серебряной пули не существует. С учетом разносторонности и атак, и каналов их реализации, и техник, и тактик, технологии тоже должны быть разнообразными. Достаточно только посмотреть на типы источников и «контейнеров» для угроз, чтобы понять, насколько непросто реализовать эту стратегию (см. табл. 1).

	Файл	TCP / IP	URL	Flow	Логи	E-mail	Домены	Метаданные веб-трафика	Активность пользователя	Активность приложений	API
Периметр	X	X	X	X	X	X	X	X	X	X	—
Серверы	X	X	—	X	X	X	X	X	X	X	—
ПК	X	X	—	X	X	X	X	X	X	X	—
Мобильные устройства	X	X	—	X	X	X	X	X	X	X	—
Внутренняя сеть	X	X	X	X	X	—	X	X	—	X	—
Wi-Fi	X	X	X	X	X	—	X	X	—	X	—
3G / 4G / 5G	X	X	X	X	—	—	X	X	—	X	—
АСУ ТП	X	X	X	X	X	—	X	X	X	X	—
Облака	X	X	X	X	X	—	X	—	X	X	X
Приложения	—	—	—	—	X	X	—	—	X	X	X
Виртуальная машина	X	X	—	X	X	X	X	X	X	X	X
Контейнер	X	X	—	X	X	X	X	X	X	X	X
Сетевые устройства	X	X	—	X	X	—	X	—	X	X	—

Табл. 1. Источники и «контейнеры» для описания угроз

УСТРАНЕНИЕ ВСЕХ УЯЗВИМОСТЕЙ

Это тоже очевидная для специалистов стратегия, которая заключается в выявлении и устранении как известных уязвимостей, так и уязвимостей нулевого дня (zero day). И здесь тоже важно не допустить ошибку, сфокусировавшись только на инфраструктурных уязвимостях, дырах в приложениях и веб-сервисах. Уязвимости также могут быть:

- в аппаратном обеспечении;
- конфигурации систем;
- архитектуре систем;
- поведении людей;
- моделях и алгоритмах;
- данных.

И все они должны быть идентифицированы для устранения слабых мест, используемых злоумышленниками разными способами — от социального инжиниринга до технической эксплуатации.

УМЕНЬШЕНИЕ ПЛОЩАДИ АТАКИ

Предыдущую стратегию можно сузить до более приземленной: вместо устранения всех уязвимостей, что иногда бывает физически невозможно, можно сфокусироваться на уменьшении площади атаки, то есть нивелировании причин, приводящих к успешной реализации атак.

К таким способам можно отнести:

- отказ от использования административных прав там, где в этом нет необходимости, для реализации принципа минимума привилегий;
- закрытие неиспользуемых портов;
- отключение ненужных сервисов;
- удаление ненужных приложений и расширений (плагинов);
- запрет неиспользуемых протоколов.

Эти достаточно очевидные и простые методы почему-то игнорируются большинством компаний. Поэтому государство не только обратило внимание на эту проблему, но и взяло ее под контроль. В частности, был принят приказ ФСБ от 11.05.2023 № 213 «Об утверждении порядка осуществления мониторинга защищенности информационных ресурсов...». Согласно ему российская спецслужба будет проводить анализ защищенности публичных ресурсов субъектов КИИ и других организаций, попадающих под действие 250-го Указа Президента России. У Роскомнадзора также есть планы по анализу защищенности внешних ресурсов на предмет имеющихся в них уязвимостей. Аналогичные планы есть и у ФСТЭК, которая в случае принятия соответствующих поправок в законодательстве сможет оценивать защищенность объектов топливно-энергетического комплекса.

ПРИВЛЕЧЕНИЕ БЕЛЫХ ХАКЕРОВ

Еще одна стратегия защиты от плохих парней — их опережение. Я имею в виду не анализ их активности с помощью threat actor intelligence, а приглашение легальных хакеров, которые смогут первыми проверить все возможные способы проникновения в организацию. Она, в свою очередь, сможет их своевременно перекрыть или устранить.

Среди этих способов:

- тесты на проникновение (пентесты) и red teaming;
- собственные программы поиска уязвимостей за вознаграждение (багбаунти);
- внешние открытые и приватные программы багбаунти;
- пентесты на платформах багбаунти;
- багбаунти на максималках — верификация недопустимых событий.

УДОРОЖАНИЕ АТАКИ

Если перефразировать известную аксиому о том, что цена защиты не должна превышать стоимость защищаемой информации, в утверждение «стоимость атаки должна превышать стоимость защищаемой информации», то у нас появляется еще одна стратегия. Она заключается в удорожании инструментов хакеров, способов их использования, привлечения и т. п. Раньше эта стратегия вполне себя оправдывала, пока хакерский инструментарий не превратился в коммодити, то есть стал доступен всем без исключения. И утечки кибераресенала из АНБ и ЦРУ, произошедшие несколько лет назад, а также регулярно выбрасываемые в свободный доступ исходники самых современных шифровальщиков только ухудшают ситуацию. Поэтому цена большинства атак сегодня стала катастрофически низкой, а дорожает только защита.

УСЛОЖНЕНИЕ АТАКИ

Если мы не можем устранить цель, хакера, уязвимости или уменьшить площадь атаки, то нам остается примириться с фактом возможности проникновения злоумышленников в инфраструктуру. Но это еще не причина опускать руки. Можно попробовать взять понемногу от всех ранее описанных стратегий, поставив себе задачу удлинить, изолировать и сделать видимым движение хакера по инфраструктуре. Это повышает вероятность его оперативного обнаружения и блокирования.

Помимо ранее перечисленных сценариев, есть следующие варианты:

- сегментация инфраструктуры;
- использование нетипичных портов для приложений и сервисов;
- шифрование данных;
- применение обманных систем.

С помощью этих методов мы устраняем самые простые точки проникновения, которыми пользуются хакеры, усложняем их перемещение по атакуемой инфраструктуре, увеличиваем время движения по цепочке атаки, делая все, чтобы оно было больше времени обнаружения за счет анализа журналов регистрации и иных источников данных.

В заключение вы, наверное, ждете рецепта и совета, какую из стратегий выбрать. Но, увы, я не дам ни того ни другого. Я не знаю ни ваших ресурсов, ни вашей текущей ситуации, ни ваших целей. А без этого любой мой совет может направить вас по ложному пути. Однако подсказку все-таки дам. У каждой из стратегий есть свои плюсы и минусы, своя область применения и ограничения. Но присмотритесь к последней описанной мной стратегии — она является комбинацией ряда предыдущих, а значит, вобрала в себя все лучшее, что в них было; правда, и не без ограничений всех этих стратегий.

7 отраслевых сегментов:

- Управляющая компания City
- Транспортная компания Heavy Logistics
- Металлургический комбинат «МеталлиКО»
- Нефтегазовая компания Tube
- Электроэнергетика
- Банковская система (ЦБ + три типа банков)
- Атомная компания Atomic Energy

NEW

>200
атакующих

15 378
офлайн-посетителей

≈150
недопустимых событий и kill chains

35 000
онлайн-участников

667
отчетов об инцидентах от синих команд

«ГОРОД ИГРУШЕЧНЫЙ, А УГРОЗЫ — ВПОЛНЕ РЕАЛЬНЫЕ»

phd 12 Positive Hack Days

Каким увидели Positive Hack Days 12 те, кто пока далек от кибербеза? Сейчас узнаете ;)

Эту статью/репортаж мы собрали из материалов, которые подготовили начинающие журналисты — ученики российских медиашкол. Мы сохранили авторский слог и внесли только минимальные редакторские правки.

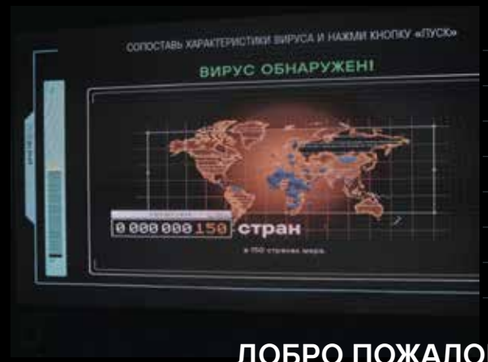
Виктория Сухолейстер
Медиацентр «Три кита»

Александр Федосов
Медиашкола «ДЮйМ»

Алиса Харская
Медиашкола «ДЮйМ»

София Поленова
Информационно-медийный центр «Школьный квартал»

Опасные вирусы, хакеры, способные взломать целый город, коварные злоумышленники, поджидающие на каждом шагу... Похоже на фильм ужасов, не так ли? На самом деле все это существует в реальности, столкнуться с которой можно было на Positive Hack Days 12 в Москве.



ДОБРО ПОЖАЛОВАТЬ В СКАМОТОПИЮ!

При входе на фестивальную площадку вы сразу попадаете в страну Скамотопию. Но будьте внимательны: здесь повсюду обман и мошенники. Банкоматы воруют карты, вместо выдачи подарков автоматы считывают ваши персональные данные, а в «Лавке диковинных товаров» ждут танцующий тостер, принтер, печатающий бесконечный текст, и непослушный вентилятор! Основная идея этой локации заключалась в том, что в наше время взломать можно абсолютно любую систему, поэтому нужно как можно внимательнее относиться к защите своих данных.

Рядом находится музей «Хранилище кибермонстров». Здесь нам рассказали все про вирусы, ловушки в социальных сетях и при телефонных звонках. Еще одним запоминающимся местом, которое произвело на нас, пожалуй, самое сильное впечатление, стал киберполигон Standoff. Здесь можно было понаблюдать за битвой за цифровое Государство F, макет которого занимал бóльшую часть зала.

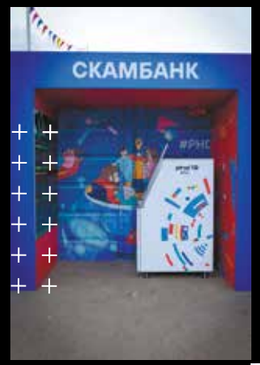
Да, непросто находиться на таких мероприятиях гуманитариям! «Что за непонятные слова?» — подумали мы поначалу. Кибератака, бэкап, логи, баги, компрометация... Это что-то на умном?! И действительно, у программистов есть свой специальный сленг, на котором они разговаривают, а большинство людей его впервые слышат. Но ничего не поделаешь, пришлось разбираться!

Как вы преобразовали форум в фестиваль? Это же первый в России фестиваль по кибербезопасности...

«Этот вопрос близок к понятию цифровой гигиены и отношению массового пользователя к безопасности. Обычно посетители форумов — это люди, которые достаточно глубоко заинтересованы в безопасности, либо те, кто уже как-то с ней связан. Конечно, это еще и клиенты, партнеры, а может, и сами компании или бизнес. В формате форума все это остается скрытым от глаз рядового человека, вот только безопасность касается каждого, поэтому организаторам пришла идея сделать фестиваль. Это очень крутая мысль, так как люди, просто гуляя в парке, видят, что существует площадка, на которой можно приобщиться к теме ИБ и вынести для себя что-то полезное».

Алексей Астахов

Руководитель продуктов Application Security Positive Technologies



ЛЕГЧЕ БЫТЬ ХАКЕРОМ...

Standoff — это место кибербитвы профессионалов за цифровое Государство F, которому угрожает опасность. Сама площадка напоминает концертный зал с яркими прожекторами, спецэффектами в виде дыма, вспышками света и огромным экраном, на котором отображались главные ходы соперников и центр зоны — большой макет виртуального города, за существование которого и идет борьба.

Поезда, заправки, офисы и атомные станции — игрушечные. Но атакующие и защитники — вполне реальные. Многие из них — серьезные специалисты в области ИБ. Чтобы бороться с реальными хакерами, нужно понимать, как они работают, изнутри.

«Мы опираемся на реальность, конечно с некоторыми упрощениями. Но если такие же атаки повторить на реальном железе, они приведут к тяжелым последствиям. Главная задача киберполигона — обучить защитников, помочь достичь определенных результатов и переложить их на реальный мир», — объясняет Владимир Назаров, руководитель отдела безопасности промышленных систем управления, Positive Technologies.

Задача хакеров — взломать инфраструктуру города, задача защитников — уберечь ее от вмешательства злоумышленников. При атаке мошенников на экране появляется сообщение: «Реализовано недопустимое событие». Также отображается макет с обозначением зданий и количества атак на эти участки. Сначала голова взрывается от непонимания: к чему эти циферки и значки? Но со временем разбираешься, и становится все интереснее и интереснее наблюдать за этой длительной схваткой.

Как фестиваль PHDays 12 помогает другим партнерам с безопасностью?

«Мне нравится этот вопрос. Есть большая ценность в этом фестивале, поскольку огромное количество компаний и людей озабочены проблемами безопасности. При этом однозначно нужна площадка, на которой они смогут не только обмениваться знаниями, но и смотреть на участников этого процесса: на людей, которые занимаются безопасностью и пентестами, и на тех, кто принимает решения, например на бизнесменов. Фестиваль дает возможность всем, кто заинтересован в построении безопасных процессов защиты своих продуктов, компаний или бизнеса, пообщаться и увидеть, как в реальности происходит построение защиты и как происходят атаки. А также понять, как правильно действовать дальше».

Алексей Астахов

Руководитель продуктов Application Security Positive Technologies

Алексей Волков рассказал, как часто совершаются хакерские атаки на социальную сеть, а также дал информацию о том, какие меры предосторожности используются в случае атак.

«На VK часто совершаются атаки, и для того, чтобы их предотвратить, мы тестируем системы безопасности 60, а то и 70 раз в год. Благодаря таким компаниям, как Positive Technologies, у нас есть собственная команда хакеров, которая регулярно нас тестирует. Также есть программа bug bounty, с помощью которой хакеры с внешних платформ за денежное вознаграждение ищут неисправности и баги в нашей системе. Мы проводим учения среди специалистов VK. Все это дает уверенность в безопасности системы».

Алексей Волков

Вице-президент, директор по информационной безопасности «ВКонтакте»

«РЕАЛИЗОВАНО НЕДОПУСТИМОЕ СОБЫТИЕ»

КРАСНАЯ НАДПИСЬ ВЫСВЕЧИВАЕТСЯ НА ОГРОМНОМ ЭКРАНЕ, ЗВУЧИТ ТРЕВОЖНЫЙ СИГНАЛ...

Звучит тревожный сигнал, на огромном экране высвечивается красная надпись о реализации недопустимого события. Это значит, что команда хакеров остановила колесо обозрения в парке, нарушила работу светофоров или взломала банк. Но тут же в бой вступает вторая команда, задачей которой было нейтрализовать недопустимые события. Команд, кстати, было неравное количество. На темной стороне сражаются 22 команды, а у защитников всего семь. И уже это свидетельствует о том, что быть на темной стороне гораздо проще.

«Легче быть хакером, но мы не выбираем легкие пути. Сейчас особенно важно, чтобы все государственные и коммерческие системы были под защитой. Поэтому для нас актуально проходить такие стрессовые мероприятия, чтобы быть готовыми к отражению подобных атак в реальности», — отметил Сергей Носков, капитан команды защитников GISCYBERTEAM из компании «Газинформсервис».

За четыре дня атакующим удалось реализовать недопустимые события 204 раза. Чаще всего это были утечки конфиденциальной информации и распространение вирусов-шифровальщиков. В реальной жизни хакеры угрожают многим сферам — страдают и огромные предприятия, и обычные люди. В большинстве случаев простых пользователей атакуют через домашние ПК или мобильные телефоны.

«Если говорить о конкретных примерах, чаще всего в реальной жизни мы сталкиваемся с вирусами-шифровальщиками, которые блокируют компьютеры и только за выкуп возвращают возможность дальше пользоваться устройством. Либо это вирусы, которые используют вычислительные мощности компьютеров, чтобы майнить криптовалюту (привет геймерам с мощными видеокартами, которые любят качать игры с торрентов). Помимо этого, через вредоносное ПО и фишинговые ссылки хакеры могут красть ваши персональные данные, пароли и даже совершать банковские транзакции», — объясняет Степан Подкасик, руководитель группы информационной безопасности промышленных систем управления, Positive Technologies.

В нашем мире противостояние защитников и хакеров будет только нарастать. Цифровой город помогает понять, что делать, чтобы избежать серьезных проблем в реальности. С каждым годом атаки становятся все изощреннее и опаснее, поэтому механизмы защиты тоже нужно отрабатывать и тренировать.

Вы не боитесь, что таким образом развиваете опасных хакеров?

«Однозначный ответ — нет. Здесь, пожалуй, я выскажу свое личное мнение. Если человек заинтересован в информационной безопасности и хочет попробовать для себя что-то из разряда пентеста, не обязательно уходить на черную сторону. Если он будет пробовать на реальной инфраструктуре, не имея скилов, это будет плохо. Нужно правильно объяснять и рассказывать, что такие специалисты действительно важны и нужны. Эти люди не получают „черную" прибыль, мы называем их „белыми шляпами". Я искренне верю в то, что Standoff позволяет показать, каким образом можно найти свой путь в информационной безопасности даже хакерам».

Алексей Астахов

Руководитель продуктов Application Security Positive Technologies

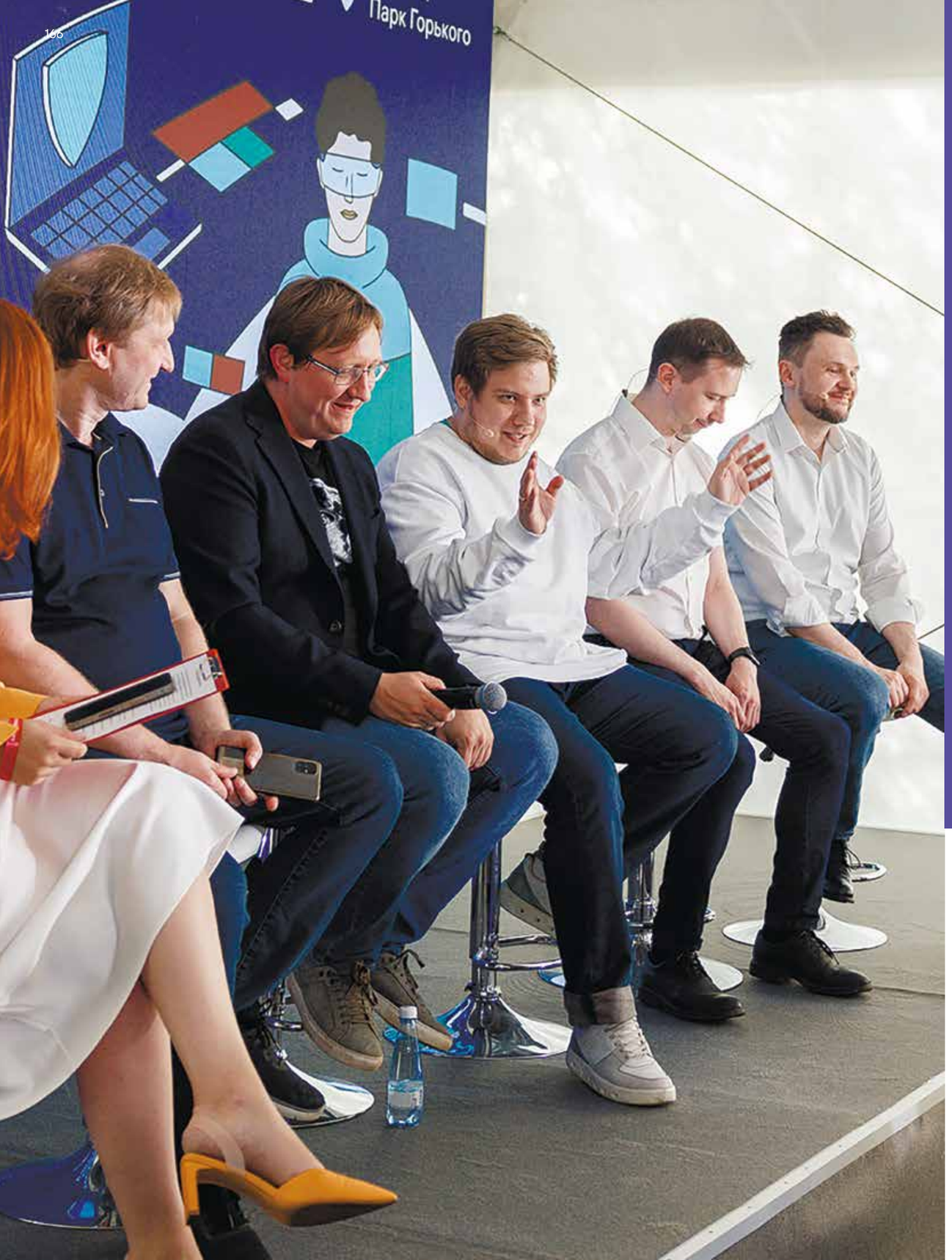
Видеорепортаж начинающих журналистов из команды «Три кита»

Как в социальной сети «ВКонтакте» предотвращают утечки данных пользователей?

«Для начала нужно понимать, что утечки бывают двух видов — реальные и ложные, фейковые. Ложные утечки появляются из-за того, что злоумышленники, пытаясь сделать себе имя, компилируют данные и стараются выдать их за свои. Также они могут брать общедоступные данные пользователей и выдавать это за взлом. Не всем сообщениям стоит верить, доверяйте только проверенной информации».

Алексей Волков

Вице-президент, директор по информационной безопасности «ВКонтакте»



NGFW ПО-РУССКИ: РЫНОК НА 100 МЛРД РУБ.

Денис Кораблев

Управляющий директор, директор по продуктам Positive Technologies

Павел Коростелев

Руководитель отдела продвижения продуктов «Кода Безопасности»

Денис Батранков

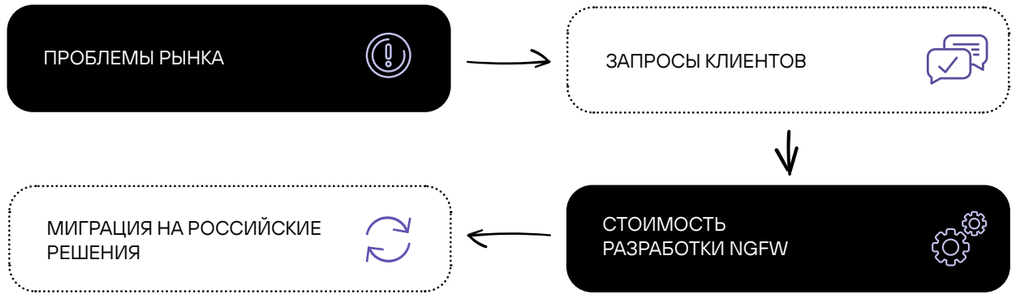
Руководитель направления сетевой безопасности Positive Technologies

Иван Чернов

Менеджер по развитию UserGate

Александр Баринов

Директор портфеля сетевых решений «РТК-Солар»



На Positive Hack Days 12 прошел круглый стол, участники которого обсудили отечественные межсетевые экраны нового поколения. Вендоры рассказали, какими функциями должен обладать качественный NGFW, можно ли безболезненно перейти на российские продукты и почему в разработку решения нужно вложить как минимум миллиард.

«Настоящий передел рынка NGFW начнется в 2025 г. Драйвером, само собой, станет Указ Президента РФ № 166. Раньше мы предполагали, что к этому моменту рынок сетевой безопасности будет оцениваться в 100+ млрд руб. и существенная доля этих средств придется на NGFW. После 24 февраля прогнозы изменились. Итоговая сумма будет выше, ведь многие западные продукты, которые клиенты уже приобрели, придется менять в рамках импортозамещения. Мы рассчитываем на существенную долю этого рынка и не ограничиваем расходы на разработку NGFW. Было бы на что тратить».

Денис Кораблев

POSITIVE TECHNOLOGIES НА PHDAYS 12 ПРЕДСТАВИЛА РАННЮЮ ВЕРСИЮ СОБСТВЕННОГО NGFW. ЗАПУСК ПРОДУКТА ЗАПЛАНИРОВАН НА ОСЕНЬ 2024 Г.

— В апреле 2023 г. на CISO-FORUM директор департамента обеспечения кибербезопасности Минцифры Владимир Бенгин сказал, что нехватка NGFW — одна из главных проблем российского рынка СЗИ. Согласны ли вы с Владимиром и что изменилось за последние несколько месяцев?

Денис Кораблев: Несмотря на то что в России есть вендоры, которые называют себя производителями NGFW, многие регуляторы чувствуют, что проблема не решена: качество отечественных продуктов оставляет желать лучшего. Я часто слышу от клиентов: «Мы купим лицензию на NGFW, повесим на стенку и будем всем показывать. На пару лет от нас отстанут, а к 2025 г., возможно, появятся рабочие решения». Все понимают, что NGFW нельзя сделать «на коленке», для этого нужно время. На рынке есть продукты, которые уже что-то умеют, но пока они не могут полноценно заменить западные решения. Всем нужна классная технология, и надеюсь, что скоро она появится. Мы над этим работаем.

Денис Батранков: Наши клиенты активно формируют запросы и требования к NGFW, поскольку к 2025 г. значительная часть российских компаний должна перейти на отечественные решения. Чаще всего они просят реализовать экспертный функционал, эффективную блокировку атак, качественный удаленный доступ и возможность блокировать вредоносный код, через который хакеры заражают инфраструктуру.

Иван Чернов: В феврале 2022 г. к нам пришли заказчики, к работе с которыми мы объективно не были готовы, — большой кровавый энтерпрайз. Они хотели закрыть все проблемы уже завтра, а в идеале — еще вчера. Извините, но это невозможно. CheckPoint в нашем положении тоже не справились бы до завтра. Привычный подход, когда технология постепенно растет и захватывает все бóльшую долю рынка, здесь не работает. Из-за этого возникло ощущение, что в России нет ничего в области NGFW. Да, определенным компаниям действительно не хватает возможностей отечественных продуктов, но мы работаем над этим. Для нас и наших конкурентов это вызов.

Павел Коростелев: Рынок NGFW — не одна большая тарелка, а скорее, пчелиные соты. Есть масса разных сегментов и клиентов, зачастую с довольно специфическими требованиями. Если в одной соте пусто, это не значит, что в остальных тоже. Версия о том, что российских продуктов на рынке нет, связана с отсутствием решений под специфические хотелки заказчиков. К тому же большинство компаний просто не готовы так быстро поменять ИТ-ландшафт. NGFW ставится вглубь сетевой инфраструктуры, поэтому перейти на новое решение без перерыва в работе невозможно.

Мы заметили интересную тенденцию: функционал, который заказчики последние пять лет называли неважным, сейчас ставится во главу угла: «Без этих возможностей мы у вас ничего не купим!» Я вижу два объяснения: либо рынок кардинально изменился, и теперь мы живем на Марсе, либо компании психологически не готовы к масштабному переходу. Они говорят, что рынок пуст, потому что хотят выиграть время и отложить трансформацию до 2025–2027 г.

Александр Баринов: Продолжу мысль Павла. В определенных «сотах» действительно нет качественных решений, но в некоторых уже сформировалась полноценная конкуренция.

Денис Кораблев: Важно разделять огромный рынок NGFW и конкретные ниши, которые закрывают российские игроки. Одно дело — организовать VPN для маленького офиса. Совсем другое — прийти в крупную компанию с предложением: «Давайте переходить на продукты отечественных вендоров». Клиенты гарантированно спросят: «Нормально же жили, ну зачем вы начинаете?»

— Каким должен быть качественный российский NGFW?

Денис Батранков: Мы фокусируемся на производительности и надежности. Многие до сих пор ожидают работу всех функций безопасности одновременно на скоростях 10 Гбит/с и выше, включая анализ расшифрованного SSL-трафика. Поддержку high availability упоминали практически все участники наших опросов.

Денис Кораблев: NGFW — это продукт, который клиент не должен замечать. Основные претензии и запросы идут не к функциональности. Если продукт начинает резать сессию, падать и глючить — это боль. Клиенты хотят, чтобы NGFW работал незаметно и прозрачно.

Иван Чернов: Первое, о чем нас спрашивают заказчики, — наличие определенных сетевых функций. Им нужны динамическая маршрутизация, построение туннелей и т. д. Второй запрос — бумажки: реестр Минпромторга, реестр отечественного ПО, ТОРП, сертификат происхождения СТ-1 и др.

Павел Коростелев: Наш опыт показывает, что самое главное — эффективное обнаружение и предотвращение вторжений. Дальше идут контроль доступа и фильтрация по категориям. Третий пункт — защищенный удаленный доступ, потому что всем нравится работать дистанционно. Четвертый — защита от вредоносного ПО, потоковый антивирус. Это ключевые критерии, за которыми идут масштабируемость, отказоустойчивость, гибкая интеграция в сетевую инфраструктуру, мониторинг и отчетность.

Александр Баринов: На первое место наши клиенты ставят надежность и отказоустойчивость. Мы также хотим, чтобы сигнатуры для IPS стали нашей конкурентной отстройкой, поэтому считаем этот момент очень важным. Еще один ключевой фактор — сертификаты для госсектора, которые никто не отменял.

NGFW — ЭТО ПРОДУКТ, КОТОРЫЙ КЛИЕНТ НЕ ДОЛЖЕН ЗАМЕЧАТЬ

«МОЙ СОВЕТ — ЗАПАСАЙТЕСЬ НЕРВАМИ»

— Где взять экспертизу, которую вы закладываете в продукт? Чем ваши решения отличаются друг от друга с этой точки зрения?

Александр Баринов: У нас самый большой по выручке коммерческий SOC в стране. Через него идет огромный поток трафика, который рождает экспертизу. Второй момент — прокси. В качестве базы для NGFW мы используем Solar webProxy, которому уже больше пяти лет. Клиенты часто просили реализовать новые фичи, которые пересекались с функционалом NGFW, и мы постепенно превращали систему в сетевой экран нового поколения.

Павел Коростелев: У нас есть четыре основных столпа. Первый — сотрудники, которые занимаются мониторингом сетей заказчиков. Второй — технологические партнеры, чьи материалы мы используем (например, «Лаборатория Касперского»). Третий столп — open source и инструменты, которые позволяют отсеивать, приоритизировать и создавать контекст. Наконец, последний пункт — наши заказчики. Они говорят: «Мы знаем все о собственной инфраструктуре. Дайте нам инструмент, который позволит загрузить эти знания в ваш продукт». Фактически мы конвертируем опыт клиентов в эффективные механизмы защиты.

Иван Чернов: У нас есть экспертный центр и ресечеры, которые анализируют угрозы и превращают эти данные в экспертизу. Чем отличается наш продукт? В первую очередь мы ориентируемся на реальные угрозы. Далеко не все тратят время на исследование малоиспользуемого софта и написание научных работ. Мы про threat hunting, изучение zero day и создание proof of concept. К примеру, на нашем сайте есть целый раздел security reports с новостями о каждой сигнатуре. Наша главная задача — сократить время реакции: от появления информации об уязвимости до попытки ее эксплуатации. У хакеров должно быть мало времени на атаку.

Денис Кораблев: Мы стараемся не растить коммерческий SOC умышленно. Наша компания долго искала баланс между экспертизой и прибылью от консалтинга. В итоге мы специально снизили поток проектов, а своим специалистам рекомендовали уделять 70% времени исследованиям: брать то, что редко ломается или встречается, и искать новые векторы угроз. В Positive Technologies есть белые хакеры, которые собирают подобную экспертизу в PT SWARM, в дальнейшем она переходит в наши продукты. Аналогично с защитой: у нас есть PT Expert Security Center (PT ESC), который занимается исследованием защиты.

— Сколько стоит сделать NGFW?

Денис Кораблев: Пока мы выделили на это 750 млн руб. Но если поймем, что нужно больше, увеличим бюджет. Важно, что Positive Technologies вкладывает в это только собственные инвестиции.

Иван Чернов: Значительные средства могут уйти на генерацию трафика. Предположим, мы ставим железо на тест: 60 Гбит/с проходит — все хорошо, даже процессор не на пределе. Сможет ли оборудование выдержать больше? Чтобы это узнать, придется потратить еще 500 млн — взять дополнительную стойку и залить трафиком.

Павел Коростелев: Как минимум миллиард придется потратить на инфраструктуру. Но сколько еще потребуется, чтобы купить необходимое железо на склад? Ведь зачастую заказчик говорит: «Нужно тысячу устройств, причем уже через три месяца». Кроме того, нельзя забывать про разработку — это отдельная история, которую сложно измерить в деньгах.

Иван Чернов: Инвестировать можно много, но самое ценное — кейсы и реальные внедрения. Если вы не поставите свой продукт в сеть заказчика и не увидите, в чем ошиблись, никакие инвестиции не помогут.

Александр Баринов: Сейчас у нас есть окно возможностей — нужно потратить столько, сколько потребуется, чтобы к 2025 г. у нас был рабочий NGFW, соответствующий ожиданиям заказчика. Сроки ограничены, и в них придется вписаться всем, иначе китайские компании заберут этот рынок.

— Вы планируете локализовывать железо и производить собственные платы для NGFW?

Денис Кораблев: Есть сегменты, в которых локализованное железо необходимо. Например, ИТ-решения, которые стоят на заправках и в продуктовых магазинах. В случае высокопроизводительного оборудования необходимость локализации не так очевидна. На данный момент мы не видим в этом никакой выгоды. Нужно фокусироваться либо на очень маленьких, но массовых системах, либо на крупных — для балансировки трафика на десятки терабит. В этих направлениях мы работаем, середины нет.

Павел Коростелев: Возьмем, к примеру, платы ASIC. Кристалл будет стоить 100 $, только если вы делаете сотни решений. Российский рынок для этого недостаточно большой, поэтому, скорее всего, мы будем опираться на процессоры, которые проектируют российские инженеры (например, в Дубне).

Александр Баринов: Перспективы параллельного импорта нельзя назвать ясными. Возможно, нам придется самостоятельно делать определенное железо: под управление, логирование, отдельные куски сетевых плат и др. Нужно закладывать эту вариативность уже сейчас, чтобы в будущем возможный переход не оказался болезненным.

— На что стоит обратить внимание при переходе на отечественные решения?

Павел Коростелев: Начинать нужно с планирования. NGFW — сложное железо, которое находится глубоко в инфраструктуре, поэтому плохо продуманный проект может остановить работу компании. Затем определите критически важные для бизнеса функции и убедитесь, что вендор планирует реализовать их к 2025 г. Также обратите внимание на техническую поддержку: есть ли 24/7, что с SLA, сколько инженеров. Четвертый пункт касается поставок: уточните, сможет ли производитель предоставить нужный вам объем в заданные сроки. Также стоит убедиться, что у решения есть все необходимые сертификаты.

Денис Кораблев: Я давно пытаюсь донести до клиентов, что самое главное — это технологический потенциал. Даже при наличии качественной поддержки, всех бумаг и обвязок вы можете получить забагованную железку, которая не будет развиваться. В первую очередь я бы оценил, что у продукта под капотом. Кроме того, нужно четко понимать, насколько железо совместимо с вашей ИТ-инфраструктурой. Если в сети останется хотя бы 10–20% устройств, которые не подружатся с новым решением, это будет боль.

Александр Баринов: Я дам «вендор-агностичный» ответ. Безусловно, для многих компаний переход на российские решения будет тяжелым. Поэтому начинать этот процесс необходимо с тотального аудита: это поможет набраться смелости перед неизбежной миграцией.

Иван Чернов: Переход с одного продукта на другой всегда проходит сложно и непредсказуемо. К тому же вы не найдете точную копию привычной системы, потому что у всех решений разный функционал и архитектура. К этому просто нужно быть готовым. Мой совет — запасайтесь нервами.

КРУГЛЫЙ СТОЛ «NGFW ПО-РУССКИ» НА YOUTUBE

КТО КОГО: PT NAD VS COBALT STRIKE И BRUTE RATEL C4

Кирилл Шипулин

Руководитель группы обнаружения атак в сети
Positive Technologies

Время прочтения: 15 минут

Для кого: сотрудники отделов ИБ, специалисты SOC, эксперты по сетям и сетевым средствам обнаружения, системные администраторы

Прокачиваем знания: сетевой трафик, фреймворки постэксплуатации, хакерский инструментарий, тактики и техники атакующих, анализ сетевого трафика, обнаружение сетевых угроз

МОДНАЯ ТЕМА

Фреймворки постэксплуатации становятся все популярнее среди злоумышленников. Это удобный инструмент для контроля зараженных машин и горизонтального перемещения внутри сети. Фреймворки помогают атакующему собрать карту сети и связи между хостами в одном интерфейсе, из которого и управляются.

В этой сфере есть своя мода и тренды: число хакерских фреймворков постоянно растет. Кроме того, в открытый доступ периодически попадают взломанные версии инструментов. Так, в 2021–2022 гг. в интернете появились бесплатные релизы платформ Cobalt Strike и Brute Ratel C4. В результате их популярность резко выросла: последние исследования говорят о тысячах серверов Cobalt Strike, в честь решения называют[1] целые группировки, а огромные корпорации борются[2] с кейсами злоупотребления инструментом.

Какие вызовы несут хакерские фреймворки для систем защиты и специалистов по ИБ? Зачастую в этих инструментах есть богатый арсенал для обхода хостовых средств защиты (в основном для противостояния антивирусам и EDR-решениям). Все фреймворки используют сеть, чтобы получать команды и устанавливать связи с управляющими серверами. При этом так называемые биконы (beacons) Cobalt Strike умеют общаться с управляющим сервером и без доступа в интернет. Они просто передают свои данные по протоколу SMB или TCP по цепочке таких же биконов, пока те не выйдут «наружу». Соединение же с самим управляющим сервером может идти более привычным образом — по HTTP или HTTPS, нередко мимикрируя под загрузку библиотеки jQuery или общение с удостоверяющими центрами. Другие фреймворки, напротив, могут «притворяться» обычными сайтами: на HTML-страницы периодически отправляются «отстуки», а результаты их выполнения выглядят как POST-запросы на PHP-скрипты. Более того, искушенным пользователям этих инструментов предлагаются классические DNS-тоннели.

Многообразие сетевых протоколов и способов маскировки помогает злоумышленникам в обходе привычных средств сетевой защиты. Но о решениях класса network traffic analysis (NTA), к которым относится наш PT Network Attack Discovery[3], хакеры пока знают не так много.

«Отстук» — периодический запрос на управляющий сервер. Его главная цель — дать понять, что бикон все еще «жив», и запросить новые команды.

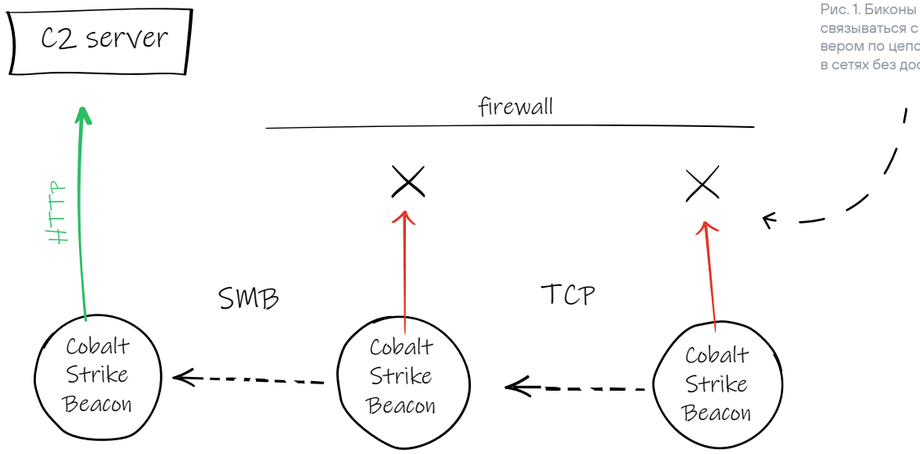


Рис. 1. Биконы Cobalt Strike умеют связываться с управляющим сервером по цепочке и работать даже в сетях без доступа в интернет

COBALT STRIKE

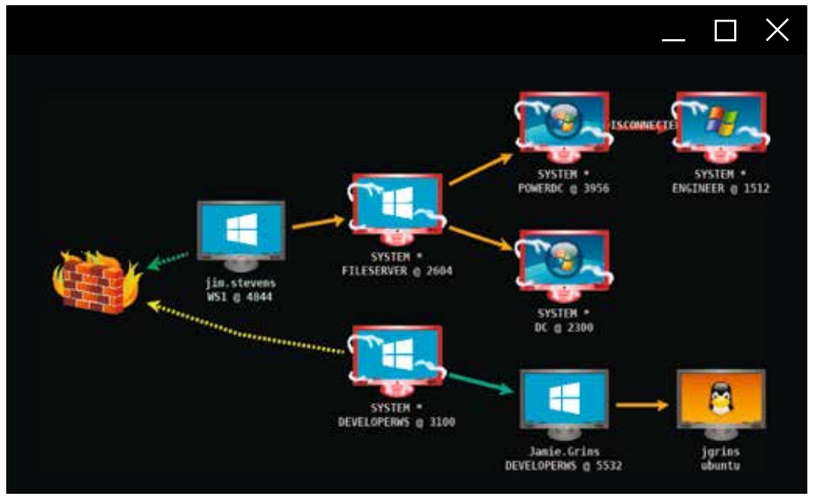


Рис. 2. Интерфейс Cobalt Strike именно так хакер видит инфраструктуру жертвы. Линии между хостами обозначают протоколы для связи между биконами

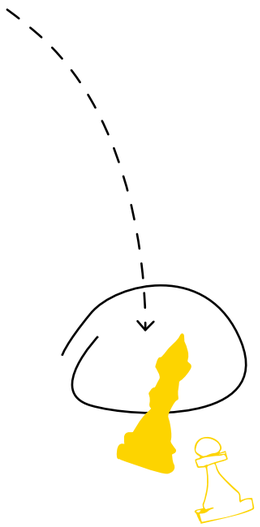
Главным фигурантом в наших делах по детекту Cobalt Strike станут метаданные, которые бикон отправляет с каждым «отстуком». Иногда «отстуки» заканчиваются для маяка новыми командами, выполнение которых требует еще нескольких запросов. При этом в каждом сетевом запросе от него передаются метаданные — 128 шифрованных байтов. Они имеют высокую энтропию, содержат разные идентификаторы и информацию о самом биконе.

Время между «отстуками» строго фиксируется и задается в Malleable C2 config — конфиг-файле, с которым запускается управляющий Team Server (C2). Этот файл содержит все параметры Cobalt Strike. Отметим, что задержка может случайным образом меняться в заданном диапазоне, например 60 ± 5% секунд.

Пара слов о PT Network Attack Discovery

PT NAD — это средство анализа сетевого трафика. Продукт не только исследует общую фактуру сети, но и определяет все стадии атак на инфраструктуру. Схема работы проста: вы передаете в систему копию сетевого трафика (по принципу «чем больше, тем лучше»), и PT NAD выявляет в нем вредоносные программы, слабые пароли, следы атак и всевозможные активности злоумышленников.

Включи VPN



Самый популярный транспорт для биконов — HTTP-протоколы. У них больше всего настраиваемых параметров: это заголовки, метод, «мусорные» данные (обычные строки, которые можно добавить до или после метаданных для маскировки) и другие. Метаданные, кстати, передаются не в открытом виде, а кодируются в разные формы. Авторы фреймворка хорошо постарались: кодировок действительно много, и они могут быть рекурсивно вложены друг в друга (при этом размер метаданных будет кратно расти). Например, метаданные сначала могут быть поксорены с четырьмя случайными байтами, затем закодированы в Base64 и потом — в NetBIOS.

Где могут передаваться метаданные? Везде: внутри HTTP-заголовков, в теле POST-запросов и даже в URL. В последнем случае, если данные будут ксориться со случайным ключом, URL будет постоянно меняться, сохраняя при этом первоначальную длину. В любой другой ситуации URL, напротив, случайным образом выбирается из имеющихся в конфиге, что также затрудняет обнаружение злоумышленников.

Я уже говорил, что Cobalt Strike маскируется под легитимные запросы? Для этого перед метаданными и после них можно добавлять «мусорные» строки. Например, сами метаданные могут отправляться в Base64 под видом JWT-токена в HTTP-куки, будучи обрамленными необходимыми конструкциями. Каждый автор волен сам придумывать способ маскировки, поэтому на GitHub можно найти сотни разных Malleable-конфигов.

Рис. 3. Cobalt Strike маскируется под запросы популярной библиотеки jQuery. Метаданные передаются в заголовке куки, при этом ответ сервера маскируется под содержимое легитимной библиотеки

Готов поспорить, вы уже устали от перечисления возможностей Cobalt Strike. Это неудивительно: он гибкий и имеет огромный потенциал для маскировки, поэтому обнаружение инструмента было для нас довольно нетривиальной задачей. Главным образом в детектировании помогает то, что метаданные передаются в каждом HTTP-запросе. Зная, как они выглядят и в какие рекурсивные формы кодируются, можно научить средство защиты копить HTTP-запросы, пока среди них не начнет прослеживаться некая периодичность, а затем определять, какое именно поле хранит метаданные. Опять же, расположение полей строго фиксируется в конфиг-файле и не меняется, что тоже играет нам на руку. Все перечисленное плюс проверка энтропии байтов метаданных и составляет алгоритм детекта активности Cobalt Strike в PT NAD. Конечно, в настоящем трафике достаточно много периодических HTTP-запросов с подходящей структурой, но мы смогли довести алгоритм до минимального количества ложных срабатываний.

Описание Cobalt Strike будет неполным без упоминания способов горизонтального перемещения фреймворка по сети. В арсенале инструмента — средства для создания сервисов на удаленной машине по протоколу SMB и SSH-сессии на Linux. Такие кейсы хорошо изучены и покрыты нашими детектами, в этой статье мы разберем особенности взаимодействия биконов Cobalt Strike по другим протоколам. Конечно, можно распространить на другие хосты и привычные биконы HTTP и HTTPS, но для машин без доступа в интернет есть только две узкоспециализированные опции — протоколы SMB и TCP. Они действуют схожим образом: это либо bind (reverse) TCP-соединение, либо SMB-пайпы (pipes). Причем появление необычного пайпа также бросается в глаза PT NAD.

После успешного захвата хоста и установки соединения (подключения к пайпу) маяк первым делом передает метаданные. Их размер — 132 байта, что соответствует XOR-шифрованию со случайным ключом, который отправляется в первых четырех байтах.

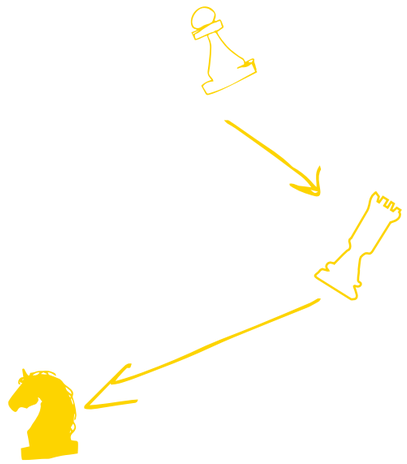
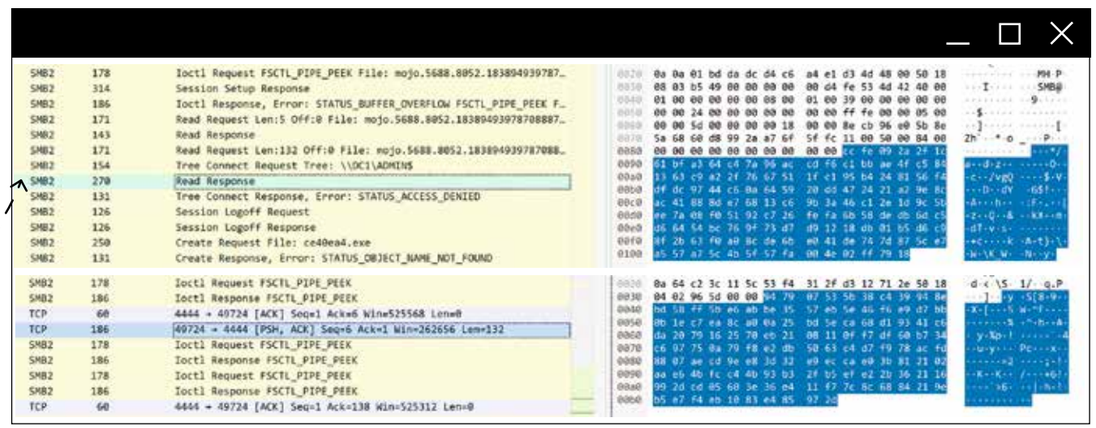


Рис. 4. Анализатор сетевого трафика Wireshark. В выделенных пакетах Cobalt Strike передает метаданные по протоколам SMB и TCP



Фреймворк PT NAD, Cobalt Strike, Brute Ratel 4, HTTP-запросы

Фреймворк PT NAD, Cobalt Strike, Brute Ratel 4, HTTP-запросы

С протоколом TCP мы наблюдаем ту же картину: 132 байта со сведениями о биконе в начале соединения. В обоих случаях метаданные предваряются полями их длины (длина составляет 4 байта), которые отправляются отдельными пакетами. Для метаданных она будет равняться 0x84 0x00 0x00 0x00 в шестнадцатеричной системе. А раз для SMB- и TCP-биконов метаданные передаются только в начале, все последующие разы хосты будут обмениваться только полями длиной по 4 нулевых байта.

Мы проанализировали трафик Cobalt Strike в лабораторных условиях методом черного ящика и можем сказать, что гибкость фреймворка поражает. При компиляции биконов на командном сервере в их коде буквально компилируется алгоритм по извлечению метаданных (как они описаны в Malleable-конфиге). Все это дает злоумышленникам огромный простор для маскировки запросов. Сетевые средства обнаружения, как правило, находятся в позиции догоняющих, ведь они могут обнаруживать только уже известные конфигурации. Тем не менее механизмы детекта в PT NAD способны находить и новые, еще неизвестные образцы фреймворка. Благодаря чему это возможно? Время между «отстуками» меняется несильно. Кроме того, метаданные:

- передаются в каждом «отстуке» бикона;
- хранятся в одном и том же месте (HTTP-заголовок или тело запроса);
- отправляются в одной и той же кодировке;
- имеют одинаковый размер.

Почему же авторы Cobalt Strike не зададут метаданным случайную длину, не добавят кодировку или не сделают время между «отстуками» максимально непредсказуемым? Причин может быть несколько:

Сложность кодовой базы	**Случайная длина упростит процесс обнаружения**	**Случайное время между «отстуками» сделает поведение биконов непредсказуемым**	**Необходимость бороться с хостовыми детектами**
Научить бикон и управляющий сервер хранить метаданные в разных частях запроса — довольно сложная задача	После того как метаданные будут зашифрованы, поле со значением их длины должно где-то храниться	Как в таком случае понять, удалил антивирус процесс на узле или маяк просто выжидает дольше обычного? Злоумышленники тоже ценят удобство	Они традиционно занимают первое место по эффективности обнаружения

Подтвердить или опровергнуть эти гипотезы нам поможет следующий фреймворк постэксплуатации — Brute Ratel C4 (BRc4).

BRUTE RATEL C4

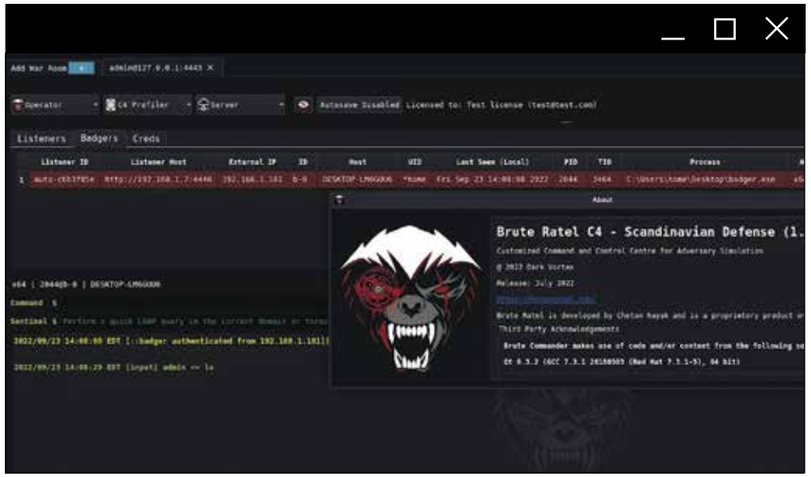


Рис. 5. Интерфейс командного сервера Brute Ratel C4

АВТОРЫ BRC4 НА СВОЕМ САЙТЕ ЗАЯВЛЯЮТ: «BADGER DOESN'T CARE. IT TAKES WHAT IT WANTS!»

У BRc4 и Cobalt Strike схожие функции, но есть и ряд различий. Например, исполняемые файлы BRc4 называются не биконами, а «барсуками» (badgers). У «барсуков» те же транспорты: HTTP(S) и SMB (или TCP), а для искушенных пользователей также существуют способы коммуникации через DNS-тоннели (но во взломанной версии 1.2.2 эта возможность отсутствует). Создатели BRc4 сделали упор на обход хостовых детектов, поэтому инструмент считался необнаруживаемым на узлах и по-настоящему опасным. Мы думаем иначе.

Рис. 6. Типичный HTTP-трафик BRc4. Все данные передаются в теле POST-запроса, остальные параметры настраиваются

С точки зрения работы в сети между Brute Ratel 4 и Cobalt Strike есть несколько различий:

- Метаданные передаются не в любом HTTP-заголовке, а строго в теле запроса.
- Данные тоже зашифрованы, но кодируются только в Base64.
- Вся информация отправляется при помощи HTTP-запросов методом POST.

Получается, что гибкости для обхода сетевых детектов у BRc4 даже меньше. Строгие правила передачи метаданных по HTTP делают его еще более легкой целью для наших алгоритмов детектирования. Схема проста:

1. Собрать побольше запросов HTTP POST.
2. Убедиться, что у них одинаковые заголовки.
3. Проверить, что тела запросов тоже совпадают.
4. Подсчитать среднее время между запросами и убедиться в регулярности «отстуков».

Что касается общения между «барсуками» по протоколу TCP или SMB, взаимодействие максимально просто. Обмен данными через SMB-пайп или TCP-соединение осуществляется напрямую, ведь спрятать что-либо в них действительно сложно.

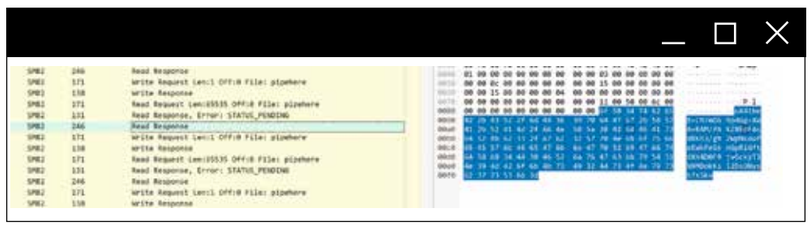


Рис. 7. Передача метаданных «барсуков» по протоколу SMB. На скриншоте видно, что информация закодирована в Base64

В SMB-трафике видно, как «барсук» отправляет метаданные в куске Base64-данных длиной 108 байт (рис. 7). После декодирования они превратятся в 80 шифрованных байтов, что соразмерно с данными

Cobalt Strike во время «отстука». В TCP-сообщениях данные будут иметь длину 110 байт за счет дополнительных символов \r\n в конце каждого сообщения.

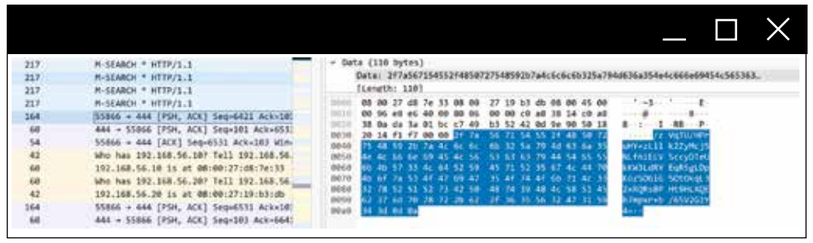


Рис. 8. Передача метаданных «барсуков» по TCP. Кодировка та же — Base64

Другие платформы

Cobalt Strike разрабатывается более десяти лет, у него уже есть устоявшаяся аудитория. Brute Ratel C4 же появился в конце 2020 г. и стал набирать популярность после выхода взломанной версии в 2022-м. Кроме того, злоумышленники продолжают активно искать аналоги привычным инструментам, которые постепенно обрастают детектами. Так, в последнее время все чаще мелькают новые названия фреймворки Sliver и Havoc (это подтверждает наша команда расследования инцидентов). Функциональность платформ не особенно отличается, а интерфейс Havoc и вовсе напоминает Cobalt Strike. Тем не менее специалисты по ИБ могут быть не готовы к атакам с использованием этих инструментов.

ПРОТИВОДЕЙСТВИЕ

Мы отслеживаем тренды развития хакерских инструментов и постоянно совершенствуем детекты, используемые в PT NAD. С технической точки зрения способы общения Cobalt Strike и Brute Ratel 4 по протоколу HTTP представляют огромный интерес, поскольку они очень вариативны и могут мимикрировать под легитимный трафик. Мы хотели показать вам, к каким мерам порой прибегают злоумышленники, чтобы специалист SOC не смог найти иголку — вредоносный запрос — в стоге легитимного трафика. Но в любом, даже самом хитром протоколе должны быть пригодные для детекта артефакты. Мы реализовали описанную выше логику детектирования в отдельных модулях PT NAD: они отлично справляются со своими задачами и присутствуют в новой версии продукта — 11.1. А для обнаружения других протоколов хорошо подходят правила без сложной логики. Например, первый DNS-запрос Sliver начинается с символов baakb, а DNS-запросы Cobalt Strike часто содержат суффикс «.180.».

ВРЕДОНОСНЫЙ ЗАПРОС — ИГОЛКА В СТОГЕ ЛЕГИТИМНОГО ТРАФИКА

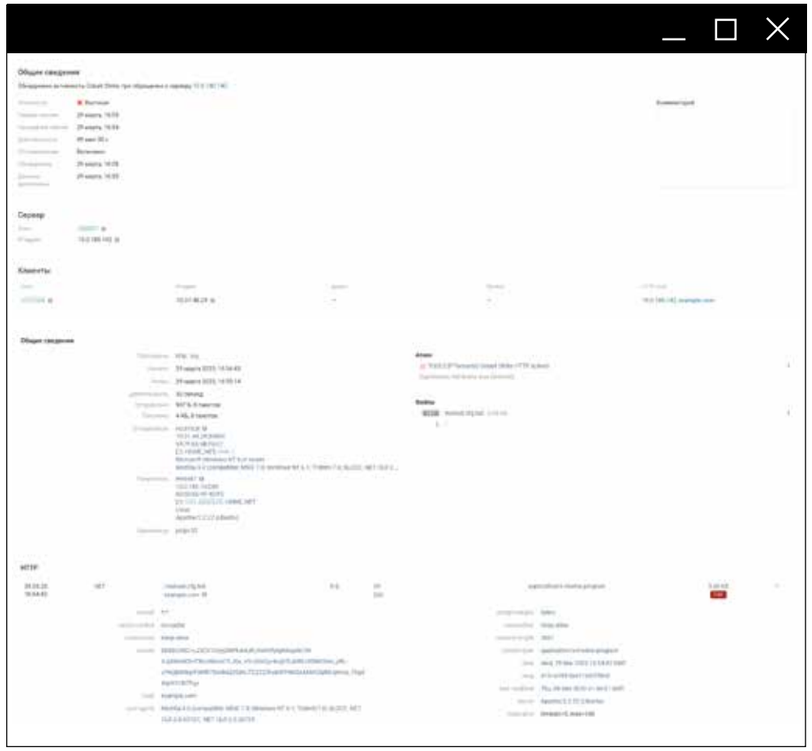


Рис. 9. Карточка атаки в интерфейсе PT NAD: обнаружена активность неизвестного образца Cobalt Strike, представлены детали HTTP-запросов

И вишенка на торте. Мы пишем детектирующие правила для всех видов коммуникации вредоносных программ. Некоторые из них удивляют (см. рис. 10).

Рис. 10. Иногда биконы Sliver кодируют байты своих данных в английские слова

ЭКСПЕРТНЫЙ КОММЕНТАРИЙ

Алексей Леднев

Руководитель отдела обнаружения атак экспертного центра безопасности Positive Technologies (PT ESC)

ПОЧЕМУ ОБЫЧНЫЙ АНТИВИРУС НЕ СПОСОБЕН ВОВРЕМЯ ОБНАРУЖИТЬ COBALT STRIKE И BRUTE RATEL C4 И ЗАБЛОКИРОВАТЬ ИХ ЗАПУСК?

Анализ сетевого трафика очень результативен при обнаружении активности инструментария злоумышленников. На это есть несколько причин. Первая достаточно банальна: за инфраструктурой очень сложно уследить. В крупных организациях новые рабочие станции и серверы могут появляться ежедневно. В какой-то момент ибэшники сталкиваются с неизвестным им устройством, на котором должным образом не настроены ни журналирование, ни агентское средство защиты (антивирус или EDR-система). С сетью такой проблемы нет. Если анализируется копия трафика, новое устройство не останется без внимания.

Вторая причина заключается в ресурсах злоумышленников. Результаты наших проектов по анализу защищенности показывают, что скрыть вредоносное ПО от средств антивирусной защиты не составляет особого труда. Злоумышленники могут как сами адаптировать свои инструменты, так и пользоваться сторонними решениями. Существует большое количество техник для обхода антивирусов, а на черном рынке можно найти готовый упаковщик, который поможет скрыть программу от проверок на конечном устройстве.

Чтобы спрятать сетевую активность, потребуется не только адаптация клиентской части вредоносного ПО, но и изменение самого управляющего сервера, так как придется пересмотреть формат взаимодействия между ними. Для злоумышленника это дополнительные временные затраты и расходы на разработку. Вот почему мало кто может позволить себе так кардинально менять инструментарий между атаками, а анализ трафика показывает высокую эффективность.

ЧТО ЕЩЕ ПОЧИТАТЬ:

- Анализ поведения трояна Pegasus в сети
- Hunting and detecting Cobalt Strike
- Где больше вирусов: в интернете или на ободке унитаза?
- Detecting and decrypting Sliver C2 — a threat hunter's guide

«ПРЕПОДАВАТЬ РАДИ ДЕНЕГ — БЕССМЫСЛЕННО»: КАК Я ЧИТАЛ КУРС ПО БЕЗОПАСНОЙ РАЗРАБОТКЕ В ВШЭ

Владимир Кочетков

Руководитель экспертизы безопасности приложений
Positive Technologies

Время прочтения:	8 минут
Для кого:	все, кто мечтает заработать свой миллион на курсах по кибербезу
Прокачиваем знания:	обучение ИБ, инфоцыганство

В обществе сложилось неоднозначное, скорее даже негативное отношение к онлайн-курсам и онлайн-образованию в целом. «Чему можно научиться в интернете? Какой специалист может вырасти из студента, который ни разу не нюхал пыль аудиторий? Ну уж нет — только парта и кафедра!» Все знают термин «инфоцыгане»... Но я убежден, что дело не в приставке «онлайн», а в том, кто, чему, зачем и где учит. В 2022–2023 гг. я провел онлайн-курс по безопасной разработке в Высшей школе экономики и приобрел преподавательский опыт, которым хочу поделиться.

ТРИ ПРИЧИНЫ СТАТЬ ПРЕПОДАВАТЕЛЕМ

В интернете можно найти массу курсов по ИБ. По запросу «онлайн-курс информационная безопасность» Google выдает больше миллиона ссылок: GeekBrains, Skillbox, «Нетология», Udemy... Десять лет назад ничего подобного не было. Чтобы хоть как-то прикоснуться к ИБ, а уж тем более к хакерской тематике, нужно было забыть все, чему тебя учили в институте, бежать на специализированные форумы, участвовать в CTF-соревнованиях и самостоятельно постигать азы, а потом и глубины кибербеза. Возможно, именно поэтому нас, практиков, так тянет в преподавание. Хочется дать будущим ИБ-специалистам то, чего мы сами были лишены, — доступные квалифицированные знания под обложкой одного курса.

На мой взгляд, есть три причины, по которым ИБ-экспертам может быть интересно преподавание. Во-первых, кибербезопасность для нас не пустой звук или формальность. Мы видим реальные кейсы и взломы, понимаем, как утекают данные, знаем, к чему это может привести и как избежать фатальных последствий.

Во-вторых, официальные вузовские ИБ-программы устаревают еще до того, как их напишут и утвердят. Наш критерий истины — практика. Мы ориентируемся на живые примеры, а не на учебники.

В-третьих, я верю в необходимость и ценность передачи знаний. Их нужно распространять, чтобы они развивались и трансформировались в нечто большее. Таков закон прогресса. Все как в науке: один ученый пишет диссертацию, второй ее читает, аккумулирует с собственным опытом и проводит новое исследование. Благодаря этому цепочка познания не прерывается. Преподавание — один из очевидных способов обеспечить непрерывную передачу знаний.

Можно сказать, что для меня это своего рода миссия. При этом я человек совершенно не академический. После окончания института я никогда не стремился вернуться в эту среду — к зачетам, конспектам и лабораторным работам. В ИТ и ИБ важно, что человек умеет и как у него работает голова, а цвет корочки, которую он получил, роли не играет. Тем не менее перспектива провести курс по безопасной разработке в одном из ведущих вузов меня зацепила так же сильно, как самих студентов — возможность освоить модную и довольно денежную профессию.

Нежданчик № 1

Студенты — это студенты, даже если речь идет о менеджерах старше 30 из солидных компаний. Желание сачкануть есть у всех, и для меня до сих пор остается загадкой, как можно отдать 800 000 руб. на курс и забить на учебу. Без списываний тоже не обошлось. Однажды я получил выполненную домашнюю работу, где в самом начале файла красовалась фраза: «Только, пожалуйста, не показывай Кочеткову в этом виде, иначе он спалит, что это я сделал». Этот человек сделал мой день.

«ВЫШКА»

В Positive Technologies мою идею сразу поддержали. Опыт преподавания у меня уже был: в 2015 г. я читал курс по введению в кибербезопасность для наших клиентов, в основном из банковской сферы. Но одно дело — провести пару семинаров в офисе заказчика, и совсем другое — стать частью полноценного обучения в вузе.

ВШЭ запустила двухгодичную магистерскую программу по кибербезопасности на платформе «Нетология». Выпускники получают дипломы государственного образца. В рамках программы мне предстояло провести курс, посвященный безопасной разработке на всех этапах жизненного цикла приложений — от проектирования до внедрения.

В обучении участвовало более 70 человек — студенты вузов и сотрудники российских компаний (в основном старше 30 лет).

Моей целью было донести до учащихся важность кибербез-составляющей разработки. В погоне за функциональностью и скоростью выхода релизов многие программисты забывают, что отвечают в том числе за безопасность кода. Разработчик не может сидеть в профессиональном пузыре и что-то вдохновенно писать в надежде, что ИБ-специалист потом закроет дыры.

3
МЕСЯЦА

70+
УЧАЩИХСЯ

12
ЛЕКЦИЙ

9
ВЕБИНАРОВ

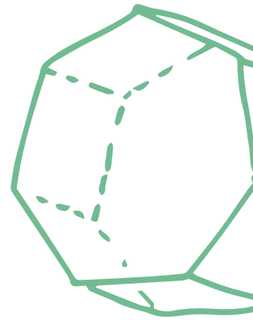
ДЕТАЛИ КУРСА

Я разработал программу с нуля. В нее входят 12 видеолекций (по 45 мин. каждая) и 9 онлайн-вебинаров (от 1,5 ч и более). В течение недели студенты изучали лекции, а по субботам мы проводили практические занятия.

О чем мы говорили на лекциях:

- основы безопасности приложений;
- моделирование угроз и оценка рисков;
- формализация уязвимостей;
- математические основы безопасности приложений;
- тестирование безопасности;
- классификация проблем безопасности;
- и многое другое.

Вебинары я постарался сделать максимально практичными — рассматривал реальные кейсы, давал много примеров, отвечал на вопросы, разбирал уязвимый код. По итогам студенты получали домашние задания двух типов.



Нежданчик № 2

Скажу по секрету: на мой взгляд, студенты вузов были чуть сообразительнее, чем уже состоявшиеся специалисты. Может быть, потому что вузовская дисциплина для них — не пустой звук. Или, может, из-за неподдельного интереса к предмету.

Первый — анкетирование. Это классические онлайн-тесты с несколькими вариантами ответов. Например:

Определите модель приложения, рассматриваемую в рамках предметной области безопасности приложений:

1. *Поток управления*
2. *Поток данных*
3. *Поток управления и множество производных от него потоков данных*
4. *Множество производных потоков управления от потока данных*
5. ***Потоки данных и управления, производные друг от друга***

Второй вид домашек — практические работы. К примеру, студентам нужно было построить модель угроз и оценить риски для конкретного приложения. Или провести поиск потенциальных уязвимостей в соответствии с построенной моделью угроз.

Все работы, подразумевающие проверку преподавателем, я смотрел сам и оценивал по десятибалльной шкале. Некоторые задания мы вместе разбирали на вебинарах. А если у студентов оставались вопросы, я проводил дополнительные консультации.

НЕМНОГО ОБ ИНСТРУМЕНТАХ

Техническая платформа курса сделана неплохо. Это большой плюс в копилку и ВШЭ, и «Нетологии». Над проектом работает профессиональная команда, и это чувствуется. Все лекции, задания и результаты доступны на специализированной платформе ВШЭ, у каждого учащегося есть свой аккаунт.

Преподавателям помогала команда «Нетологии». За мной был закреплен методист, который давал рекомендации по проработке материалов курса. Каждый слайд лекций проходил через редактора-корректора; кроме того, коллеги помогали при записи видео.

Онлайн-площадка выручала: общаться с такой большой и разнообразной аудиторией вживую было бы куда сложнее. Но без проблем, конечно, не обошлось. Мой курс был частью большой программы — одним из модулей, которые студенты сами выбирали и комбинировали. При этом в программе были взаимоисключающие модули. К примеру, учащимся нужно было выбрать либо программирование на Python, либо криптографию. Проблема в том, что задания моего модуля предполагали владение Python, поэтому те, кто выбрал криптографию, не могли делать домашку. Мы обсудили этот косяк с организаторами и договорились исправить его в будущем.

ЧТО ДАЛЬШЕ

Нынешняя группа студентов продолжает обучение, а в ноябре ко мне придет новый поток. В этот раз я планирую привлечь к работе коллег из Positive Technologies. Все-таки каждый раз проверять 70 домашек — это долго и не так уж просто.

Кроме того, я планирую активнее использовать на вебинарах наши ИБ-продукты. Это отличный способ показать, как та или иная функциональность реализуется на практике. Тем более что я знаю все нюансы. Например, в новом учебном году я хочу использовать PT Application Inspector на практических занятиях, посвященных поиску и устранению уязвимостей в коде.

Реверанс

Хочу отдать должное команде «Нетологии». При чтении докладов я привык ориентироваться на слайды, но в случае с видеолекциями так нельзя — нужно смотреть в камеру. Мы отдельно записывали каждый абзац, иногда на сорокаминутный ролик уходил целый день. Сначала я то и дело косился на слайды, жутко нервничал, и мы снова и снова переписывали один и тот же кусок. Коллеги меня поддерживали, помогали и отпаивали чаем :) Постепенно я освоился, дело пошло веселее, и во многом это заслуга команды «Нетологии». В будущем я бы не хотел менять платформу, хотя есть и другие, не менее эффективные инструменты.

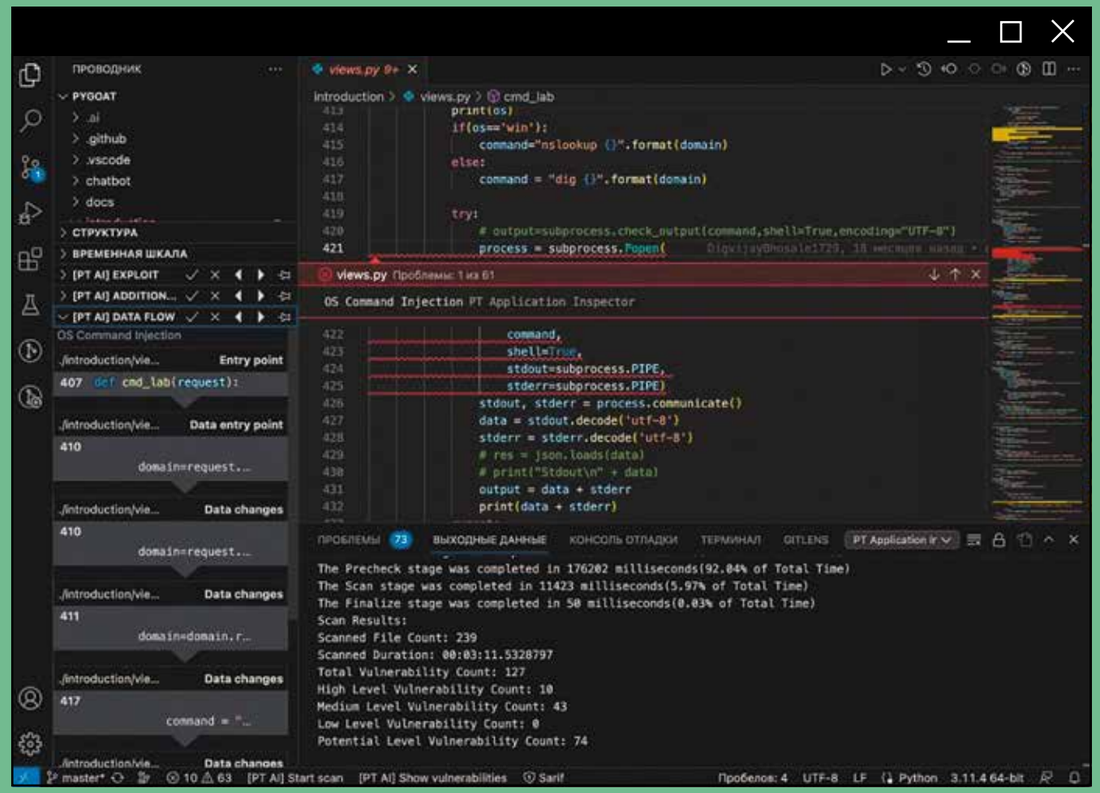


Рис. 1. Интерфейс PT Application Inspector

БУДУЩИМ ПРЕПОДАВАТЕЛЯМ ПОСВЯЩАЕТСЯ

- Не стоит идти в преподавание только ради денег. Гонорары составляют примерно 1/10 часть моей зарплаты, а времени зачастую уходит даже больше. Кроме того, я убежден, что одной из главных задач преподавателя является «зажигание глаз» студентов, пробуждение интереса к предметной области. Если вы работаете только ради денег, у вас вряд ли это получится.
- «Метод прогрессивного джипега» от Артемия Лебедева прекрасно ложится на подход к разработке учебных материалов. Нужно идти от общего к частному, чтобы в любой момент у вас была возможность использовать уже имеющиеся материалы при проведении занятий.
- В области безопасности приложений (как, впрочем, и во всей ИБ) практика имеет куда большее значение, чем теория. А негативный опыт играет не меньшую роль, чем позитивный. Чтобы освоить язык программирования, нужно писать код, в котором поначалу будет много багов и ошибок. Чтобы научиться администрировать сеть, нужно завалить несколько инфраструктур в попытке настроить конфигурацию. Чтобы разобраться в безопасности приложений, нужно пропустить приличное количество уязвимостей. Любой подобный курс должен строиться вокруг практики, которую дополняет теория, а не наоборот.

Да пребудет с нами сила AppSec!

БИЗНЕС-ТРЕК
ВНДАУ'23

— **Какое количество — в процентах — руководителей крупнейших компаний понимают реальное положение дел в своей безопасности, в том числе знают степень взломанности?**

— Берем 100 компаний.

Юрий Максимов

Сооснователь, мажоритарный акционер и председатель совета директоров Positive Technologies

— Я считаю, максимум — каждый пятый. Максимум.

Максут Шадаев

Министр цифрового развития, связи и массовых коммуникаций РФ

— Я думаю, никто не знает… Это такой же вопрос, как «Сколько руководителей знают о проблемах или нарушениях требований пожарной безопасности у них в организации?».

Виталий Лютиков

Заместитель директора федеральной службы по техническому и экспортному контролю России

— Я думаю, 5–10%. Каждый десятый или двадцатый.

Денис Баранов

Генеральный директор Positive Technologies

— 30% максимум.

Алексей Волков

Вице-президент, директор по информационной безопасности «ВКонтакте»

ЧТО ЭТО?

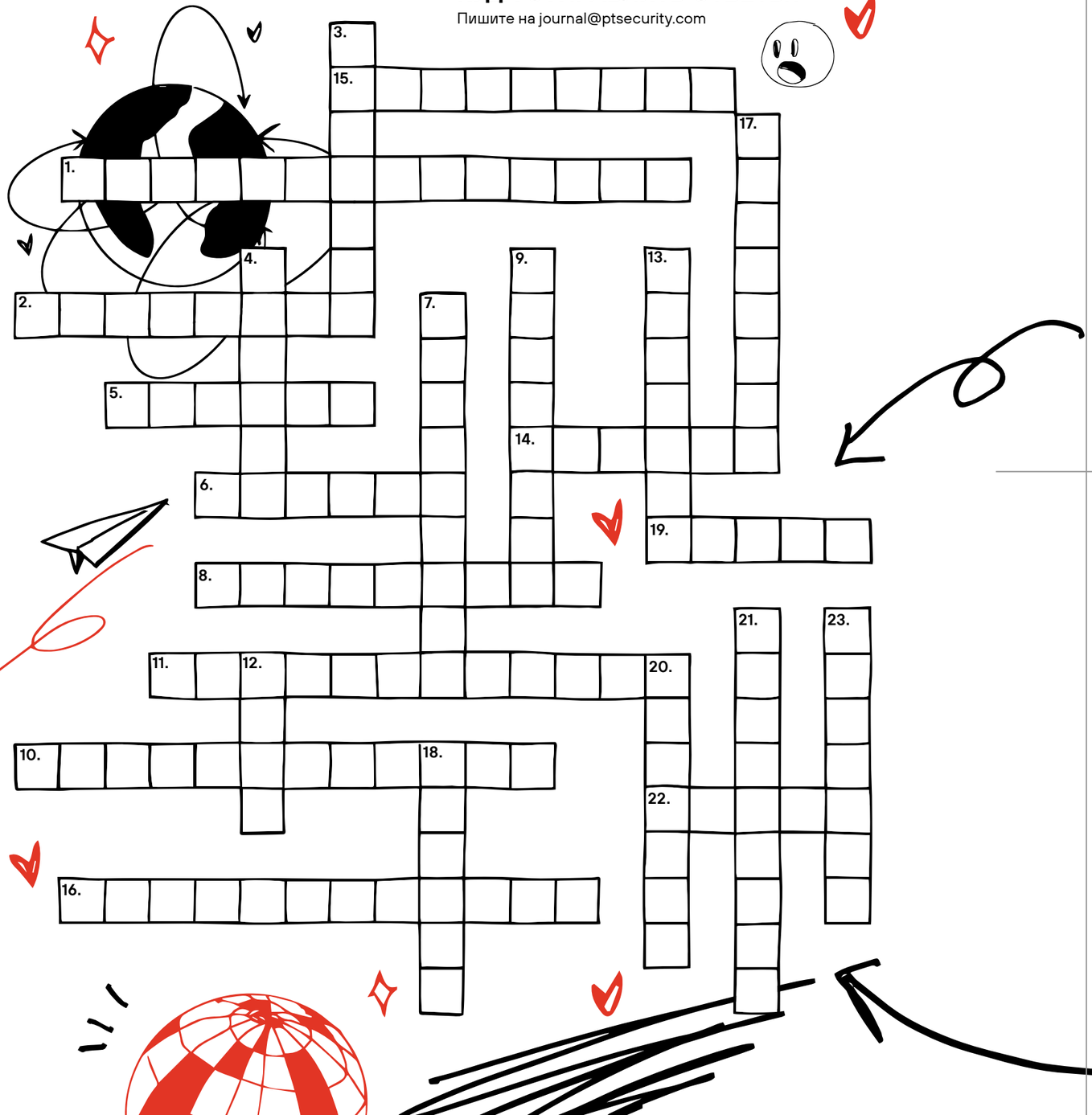
Кроссворд от детей наших сотрудников. Мы взяли известные ИБ-термины и попросили ребят — дошкольников и младшеклассников — дать им свои определения. В итоге получили неожиданные, нестандартные и прикольные обозначения. Правильных и неправильных ответов здесь нет ;)

ЧТО ДЕЛАТЬ?

Удивиться, улыбнуться и попробовать разгадать кроссворд. Первые 30 человек, которые правильно отгадают хотя бы 12 слов, получат мерч Positive Technologies и респект от редакции Positive Research.

КУДА ОТПРАВЛЯТЬ ОТВЕТЫ?

Пишите на journal@ptsecurity.com



ПО ГОРИЗОНТАЛИ:

1. Живой компьютер, который строит социальные сети.
   Цветок, похожий на акацию.
   Это чтобы безопасно зайти и заработать много денег.
2. Экспедиция на другую планету.
   Чтобы чинили компьютер, который ломают мошенники.
   Вид бабочек.
5. Это тот, кто пересидел в «Майнкрафте».
   Тот, кто варит манную кашу.
   Монетки.
6. Перемещение воды, уток и рыбы.
   Когда выгоняют плохих сотрудников.
   Название речки.
8. Когда баг ухаживает за маленьким багом.
   Такой банк есть, он охраняется, чтобы не сломали мошенники.
   Шлагбаум.
10. Звук, когда вылезает фото из фотика.
    Самое главное государство, Россия, чтобы оплатить коммунальные услуги.
    Грибное расписание.
11. Такой работник, который быстро работает и ему дают миллион денег.
    Число.
14. Например, с карточкой возьмем. Ты сделал себе запретную карточку. На первый день у тебя все хорошо, а на следующий день заходишь в «Сбер» «и видишь, что ее уже нет. Она не она.
    Огнедышащий дракон.
    Это просто программа для компьютера.
15. Человек превратился в зомби, и нашли противоядие.
    Это античит, у кого читы, того сразу забанят и он не сможет играть. Он у тебя удаляется сразу.
    Когда скачиваешь что-то запретное.
    Микроб, который лечит.
16. Перемещение поля с гиперскоростью.
    Гоночная машина, которая палит огнем.
    Погоны.
19. Вид рояля, на котором играют втроем.
    Это тот, кто плохо учится, на тройки.
    Радиоактивность. Вирус, может поломать компьютер и весь интернет.
    Лошадка с неприятным сюрпризом.
22. Динозавр какой-то.
    Изменение интерьера (дома меняются).
    Это когда тебя могут ограбить — когда у тебя умный дом с Алисой. Это номера домов для злоумышленников. И они могут выяснять, кто где живет.

ПО ВЕРТИКАЛИ:

3. Это когда потеет хорек.
   Работа, где задают вопросы.
   Когда украли капот.
4. Это когда в магазине батона нет.
   Робот, внутри которого сидит пришелец.
   Это что-то на английском.
7. Сиплый голос у человека.
   Гонщик патруля, который гоняется за мошенником.
   Когда полиция подает сигнал.
   Сигнал отправляют на планету Тура.
9. Племя индейцев / индейский президент.
   Может, обед какой-то. Институт. Мне кажется, это хорошее слово.
   Это студент-пинцет, с тонкими ножками как у пинцета.
12. Пофигический крот.
    Проникновение бактерий через рот.
    Кличка питомца.
13. Чтобы деньги хранить в банке.
    Пенная проверка.
17. Мошенники, пытаются сломать компьютеры, телефоны, запчасти и весь город.
    Болезнь в носу, или что-то в носу застряло и трудно вытащить.
    Вред для твоего телефона. Ты сможешь поиграть 2–3 дня, а потом не сможешь разблокировать телефон.
18. Ошибка, телевизор зависает, комп подает вирус, и не можешь смотреть мультики.
    Это когда рыбак устает рыбачить и пора домой, а вдруг смотрит — и у него много рыб.
    Процесс игры в ходилки.
20. Пудинг для пришельцев.
    Что-то связанное с координатами.
    Прием в вокале.
21. Программа, чтобы мама смогла говорить с другими сотрудниками.
    Сладкий фрукт.
    Кроссворд.
23. Папин блокнот для записей.
    Подарок для пришельца.
    Может, это Дарт Вейдер?
    Злоумышленники Google взламывают.
    Темная сова.

САЙТ POSITIVE RESEARCH