АВТОМАТИЧЕСКАЯ СИМУЛЯЦИЯ КИБЕРАТАК

НОВЫЙ ПОДХОД К РАЗВИТИЮ СИСТЕМЫ КИБЕРЗАЩИТЫ
НА БАЗЕ ОБЪЕКТИВНЫХ ДАННЫХ

CTRLHACK

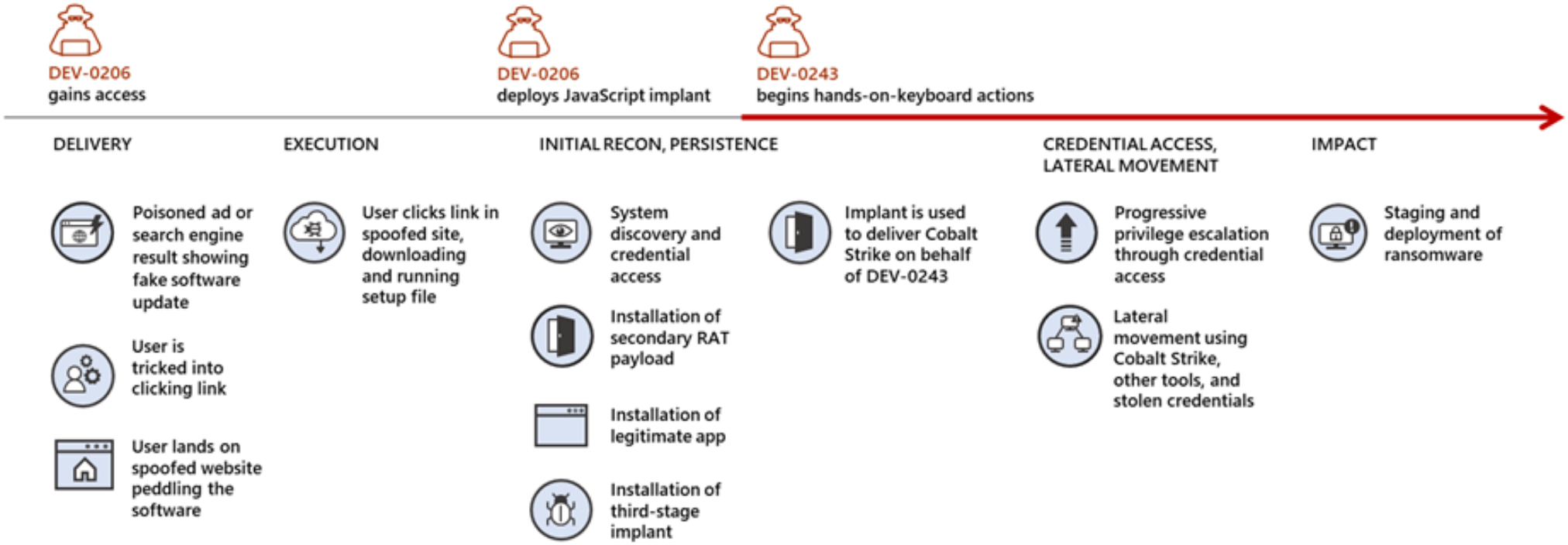
План

01 Примеры актуальных хакерских техник

02 Процесс повышения эффективности детектирования хакерских техник

03 Платформа CtrlHack, как основной инструмент построения процесса

Схема развития атаки на примере RaaS



https://www.microsoft.com/security/blog/2022/05/09/ransomware-as-a-service-understanding-the-cybercrime-gig-economy-and-how-to-protect-yourself/

Схема развития атаки на примере RaaS

- Получив первичный доступ, злоумышленник может долгое время не использовать его и не передавать третьим лицам

- В случае высоко-приоритетных целей цикл от первичного доступа до нанесения урона может занять всего лишь **несколько часов**

MSHTA и легитимные подписанные DLL

- Не нарушая целостности подписи, вредоносный контент размещался в легитимных подписанных DLL от Microsoft
- Созданные DLL были полиглотами (одновременно DLL и HTA-скрипт)
- Запуск посредством штатной утилиты mshta.exe (T1218.005)

https://research.checkpoint.com/2022/can-you-trust-a-files-digital-signature-new-zloader-campaign-exploits-microsofts-signature-verification-putting-users-at-risk/

MSHTA и легитимные подписанные DLL

Последствия:

- Уклонение от детектирования вредоносного контента засчет повышенного доверия к валидным подписанным файлам

Принятие мер:

- Выявление нетипичного для инфраструктуры использования mshta.exe

Там где не ждали: вредоносы в логах

- Основное вредоносное тело сохранялось в виде записей eventlog
- Запуск вредоносного тела осуществлялся с помощью скопированного werfault.exe (штатный) и wer.dll (нештатный) в одну папку (T1036)
- Wer.dll извлекала вредоносное тело из логов и запускала его
- В целях закрепления такой запуск werfault.exe прописывался в реестре CurrentVersion\Run (T1547.001)

https://securelist.com/a-new-secret-stash-for-fileless-malware/106393/

Там где не ждали: вредоносы в логах

Последствия:

- Уклонение от обнаружения
- Минимальное количество файловых вредоносных артефактов

Принятие мер:

- Отслеживание запуска штатных утилит из нетипичных мест
- Отслеживание создания ключей реестра, отвечающих за автозапуск

LNK-файлы как инструмент фишинга

- Microsoft усилила защиту для документов с макросами, в ответ злоумышленники ищут новые форматы
- Рассылка в емейлах LNK-файлов в чистом или архивном виде
- LNK-файл обычно содержит oneliner для получения или извлечения нагрузки и её запуска посредством regsvr32/rundll32 (T1218.010/T1218.011)

https://www.netskope.com/blog/emotet-new-delivery-mechanism-to-bypass-vba-protection

LNK-файлы как инструмент фишинга

Последствия:

- Обход почтовых антивирусов и песочниц
- Упрощение фишинга

Принятие мер:

- Регулярная проверка и обновление средств защиты
- Выявление нетипичного для инфраструктуры использования regsvr32.exe/rundll32.exe в корреляции с другими событиями

Доверяй, но проверяй (часть 1)

- Злоумышленники приносили с собой и использовали исполняемые файлы, входящие в состав средств защиты, для запуска вредоносной нагрузки
- Использовалась техника dll-sideloading, что позволяло выполнять активность от доверенного процесса (T1574.002)
- Далее выполнялось закрепление с помощью задач планировщика (T1053.005) или создания службы (T1543.003)

https://www.sentinelone.com/labs/moshen-dragons-triad-and-error-approach-abusing-security-software-to-sideload-plugx-and-shadowpad/

Доверяй, но проверяй (часть 1)

Последствия:

- Уклонение от блокирования средствами защиты
- Сложность выявления атакующей активности

Принятие мер:

- Выявление атак dll-sideloading (может быть нецелесообразным и сложным)
- Отслеживание создания задач планировщика и служб, нетипичных для инфраструктуры

Доверяй, но проверяй (часть 2)

- Драйвер, входящий в поставку средства защиты, позволял злоумышленникам уничтожать любые, даже защищенные, процессы
- Установка драйвера осуществлялась посредством создания службы (T1543.003)
- Далее точечно ликвидировались процессы, мешающие вредоносной активности (T1562.001)

https://cyber.aon.com/aon_cyber_labs/yours-truly-signed-av-driver-weaponizing-an-antivirus-driver/

Доверяй, но проверяй (часть 2)

Последствия:

- Нарушение функций средств защиты

Принятие мер:

- Отслеживание загрузки нетипичных для инфраструктуры драйверов
- Отслеживание создания задач планировщика и служб, нетипичных для инфраструктуры

Детектирование техник. Реальность

5 из 14 топ техник MITRE детектируются

80% техник MITRE не покрываются правилами, поставляемыми вендорами SIEM

15% правил, поставляемыми вендорами SIEM не работают корректно в реальной инфраструктуре

190+ описаний различных техник в матрице MITRE

Отчет компании Cardinalops «2022 The state of SIEM detection risk»

Текущая ситуация

Как повысить эффективность выявления хакера в сети?

1. Детектировать как можно больше различных техник (TTP)
2. Постоянно проверять и настраивать компоненты системы защиты:
   - реагирование средств защиты
   - полнота сбора информации в систему мониторинга
   - обработка, приоритезация событий и правила реагирования
3. Покрывать всю инфраструктуру

Как убедиться, что техники хакеров детектируются?

Нужно запускать техники в сети ... ... Автоматически

Существующие процессы

- Построен процесс управления уязвимостями

  Наиболее качественный процесс. Но совсем не работает с TTP

- Построен процесс Threat Intelligence

  Работа только с конкретными экземплярами. В продвинутых случаях ручная работа по TTP

- Проводятся пентесты (возможно red team)

  Работа только по отдельным векторам и части инфраструктуры. Эпизодическая работа

- Проводятся киберучения

  Не на своей инфраструктуре. Только тренировка персонала

Нужен отдельный процесс совершенствования правил детектирования TTP

Новый процесс

- Получение и анализ информации по техникам
  - Платформы TI + отдельные отчеты об атаках
- Запуск техник
  - Искать в сети или писать самим тесты для проверки техник
- Анализ результатов и написание правил детектирования
  - Анализ событий для всей сети и написание правил
- Повторный запуск техник для проверки
  - Проверка созданных правил

И все эти шаги нужно выполнять постоянно
(инфраструктура меняется, появляются новые техники)

Для непрерывного процесса нужны средства автоматической симуляции, покрывающие всю инфраструктуру

CTRLHACK

базовый инструмент для построения процесса

Общее описание

CTRLHACK – российский продукт класса Breach and Attack Simulation.

CTRLHACK позволяет автоматически выполнять симуляции техник, используемых хакерами.

Действия атакующих имитируются для того, чтобы определить, как на них реагируют средства защиты и насколько эффективны процессы детектирования и реагирования.

Минцифры России

В реестре Российского ПО

Как это работает

Для симуляций необходимы агенты

- Дистрибутив агента скачивается из интерфейса управления
- Windows, Linux, MacOS

Симуляция – заданная в скрипте последовательность действий

- Скриптовый язык для симуляции
- Подробный протокол активности в рамках симуляции
- Откат внесенных атакующей техникой изменений

Для работы CTRLHACK не нужно вносить изменения в инфраструктуру

Основные функции

Проверка средств защиты периметра и конечных точек

Модуль предназначен для проверки блокирования средствами защиты актуальных вредоносных файлов и попыток соединения с адресами из «черных» списков

Основной модуль – атакующие техники по стадиям матрицы MITRE

Модуль предназначен для проверки детектирования актуальных атакующих техник, применяемых хакерами после проникновения в сеть

Первичный доступ

Симуляция – работа с реальными вредоносными адресами и файлами

Без запуска. Только соединение, скачивание и выкладывание на диск

- Посещение «вредоносных» сайтов
- Скачивание через web вредоносных файлов
- Получение вредоносных файлов по e-mail

Постоянно обновляемая база адресов и экземпляров вредоносных файлов

ABUSE.ch

Базовые атакующие техники

Симуляция – действия в ОС, специфичные для атакующих техник

ФАЙЛЫ	РЕЕСТР	ПРОЦЕССЫ	СЕТЬ

Симуляция техник по разным стадиям атаки

MITRE | ATT&CK®

- Более 200 реализаций техник
- Техники для ОС Windows, Linux, MacOS
- Атомарные техники и сценарии

Постоянно обновляемая база атакующих техник

Какие задачи решает?

CTRLHACK поможет определить реальный уровень защищенности инфраструктуры

Проведение симуляций атакующих действий хакеров на постоянной основе позволяет выявить и устранить проблемы в работе средств защиты и повысить эффективность SOC.

ПРОВЕРКА СРЕДСТВ ЗАЩИТЫ

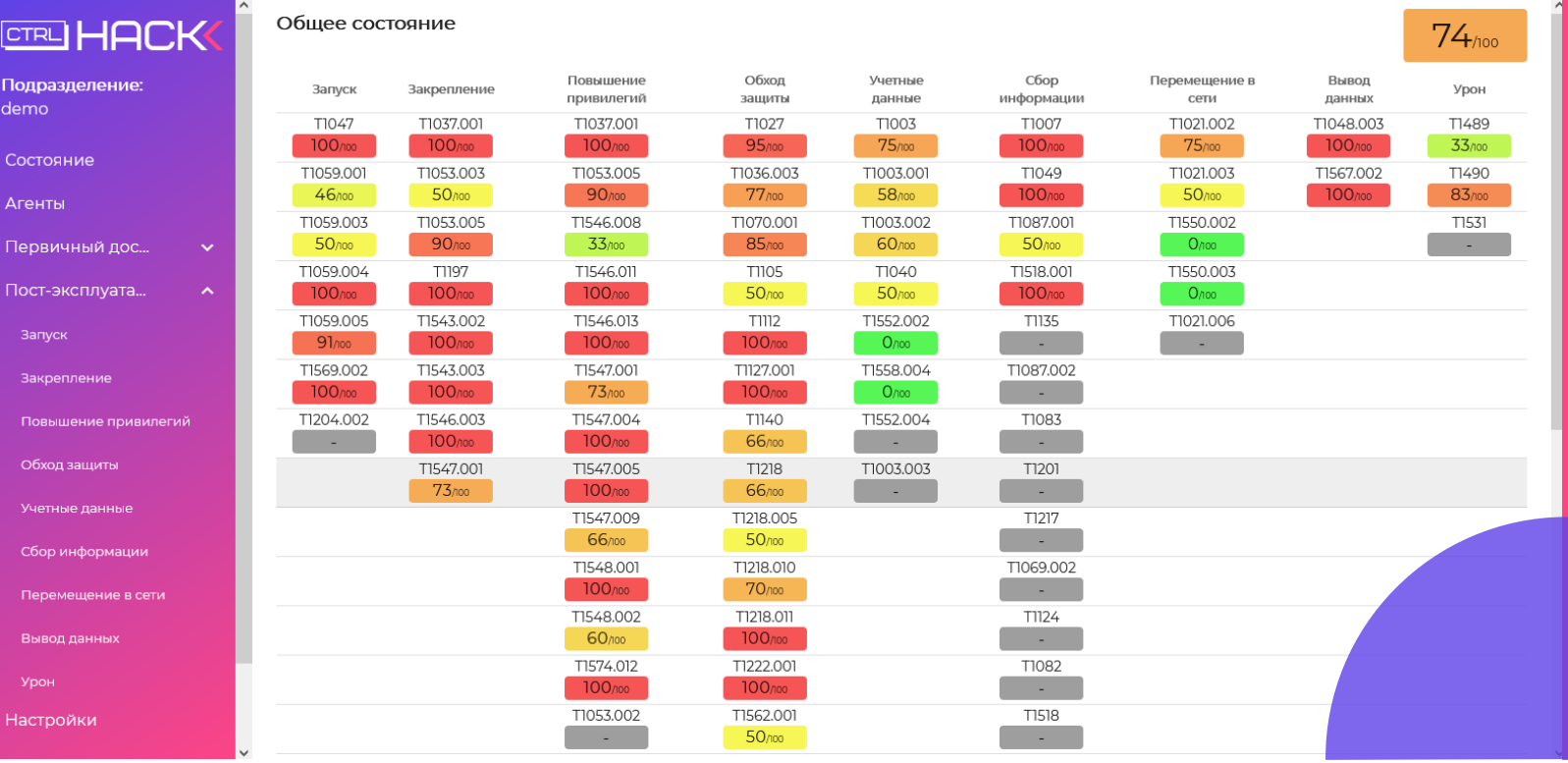
Какие из атакующих действий блокируют средства защиты?
Как работают средства защиты в разных сегментах сети?

ДЕТЕКТИРОВАНИЕ ТЕХНИК

Какие атакующие техники не детектируются? Какие события для каждой атакующей техники есть в SOC, а каких не хватает?

РАЗВИТИЕ SOC

Формируются ли инциденты в SOC? Как команда реагирует на инциденты? Как быстро устраняются инциденты?



Управление платформой запуск симуляций и анализ результатов проводится в удобном и понятном интерфейсе

ИНТЕРФЕЙС

Результаты выполнения симуляций и оценка рисков для каждой стадии проведения атаки отображаются в графическом виде.

Можно получить как сводную оценку текущего состояния киберзащиты, так и детальный отчет по каждой атакующей технике.

Процесс на базе CtrlHack

Непрерывной процесс совершенствования правил детектирования хакерских техник

Стадии процесса:

1. первичный контроль средств защиты и системы мониторинга

   (выделенный сегмент 10-20 машин)

2. настройка СЗИ, расширение объема событий, разработка правил детектирования
3. поэтапная отработка техник на разных стадиях во всей инфраструктуре
4. быстрая проверка детектирования новых техник

Суровая реальность...

Результаты применения CtrlHack

01 30-70% техник не блокируется

02 Для Linux до 90% техник не блокируется и не детектируется

03 Для 30% техник нет событий в SIEM

04 Различия в настройках СЗИ в разных сегментах сети

05 Отличия в полноте сбора событий в разных сегментах сети

WWW.CTRLHACK.RU

СПАСИБО!

ООО «КонтролХак»

127299, г. Москва, ул. Космонавта Волкова, д.20

+7 495 789-72-97

info@ctrlhack.ru