Таск №1

Изучая содержимое файловой системы, находим в директории пользователя файл менеджера паролей Keepass (credentials.kdbx):

```
scp@scp:~$ ls /home/scp/
credentials.kdbx
scp@scp:~$
```

```
scp@scp:~$ sudo file credentials.kdbx
credentials.kdbx: Keepass password database 2.x KDBX
```

Перекинем файл с виртуальной системы на другую виртуалку с необходимым инструментарием. Для получения доступа к содержимому kdbx-файла необходимо знать мастер-пароль. Можно попробовать восстановить пароль.

Сначала извлечем хеш мастер-пароля с помощью инструмента keepass2john:

```
┌──(kali㉿kali)-[~]
└─$ sudo keepass2john credentials.kdbx > keepass_hash.txt

┌──(kali㉿kali)-[~]
└─$ cat keepass_hash.txt
credentials:$keepass$*2*60000*0*60cd8aac1aee268064eff6509a11c97faecf9e73564a4
4c72e82ac802273db46*2f55fcbc605183c5912a4d76e474340986d429ac08a9ac1b00889524d
543b4a5*3ffd11ad0e7ff101d8bf4f3c039714f4*a2c61cd079d781464c16b087af6fe9697f09
a1744121db97df0c3c413c96e291*8818e3ecaec7140136ccf4d6d4e987764b75e36806e7d175
fc20f93b496b0c67

┌──(kali㉿kali)-[~]
└─$
```

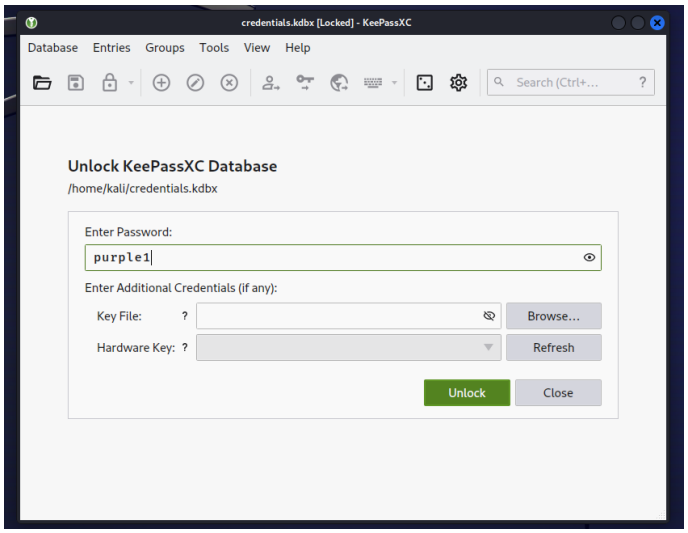
Удалим лишние данные из начала файла (название базы «credentials:»), чтобы получить чистую хеш-сумму:

```
┌──(kali㉿kali)-[~]
└─$ cat keepass_hash.txt
$keepass$*2*60000*0*60cd8aac1aee268064eff6509a11c97faecf9e73564a44c72e82ac802
273db46*2f55fcbc605183c5912a4d76e474340986d429ac08a9ac1b00889524d543b4a5*3ffd
11ad0e7ff101d8bf4f3c039714f4*a2c61cd079d781464c16b087af6fe9697f09a1744121db97
df0c3c413c96e291*8818e3ecaec7140136ccf4d6d4e987764b75e36806e7d175fc20f93b496b
0c67
```

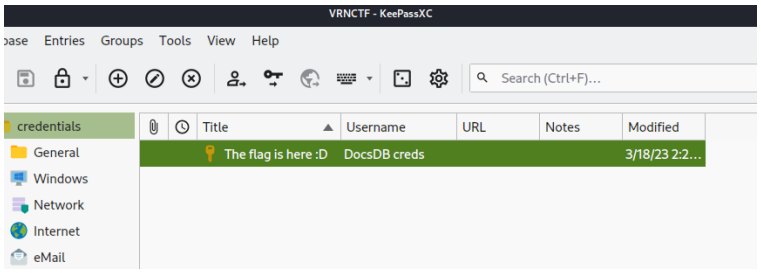
Для взлома хеш-суммы можно воспользуемся инструментом hashcat (можно также использовать любой аналог, поддерживающий хеш-суммы данного типа, например john the ripper). Для взлома необходимо указать тип хеш-суммы (режим hashcat 13400) и словарь для перебора, например rockyou.txt:

```
└─$ hashcat -m 13400 keepass_hash.txt -a0 /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LLV
M 14.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================
* Device #1: pthread-sandybridge-AMD Ryzen 9 5900X 12-Core Processor, 2916/58
97 MB (1024 MB allocatable), 6MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

$keepass$*2*60000*0*60cd8aac1aee268064eff6509a11c97faecf9e73564a44c72e82ac802273db46*2f55fcbc605183c5912a4d76e474340986d429ac08a9ac1b00889524d543b4a5*3ffd11ad0e7ff101d8bf4f3c039714f4*a2c61cd079d781464c16b087af6fe9697f09a1744121db97df0c3c413c96e291*8818e3ecaec7140136ccf4d6d4e987764b75e36806e7d175fc20f93b496b0c67:purple1
```

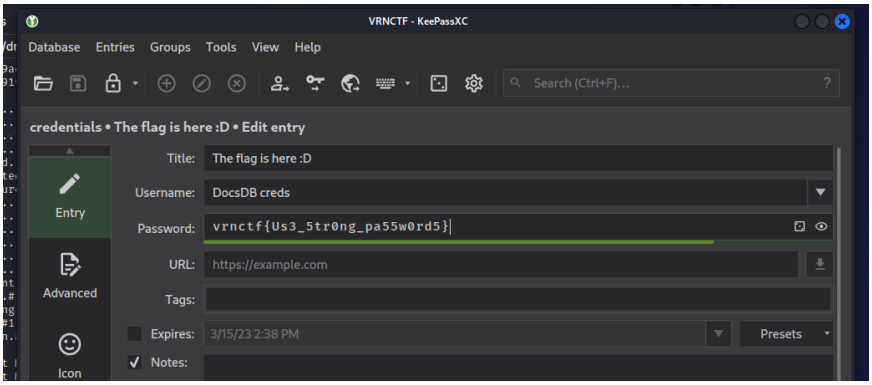
В результате перебора мы получаем мастер-пароль от базы Keepass. Остается только импортировать базу в Keepass и посмотреть содержимое:



Внутри видим сохраненные учетные данные:



Открываем редактирование записи и в поле с паролем находим флаг:



Флаг: vrnctf{Us3_5tr0ng_pa55w0rd5}

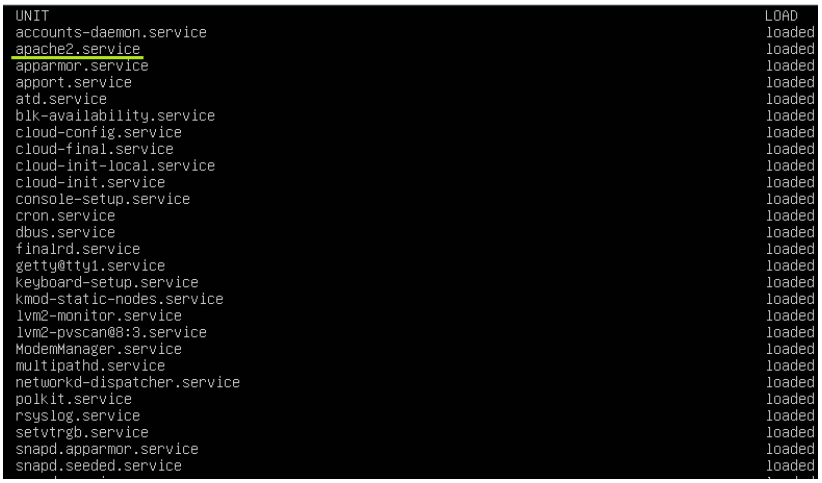
Таск №2

Одним из способов закрепления в системе является использование веб-оболочек.

С помощью команды получим список запущенных сервисов:

- sudo systemctl list-units --type service

Интерес представляет запущенный веб-сервер apache:

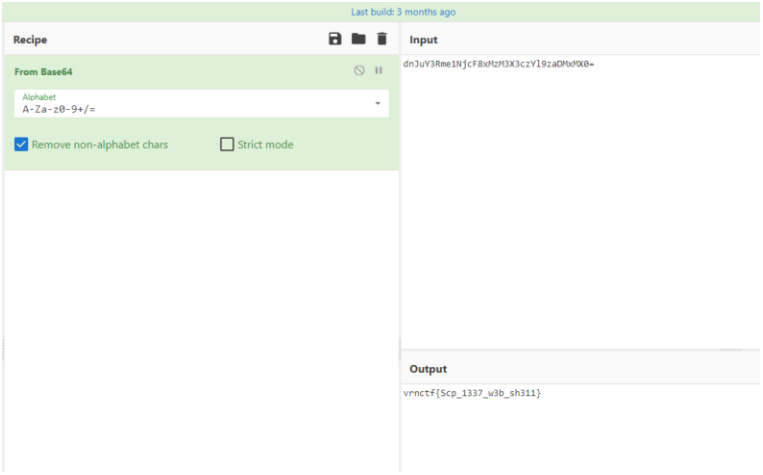


Изучив содержимое стандартной директории веб-сервера /var/www/html/, мы видим интересный файл «index.php»:

Изучив файл, становится понятно, что перед нами модифицированная веб-оболочка p0wny-shell:

Покопавшись в исходном коде, замечаем наличие функции genFlag и собираемой строки в кодировке base64:

Собираем, декодируем строку и получаем флаг:



Флаг: vrnctf{Scp_1337_w3b_sh311}

Таск №3

Из условия задания становится понятно, что некий файл запускается регулярно несмотря на то, что администратор завершал процесс. Добиться подобного закрепления в скомпрометированной системе можно с помощью планировщика задач. В нашем случае – cron.

Проверим список запланированных заданий пользователя scp:

```
scp@scp:~$ crontab -l
no crontab for scp
scp@scp:~$
```

Хмм… пусто, что же по root-пользователю:

```
scp@scp:~$ sudo crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 3 * * * /usr/share/docs/beacon > /dev/null 2>&1
scp@scp:~$ _
```

Видим, что раз в сутки осуществляется запуск файла «beacon» из директории /usr/share/docs/:

```
scp@scp:~$ ls /usr/share/docs/
beacon
```

```
scp@scp:/usr/share/docs$ sudo file beacon
beacon: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, stripped
scp@scp:/usr/share/docs$
```

Сохраним файл себе и откроем его файл в HEX-редакторе. Поищем сочетание «vrnctf» и получим флаг (то же самое можно сделать с помощью утилит strings и grep):

```
00D63830  5C E8 2D 70 6F 6A 01 00 20 00 00 00 00 00 00 00  \и-poj.. .......
00D63840  D7 76 00 00 F2 76 00 00 C5 C0 0B 00 04 00 00 00  Чv..тv..ЕА......
00D63850  B9 01 00 00 05 00 00 00 00 00 00 07 00 00 00 00  №...............
00D63860  00 00 00 00 00 00 00 F2 76 00 00 60 14 00 00     ........тv..`...
00D63870  76 72 6E 63 74 66 7B 4E 31 63 33 5F 63 32 5F 66  vrnctf{N1c3_c2_f
00D63880  72 61 6D 65 77 30 72 6B 7D C4 DE B7 8C 6A 01 00  ramew0rk}ДЮ·Њj..
00D63890  20 00 00 00 00 00 00 00 C8 C0 0B 00 6D 11 00 00   .......ИА..m...
00D638A0  D1 C0 0B 00 04 00 00 00 B9 01 00 00 20 00 00 00  СА......№... ...
00D638B0  00 00 00 07 26 6F 05 00 D8 C0 0B 00 00 00 00 00  ....&o..ША......
```

Еще один вариант решения предусматривает использование YARA-правила. Пример составленного правила для поиска HEX от «vrnctf» в структуре файла:

```
rule vrnctf
{
    meta:
        description = "Rule to detect vrnctf{ bytes"

    strings:
        $pattern_0 = { 76 72 6e 63 74 66 7b }

    condition:
        1 of them
}
```

В результате поиска мы получаем номер строки, содержащей искомое значение:

```
Windows PowerShell
PS C:\Users\Public\123\Samples\Tools\Program_Files\malware_analysis\detection\yara\yara-v4.0.0-1305-win64> .\yara64.exe
.\vrnctf.yar C:\Users\scst\Desktop\sample\ -s
vrnctf C:\Users\scst\Desktop\sample\\scp_beacon
0xd63870:$pattern_0: 76 72 6E 63 74 66 7B
```

Флаг: vrnctf{N1c3_c2_framew0rk}

Таск 4

Посмотрим список пользователей, доступных в системе:

```
scp@scp:~$ sudo cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
scp:x:1000:1000:scp:/home/scp:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
scp@scp:~$ _
```

Ничего подозрительного нет. Может остались какие-то следы создания учетных записей? Проверим историю вводимых команд на сервере (sudo nano .bash_history):

```
reboot
git clone https://github.com/iagox86/dnscat2.git
ls
cd dnscat2
ruby -W0 dnscat2.rb --security=open --no-cache\
ls
cd server
ls
ruby -W0 dnscat2.rb --security=open --no-cache\
sudo ruby -W0 dnscat2.rb --security=open --no-cache\
sudo apt-get update
ip a
sudo apt-get upgrade
sudo apt-get upgrade -y
sudo ruby -W0 dnscat2.rb --security=open --no-cache\
useradd vrnctf -p vrnctf{5cp_wa5_cr3at3d}
gem install ecdsa\
sudo gem install ecdsa\
sudo ruby -W0 dnscat2.rb --security=open --no-cache\
ls
sudo bundle install
sudo ruby -W0 dnscat2.rb --security=open --no-cache\
reboot
ls
cd /usr/share/wordlists/
ls
sudo gzip -d rockyou.txt.gz
ls
userdel vrnctf
cd dnscat2/server
sudo ruby -W0 dnscat2.rb --security=open --no-cache\
cd /usr/share/
sudo git clone https://github.com/BlackArch/webshells.git
```

Флаг: vrnctf{5cp_wa5_cr3at3d}