Ethical Hacking, Pen Testing, Red Teaming and Bug Hunting Deep Dive

DAY 2

Omar Santos

@santosomar

About // Omar Ωr Santos

https://h4cker.org

@santosomar

https://h4cker.org/discord

/in/santosomar

h4cker.org/github

Omar Santos is an active member of the security community, where he leads several industry-wide initiatives and standard bodies. His active role helps businesses, academic institutions, state and local law enforcement agencies, and other participants that are dedicated to increasing the security of the critical infrastructure.

Omar is the author of over 20 books and video courses; numerous white papers, articles, and security configuration guidelines and best practices. Omar is a Principal Engineer of Cisco's Product Security Incident Response Team (PSIRT) where he mentors and lead engineers and incident managers during the investigation and resolution of security vulnerabilities.

Omar is often presenting at many cybersecurity conferences and he is the co-lead of the DEF CON Red Team Village (redteamvillage.io). He is also the chair of the OASIS Common Security Vulnerability Framework (CSAF) Technical Committee and the co-chair of the Forum of Incident Response and Security Teams (FIRST) PSIRT Open Source Security Working Group.

Omar has been quoted by numerous media outlets, such as TheRegister, Wired, ZDNet, ThreatPost, CyberScoop, TechCrunch, Fortune Magazine, Ars Technica, and more.

Omar's PGP Key: 0x8e19a9d13af27edc

DISCLAIMER/ WARNING

- The information provided on this training is **for educational purposes only**. The **author**, O'Reilly, or any other entity **is in no way responsible for any misuse of the information**.

- Some of the tools and technologies that you will learn in this training class may be illegal depending on where you reside. Please check with your local laws.

- Please practice and use all the tools that are shown in this training in a lab that is not connected to the Internet or any other network.

Agenda – Day 1

- An Overview of Ethical Hacking and Penetration Testing Methodologies
- Red Teaming vs. Pen Testing vs. Bug Bounties
- Building your own hacking lab with WebSploit Labs
- Passive Reconnaissance and Open-Source Intelligence (OSINT)
- Active Reconnaissance, Scanning, and Fuzzing
- Introduction to Hacking Modern Web Applications
- Introduction to Hacking User Credentials and Cracking Passwords
- Introduction to Hacking Databases and SQL Injection

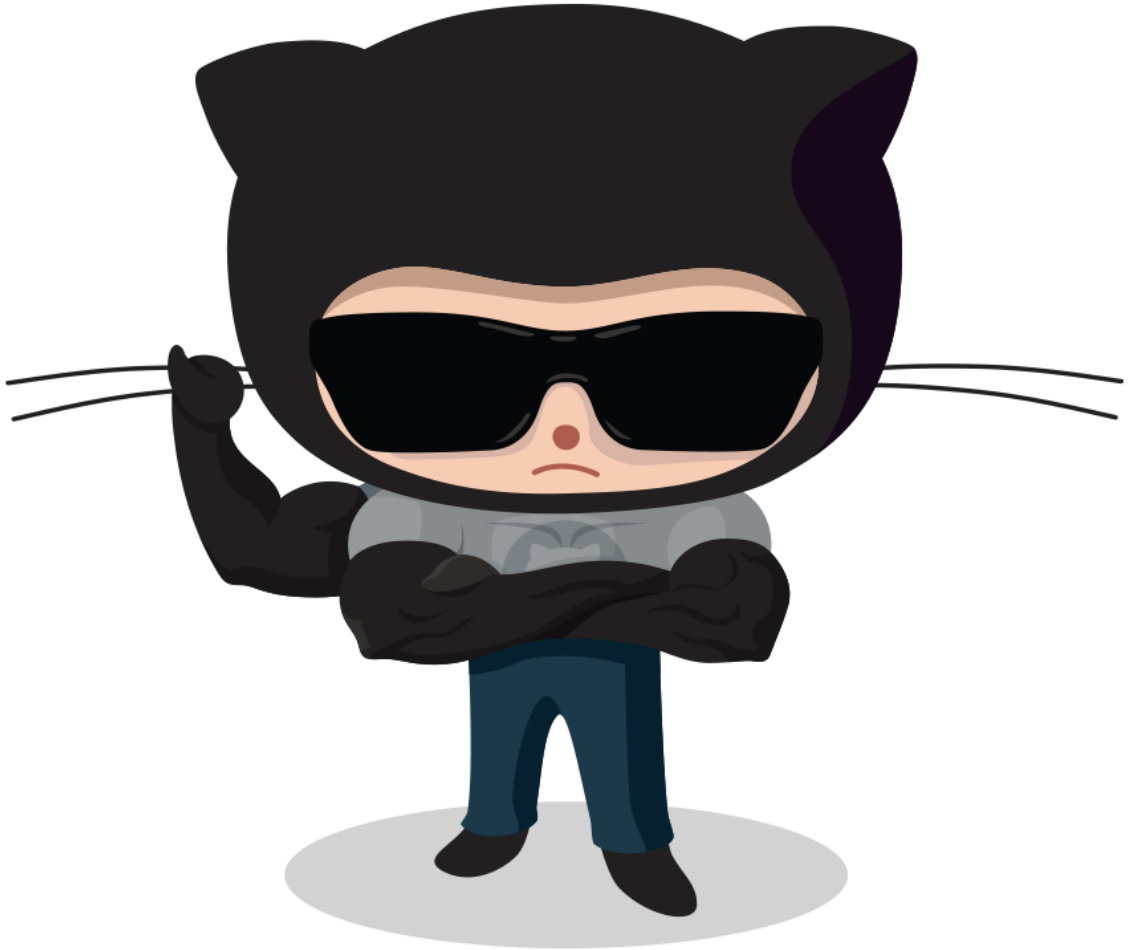
Agenda – Day 2

- Introduction to Hacking Networking Devices
- Fundamentals of Wireless Hacking
- Introduction to Buffer Overflows and Creating Payloads for Code Execution
- Introduction to Social Engineering
- Fundamentals of Evasion and Post Exploitation Techniques
- Command and Control, Exfiltration, and Privilege Escalation
- Best Practices on How to Write Penetration Testing Reports

Learning Path: https://h4cker.org/learning-path

GitHub Repo

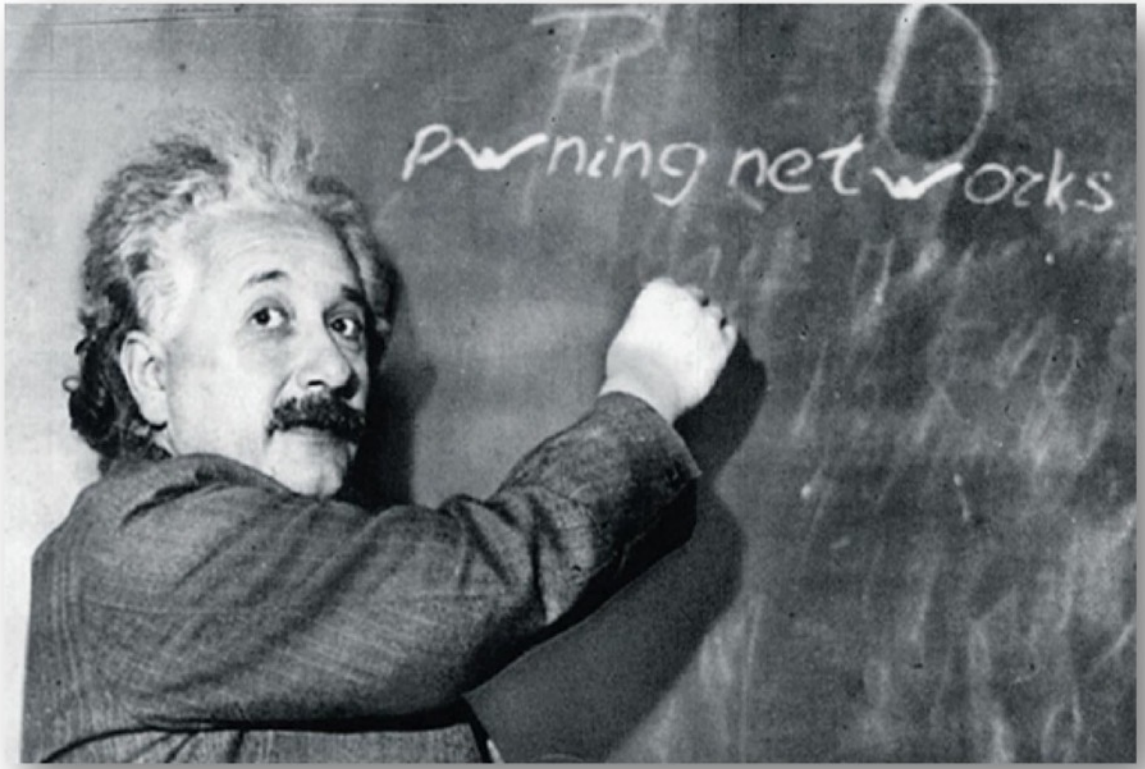
https://hackerrepo.org



Hacking Labs

- https://hackingscenarios.com

Introduction to Hacking Networking Devices



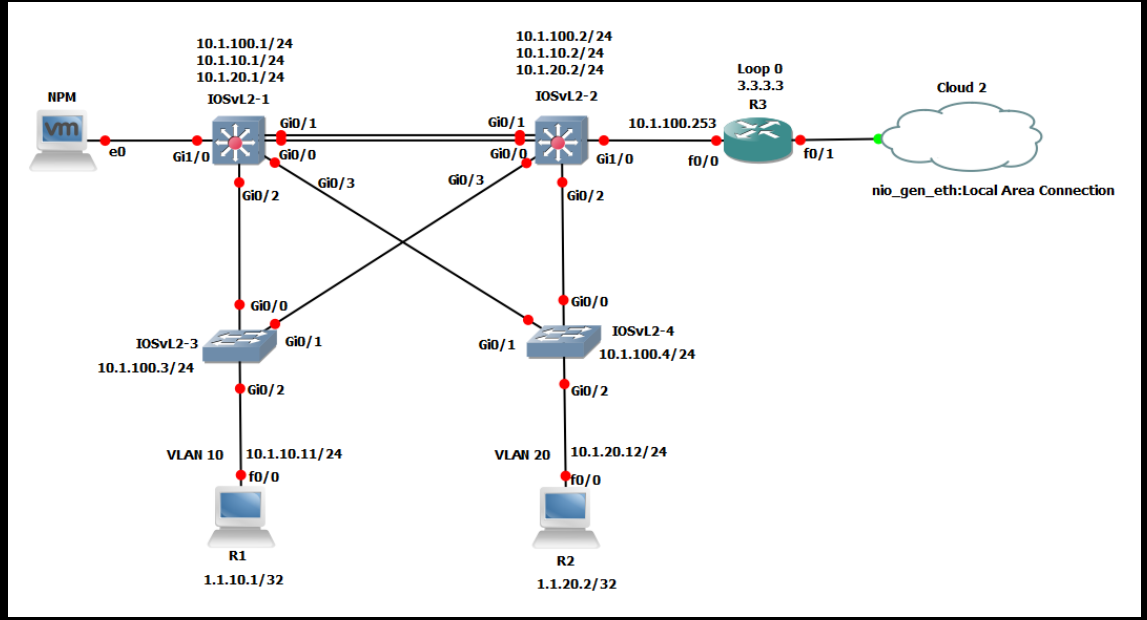
Why Hack Network Devices?

- Used as stepping stones.
- Mass surveillance.
- Sometimes not monitored as closed as your hosts.
- Longer system lifecycle.
- No malware detection.
- Sometimes running protocols designed back in the 80's.
- Take advantage of features like port mirroring, tunneling, lawful intercept to infiltrate and exfiltrate data.

https://theartofhacking.org/go/hacking_networks.html

GNS3 & Cisco Modeling Labs for Pen Testing

- https://developer.cisco.com/modeling-labs/
- https://www.gns3.com/
- https://www.netacad.com/courses/packet-tracer

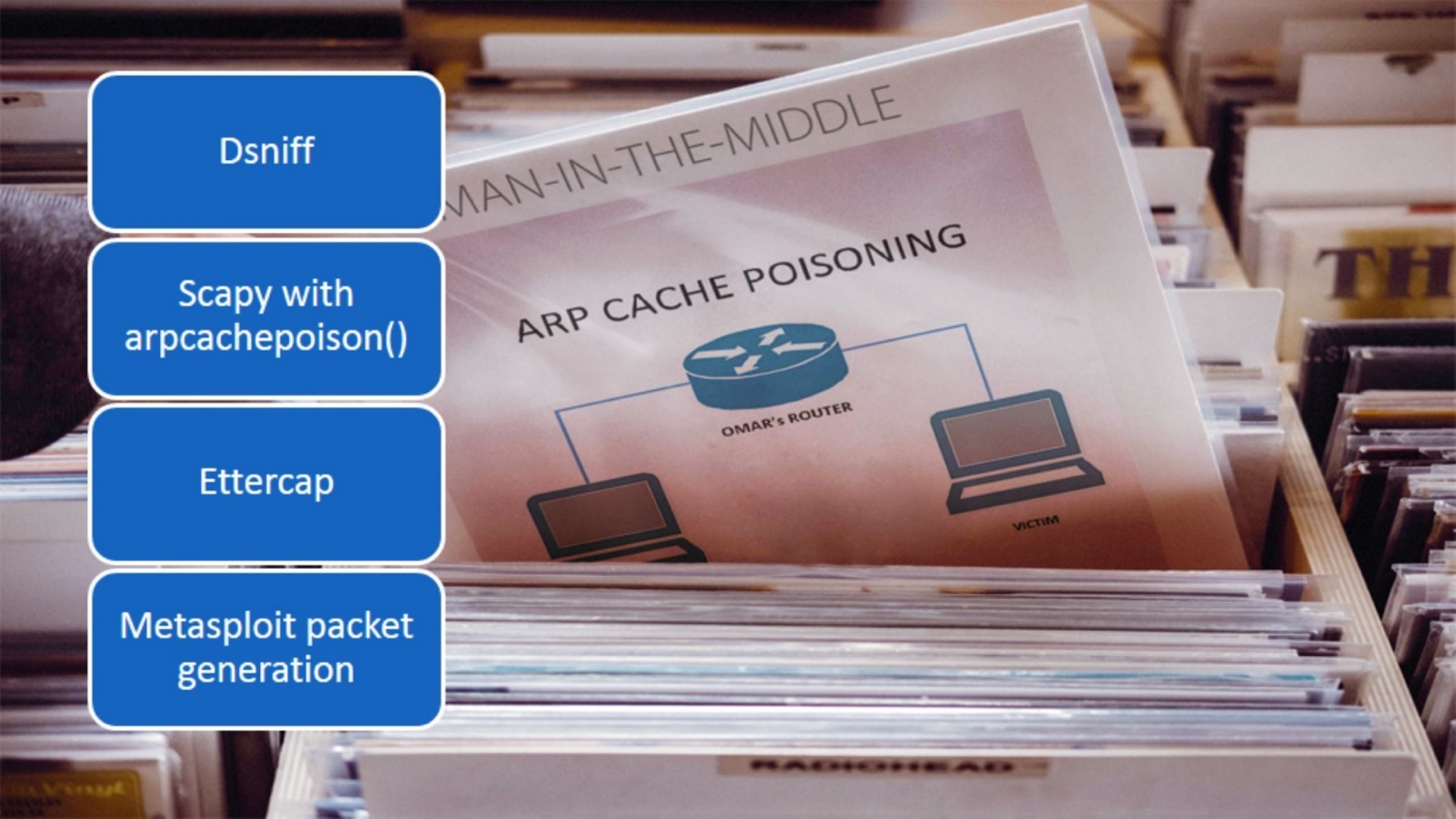


Example Attacks

- Known vulnerabilities (exploit-db, Metasploit, etc.)
- VTP Attacks
- DHCP Attacks
- ARP Cache Poisoning
- Routing Protocol Hijack
- Default Passwords!
- Weak Configurations

- Rogue DHCP Servers
- MiTM
- Firewall Evasion and Tunneling
- ARP Spoofing
- HSRP Attacks
- Spanning Tree Attacks
- MPLS Attacks
- 802.1Q Attacks
- 801.2X Attacks

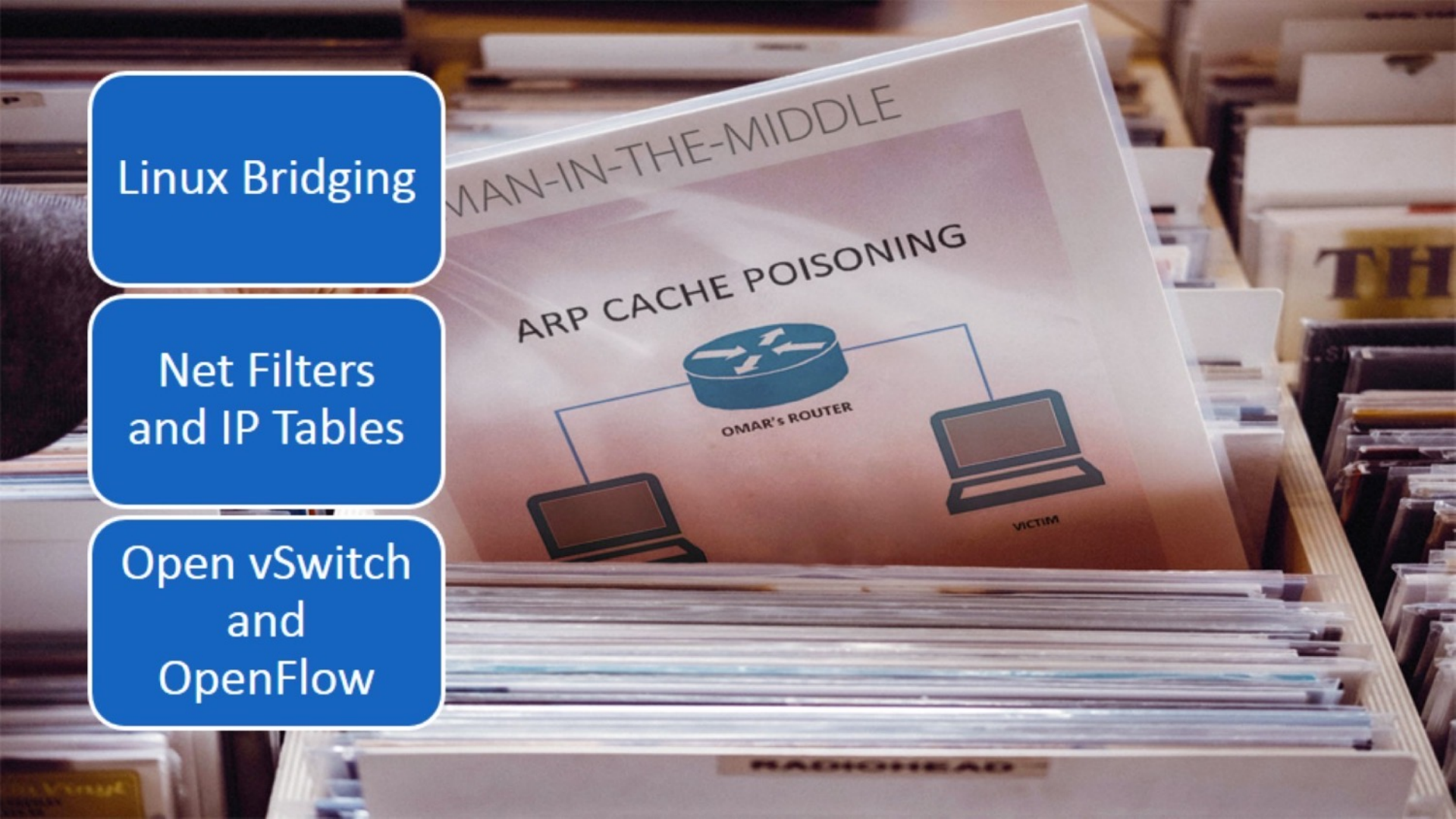
- Dsniff
- Scapy with arpcachepoison()
- Ettercap
- Metasploit packet generation



Linux Bridging

Net Filters and IP Tables

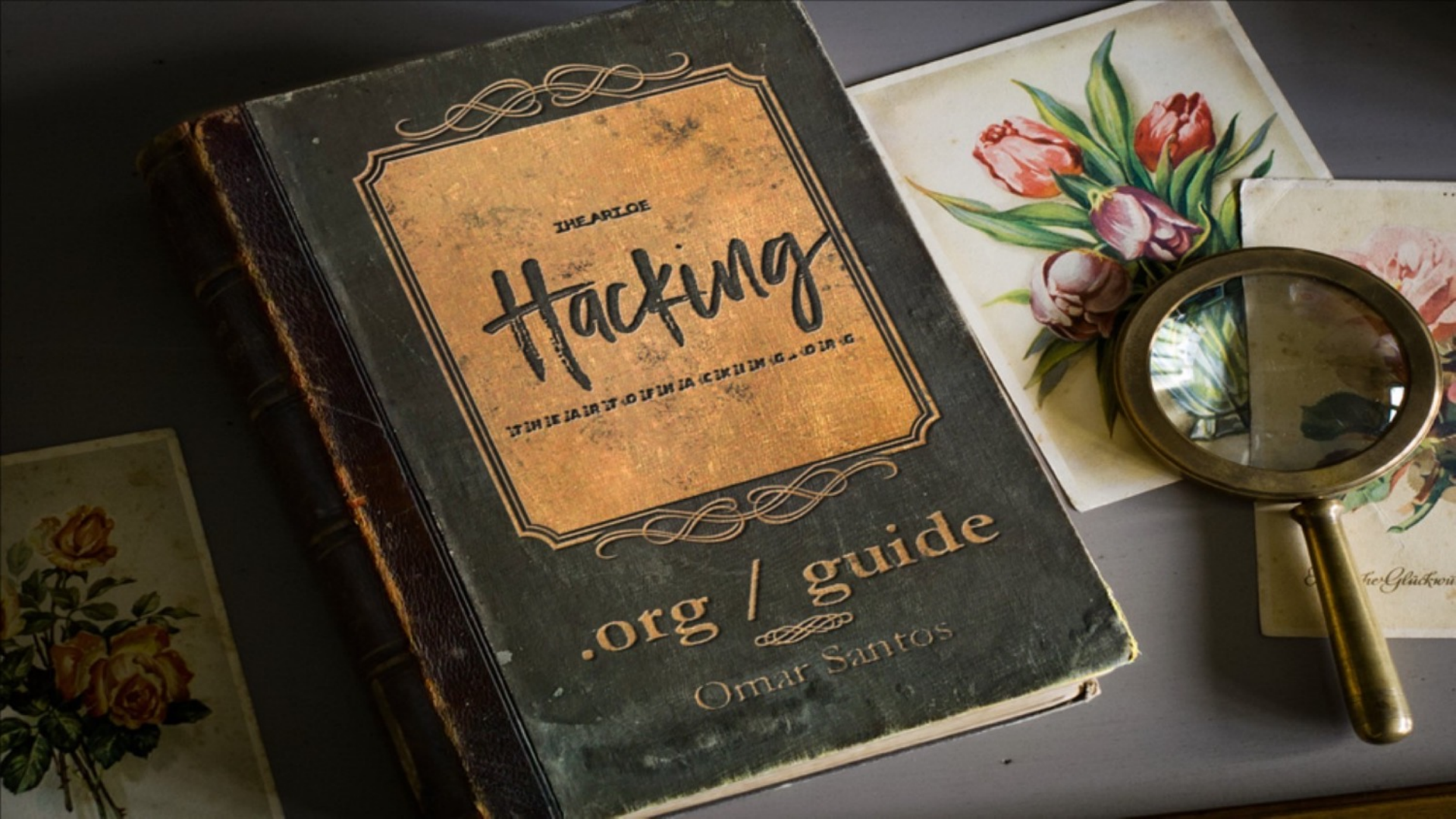
Open vSwitch and OpenFlow



```
┌──[omar@websploit]─[~]
└─$sudo scapy
[sudo] password for omar:
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)

                     aSPY//YASa
             apyyyyCY//////////YCa       |
            sY//////YSpcs  scpCY//Pp     | Welcome to Scapy
 ayp ayyyyyyySCP//Pp           syY//C    | Version 2.4.3
 AYAsAYYYYYYYY///Ps              cY//S   |
         pCCCCY//p          cSSps y//Y   | https://github.com/secdev/scapy
         SPPPP///a          pP///AC//Y   |
              A//A            cyP////C   | Have fun!
              p///Ac            sC///a   |
              P////YCpc           A//A   | Craft packets like it is your last
       scccccp///pSP///p          p//Y   | day on earth.
      sY/////////y  caa           S//P   |                     -- Lao-Tze
       cayCyayP//Ya              pY/Ya   |
        sY/PsY////YCc           aC//Yp
         sc  sccaCY//PCypaapyCP//YSs
                  spCPY//////YPSps
                       ccaacs

                                       using IPython 7.18.1

>>>
```


Applications
Places
Terminal
Sat 14:22
root@kali: ~/bo_example
File Edit View Search Terminal Help
yersinia 0.8.2 by Slay & tomac - VTP mode
[14:19:24]
Code Domain MD5 Iface Last seen
YERSINIA DEMO
Attack Panel
No DoS Description
0 sending VTP packet
1 X deleting all VTP vlans
2 X deleting one vlan
3 adding one vlan
4 X Catalyst zero day
Select attack to launch ('q' to quit)
Total Packets: 0
VTP Packets: 0
MAC Spoofing [X]
Those strange attacks...
VTP Fields
Source MAC 02:C2:DC:7F:8E:F3 Destination MAC 01:00:0C:CC:CC:CC
Version 01 Code 03 Domain
MD5 00000000000000000000000000000000 Updater 010.013.058.001
Revision 0000000001 Timestamp Start value 00001
Followers 001 Sequence 001


WIRELESS HACKING
Fundamentals

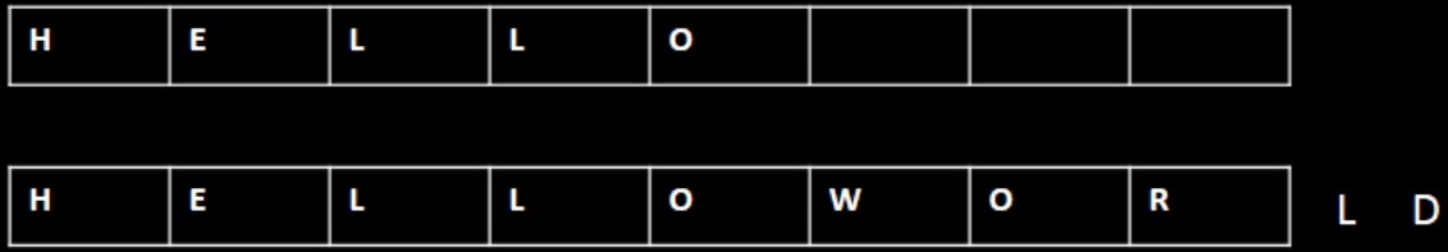

Hacking
.org / guide
Omar Santos

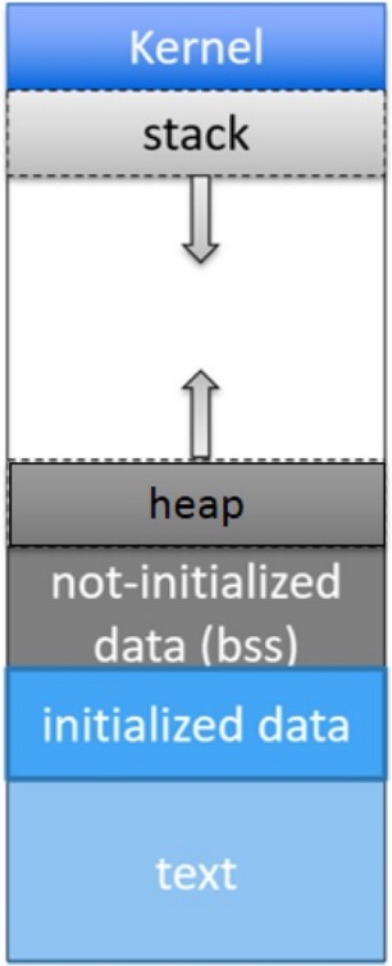
BREAK

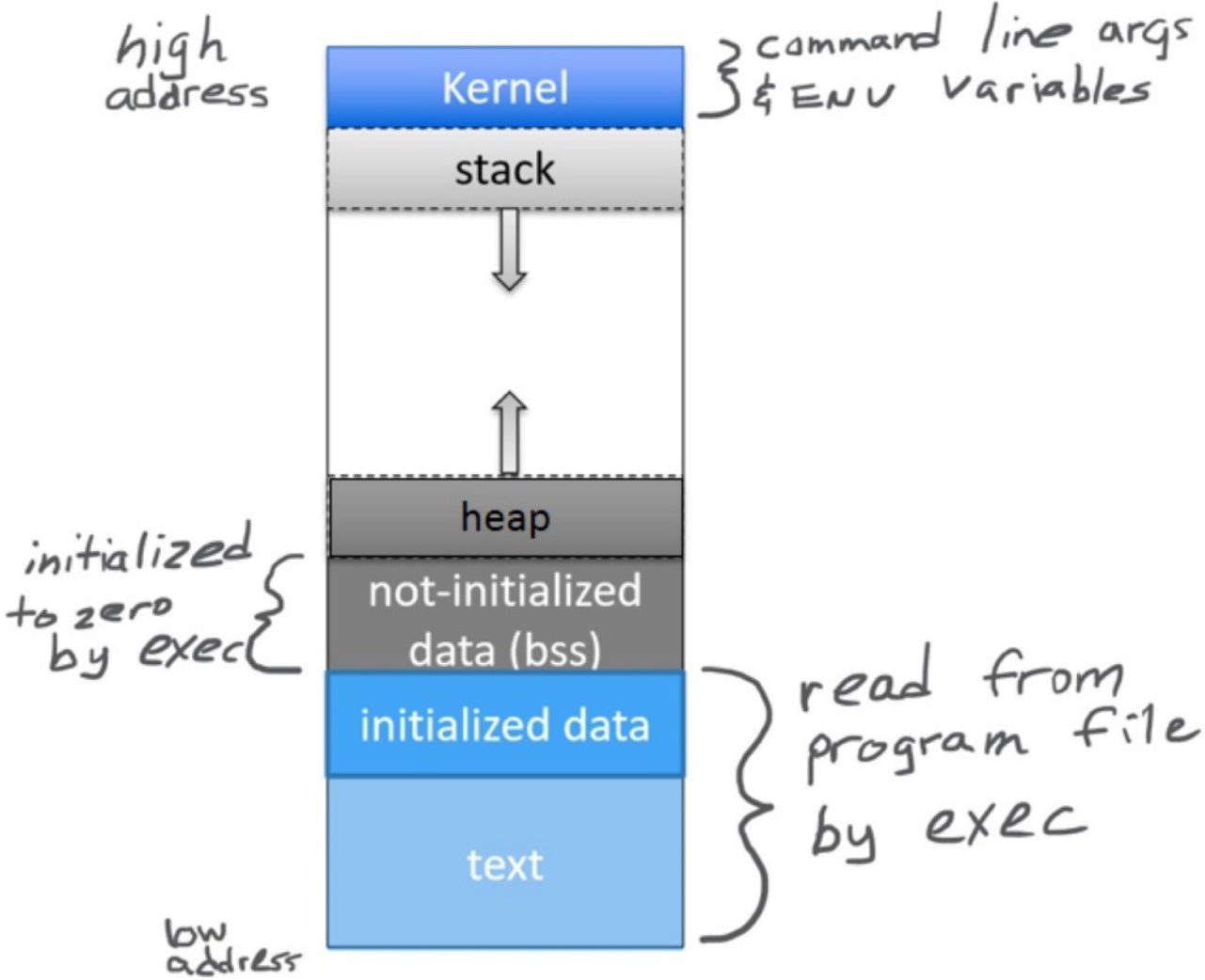
10 MINUTES

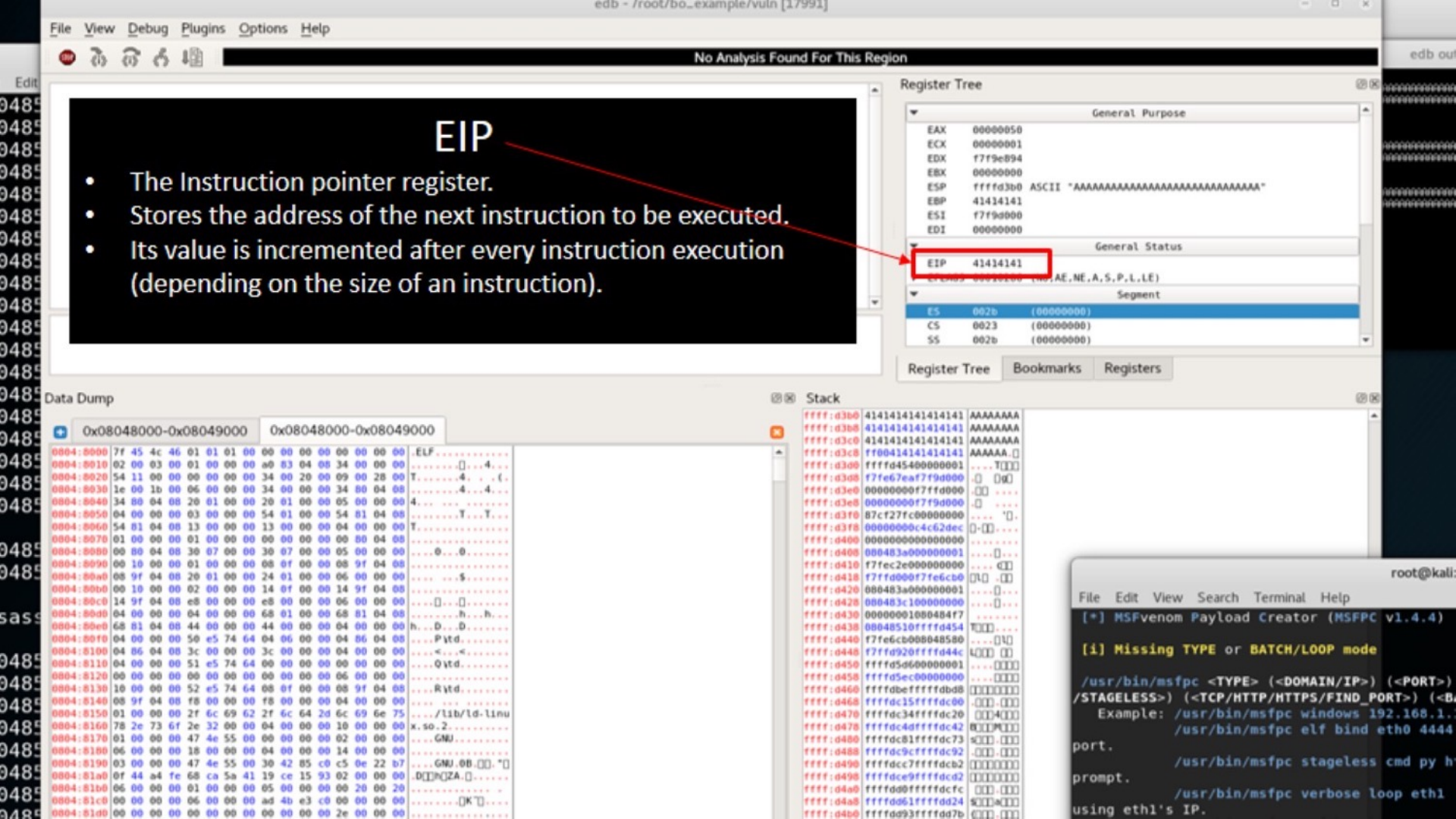
Don’t forget about the resources at: https://h4cker.org

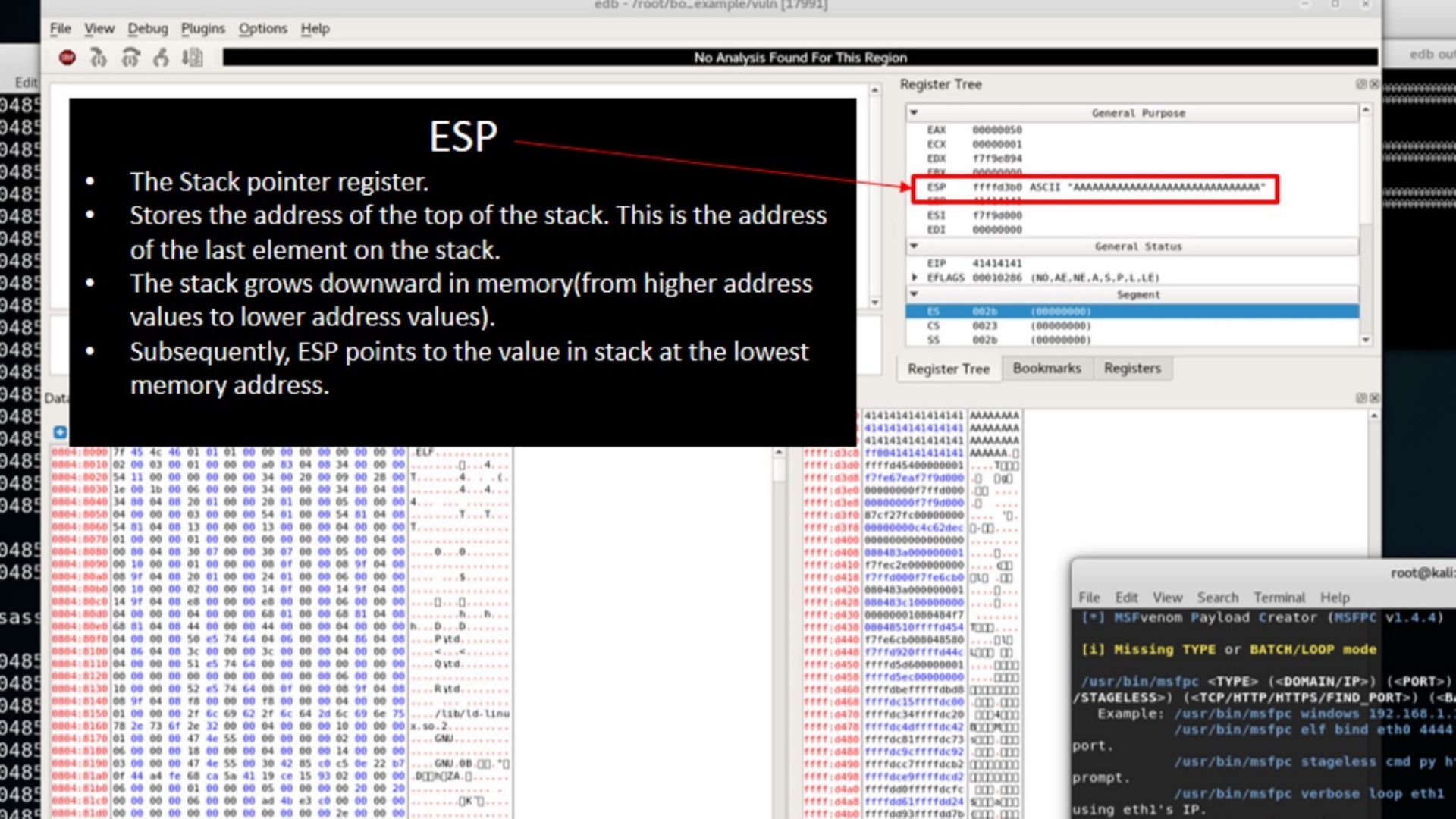
Introduction to Buffer Overflows

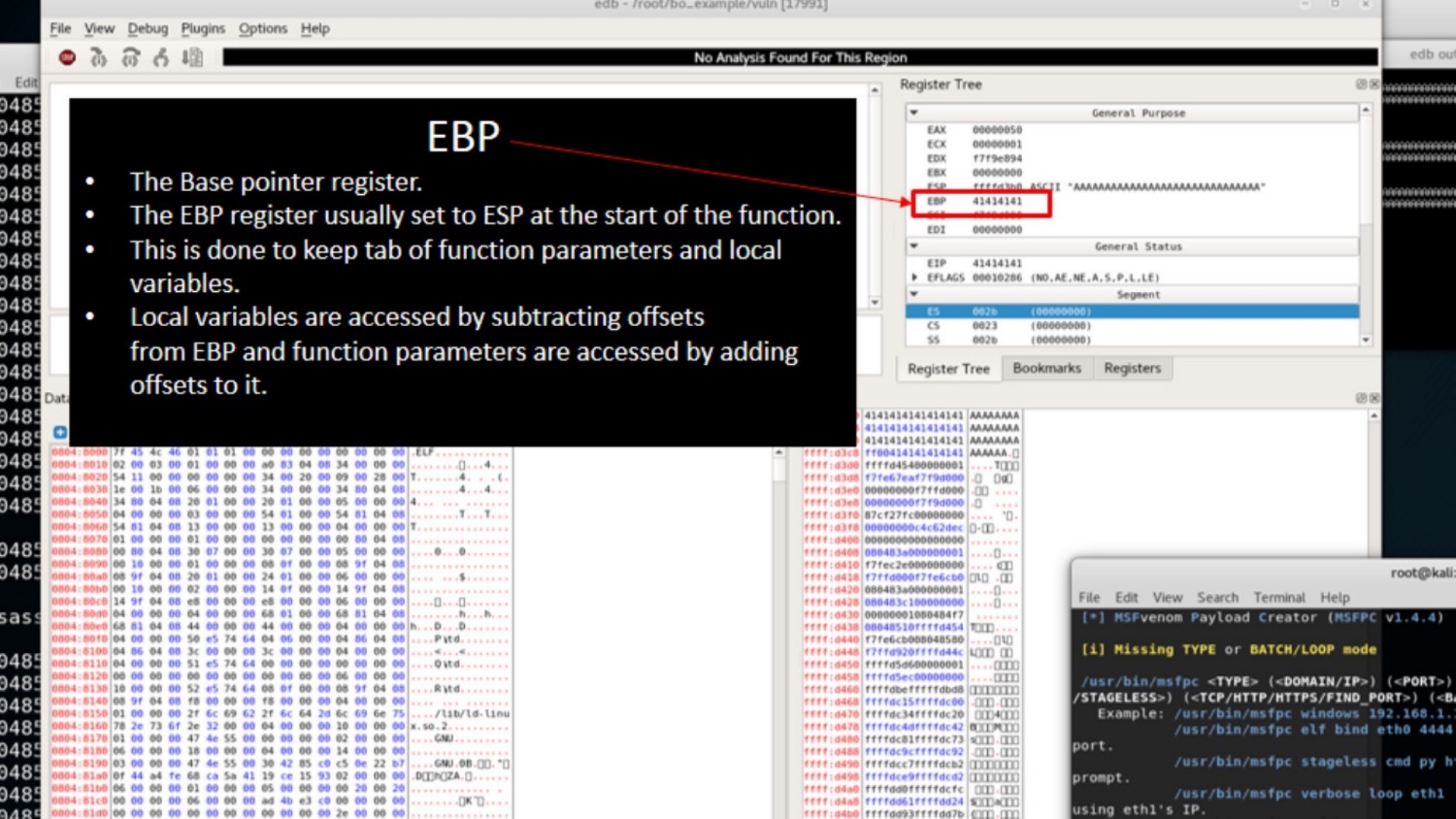

H E L L O
H E L L O W O R L D


Kernel
stack
heap
not-initialized
data (bss)
initialized data
text


high address
Kernel
command line args & ENV variables
stack
heap
initialized to zero by exec
not-initialized data (bss)
initialized data
text
read from program file by exec
low address

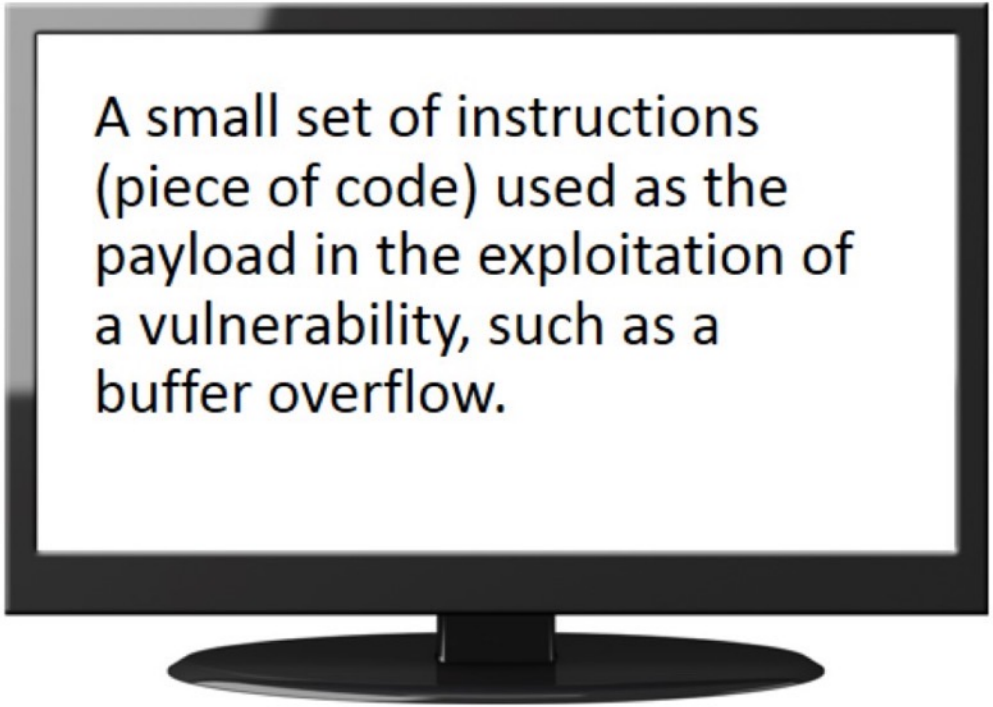

EIP
The Instruction pointer register.
Stores the address of the next instruction to be executed.
Its value is incremented after every instruction execution (depending on the size of an instruction).
File View Debug Plugins Options Help
No Analysis Found For This Region
Register Tree
General Purpose
EAX 00000050
ECX 00000001
EDX f7f9eb94
EBX 00000000
ESP ffffd3b0 ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAA"
EBP 41414141
ESI f7f9d000
EDI 00000000
General Status
EIP 41414141
Segment
ES 002b (00000000)
CS 0023 (00000000)
SS 002b (00000000)
Register Tree
Bookmarks
Registers
Data Dump
0x08048000-0x08049000
0x08048000-0x08049000
Stack
root@kali
File Edit View Search Terminal Help
[*] MSFvenom Payload Creator (MSFPC v1.4.4)
[i] Missing TYPE or BATCH/LOOP mode
/usr/bin/msfpc <TYPE> (<DOMAIN/IP>) (<PORT>)
/STAGELESS>) (<TCP/HTTP/HTTPS/FIND_PORT>) (<BA
Example: /usr/bin/msfpc windows 192.168.1.
/usr/bin/msfpc elf bind eth0 4444
port.
/usr/bin/msfpc stageless cmd py h
prompt.
/usr/bin/msfpc verbose loop eth1
using eth1's IP.

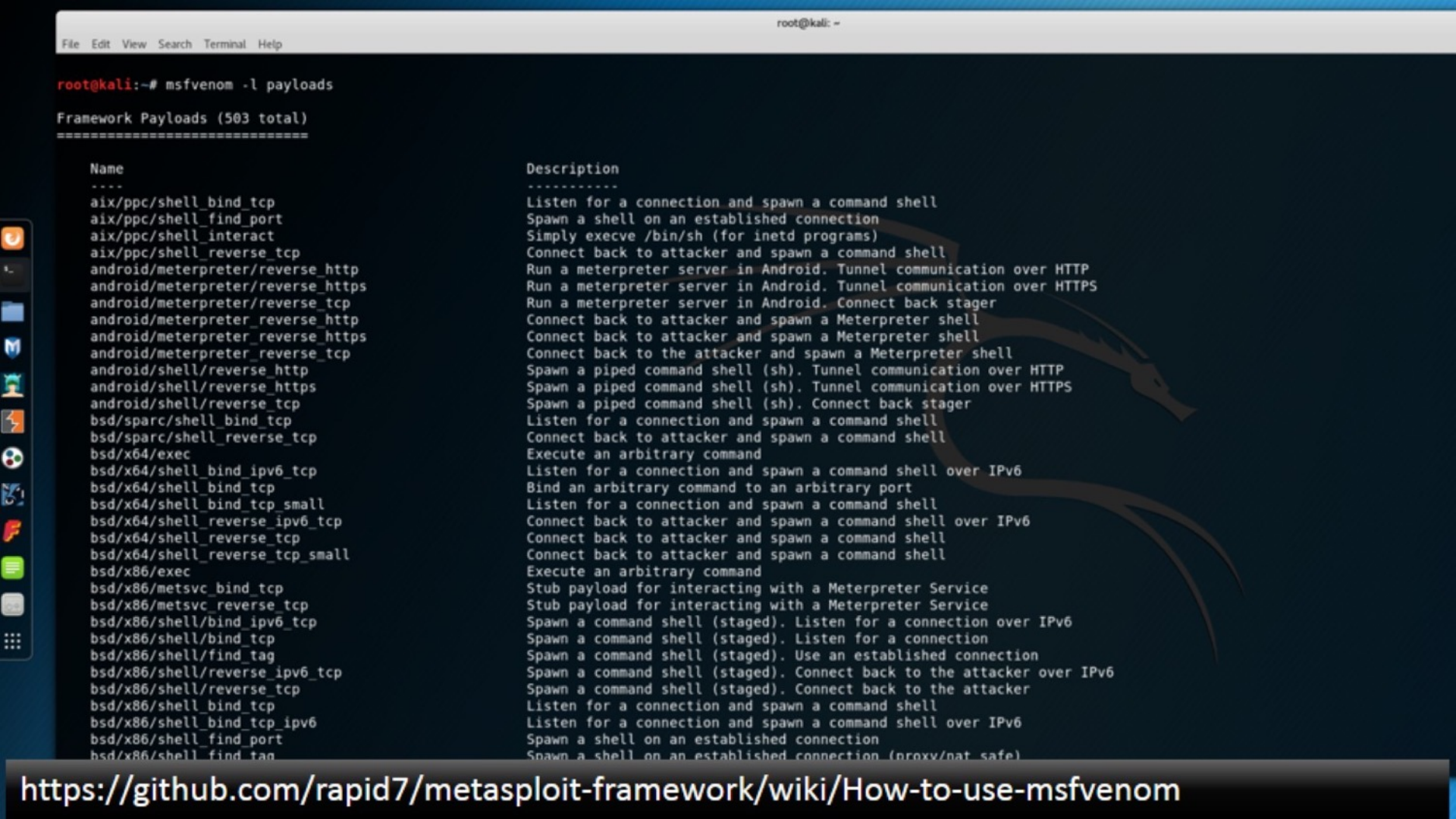

ESP
The Stack pointer register.
Stores the address of the top of the stack. This is the address of the last element on the stack.
The stack grows downward in memory(from higher address values to lower address values).
Subsequently, ESP points to the value in stack at the lowest memory address.
File View Debug Plugins Options Help
No Analysis Found For This Region
Register Tree
General Purpose
EAX 00000050
ECX 00000001
EDX f7f9eb94
ESP ffffd3b0 ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAA"
ESI f7f9d000
EDI 00000000
General Status
EIP 41414141
EFLAGS 00010286 (NO,AE,NE,A,S,P,L,LE)
Segment
ES 002b (00000000)
CS 0023 (00000000)
SS 002b (00000000)
Register Tree
Bookmarks
Registers
root@kali
File Edit View Search Terminal Help
[*] MSFvenom Payload Creator (MSFPC v1.4.4)
[i] Missing TYPE or BATCH/LOOP mode
/usr/bin/msfpc <TYPE> (<DOMAIN/IP>) (<PORT>)
Example: /usr/bin/msfpc windows 192.168.1.
/usr/bin/msfpc elf bind eth0 4444
/usr/bin/msfpc stageless cmd py h
/usr/bin/msfpc verbose loop eth1

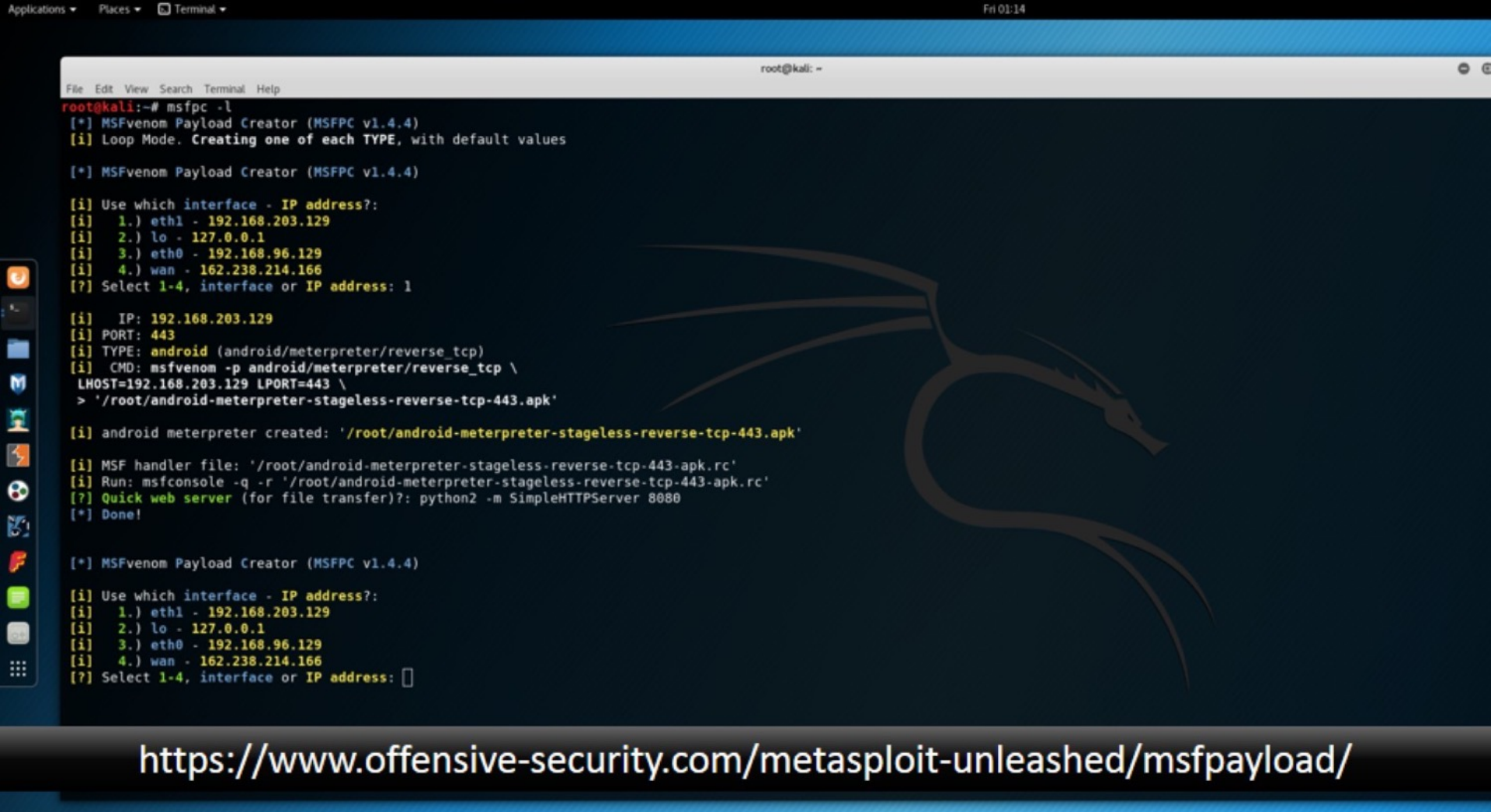

EBP
The Base pointer register.
The EBP register usually set to ESP at the start of the function.
This is done to keep tab of function parameters and local variables.
Local variables are accessed by subtracting offsets from EBP and function parameters are accessed by adding offsets to it.
File View Debug Plugins Options Help
No Analysis Found For This Region
Register Tree
General Purpose
EAX 00000050
ECX 00000001
EDX f7f9eb94
EBX 00000000
EBP 41414141
EDI 00000000
General Status
EIP 41414141
EFLAGS 00010286 (NO,AE,NE,A,S,P,L,LE)
Segment
ES 002b (00000000)
CS 0023 (00000000)
SS 002b (00000000)
Register Tree
Bookmarks
Registers
root@kali
File Edit View Search Terminal Help
[*] MSFvenom Payload Creator (MSFPC v1.4.4)
[i] Missing TYPE or BATCH/LOOP mode
/usr/bin/msfpc <TYPE> (<DOMAIN/IP>) (<PORT>)
Example: /usr/bin/msfpc windows 192.168.1.
/usr/bin/msfpc elf bind eth0 4444
/usr/bin/msfpc stageless cmd py h
/usr/bin/msfpc verbose loop eth1
using eth1's IP.


CINEMA
IN CINEMA TODAY
OMAR'S BUFFER
OVERFLOW DEMO

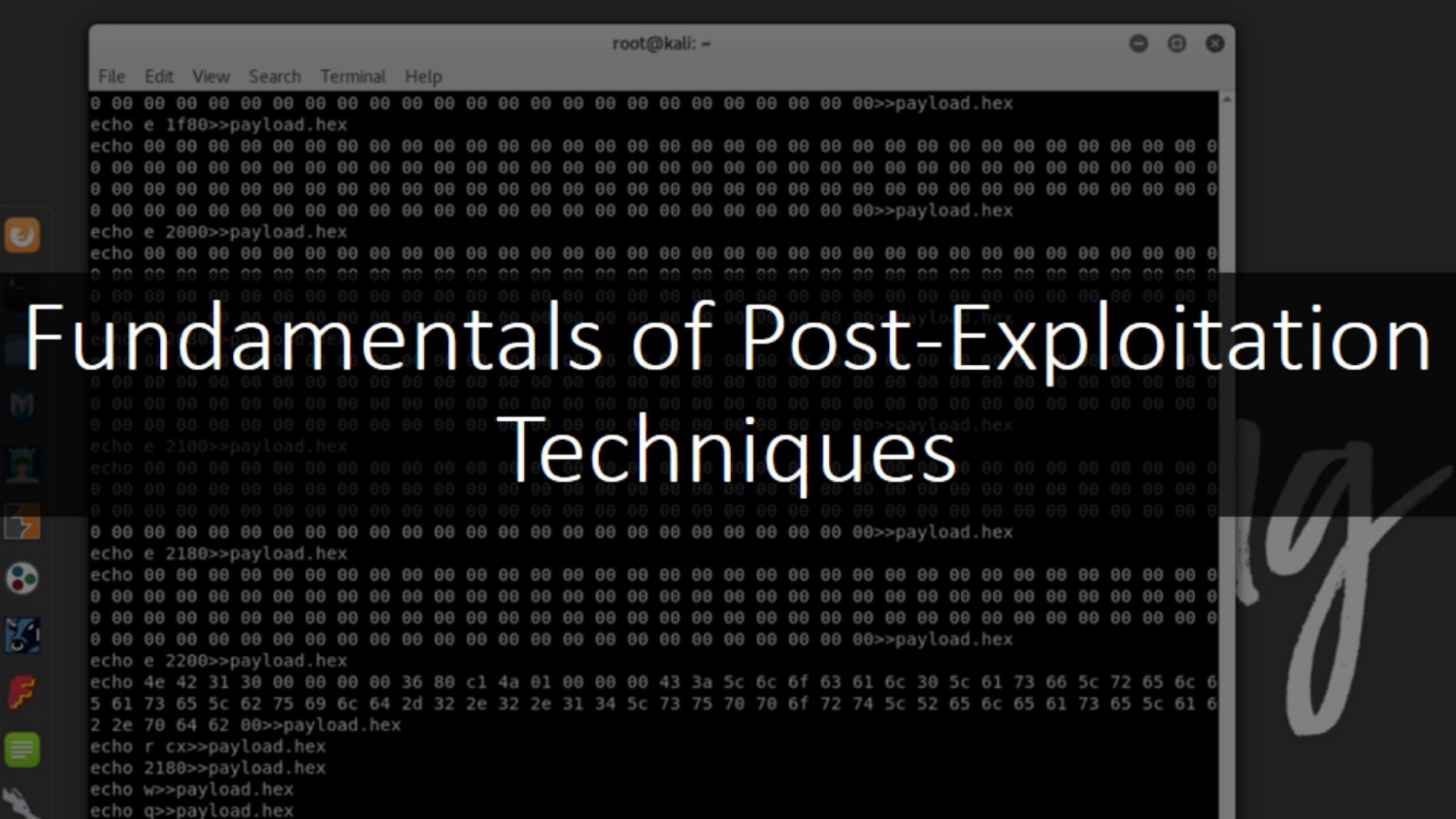
WHAT IS SHELLCODE?




root@kali: ~
File Edit View Search Terminal Help
root@kali:~# msfvenom -l payloads
Framework Payloads (503 total)
Name Description
aix/ppc/shell_bind_tcp Listen for a connection and spawn a command shell
aix/ppc/shell_find_port Spawn a shell on an established connection
aix/ppc/shell_interact Simply execve /bin/sh (for inetd programs)
aix/ppc/shell_reverse_tcp Connect back to attacker and spawn a command shell
android/meterpreter/reverse_http Run a meterpreter server in Android. Tunnel communication over HTTP
android/meterpreter/reverse_https Run a meterpreter server in Android. Tunnel communication over HTTPS
android/meterpreter/reverse_tcp Run a meterpreter server in Android. Connect back stager
android/meterpreter_reverse_http Connect back to attacker and spawn a Meterpreter shell
android/meterpreter_reverse_https Connect back to attacker and spawn a Meterpreter shell
android/meterpreter_reverse_tcp Connect back to the attacker and spawn a Meterpreter shell
android/shell/reverse_http Spawn a piped command shell (sh). Tunnel communication over HTTP
android/shell/reverse_https Spawn a piped command shell (sh). Tunnel communication over HTTPS
android/shell/reverse_tcp Spawn a piped command shell (sh). Connect back stager
bsd/sparc/shell_bind_tcp Listen for a connection and spawn a command shell
bsd/sparc/shell_reverse_tcp Connect back to attacker and spawn a command shell
bsd/x64/exec Execute an arbitrary command
bsd/x64/shell_bind_ipv6_tcp Listen for a connection and spawn a command shell over IPv6
bsd/x64/shell_bind_tcp Bind an arbitrary command to an arbitrary port
bsd/x64/shell_bind_tcp_small Listen for a connection and spawn a command shell
bsd/x64/shell_reverse_ipv6_tcp Connect back to attacker and spawn a command shell over IPv6
bsd/x64/shell_reverse_tcp Connect back to attacker and spawn a command shell
bsd/x64/shell_reverse_tcp_small Connect back to attacker and spawn a command shell
bsd/x86/exec Execute an arbitrary command
bsd/x86/metsvc_bind_tcp Stub payload for interacting with a Meterpreter Service
bsd/x86/metsvc_reverse_tcp Stub payload for interacting with a Meterpreter Service
bsd/x86/shell/bind_ipv6_tcp Spawn a command shell (staged). Listen for a connection over IPv6
bsd/x86/shell/bind_tcp Spawn a command shell (staged). Listen for a connection
bsd/x86/shell/find_tag Spawn a command shell (staged). Use an established connection
bsd/x86/shell/reverse_ipv6_tcp Spawn a command shell (staged). Connect back to the attacker over IPv6
bsd/x86/shell/reverse_tcp Spawn a command shell (staged). Connect back to the attacker
bsd/x86/shell_bind_tcp Listen for a connection and spawn a command shell
bsd/x86/shell_bind_tcp_ipv6 Listen for a connection and spawn a command shell over IPv6
bsd/x86/shell_find_port Spawn a shell on an established connection
https://github.com/rapid7/metasploit-framework/wiki/How-to-use-msfvenom



https://www.offensive-security.com/metasploit-unleashed/msfpayload/



Fundamentals of Post-Exploitation Techniques

Evasion Techniques

Encryption & Obfuscation Demo Using the Eternalblue Exploit

Pivoting

Whiteboard Explanation

Pivoting

Meterpreter Demo

Exfil

egressbuster & just a simple letmeout script...


Omar's
Demo & Whiteboard

Break

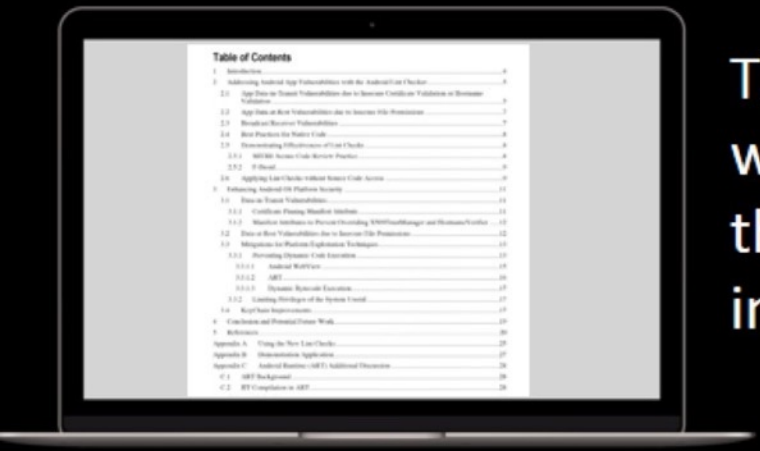
10 minutes

Introduction to Social Engineering


Omar's
Demo & Whiteboard

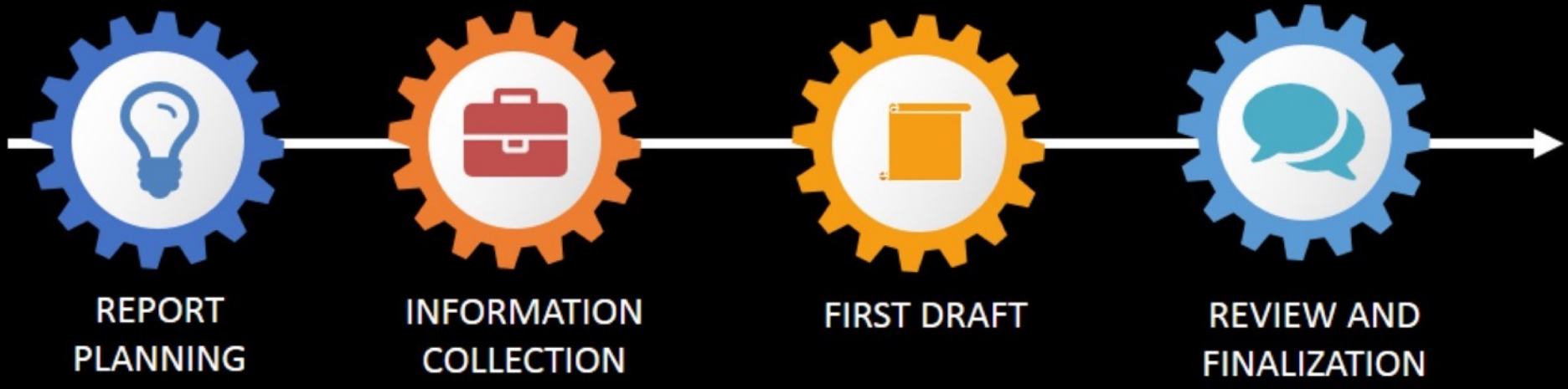
How to Write Penetration Testing Reports

The penetration testing report must be clear, detail the outcome of the tests, and in most cases include recommendations.



The audience will vary, executive summary will be read by the senior management and the technical details will be read by the IT or information security stakeholders.

Pen Testing Report Development Stages



Report Planning

Consider the target audiences

- Their need for the report (i.e. operational planning, resource allocation, approval),
- Position in the organization
- Knowledge of the report topic(i.e. purpose),
- Responsibility or authority to make decision based on the report, and
- Personal demographics (i.e. age, alliances, attitudes).

Report Planning

Report Classification

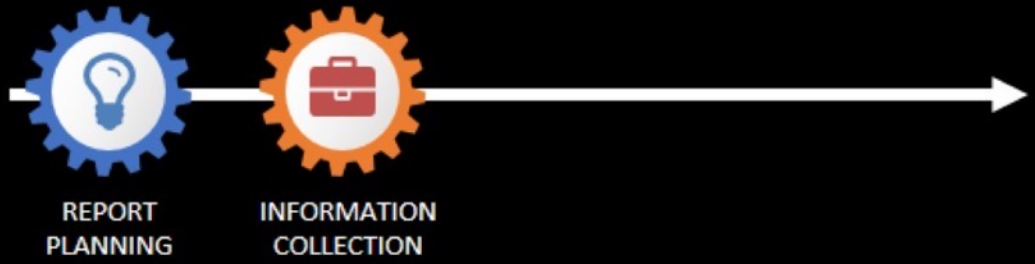
- Be aware of sensitive information
- The report classification should be based on the underlying organization’s information classification policy.

Report Planning

Report Distribution

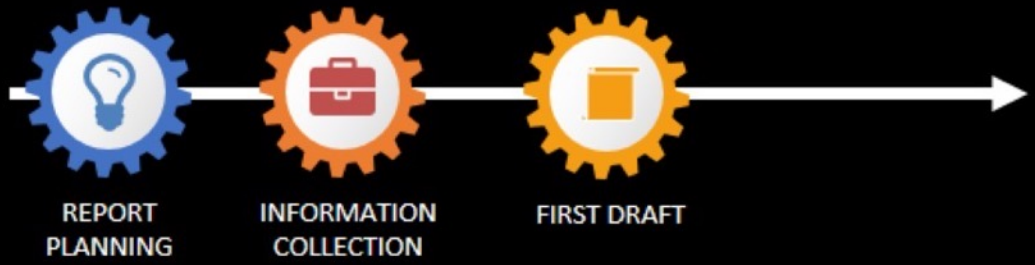
- The type of report delivery, recipients, number of copies and report distribution should be addressed in the scope of work.
- You should perform due diligence to ensure the confidentiality of the test results.

Information Collection



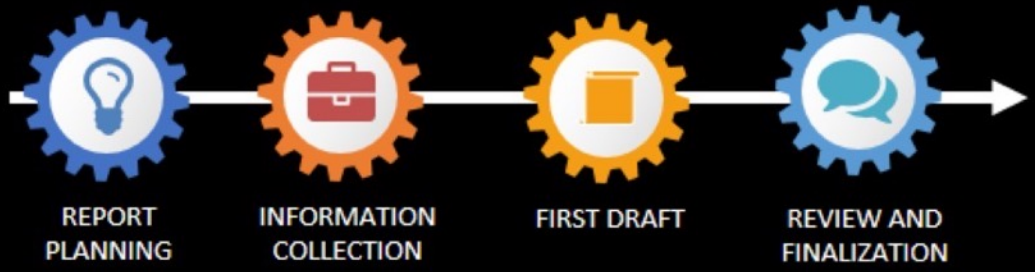
- Make sure that you collected all the information in all stages, system used and tools.
- Take notes, capture screenshots, log all activities, and keep all packet captures, scan reports, and any other results of your pen testing activities.

Report First Draft



Write a rough draft report using all relevant information gathered in the “information collection” stage.

Review and Finalization



- Peer review is very important!
- If you are a one-man pen testing shop, make sure to hire someone to proof read your report.

RISK RATINGS

The most common risk rating for vulnerability assessment is the Common Vulnerability Scoring System (CVSS).

https://first.org/cvss

RISK RATINGS

The standard risk model:

Risk = Likelihood * Impact

OWASP has a great resource that describes a risk rating methodology:

https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology


Omar's
Demo & Whiteboard


THE ART OF
Hacking
THEARTOFHACKING.ORG
~ 325 ~

Thank you!

Don’t forget about the additional resources at:

https://h4cker.org/resources