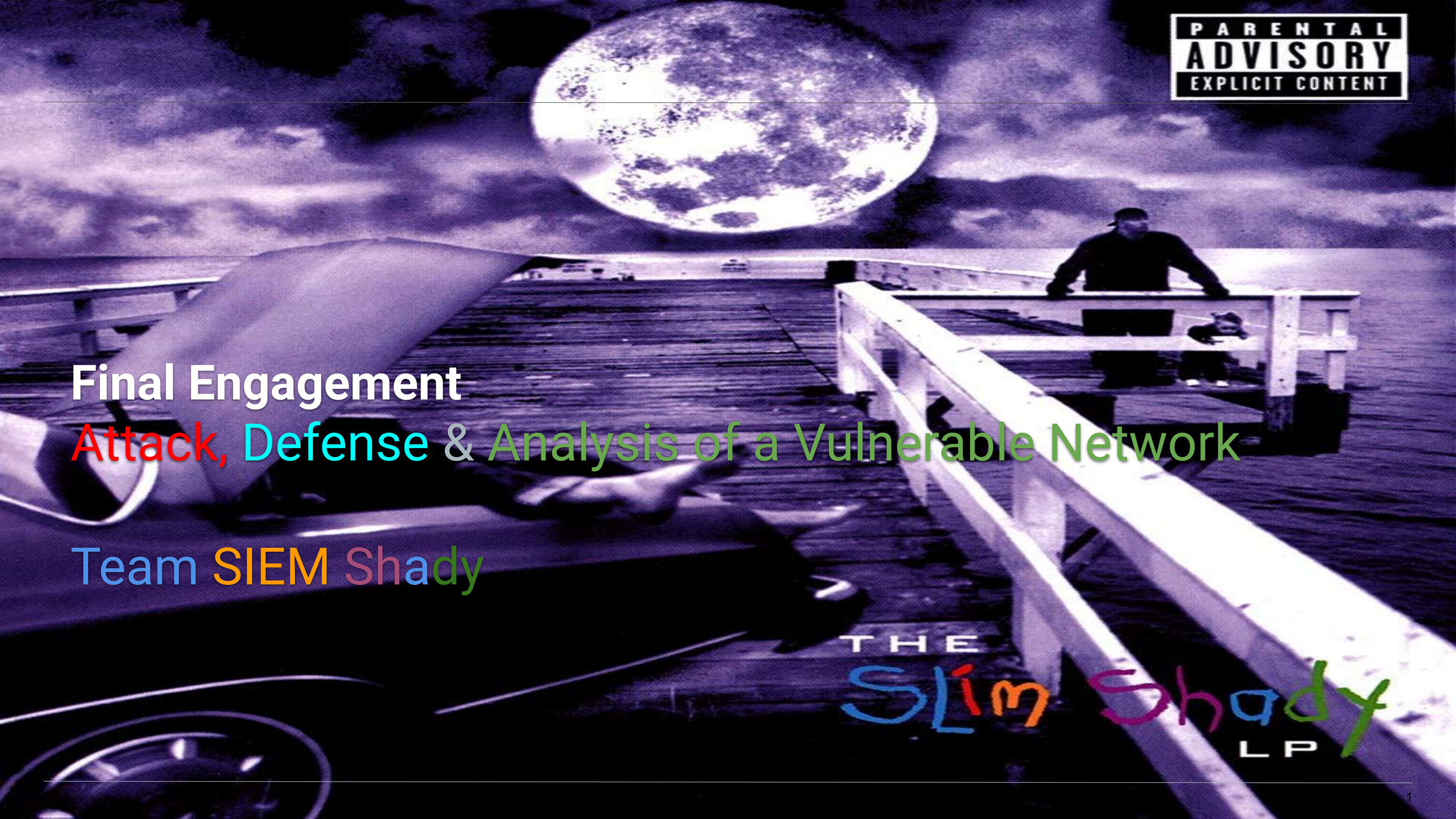


Final Engagement

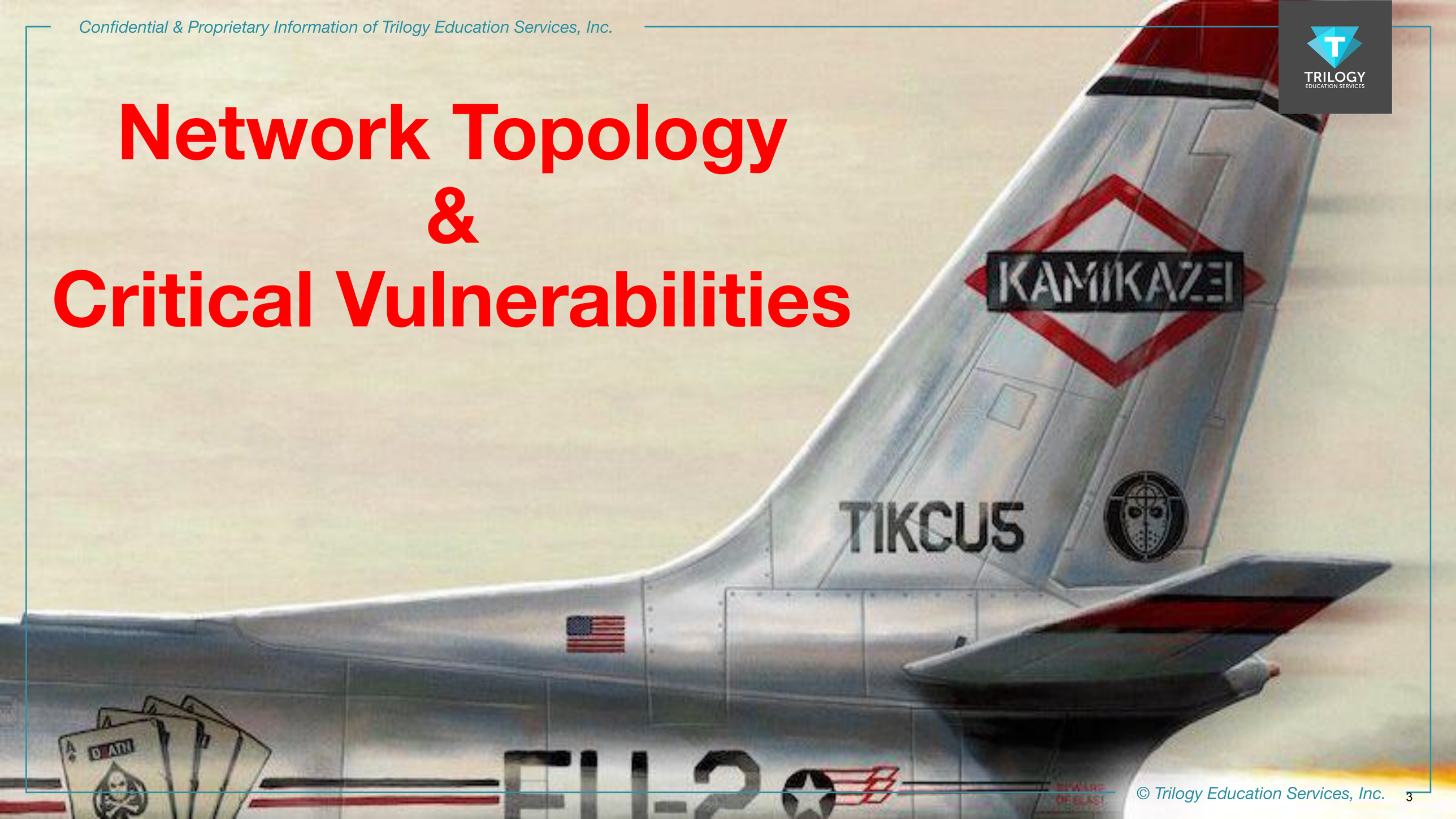
Attack, Defense & Analysis of a Vulnerable Network

Team SIEM Shady

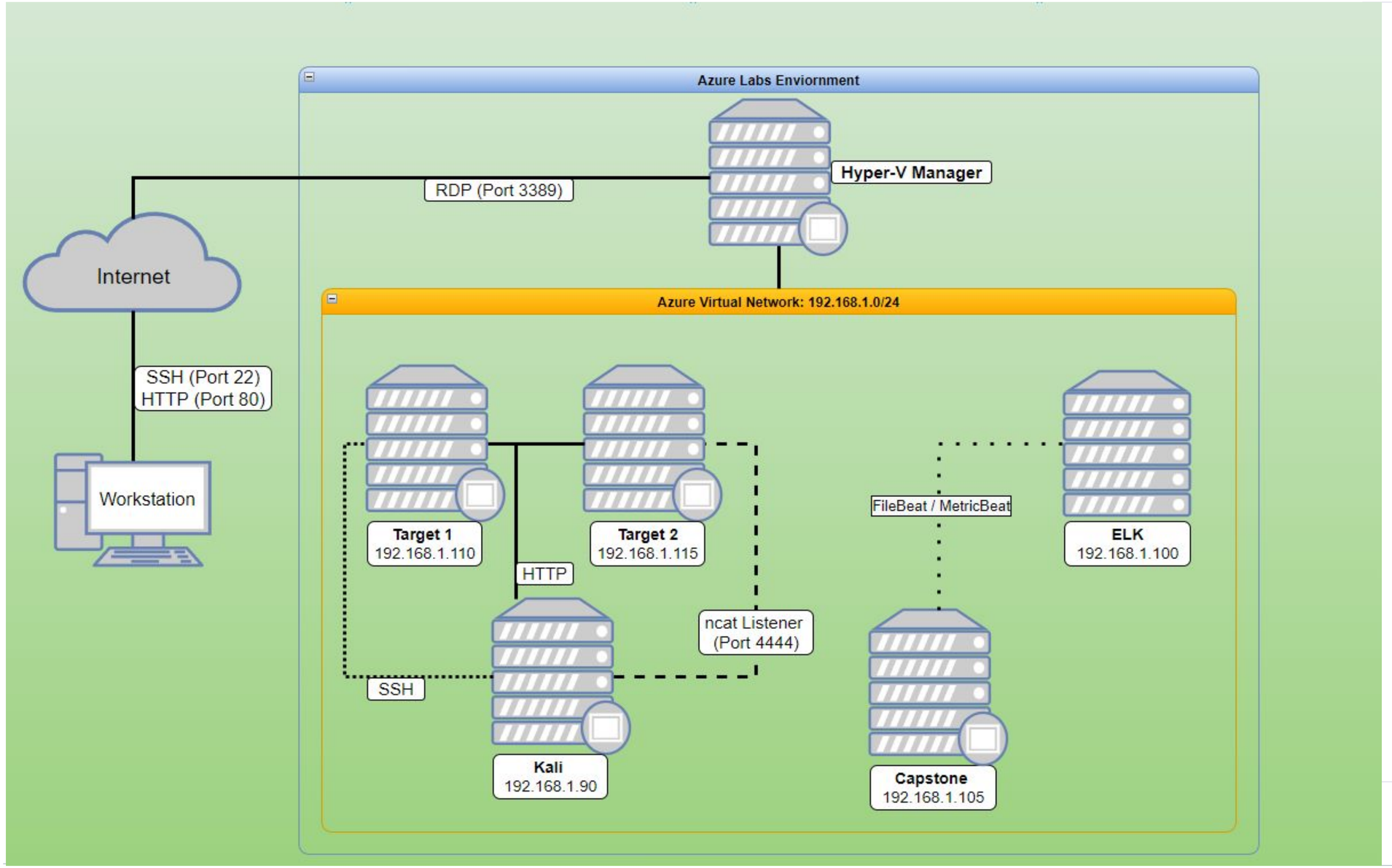
Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities



Network Topology



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.225.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.1
OS: Windows 10
Hostname: ML-REFVM-684427

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Weak Passwords	Michael’s password is michael	Was able to gain access to protected web pages
Vulnerable WordPress	Not updated, made it easy to enumerate users	Was able to find the user Michael
Unsalted Passwords	Made it easy for John to rip then R.I.P	Made it easy to gather their passwords from common word lists

Exploits Used

Exploitation: Weak Password

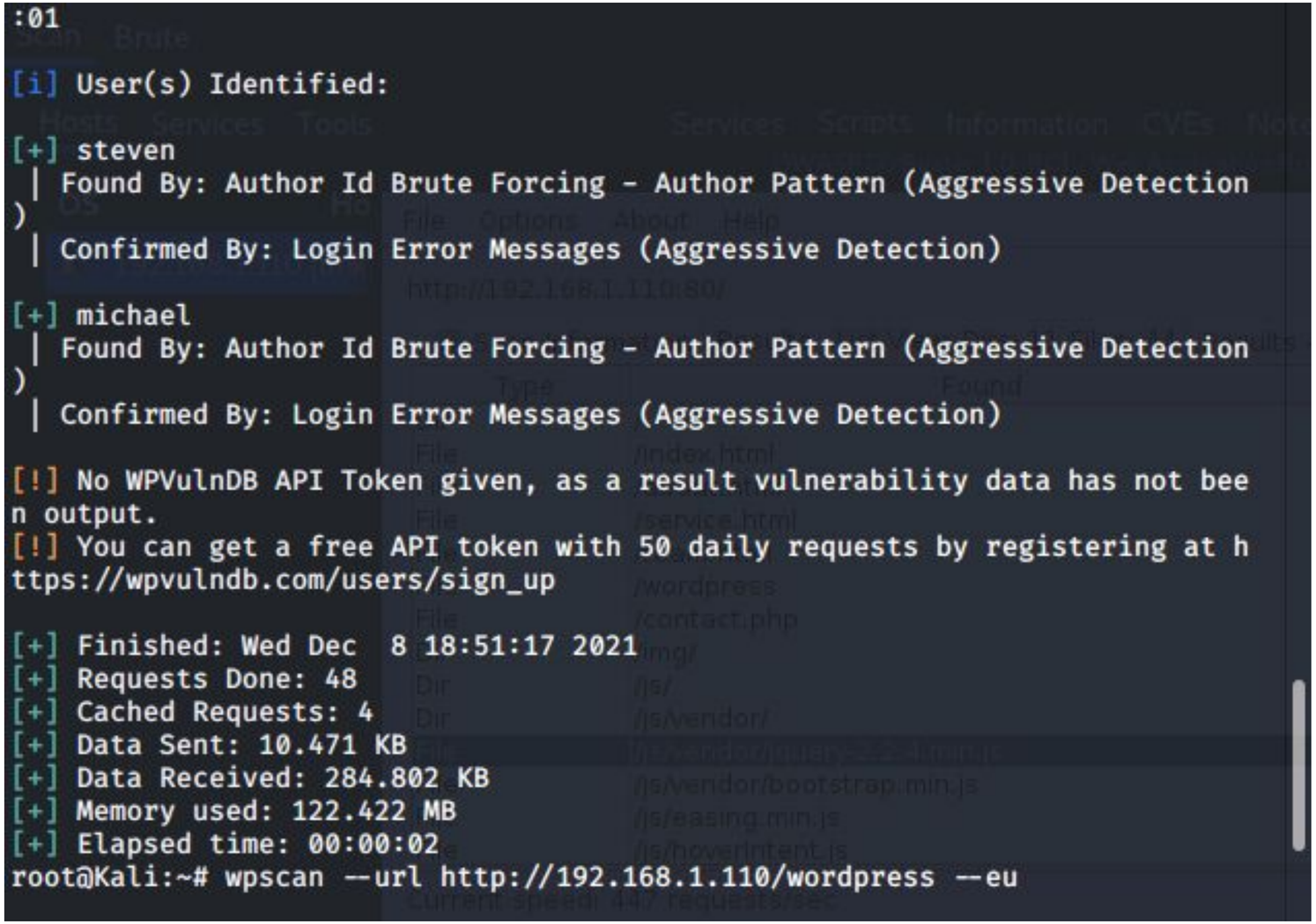
Summarize the following:

- Troubleshooting the password with common password malpractices.
  - michael:michael **(Very Weak!)**
- This enables the attacker to SSH into 192.168.1.110, giving access to **Target 1.**

Exploitation: Vulnerable WordPress

- With an outdated WordPress, attackers are able to enumerate all the users.
- The Cmd to enumerate

  `wpscan --url 192.168.1.110/wordpress -eu`
- After enumeration, attacker is able to locate the user Michael.



Exploitation: Unsalted Passwords

- John the Ripper is able to crack unsalted passwords; using a common wordlist made it easy for John to find the password.
- After John found the passwords we were able to gain access to Steven with the password **pink84**

```
root@Kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt wp_hashes
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (?)
```

Avoiding Detection

Stealth Exploitation of Vulnerable Wordpress

Monitoring Overview

- Which alerts detect this exploit?

➔ WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes

- Which metrics do they measure?

➔ http.response.status_code (Excessive HTTP Errors)

- Which thresholds do they fire at?

➔ Above 400

Mitigating Detection

- How can you execute the same exploit without triggering the alert?

➔ Spread out the time of the attack, 50 attempts in 1 minute.

Stealth Exploitation of Local File Inclusion (LFI)

Monitoring Overview

- Which alerts detect this exploit?

  ➔ WHEN sum()OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 30 seconds

- Which metrics do they measure?

  ➔ http.request.bytes

- Which thresholds do they fire at?

  ➔ Above 3500

Mitigating Detection

- How can you execute the same exploit without triggering the alert?

  ➔ Limit size of gathering less than 3500 bytes every 30 seconds

Stealth Exploitation of Directory Exploration

Monitoring Overview

- Which alerts detect this exploit?
  - ➔ WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - ➔ system.process.cpu.total.pct (Total CPU Usage)
- Which thresholds do they fire at?
  - ➔ 0.5 (50%)

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - ➔ Utilizing Google Dorking to find “invisible” directories and/or text documents that can provide information without setting off any alarms.


Fin