Final Engagement

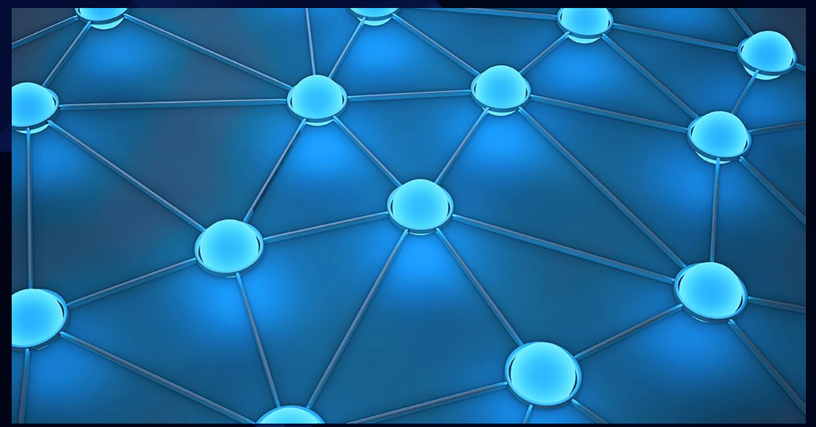
Attack, Defense & Analysis of a Vulnerable Network

Kellen Piro, Nicholas Williams, Richard Nguyen, Leo Martinez, Lucas Busche

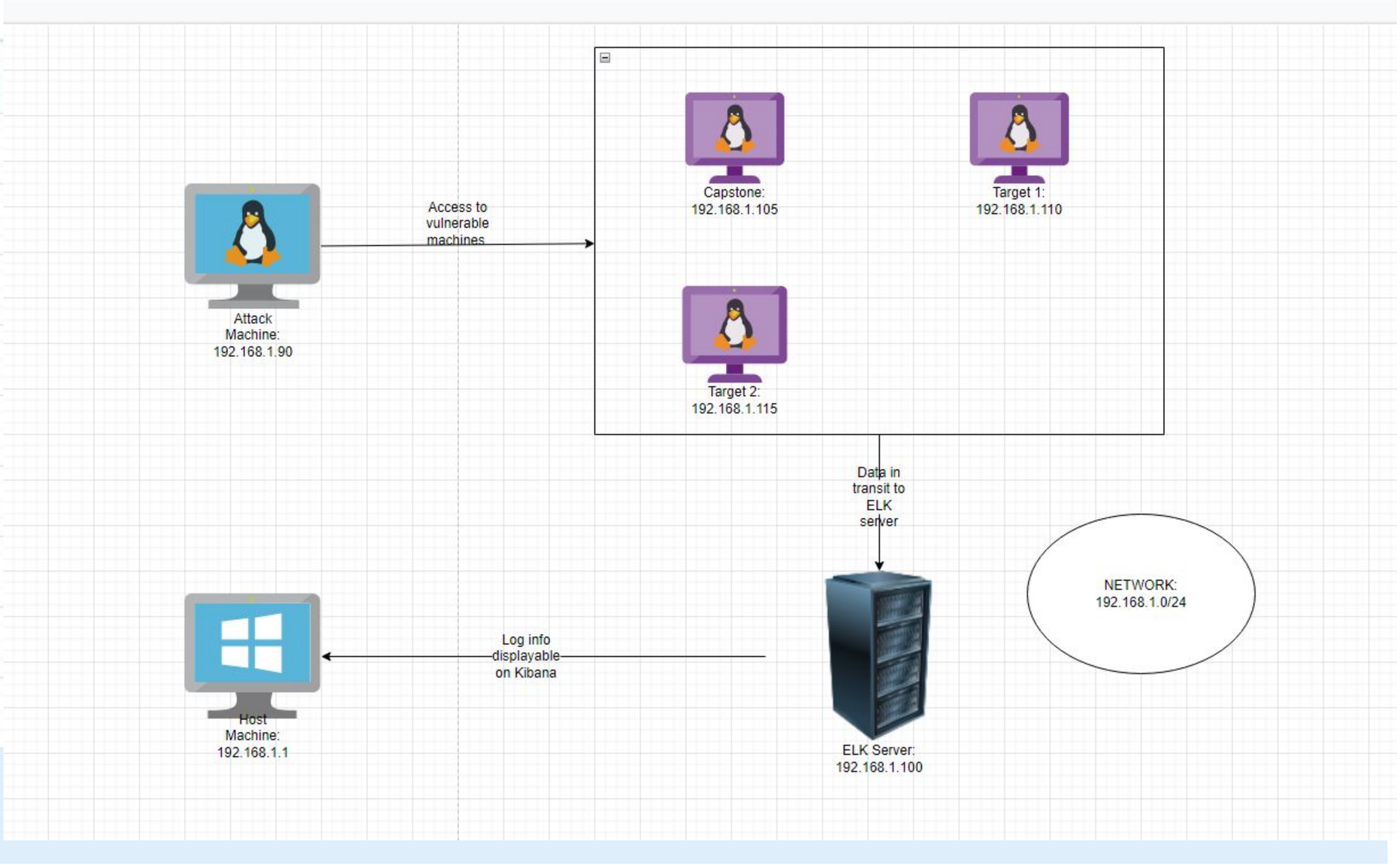
Table of Contents

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:Windows
Hostname:Capstone

IPv4:192.168.1.110
OS:Linux
Hostname:Target 1

IPv4:192.168.1.115
OS:Linux
Hostname:Target 2

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
WordPress Enumeration	wp-scan enumeration will scan for exploitable vulnerabilities in the WP site allowing for enumeration of weakly secured databases.	Using a linux based terminal, an attacker can generate a list of vulnerabilities as well as enumerate a list of users on a system. The attacker was able to gather critical login information during this step.
Open and vulnerable SSH access	Having a poorly secured network and allowing open ports, drastically increases the number of potential entry vectors into a network.	Attacker successfully gained shell access to our targeted machine.
Weak login credentials	Poor/predictable login credentials create a great amount of risk concerning unauthorized network access.	The attacker was able to simply guess the login password for a critical network user. This allowed for further lateral movement in our network.

Critical Vulnerabilities: Target 1 (continued)

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

Vulnerability	Description	Impact
Weakly stored credentials (plaintext)	Critical user data such as admin login credentials, was stored in plaintext (no encryption or additional security measures) on a system database.	Attacker was able to locate the credentials in the Raven security file
Exposed password hashes	The hashed passwords stored on the system can be ran against dedicated lists of known/pre-hashed inputs (Rainbow Table).	Using a tool such as john, the attacker was successful in cracking the hashes, thereby gaining a raw/plaintext passwords.
User Privilege Management allowing a Python script exploit	Using Python, it is possible to spawn a TTY shell. Once this pseudo-terminal is established, the module can make it seem like certain commands are being sent from a legitimate source.	This exploit allowed the attacker to setup a shell on our network and escalate their privilege to a root user.

Alerts Implemented

HTTP Request Size Monitor

Summarize the following:

- This alert will monitor the network for HTTP request byte size
- The alert will fire once it hits this threshold:
  - WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute

Time	metadata.name	result.condition.met	state
Mar 19, 2022 @ 15:03:42.329	HTTP Request Size Monitors	true	executed
Mar 17, 2022 @ 02:15:59.416	HTTP Request Size Monitors	true	executed
Mar 17, 2022 @ 02:12:59.314	HTTP Request Size Monitors	true	executed
Mar 17, 2022 @ 02:10:59.239	HTTP Request Size Monitors	true	executed
Mar 17, 2022 @ 02:08:59.176	HTTP Request Size Monitors	true	executed
Mar 19, 2022 @ 15:04:42.390	HTTP Request Size Monitors	false	execution_not_needed
Mar 19, 2022 @ 15:03:42.329	HTTP Request Size Monitors	true	executed
Mar 17, 2022 @ 03:29:59.551	HTTP Request Size Monitors	false	execution_not_needed
Mar 17, 2022 @ 03:28:59.518	HTTP Request Size Monitors	false	execution_not_needed
Mar 17, 2022 @ 03:27:59.466	HTTP Request Size Monitors	false	execution_not_needed

CPU Usage Monitor

Summarize the following:

- This alert will monitor the Percentage CPU usage
- The alert will fire once it hits this threshold:
  - WHEN max() of system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes

Time	metadata.name	result.condition.met	state
Mar 19, 2022 @ 15:07:42.504	CPU Usage Monitor	true	executed
Mar 19, 2022 @ 15:06:42.468	CPU Usage Monitor	true	executed
Mar 19, 2022 @ 15:05:42.427	CPU Usage Monitor	true	executed
Mar 19, 2022 @ 15:04:42.390	CPU Usage Monitor	true	executed
Mar 19, 2022 @ 15:03:42.329	CPU Usage Monitor	true	executed
Mar 16, 2022 @ 23:38:58.170	CPU Usage Monitor	true	executed

Mar 19, 2022 @ 15:05:42.427	CPU Usage Monitor	true	executed
Mar 19, 2022 @ 15:04:42.390	CPU Usage Monitor	true	executed
Mar 19, 2022 @ 15:03:42.329	CPU Usage Monitor	true	executed
Mar 17, 2022 @ 03:29:59.551	CPU Usage Monitor	false	execution_not_needed
Mar 17, 2022 @ 03:28:59.518	CPU Usage Monitor	false	execution_not_needed

HTTP Errors

Summarize the following:

- This alert will monitor the network’s HTTP Response Codes
- The alert will fire once it hits this threshold:
  - WHEN count() GROUPED OVER top 5 ‘http.response.status_code’ IS ABOVE 400 FOR THE LAST 5 minutes

Time	metadata.name	result.condition.met	state
Mar 15, 2022 @ 02:54:25.422	HTTP Errors	true	executed
Mar 15, 2022 @ 02:53:25.386	HTTP Errors	true	executed
Mar 15, 2022 @ 02:52:25.356	HTTP Errors	true	executed
Mar 15, 2022 @ 02:51:25.323	HTTP Errors	true	executed

Mar 15, 2022 @ 02:46:25.647	HTTP Errors	false	execution_not_needed
Mar 15, 2022 @ 02:45:25.618	HTTP Errors	false	execution_not_needed
Mar 15, 2022 @ 02:44:25.588	HTTP Errors	false	execution_not_needed
Mar 15, 2022 @ 02:43:25.553	HTTP Errors	true	executed
Mar 15, 2022 @ 02:42:25.523	HTTP Errors	true	executed
Mar 15, 2022 @ 02:41:25.491	HTTP Errors	true	executed

Hardening

Hardening Against WordPress Enumeration on Target 1

Explain how to patch Target 1 against Vulnerability 1. Include:

- This patch makes it so that it is more difficult for wpscan to enumerate by detecting requests to the wp-config.php file and block the user if they are unauthorized.
- # Detect that someone is accessing those strange files
  - RewriteRule ^wp-config\.php\.save$ index.php?wp_config_enumeration=1 [L]
  - $transient_name = 'wce_block_'.$_SERVER['REMOTE_ADDR'];
  - $transient_value = get_transient($transient_name);
  - if ($transient_value !== false) {
  - die('BANNED!');}
  - if (isset($_GET['wp_config_enumeration'])) {
  - set_transient($transient_name, 1, DAY_IN_SECONDS);
  - die('BANNED!');

Hardening Against Open and Vulnerable SSH Port on Target 1

Explain how to patch Target 1 against Vulnerability 2. Include:

- This patch makes it so only a pre-made white list of IP addresses can SSH onto through the port.
- #First black listing all IP addresses.
- in /etc/hosts.deny add sshd:ALL
  - #Whitelisting certain IP addresses.
- in /etc/hosts.allow add sshd: ‘IP_Address(es)_to_Allow’

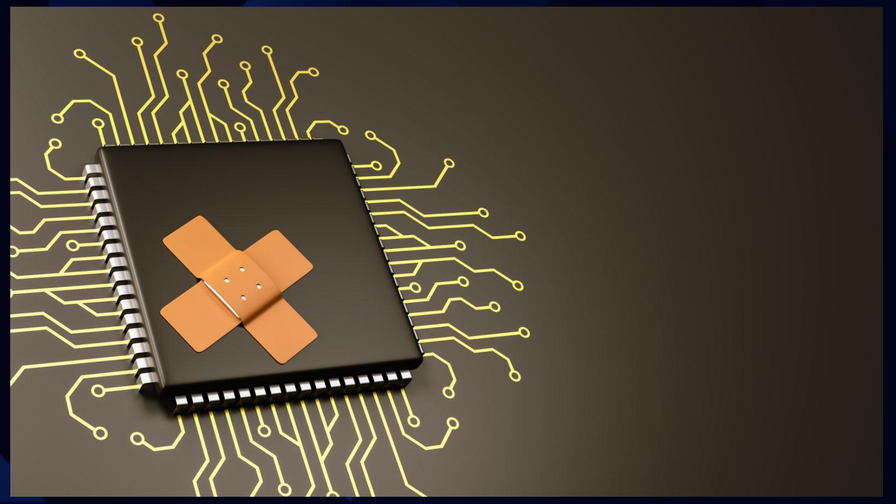
Hardening Against Weak Login Credentials on Target 1

Explain how to patch Target 1 against Vulnerability 3. Include:

- This patch forces users to create a complex password and forces them to change their password after a set amount of days.
- #Configuring password strength

- within the /etc/pam.d/passwd file to enable use of pam_quality
  password required pam_pwquality.so retry=3
- create a pwquality.conf within /etc/security with following lines:
  - (For minimum Length requirement):minlen = (amount#) ie. 8
  - (For restricting sequential characters): maxsequence = (amount#) ie. 3
  - (For restricting repeating characters): maxrepeat = (amount#) ie. 2
- #Configuring Password aging
  - change -M (amount_of_days)(username) ie. 60 michael

Implementing Patches



Implementing Patches with Ansible

Playbook Overview

```
- hosts: localhost
  connection: local
  tasks:
    - name: stop httpd
      systemd:
        name: httpd
        state: stopped
      become: true

    - name: backup html files
      archive:
        path: /var/www/html
        dest: "/home/centos/backups/wordpress-bck-{{ansible_date_time.iso8601_basic_short}}.tgz"
        format: gz
      become: true

    - name: backup wordpress database
      command: /etc/backup-wpdb.sh
      become: true

    - name: get latest wordpress
      unarchive:
        src:  https://wordpress.org/latest.zip
        dest: /tmp/
        remote_src: yes
      become: true

    - name: Wait until wordpress has been downloaded
      wait_for:
        path: /tmp/wordpress/index.php
        state: present

    - name: copy wordpress to website
      shell: /bin/cp -rf /tmp/wordpress/* /var/www/html/
      become: true

    - name: delete tmp wordpress
      file:
        path: /tmp/wordpress
        state: absent
      become: true

    - name: start httpd
      systemd:
        name: httpd
        state: started
        daemon_reload: yes
      become: true
```