Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

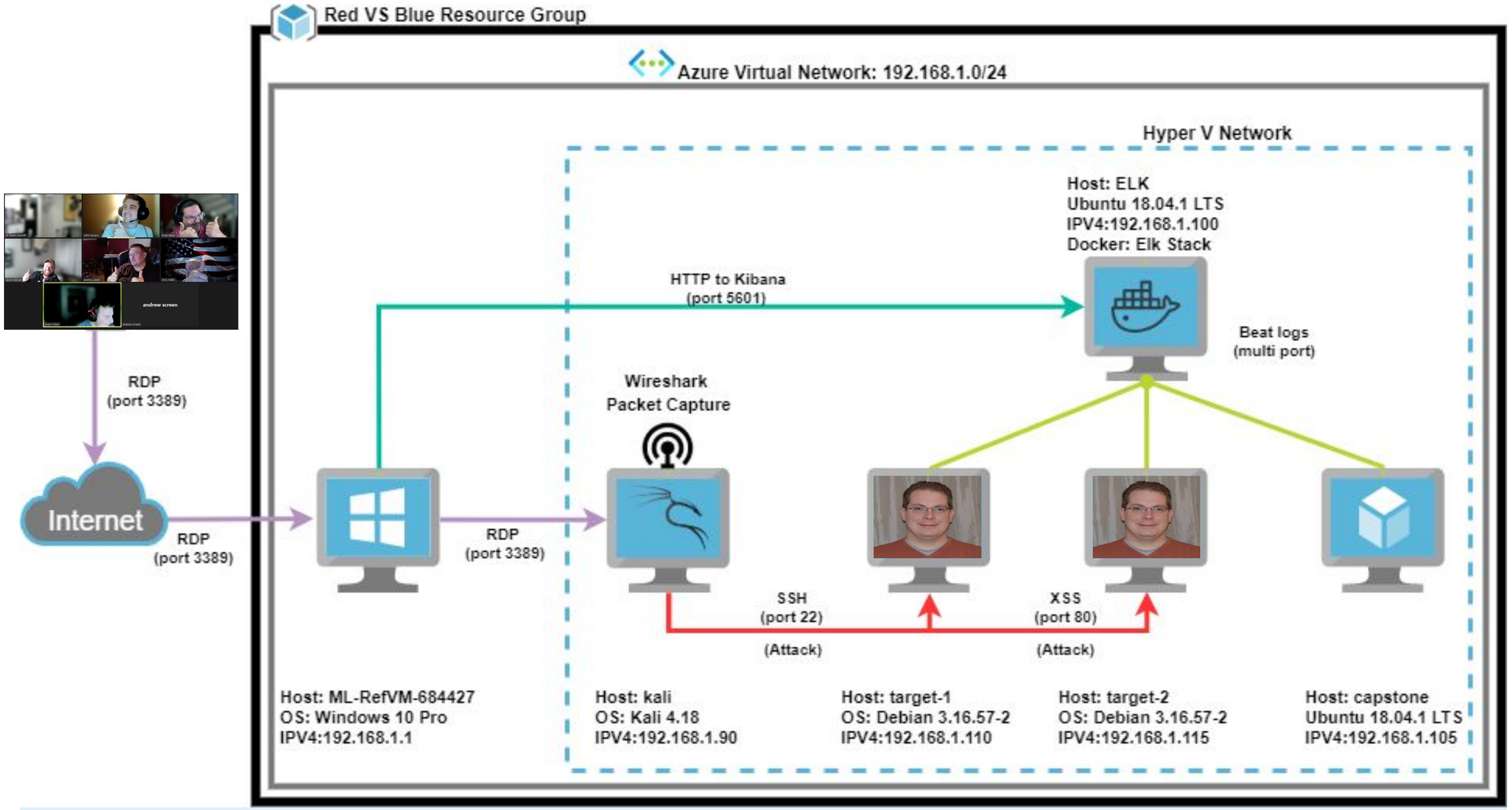
By: Andrew Anelli, Dean Baker, Myles Everett, Ostyn Fisher, Chris Miller, and John Sowers

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
IPv4:192.168.1.100
OS:Ubuntu Linux
Hostname:ELK

IPv4:192.168.1.105
OS:Ubuntu Linux
Hostname:Capstone

IPv4:192.168.1.110
OS:Debian Linux
Hostname:Target 1

IPV4: 192.168.1.115
OS:Debian Linux
Hostname:Target 2

IPv4:192.168.1.90
OS:Kali Linux
Hostname:Kali

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
User Enumeration	See user names in Wordpress site	Able to see list of users on the site.
Weak passwords	No password requirements for users	Able to brute force into accounts found in Wordpress
Unsalted User Password Hash	Unique value that can be added to the end of the password to create a different hash value.	Able to crack hashed passwords using John the RIpper
Privilege Escalation	Users able to get sudo access	Able to use python script to become a super user

Critical Vulnerabilities: Target 2

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

Vulnerability	Description	Impact
Unrestricted access to wordpress directories	Once on the system there were no restrictions to the files or directories.	Anyone who gained unauthorized or authorized access has unrestricted access to all the system and its directories.
Cross site scripting not disabled	XSS was not disabled and could be used to gain unauthorized access to information	User accounts can be hijacked, credentials could be stolen, and sensitive data could be exfiltrated.
Enumerating Users and Directories	Applications are able to scan for both user credentials and hidden/visible directories.	Information left out in the directories and combination of user info can lead to a breach.
Account Privileges	Account used to perform RCE (www-data) has read permissions on sensitive files.	Being able to read etc/passwd, etc/shadow compromises the integrity of the system.

Alerts Implemented

Excessive HTTP Errors

Summarize the following:

- Which **metric** does this alert monitor?

  -WHEN count()GROUPED OVER top 5 'http.response.status_code'

- What is the **threshold** it fires at?

  -IS ABOVE 400 FOR THE LAST 5 minutes

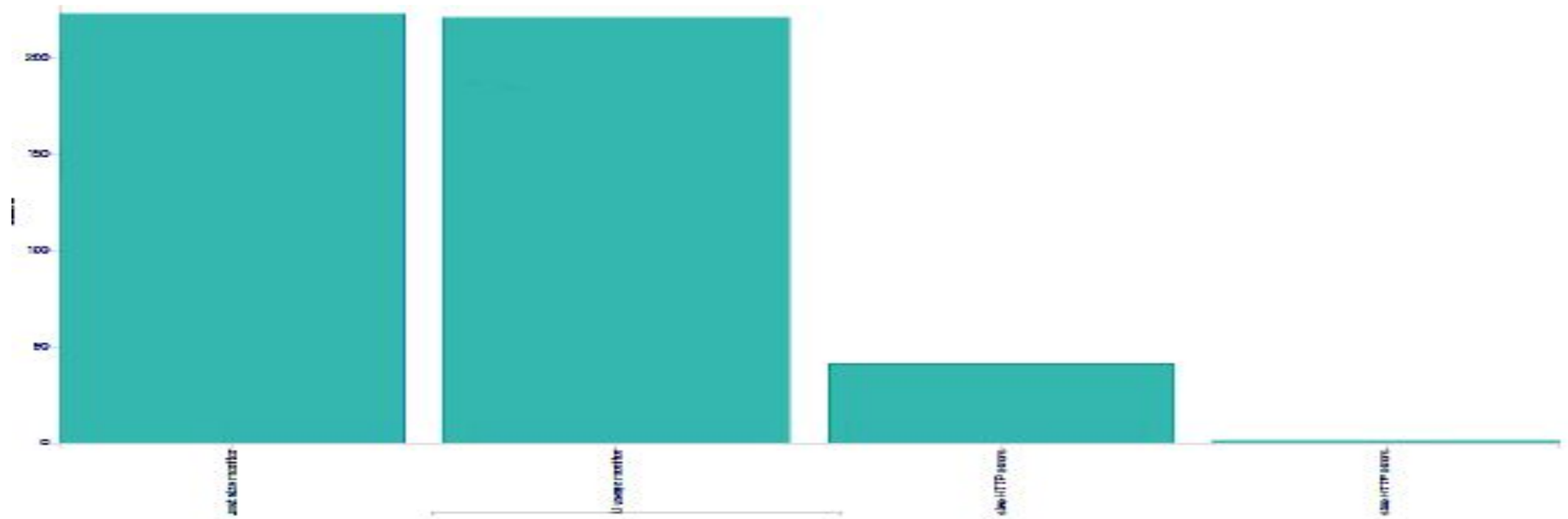
HTTP request size monitor

Summarize the following:

- Which **metric** does this alert monitor?

  -WHEN sum()OF http.request.bytes OVER all documents
- What is the **threshold** it fires at?

  -IS ABOVE 3500 FOR THE LAST 1 minute



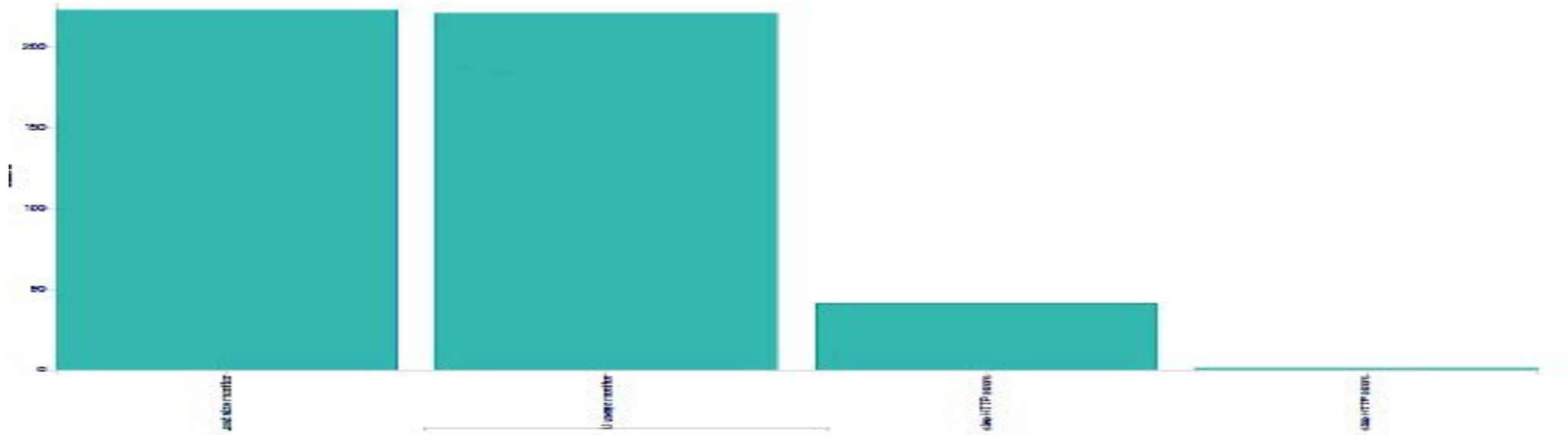
CPU usage monitor

Summarize the following:

- Which **metric** does this alert monitor?

  -WHEN max()OF system.process.cpu.total.pct OVER all documents
- What is the **threshold** it fires at?

  -IS ABOVE 0.5 FOR THE LAST 5 minutes



Hardening

Hardening Against User Enumeration on Target 1

- Wordpress has plugins to stop user enumeration that can be installed through wordpress itself
- To stop user enumeration in WordPress by just one click, use the free plugin WP Hardening. Install and activate the plugin.
- In the Wordpress WebGUI
- Go to the “**Security Fixers**” tab.
- Toggle the key next to “**Stop user enumeration**” and it’s done.

This plug in will fix the enumeration issue. No changes in the code is required. It is free and easy to install.

Hardening Against Weak User Password on Target 1

- Make sure that user passwords are at least 10 characters long.
- Passwords must contain a capitol, a number, and special character.
- Passwords must be changed every 60 to 90 days
- Passwords cannot be reused

Hardening Against Unsalted User Password Hash on Target 1

- Install Wordpress add on #2394
- This will add salts to the password hashes, leaving standard hash crackers much less effective.
- By adding random data to the input of a hash function to guarantee a unique output.

Hardening Against Unrestricted access to WordPress directories on Target 2

- How to patch:
  - First step would be to make sure wordpress is updated and running the most current version.
  - Next would be to audit who has what privileges
    - When it comes to wordpress there are 6 predefined roles. Each user should have the bare minimum access.
  - Password rules need to be made/enforced

Hardening Against XSS on Target 2 Webserver

- How to Patch:
  - Edit Apache config file (/etc/httpd/conf/httpd.conf)
    - Add:
      - Header always set X-XSS-Protection "1; mode=block”
    - Restart Apache
      - sudo systemctl restart apache2
  - Test this by opening up the webpage and pressing F12 and open the network tab
  - Refresh the page and look under headers
  - Look for the added line above.

Hardening Against Enumerating Users and Directories on Target 2

- User Enumeration
  - Navigate to the WordPress directory and add the following code to functions.php file.

```
if (!is_admin()) {
// default URL format
if (preg_match('/author=([0-9]*)/i', $_SERVER['QUERY_STRING']))
die(); add_filter('redirect_canonical', 'shapeSpace_check_enum',
10, 2);
}
function shapeSpace_check_enum($redirect, $request) {
// permalink URL format
if (preg_match('/\?author=([0-9]*)(\/*)/i', $request)) die();
else return $redirect;
}
```

- Directory Enumeration
  - In the .htaccess file add the following line:
    - Options -Indexes

Note: Downside is this disables all directory browsing

Best way is to audit all directories that is uploaded to wordpress.

Hardening Against Security Misconfiguration on Target 2

- How to patch:
  - Login as admin (or root)
  - Set so that user www-data cannot read sensitive data
    - sudo setfacl -m g:www-data:- /etc/passwd
    - Broaden restrictions to all of /etc
      - sudo setfacl -R -m g:www-data:- /etc/
  - Check permissions of sensitive files with “getfacl [filename]”

```
└─# getfacl /etc/passwd
getfacl: Removing leading '/' from absolute path names
# file: etc/passwd
# owner: root
# group: root
user::rw-
group::r--
group:www-data:---
mask::r--
other::r--
```

Implementing Patches

Implementing Patches with Ansible

This playbook fixes the vulnerabilities associated with outdated versions of Wordpress.

```
---

- name: update wordpress

  hosts: target1,target2 # connect to Target1 and Target2

  become: true # preform the following actions as root

  apt: # this module uses the package manager "apt" available on debian and ubuntu distributions

    update_cache: yes # preforms "apt update"

    name: wordpress # name of package to be affected

    state: latest # preforms "apt update wordpress"
```


LOONEY TUNES