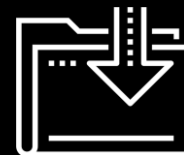




# Blue Team Report

By John Steskal, Melton McWilliams, Valentin Meica

Cybersecurity  
Final Project WeekBy



---

## Items we will cover in this report

01

Current Network Topology

02

Alerts Implemented

03

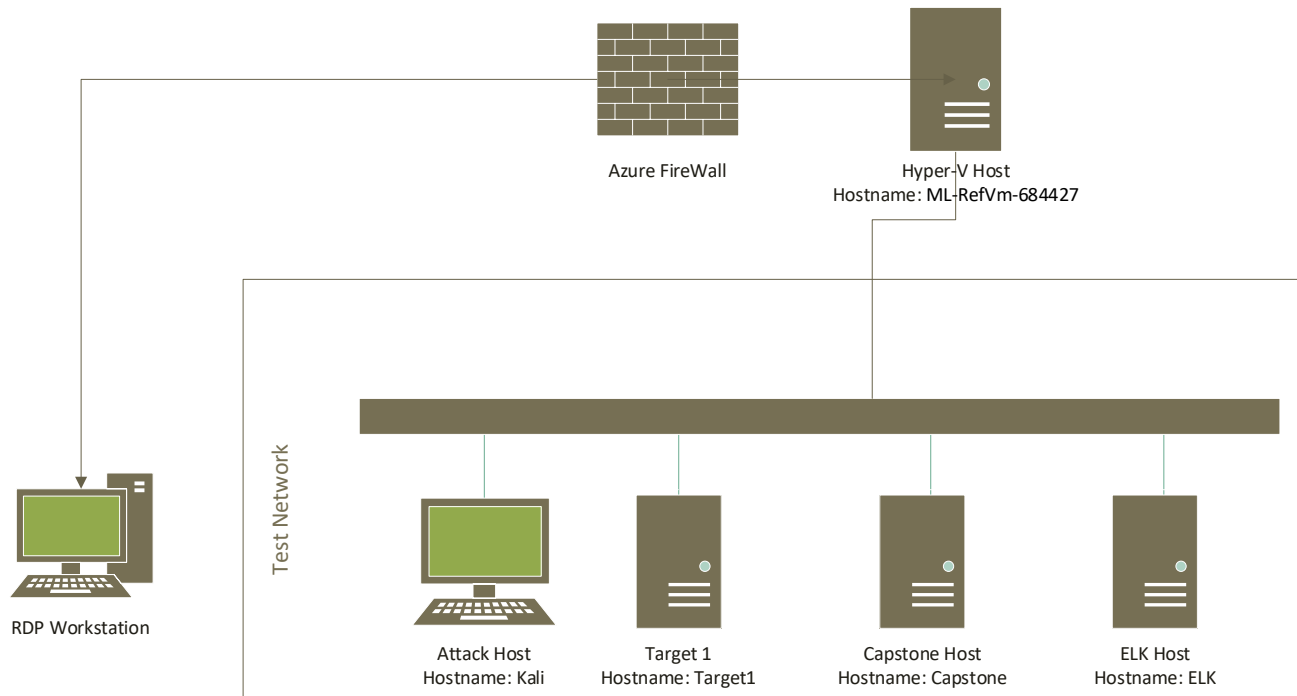
Hardening/Mitigation

04

Questions

---

# Network Topology



HostName	OS	IP Address	Purpose
ML-RefVm-684427	Windows 10	192.168.1.1	Hyper-V Host
Kali	Kali Linux 2020.1	192.168.1.90	Attacker Machine
Target 1	Debian Linux 8	192.168.1.110	Wordpress/Target Machine
Capstone	Ubuntu Linux 18.04	192.168.1.105	Log forwarder
ELK	Ubuntu Linux 18.04	192.168.1.100	ELK Stack Server



Alerts Implemented



Alert 1

## Alert 1: Excessive HTTP Errors

---

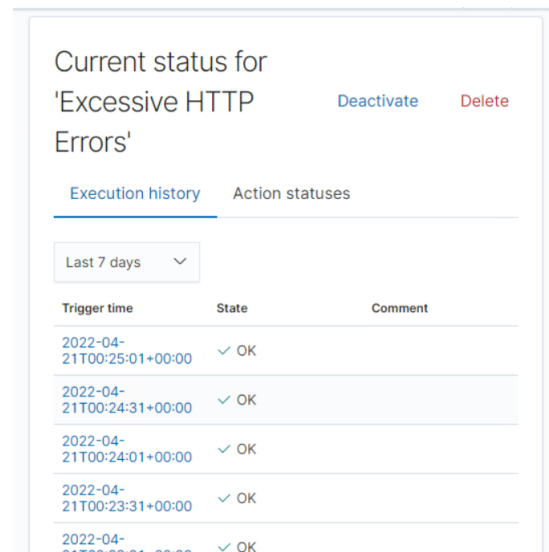
01

### Implementation

- **Metric:** http.response.status\_code
- **Threshold:** 400
- **Vulnerability Detected:** Brute Force Attack
- **Reliability:** High

02

### Visualization



The screenshot displays a dashboard for monitoring 'Excessive HTTP Errors'. At the top, it shows the 'Current status for' with a title 'Excessive HTTP Errors' and two action buttons: 'Deactivate' (blue) and 'Delete' (red). Below this, there are two tabs: 'Execution history' (active) and 'Action statuses'. Under the 'Execution history' tab, there is a dropdown menu set to 'Last 7 days'. A table follows, listing the trigger times, states, and comments for the execution history.

Trigger time	State	Comment
2022-04-21T00:25:01+00:00	✓ OK	
2022-04-21T00:24:31+00:00	✓ OK	
2022-04-21T00:24:01+00:00	✓ OK	
2022-04-21T00:23:31+00:00	✓ OK	
2022-04-21T00:23:01+00:00	✓ OK	

# Hardening for Alert 1



# Vulnerability: Wordpress Brute Force Attack

---

01

## Implementation

a brute-force attack is a trial of each and every possible combination of username and password to bypass the website admin login. These attacks are called brute-force because they use extensively forceful methods to break in. Unlike other attacks, they don't rely on the weaknesses or vulnerabilities of the website. Instead, they prey on easy passwords, unlimited login attempts, etc.

02

## Mitigation

1. Use strong login credentials
2. Hide WordPress login page
3. Two-factor Authentication
4. Limit login attempts
5. Whitelisting Access



Alert 2

## Alert 2: HTTP Request Size Monitor

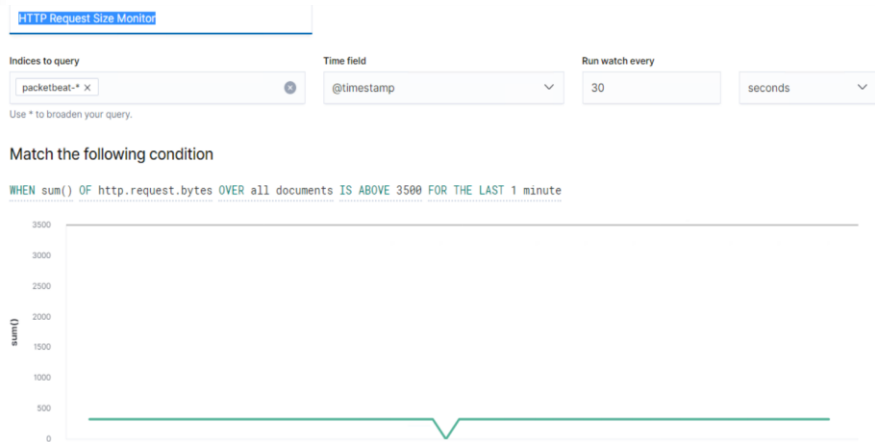
01

### Implementation

- **Metric:** http.request.bytes
- **Threshold:** 3500
- **Vulnerability Detected:** Code injection in HTTP requests (XSS and CRLF) or DDOS
- **Reliability:** Medium

02

### Visualization



# Hardening for Alert 2

# Vulnerability: Code injection in HTTP requests (XSS and CRLF) or DDOS

---

01

## Implementation

Code Injection:

An object injection vulnerability occurs when you fail to sanitize user-supplied input correctly before passing it to the unserialize() PHP function. Since PHP permits object serialization, attackers can potentially pass ad-hoc serialized strings to an unserialize() call. This may result in arbitrary PHP object(s) being injected into your application's scope.

DDOS:

DDoS attack, short for Distributed Denial of Service attack, is a type of cyber attack that uses compromised computers and devices to send or request data from a WordPress hosting server. The purpose of these requests is to slow down and eventually crash the targeted server.

02

## Mitigation

1. Keep WordPress Core Up-to-Date
2. Use a Web Application Firewall
3. Sanitize Your Data
4. Disable XML RPC in WordPress
5. Disable REST API in WordPress

Alert 3

## Alert 3: CPU Usage Monitor

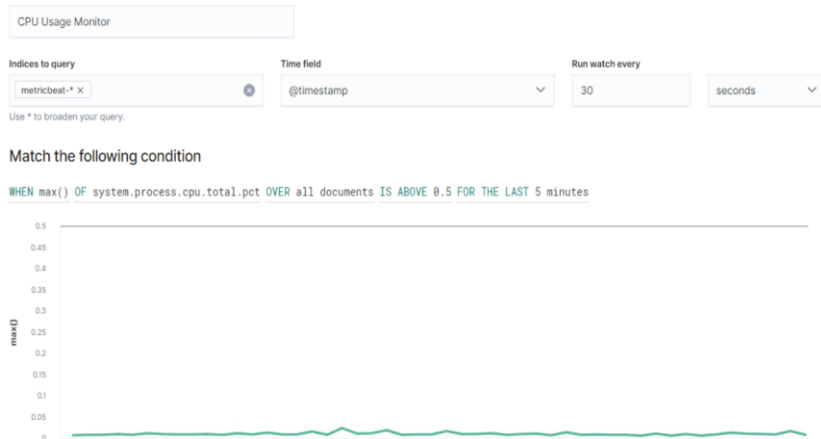
01

### Implementation

- **Metric:** system.process.cpu.total.pct
- **Threshold:** .5
- **Vulnerability Detected:** Virus/Malware or DDOS
- **Reliability:** High

02

### Visualization



# Hardening for Alert 3



# Vulnerability: Virus/Malware

---

01

## Implementation

Common web coding languages like PHP and Java allow programmers to refer to external files and scripts from within their code. The “include” command is the generic name for this type of activity.

In certain situations, a hacker can manipulate a website’s URL to compromise the “include” section of the code and gain access to other parts of the application server. Certain plug-ins for the WordPress platform have been found to be vulnerable against file inclusion attacks. When such hacks occur, the infiltrator can gain access to all data on the primary application server.

02

## Mitigation

1. Keep WordPress Core Up-to-Date
2. Use a Web Application Firewall
3. Sanitize Your Data
4. Disable XML RPC in WordPress
5. Disable REST API in WordPress
6. Install a Virus Scanner



Questions?