Blue Team: Summary of Operations

Table of Contents


Network Topology

The following machines were identified on the network:

- Name of Kali
  - Operating System: kali release 2020.1/kernel: Linux 5.4.0
  - Purpose: Attacker/ Pen Test Machine
  - IP Address: 192.168.1.90

- Name of ELK
  - Operating System: Ubunto 18.04
  - Purpose: Attacker/pentest
  - IP Address: 192.168.1.100


- Name of Target1
  - Operating System: Debian GNU/Linux 8/v3.16.0-6
  - Purpose: WordPress Host
  - IP Address: 192.168.1.110

- Name of Target2
  - Operating System: Debian GNU/Linux 8/v3.16.0-6
  - Purpose: WordPress host
  - IP Address: 192.168.1.115

- Name of Capstone
  - Operating System: Ubuntu 18.04
  - Purpose: Vulnerable web-server
  - IP Address: 192.168.1.105

* Description of Target

The target of this attack was: Target 1 (192.168.1.110).

* There were two VMs on the network which were vulnerable to attack Target-1 <192.168.1.110> and Target-2 <192.168.1.115>. Whereas we will only be focusing on Target-1 in the report.
* Target 1 is an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers. As such, the following alerts have been implemented:

* Monitoring the Targets

Target-1
* IP- 192.168.1.110
* Port open –22/TCP  open ssh
* Port open -80/TCP Open HTTP Apache

Alert 1: Excessive HTTP Errors
  - Metric: WHEN count() GROUPED OVER top 5 'http.response.status_code'
  - Threshold: Above 400
  - Vulnerability Mitigated:Enumeration
  - Reliability: Fairly reliable, as Kibana will only be trigger the alert for any error code above 400. Since anything over 400 is the one we need to be concerned about. Especially when considering this error going off at high rate. WP-scan

HTTP Request Size Monitor :
  - Metric: WHEN sum() of http.request.bytes OVER all documents
  - Threshold:  ABOVE 3500
  - Vulnerability Mitigated: Measuring Heavy traffic events, showing an attack is happening.
  - Reliability: Highly reliable, proven with john the ripper and hydra

CPU Monitor :
  - Metric: WHEN max() OF system.process.cpu.total.pct OVER all documents
  - Threshold: Above 0.5
  - Vulnerability Mitigated: John the ripper
  - Reliability: Not highly reliable as it spikes if if no attack is happening

Suggestions for Going Further:

Each alert above pertains to a specific vulnerability/exploit. Recall that alerts only detect malicious behavior, but do not stop it. For each vulnerability/exploit identified by the alerts above, suggest a patch. E.g., implementing a blocklist is an effective tactic against brute-force attacks. It is not necessary to explain _how_ to implement each patch.

The logs and alerts generated during the assessment suggest that this network is susceptible to several active threats, identified by the alerts above. In addition to watching for occurrences of such threats, the network should be hardened against them. The Blue Team suggests that IT implement the fixes below to protect the network:

- Vulnerability 1: Easily brute-forced passwords/ SSH Login
* Mitigation Techniques:
* Set a custom SSH port
* Why It Works - Since every hacker in the world knows about port-22 being a listening port for ssh, if we configure the port to a random one say 776, a level of redundancy can be added.
* Allow specific users
* Why It Works - Configuring the firewall to allow only specific users, means only certain IP would be allowed.
Vulnerability 2:
* Mitigation Technique
* 2-step: Disable scans and Block User Enumeration via .Htcaccess.
* Why it works: The will add another layer of security.
* Adding a code snippet to your theme’s functions.php file.

Vulnerability:3 Weak wp-config.php security implementation
* Mitigation Technique:
* Setting up a proper file security permission within user files & Protect the wp-config.php file with  .htacaccess file
* Why it works: Setting up the proper file security permissions could have mitigated reading the file when we performed the cat/nano command. This left the wp-config.php that listed a password in plaintext.