Blue Team: Summary of Operations

Table of Contents

- Network Topology
- Description of Targets
- Monitoring the Targets
- Suggestions for Going Further

Network Topology

The following machines were identified on the network:

- Target 1
  - **Operating System**: Debian GNU/Linux
  - **Purpose**: First target
  - **IP Address**: 192.168.1.110
- Target 2
  - **Operating System**: Debian GNU/Linux
  - **Purpose**: Second target
  - **IP Address**: 192.168.115
- ELK
  - **Operating System**: Debian GNU/Linux
  - **Purpose**: Monitoring
  - **IP Address**: 192.168.1.100
- Kali
  - **Operating System**: Kali Linux
  - **Purpose**: Penetration Testing System
  - **IP Address**: 192.168.1.90
- Capstone
  - **Operating System**: Debian GNU/Linux
  - **Purpose**: Alert Testing
  - **IP Address**: 192.168.1.105
- ML-REFVM-684427
  - **Operating System**: Windows
  - **Purpose**: NATSwitch
  - **IP Address**: 192.168.1.1

Blue Team: Summary of Operations

Description of Targets

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

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

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

Excessive HTTP Errors

Alert 1 is implemented as follows:

- **Metric**: http.response.status_code
- **Threshold**: Above 400 for the last 5 minutes
- **Vulnerability Mitigated**: Brute Force Attacks/Resource Usage Issues.
- **Reliability**: High reliability

HTTP Request Size Monitor

Alert 2 is implemented as follows:

- **Metric**: http.request.bytes
- **Threshold**: Above 3500 for the last minute
- **Vulnerability Mitigated**: DOS (Denial of Service) Attacks.
- **Reliability**: High reliability.

CPU Usage Monitor

Alert 3 is implemented as follows:

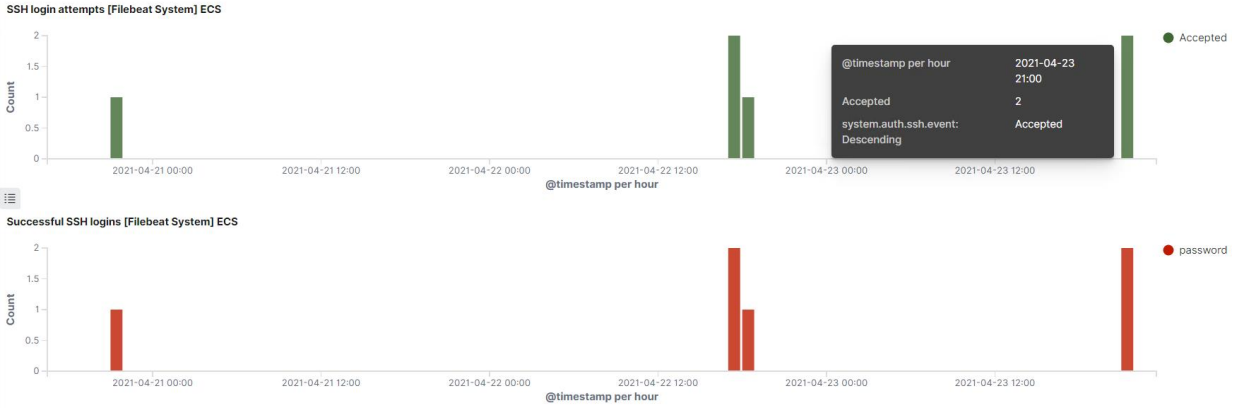
- **Metric**: system.process.cpu.total.pct
- **Threshold**: Above 0.5 for the last 5 minutes.
- **Vulnerability Mitigated**: Resource Management/Excessive CPU Usage.
- **Reliability**: Medium reliability.

By monitoring excessive HTTP errors, we can quickly detect and identify potential brute force attacks/resource usage issues on the network. This alert will catch any error codes above 400 that have been caused by the client (4XX), and server (5XX).

By monitoring the maximum HTTP request body size can help to identify large file uploads that could result in DoS attacks.

Blue Team: Summary of Operations

SSH Logins through port 22 is extremely vulnerable, especially given the fact that this machine is highly susceptible to brute force attacks. 192.168.1.90 was able to gain access through ssh login for both users Michael and Steven. This ultimately led to the attacker gaining root access to the system.

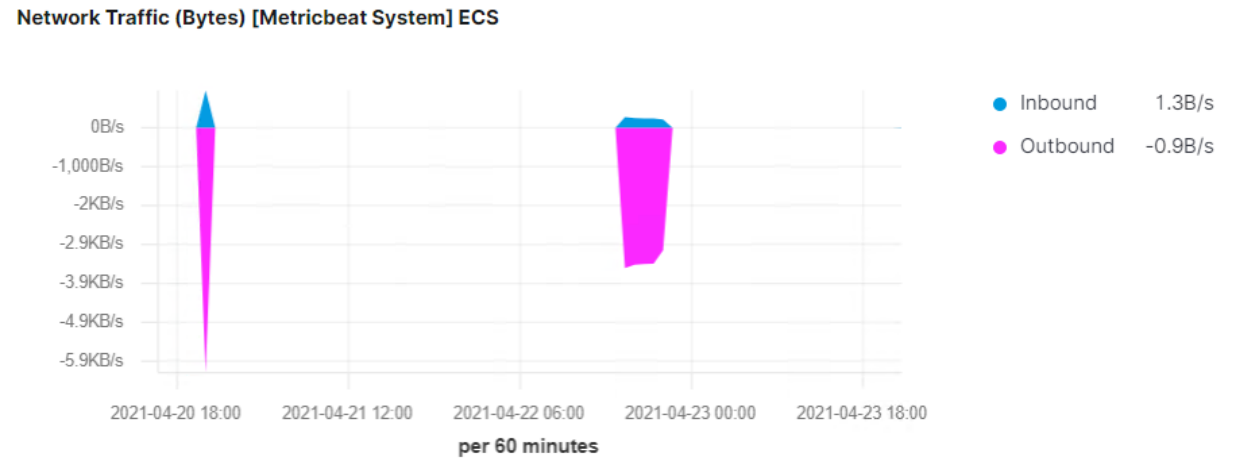


SSH login attempts [Filebeat System] ECS

Time	system.auth.ssh.event	system.auth.ssh.method	user.name	source.ip
Apr 23, 2021 @ 21:32:11.000	Accepted	password	michael	192.168.1.90
Apr 23, 2021 @ 21:30:05.000	Accepted	password	michael	192.168.1.90
Apr 22, 2021 @ 18:13:17.000	Accepted	password	michael	192.168.1.90
Apr 22, 2021 @ 17:44:42.000	Accepted	password	steven	192.168.1.90
Apr 22, 2021 @ 17:04:00.000	Accepted	password	michael	192.168.1.90

Top sudo commands [Filebeat System] ECS

system.auth.sudo.command: Descending	user.name: Descending	Count
/bin/sh	steven	1
/usr/bin/python -c import pty;pty.spawn("/bin/bash");	steven	1



Monitoring CPU usage spikes can help identify inefficient processes like heavy search or indexing workload and can provide essential clues to potential problems within the system, or opportunities to optimize performance. This can also help reduce false alarms and better identify anomalies.

Blue Team: Summary of Operations

Suggestions for Going Further

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: WordPress Username Disclosure/Brute Force Attack *CVE-2009-2335***

  https://cve.mitre.org/cgi-bin/cvename.cgi?name=2009-2335

  - **Patch**:
    - Implement invalid credentials lock out policy/Enforce strong passwords
    - Multi-Factor Authentication
    - Limit activity by Whitelisting specified IP addresses or ranges
    - Install SSHGuard/Fail2Ban
  - **Why It Works**:
    - A lockout policy limits the number of attempts an attacker can make on the network during a brute force attack. However, a lockout policy can increase the workload for other departments, and an attacker could also use this feature to affect service availability by locking out legitimate users.
    - Multi-factor authentication offers a better way to secure the login process. By requiring users to submit more than one authentication factor before gaining access, it mitigates the inherent risks of using a single password and is an effective defense against automated attacks.
    - Whitelisting IP addresses will only allow connections from trusted addresses.
    - SSHGuard is a fast and lightweight monitoring tool that monitors and protects servers from brute force attacks using their logging activity. If someone continuously trying to access a server via SSH with many unsuccessful attempts, the SSHGuard will temporarily block by putting their IP address in iptables.
    - Fail2ban is an open-source intrusion prevention system that can be used to prevent brute force attacks and other suspicious malicious attacks. It scans log files and bans IP's that exhibit malicious signatures, such as too many password failures, seeking for exploits etc.

Blue Team: Summary of Operations

- **Vulnerability 2: WordPress Password Hash Disclosure/Cracked Credentials**
  - **Patch**:
    - Protect the wp-config.php file by copying it to the .htaccess file, then updating the .htaccess to the root of the website.
    - Remove sensitive information by creating a new file called 'config.php' in a non-WWW accessible directory/above root directory.
  - **Why It Works**:
    - Adding the wp-config.php file to the .htaccess will deny access to your wp-config.php file and protect its contents.
    - Creating/moving the configuration file will ensure the file cannot be reached by any site visitors.

- **Vulnerability 3: Misconfigured SUDO Rights/Privilege Escalation Exploit**
  - **Patch**:
    - Revoke SUDO rights to any program which lets a user escape to the shell.
    - Revoke SUDO rights to vi, more, less, nmap, perl, ruby, python, gdb and others.
    - Revoke SUDO rights to any of the programming language compiler, interpreter and editors.
  - **Why It Works**:
    - Properly configuring the sudoers file will mitigate these privilege escalation techniques.