Blue Team: Summary of Operations - Daniel Stillings

Table of Contents

Network Topology

TODO: Fill out the information below.

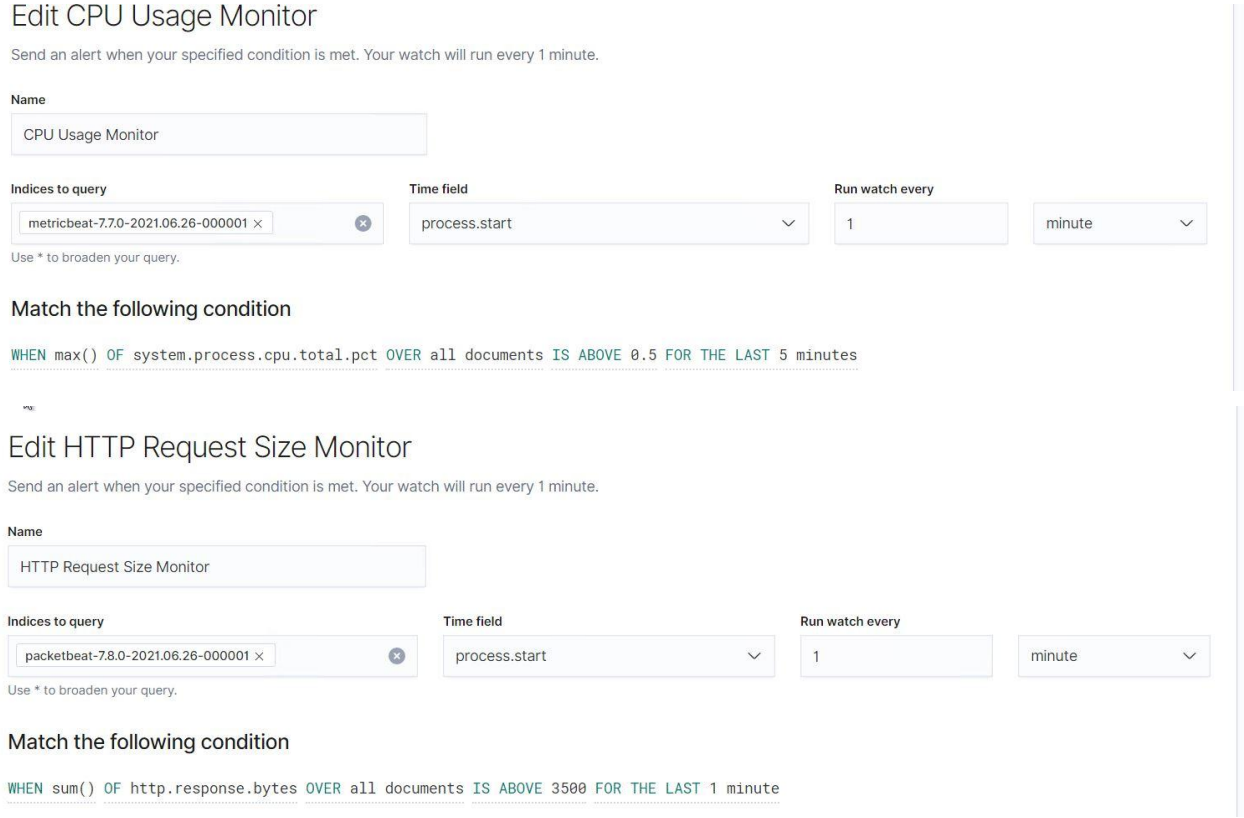
The following machines were identified on the network:

- Name of VM 1 - *Kali*
    - **Operating System**: *Linux*
    - **Purpose**: *Attacking Machine*
    - **IP Address**: *192.168.1.90*
- Name of VM 2 - *Elk*
    - **Operating System**: - *Ubuntu*
    - **Purpose**: - *Elk machine is used for obtaining event logs used with Kibana*
    - **IP Address**: - *192.168.1.100*
- Name of VM 3 - *Target 1*
    - **Operating System** - *Linux*
    - **Purpose** - *Wordpress Vulnerable machine*
    - **IP Address** - *192.168.1.110*
- Name of VM 4 - *Capstone*
    - **Operating System -** *Ubuntu*
    - **Purpose - *Machine was used as the vulnerable web server***
    - **IP Address -** *192.168.1.105*

TODO: Answer the questions below.

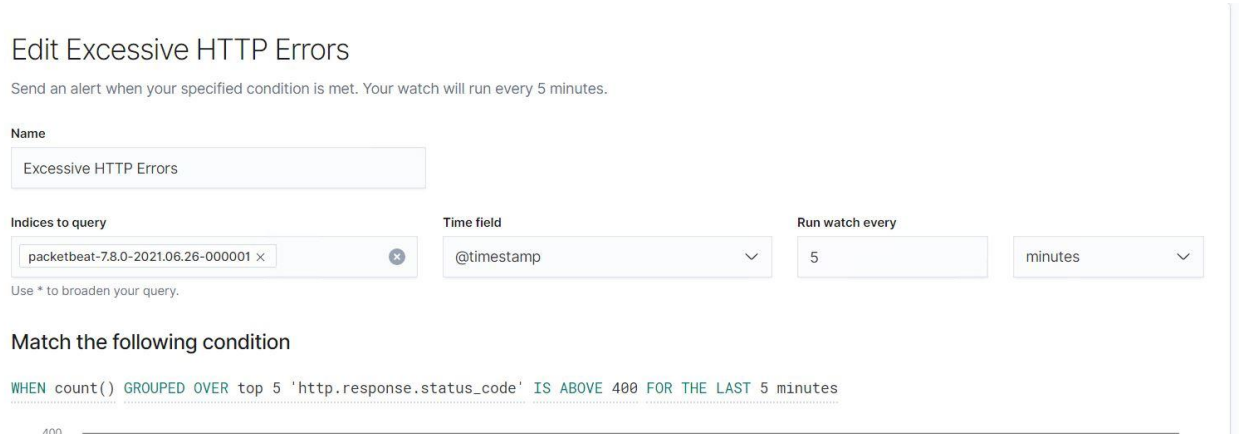
The target of this attack was: Target 1 (TODO: IP Address). ***192.168.1.110***

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: ***CPU Usage Monitor and HTTP Request Size Monitor.***



Monitoring the Targets

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



Name of Alert 1

CPU Use Monitor

Alert 1 is implemented as follows:

- **Metric**: When max () OF system.process.cup.total.pct
- **Threshold**: IS ABOVE 0.5 FOR THE LAST 5 minutes
- **Vulnerability Mitigated**: This would detect if there is abnormally higher CPU usage than normal.
- **Reliability**: The reliability would be low as the small threshold would allow for false positives.

HTTP request size

Alert 2 is implemented as follows:

- **Metric**: WHEN sum () OF http.response.bytes OVER all documents
- **Threshold**: IS ABOVE 3500 FOR THE LAST 1 minute
- **Vulnerability Mitigated**: An excessive number of HTTP requests can create a backlog of HTTP requests that could cause users to experience long load time.
- **Reliability**: This alert would be rated at medium as it could create false negatives.

Excessive HTTP Errors

Alert 3 is implemented as follows:

- **Metric**: WHEN count () GROUPED OVER TOP 5 http.response.status_code
- **Threshold**: IS ABOVE 400 FOR THE LAST 5 MINUTES
- **Vulnerability Mitigated**: The vulnerability mitigated here is Brute Force attacks.
- **Reliability**: This alert is highly reliable. It sends out 400 error codes to the attack preventing them from gaining access to a certain page.

Suggestions for Going Further (Optional)

- 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:

- Cpu Use Monitor
    - **Patch**: Harden the malware and antivirus systems
    - **Why It Works**: Antivirus systems (software) is to help prevent viruses and malware from getting onto your system.
- HTTP Request Size Monitor
    - **Patch**: I would recommend a DDOS hardening patch.
    - **Why It Works**: If the URL or HTTP request is hitting the threshold, it will send out a 400 type message, likely a 403 meaning access is forbidden.
- Excessive HTTP Errors
    - **Patch**: I would recommend to the staff to harden the WordPress server.
    - **Why It Works**: Hardening the server with regular updates could potentially prevent Brute Force attacks from happening and can also fix any vulnerabilities the server may have.