Blue Team: Summary of Operations

Table of Contents

- Network Topology
- Description of Targets
- Monitoring the Targets
- Patterns of Traffic & Behavior
- Suggestions for Going Further

Network Topology

The following machines were identified on the network:

The following machines were identified on the network:

- Kali
    - Operating System: Kernel: Linux 5.4.0
    - Purpose: Tool to Penetrate vulnerable web servers; Attaker
    - IP Address: 192.168.1.90
- ELK
    - Operating System: ubuntu 18.04
    - Purpose: Deployed ELK server to collect Logs of FileBeat, MetricBeat, and PacketBeat from the targeted web servers
    - IP Address: 192.168.1.100
- Capstone
    - Operating System: Ubuntu 18.04
    - Purpose: The Vulnerable Web Server
    - IP Address: 192.168.1.105
- Target 1
    - Operating System: Debian GNU/Linux 8/v3.16.0-6
    - Purpose: WordPress host
    - IP Address: 192.168.1.110

Description of Targets

TODO: Answer the questions below.

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:

- ☐ **SSH Login Alert**
  - Monitors any SSH brute force attack through credentials
  - Any user attempts to access system over port 22, an alert will trigger
  - Monitor SSH ports for any unauthorized access from people
- ☐ **SQL Database Alert**
  - Monitor unauthorized access from any SQL database
  - Alert should trigger when any external unauthorized IP tries to get access from SQL database or any files
- ☐ **Escalation Alert**
  - Monitor any unauthorized access to all the attempts
  - Any secret files or directories that are accessed unauthorized by users will be alerted.

Monitoring the Targets

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

Name of Alert 1

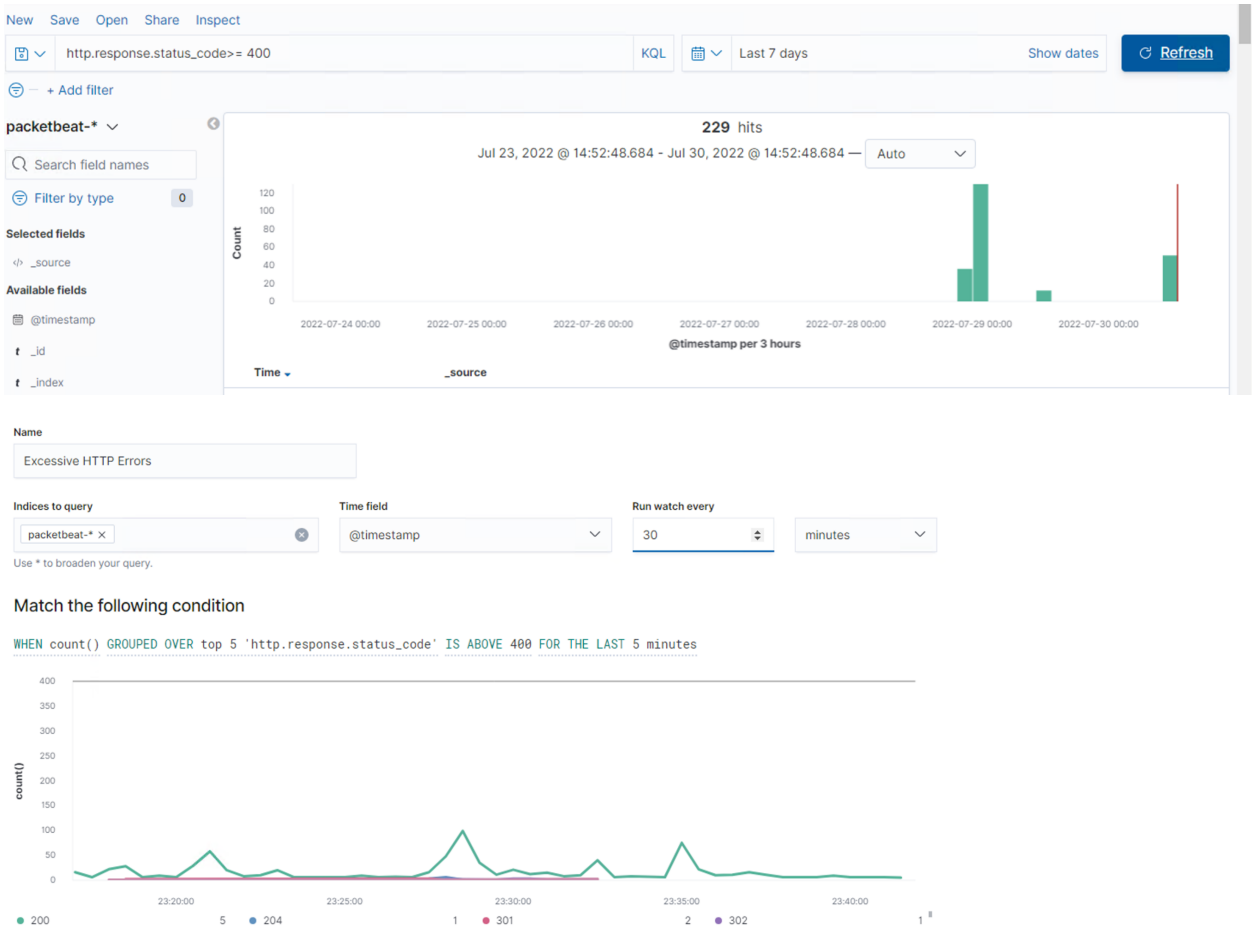
TODO: Replace Alert 1 with the name of the alert.

Excessive HTTP Error

Excessive HTTP Error is implemented as follows:

- **Metric**: Packetbeat
- **Threshold**: http.response.status_code above 400 for the last 5 mins
- **Vulnerability Mitigated**: BruteForce Attacks, Enumeration,phishing

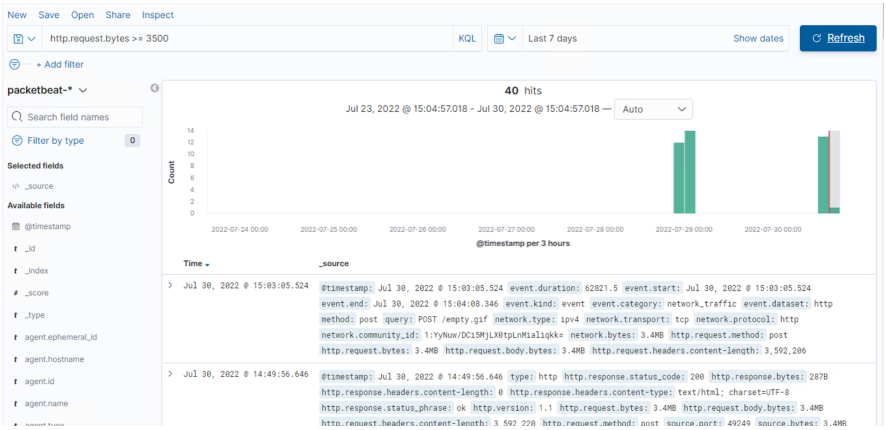
- **Reliability**: This results does generate lots of True positives because more than 300+ codes were from clients. With the threshold set, the reliability is highly.

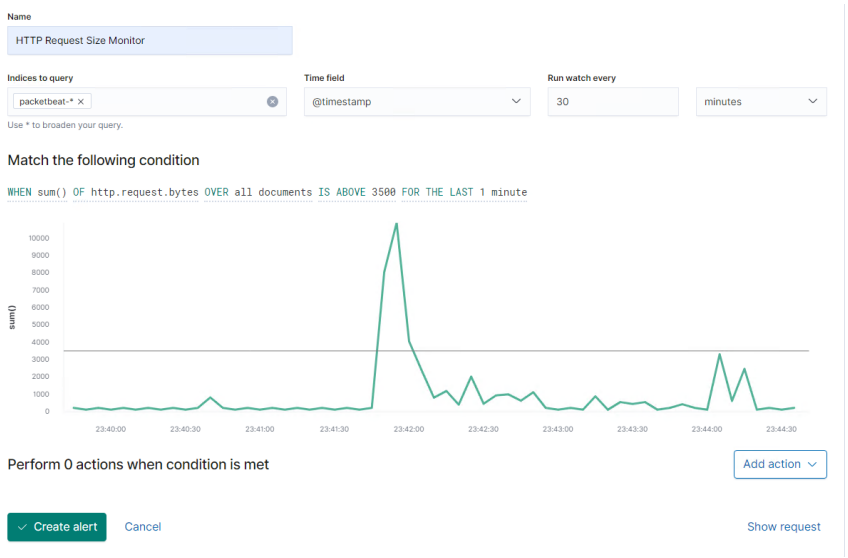


HTTP Request Size Monitor

HTTP Request Size is implemented as follows:

- **Metric**: Packetbeat
- **Threshold**: http.request.bytes above 3500 for the last 1 min
- **Vulnerability Mitigated**: Large files are transferred, High traffic events which could be a vulnerability
- **Reliability**: This would be a medium reliability, as alerts would generate a little bit of false positives. There might be a large file that is being transferred over within the network for work purposes but it gets alerted because of the threshold. It's not too reliable to catch malicious files.

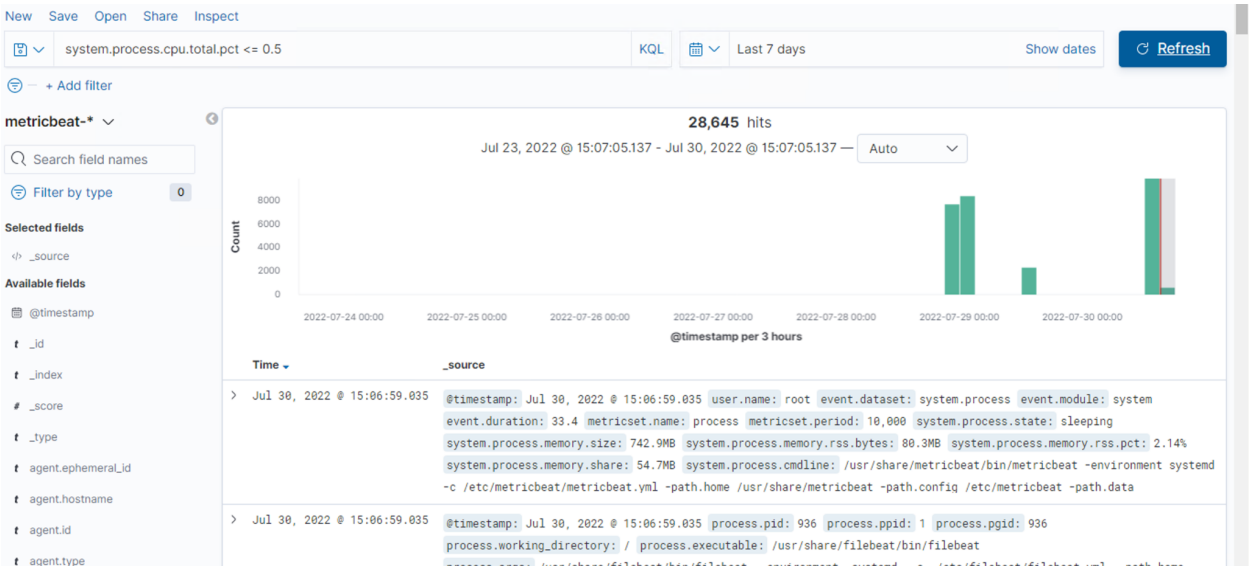


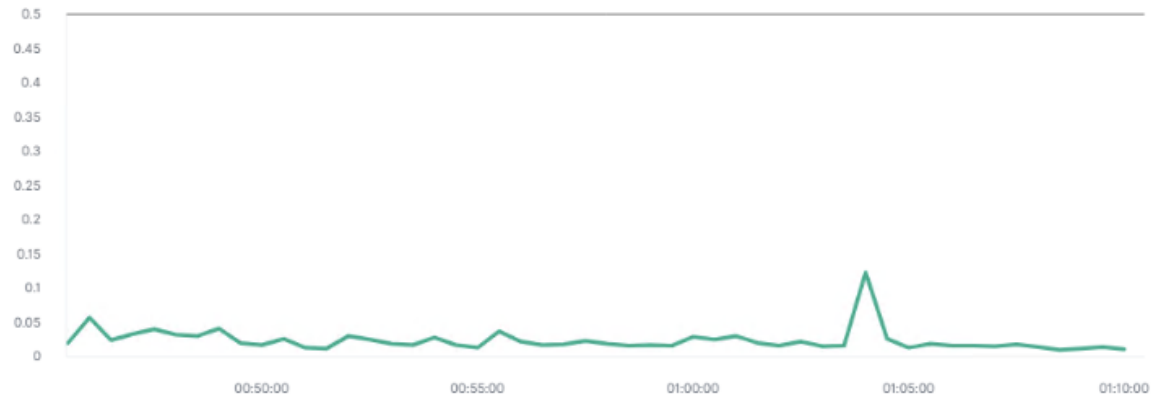


CPU Usage Monitor

CPU Usage Monitor is implemented as follows:

- **Metric**: Metricbeat
- **Threshold**: system.process.cpu.total.pct above 0.5 for the last 5 mins
- **Vulnerability Mitigated**: DoS attack, John the Ripper
- **Reliability**: The reliability is very low. The cpu triggered unnecessary alerts even if it was not attacked.


New
Save
Open
Share
Inspect
system.process.cpu.total.pct <= 0.5
KQL
Last 7 days
Show dates
Refresh
+ Add filter
metricbeat-*
Search field names
Filter by type
0
Selected fields
_source
Available fields
@timestamp
_id
_index
_score
_type
agent.ephemeral_id
agent.hostname
agent.id
agent.type
28,645 hits
Jul 23, 2022 @ 15:07:05.137 - Jul 30, 2022 @ 15:07:05.137 —
Auto
Count
8000
6000
4000
2000
0
2022-07-24 00:00
2022-07-25 00:00
2022-07-26 00:00
2022-07-27 00:00
2022-07-28 00:00
2022-07-29 00:00
2022-07-30 00:00
@timestamp per 3 hours
Time
_source
Jul 30, 2022 @ 15:06:59.035
@timestamp: Jul 30, 2022 @ 15:06:59.035 user.name: root event.dataset: system.process event.module: system event.duration: 33.4 metricset.name: process metricset.period: 10,000 system.process.state: sleeping system.process.memory.size: 742.9MB system.process.memory.rss.bytes: 80.3MB system.process.memory.rss.pct: 2.14% system.process.memory.share: 54.7MB system.process.cmdline: /usr/share/metricbeat/bin/metricbeat -environment systemd -c /etc/metricbeat/metricbeat.yml -path.home /usr/share/metricbeat -path.config /etc/metricbeat -path.data
Jul 30, 2022 @ 15:06:59.035
@timestamp: Jul 30, 2022 @ 15:06:59.035 process.pid: 936 process.ppid: 1 process.pgid: 936 process.working_directory: / process.executable: /usr/share/filebeat/bin/filebeat


0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
00:50:00
00:55:00
01:00:00
01:05:00
01:10:00

Suggestions for Going Further (Optional)

TODO:

- 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: Weak passwords vulnerabilities and open port 22 SSH Login**
    - **Disallow access to port 22 openSSH**
    - **Why It Works**: Shutting this port would stop the SSH associations with the server, forestalling the unapproved access. Common tools for securing ports are firewalld, ufw, or any 3rd party firewall.
    - **Allow Only Specific IP Connections**
    - **Why It Works**: Whitelist all known and safe IP for SSL. Using firewalld or similar programs will allow for ease in implementation. Close port 80 HTTP and open port 443 HTTPS for more secure/encrypted traffic.
    - **Making critical passwords should be Mandatory**
    - **Why It Works:** The Red Group had the option to guess Michael's record with such ease as the password was so basic. This can be assumed that we can execute a strategy for passwords that would assist workers with thinking of muddled passwords. Likewise, it would be a solid match if there is a training course for security passwords to assist workers with grasping the significance of safety.
- **Vulnerability 2: WordPress User Enumeration**
    - **Patch**: use a free plugin "WP Hardening" to disable User Enumeration in WordPress Install and activate plugin > 'Security Fixers' tab > Stop user enumeration.
    - **Why It Works**: By disabling the User Enumeration function, it will stop the attacker from counting user records.
- **Vulnerability 3: Escalation Vulnerabilities**
    - **Patch**: ONLY allow users that are responsible for the task
    - **Why It Works**: By having right record consents for user accounts, we can keep up with command over assigned jobs and authorizations for any records.
- **Vulnerability 4: WordPress Configuration & SQL Database**
    - **Patch**: Hashed wordpress database login information from wp-config.php
    - **Why It Works** :By using the encryption, the unauthorized person gets into someone else’s system, but they won’t be able to easily grab the login credentials to access the database.