Blue Team

Summary of Operations

Submitted by: Bill, Matt, Shannon, Tanya

Table of Contents

1. Network Topology

The following machines were identified on the network:

- **Kali**
  - Operating System: Linux
  - Purpose: Penetration Testing Machine/Attacking VM
  - IP Address: 192.168.1.90
- **ELK**
  - Operating System: Linux
  - Purpose: SIEM
  - IP Address: 192.168.1.100
- **Capstone**
  - Operating System: Linux
  - Purpose: Target VM
  - IP Address: 192.168.1.105
- **Target 1**
  - Operating System: Linux
  - Purpose: Vulnerable WordPress Server
  - IP Address: 192.168.1.110
- **Target 2**
  - Operating System: Linux
  - Purpose: Vulnerable WordPress Server
  - IP Address: 192.168.1.115

2. Description of Targets

The target of this attack was: **Target 1** (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:

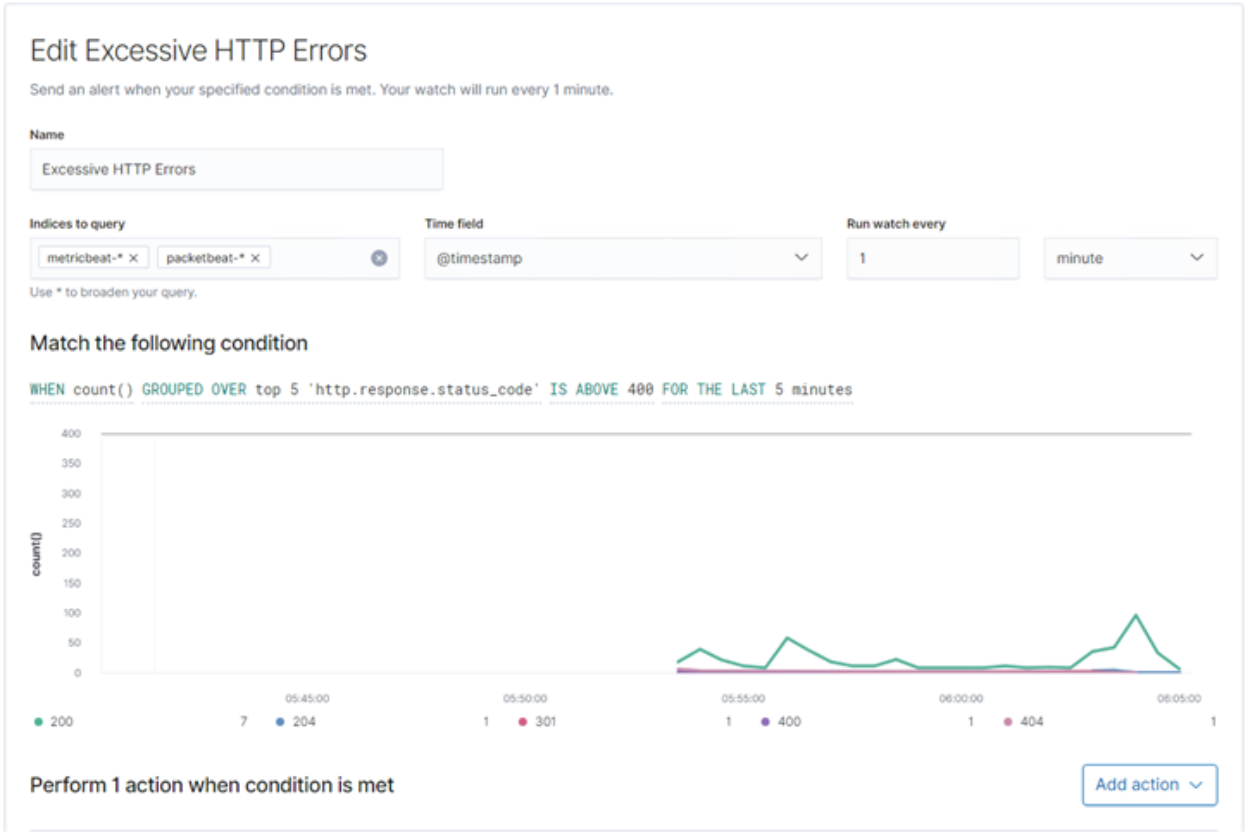
3. Monitoring the Targets

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

Name of Alert 1 : Excessive HTTP Errors

Alert 1 is implemented as follows:

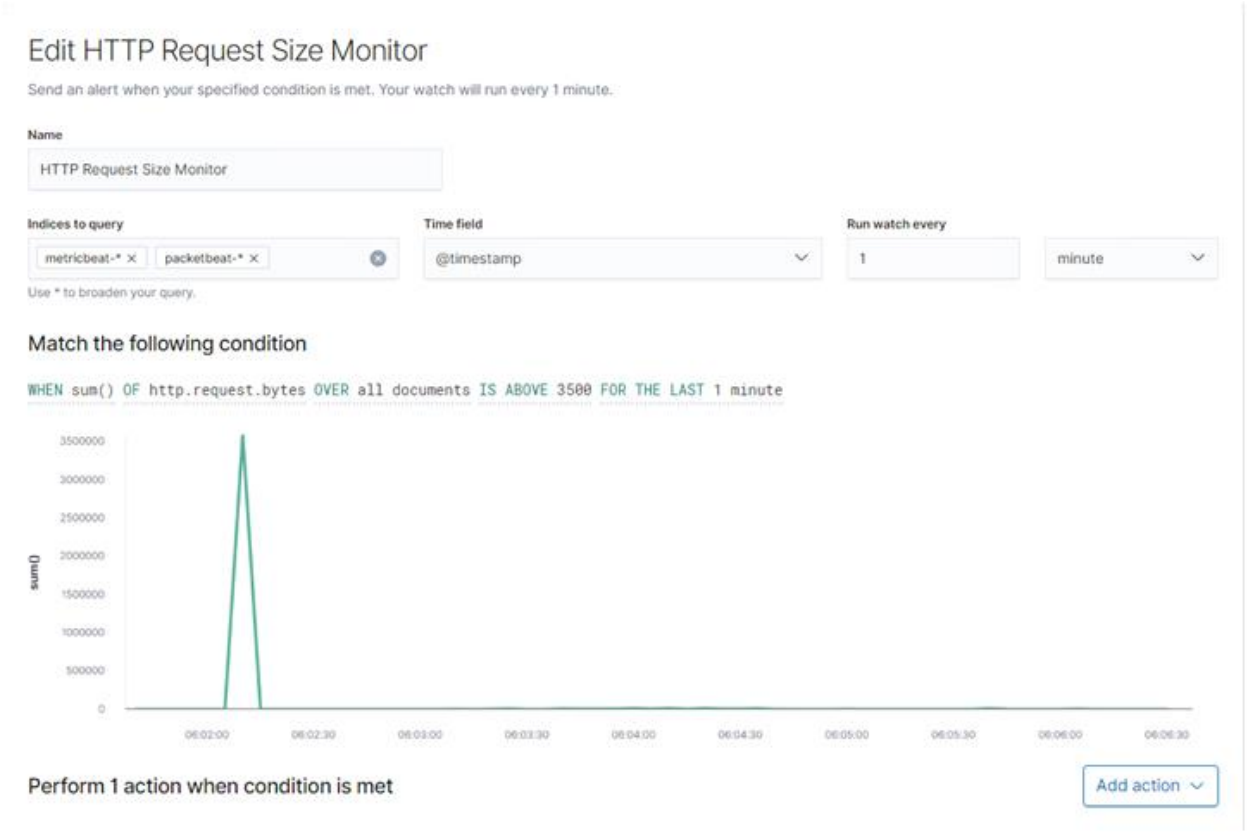
- **Metric**: WHEN count() GROUPED OVER top 5 'http.response.status_code'
- **Threshold**: IS ABOVE 400 FOR THE LAST 5 minutes
- **Vulnerability Mitigated**: Brute Force, DoS, DDoS
- **Reliability**: This alert could generate false positives, with a rating of medium to high reliability.



Name of Alert 2 : HTTP Request Size Monitor

Alert 2 is implemented as follows:

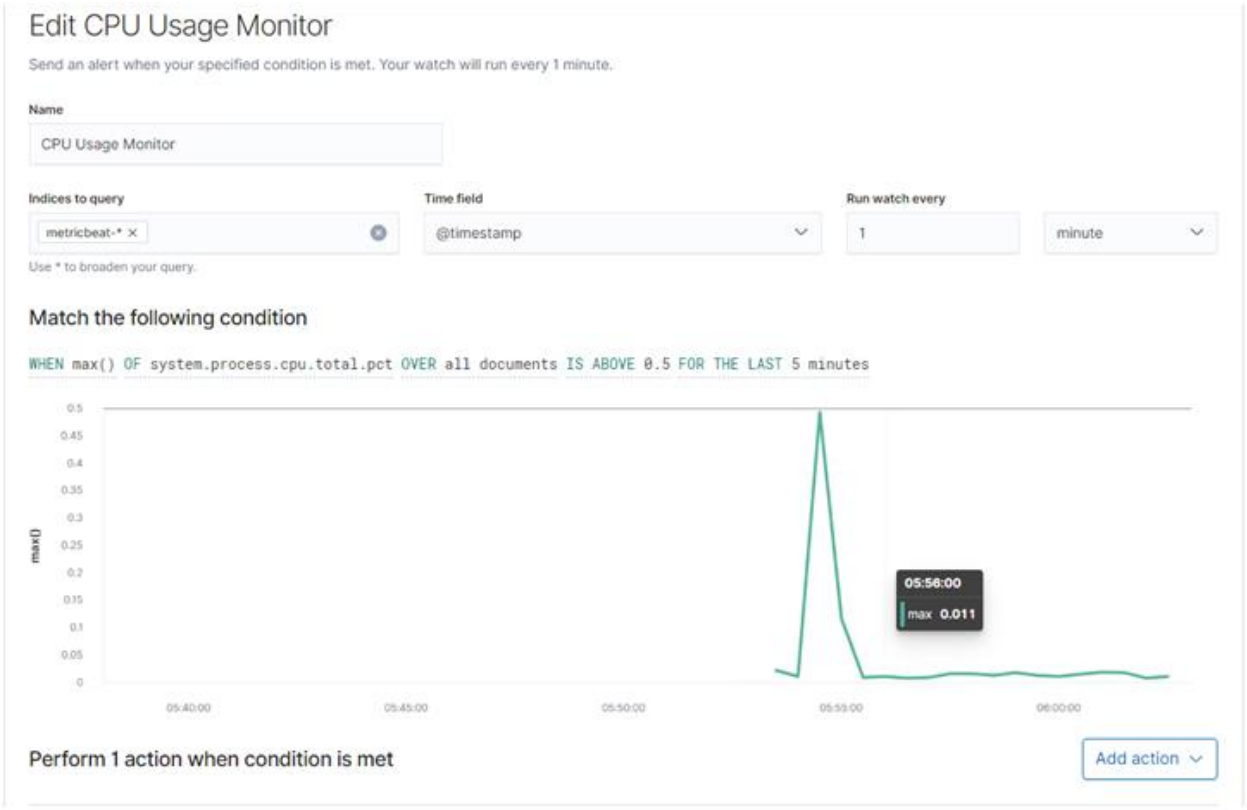
- **Metric**: WHEN sum() of http.request.bytes OVER all documents
- **Threshold**: IS ABOVE 3500 FOR THE LAST 1 minute
- **Vulnerability Mitigated**: SQL Injection, Cross-Site Scripting, HTTP Request Smuggling
- **Reliability**: This alert could generate both false positives and false negatives, with a Reliability Rating of medium or high dependent upon the HTTP method



Name of Alert 3 : CPU Usage Monitor

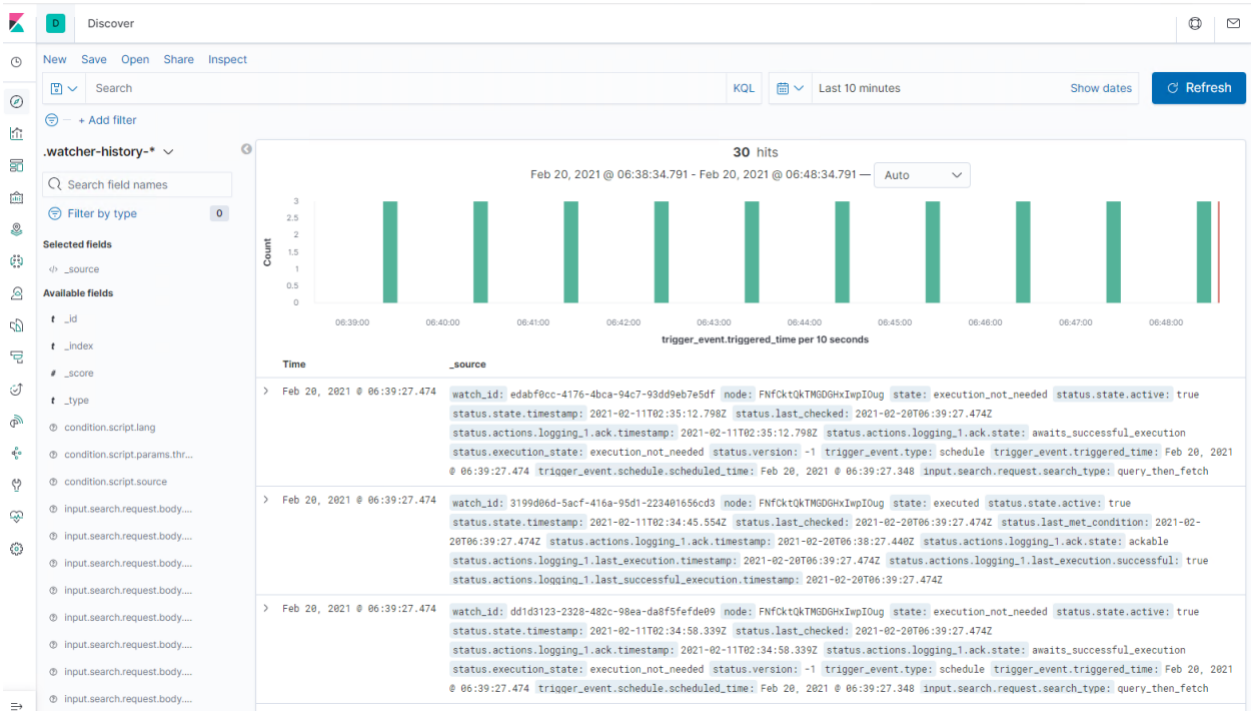
Alert 3 is implemented as follows:

- **Metric**: WHEN max() OF system.process.cpu.total.pct OVER all documents
- **Threshold**: IS ABOVE 0.5 FOR THE LAST 5 minutes
- **Vulnerability Mitigated**: Alerts to hidden processes running in the background
- **Reliability**: This alert will not generate false positives or false negatives, as it is strictly monitoring the overall CPU usage, and not specific processes



4. Pattern of Traffic and Behavior

Configured Alerts Monitoring:



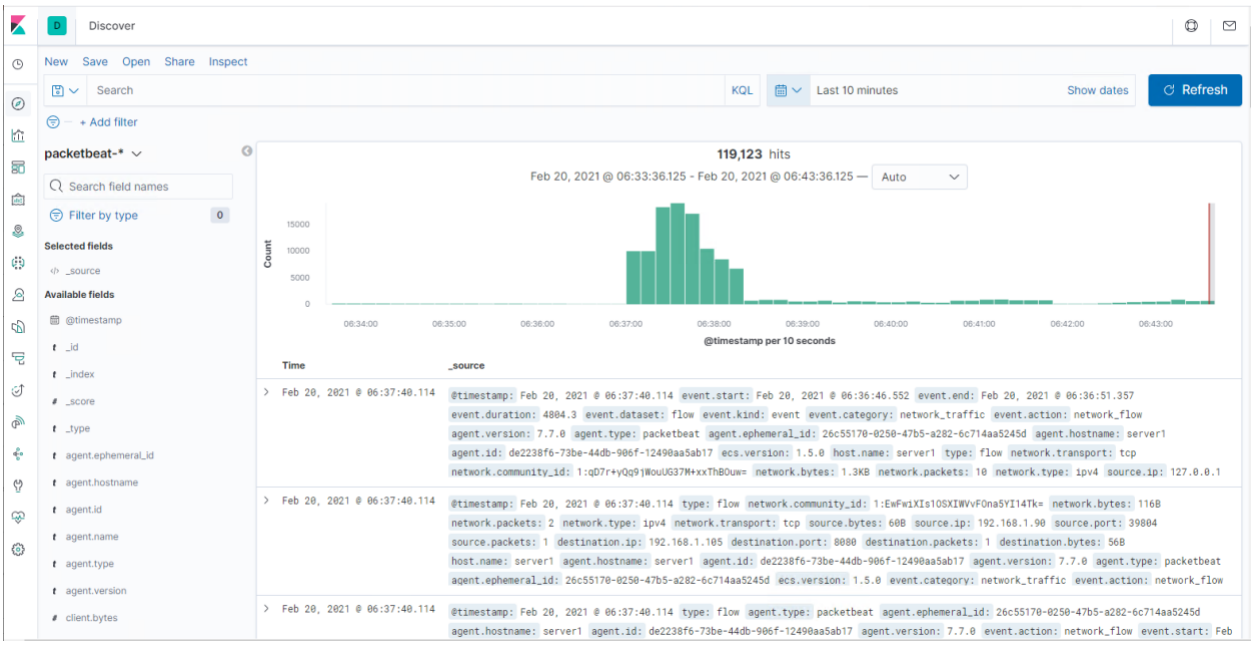
The below 3 Monitoring keeps checking the events to see if any of the alert conditions is met.

Feb 20, 2021 @ 06:43:27.615
watch_id: 3199d06d-5acf-416a-95d1-223401656cd3 node: FNfCktQkTMGDGHxIwpIOug state: executed status.state.active: true status.state.timestamp: 2021-02-11T02:34:45.554Z status.last_checked: 2021-02-20T06:43:27.615Z status.last_met_condition: 2021-02-20T06:43:27.615Z status.actions.logging_1.ack.timestamp: 2021-02-20T06:43:27.615Z status.actions.logging_1.ack.state: ackable status.actions.logging_1.last_execution.timestamp: 2021-02-20T06:43:27.615Z status.actions.logging_1.last_execution.successful: true status.actions.logging_1.last_successful_execution.timestamp: 2021-02-20T06:43:27.615Z

Feb 20, 2021 @ 06:43:27.615
watch_id: edabf0cc-4176-4bca-94c7-93dd9eb7e5df node: FNfCktQkTMGDGHxIwpIOug state: execution_not_needed status.state.active: true status.state.timestamp: 2021-02-11T02:35:12.798Z status.last_checked: 2021-02-20T06:43:27.615Z status.actions.logging_1.ack.timestamp: 2021-02-11T02:35:12.798Z status.actions.logging_1.ack.state: awaits_successful_execution status.execution_state: execution_not_needed status.version: -1 trigger_event.type: schedule trigger_event.triggered_time: Feb 20, 2021 @ 06:43:27.615 trigger_event.schedule.scheduled_time: Feb 20, 2021 @ 06:43:27.348 input.search.request.search_type: query_then_fetch

Feb 20, 2021 @ 06:43:27.615
watch_id: dd1d3123-2328-482c-98ea-da8f5fefde09 node: FNfCktQkTMGDGHxIwpIOug state: execution_not_needed status.state.active: true status.state.timestamp: 2021-02-11T02:34:58.339Z status.last_checked: 2021-02-20T06:43:27.615Z status.actions.logging_1.ack.timestamp: 2021-02-11T02:34:58.339Z status.actions.logging_1.ack.state: awaits_successful_execution status.execution_state: execution_not_needed status.version: -1 trigger_event.type: schedule trigger_event.triggered_time: Feb 20, 2021 @ 06:43:27.615 trigger_event.schedule.scheduled_time: Feb 20, 2021 @ 06:43:27.348 input.search.request.search_type: query_then_fetch

Packetbeat logs in Kibana: For port scan on 192.168.1.0/24



5. 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 supports the ***Pingback XML-RPC API***.
    - **More info:** https://www.avsecurity.in/wordpress-xml-rpc-pingback-vulnerability/
    - **Impact:** xmlrpc.php poses a security risk. It creates an additional access point to your site, which could leave it vulnerable to external attacks. Every time you authenticate XML-RPC, you need to supply your username and password. As you can imagine, this isn't exactly ideal for security purposes.

      For example, in order to prevent brute force attacks, you can limit login attempts on your WordPress site. However, with XML-RPC enabled, that limit does not exist. There's no capping on login attempts, which means it's only a matter of time before a determined cybercriminal gains access.

    - **Patch**: Use a plugin which disables this feature, this way you are closing a potential area of entry for hackers.
      https://wordpress.org/plugins/simple-xml-rpc-pingback-disabler/
    - **Why It Works**: The plugin simply disables only the XML-RPC API Pingback Methods used by hackers on a WordPress site, providing an easy and simple way

to disable/enable XML-RPC API Pingback Methods without completely disabling the XML-RPC API, which is used by some plugins and applications (i.e. mobile apps or a few Jetpack modules).
Removes the following methods from the XML-RPC API interface.

- pingback.ping
- pingback.extensions.getPingbacks
- X-Pingback from HTTP headers

- **Vulnerability 2:** Wordpress supports ***wp-cron.php***
  - **Impact:** depending on the amount of traffic to your site, using the built-in cron handler can actually start to impact your page load times. Potential use for DoS/DDoS attacks.
  - **Patch**: Disable WP-Cron (wp-cron.php) and instead use a system cron for faster performance
  - **Why It Works**: wp-cron.php doesn't run at page load, thus providing better performance.

- **Vulnerability 3:** Use of weak passwords
  - **Patch**: Enforce password policy to have complex and long passwords
  - **Why It Works**: Complex and long passwords are difficult to break

- **Vulnerability 4**: Wordpress security vulnerabilities
  - **Patch:**
    - Updating wordpress to latest version
    - Installing security plugins
    - Enable Web Application Firewall (WAF)
    - Move wordpress site to SSL/HTTPS
  - **Why it works**: They will protect the WordPress site from malware, brute force attacks, and hacking attempts and patches vulnerabilities which exist in the prior versions