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:

- **Network**
  - Address Range: 192.168.1.0/24
  - Netmask:255.255.255.0
  - Gateway: 192.168.1.1
  - Cloud Provider: Azure
- **Machines**
  - IPv4: 192.168.1.90
  - OS: Kali Linux 5.4.0
  - Hostname: Kali
  - Purpose: Penetration Testing

  - IPv4: 192.168.1.110
  - OS: Linux
  - Hostname: Target 1
  - Purpose: Target Machine with Wordpress Vulnerabilities

  - IPv4: 192.168.1.100
  - OS: Linux
  - Hostname: Elk
  - Purpose: Metricbeat, Filebeat, Packetbeat, Watcher Log Collection

  - IPv4: 192.168.1.105
  - OS: Linux Ubuntu
  - Hostname: Capstone
  - Purpose: Kibana

Description of Targets

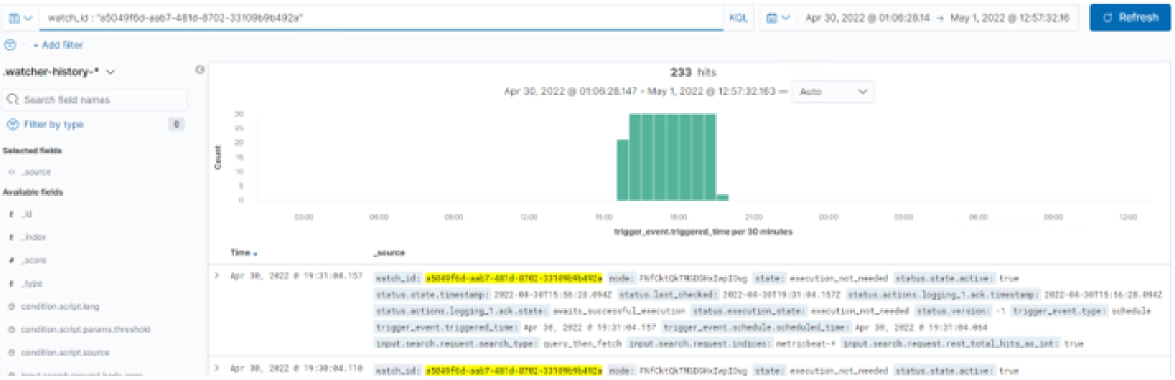
TODO: Answer the questions below.

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:

Monitoring the Targets

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

Alert 1 - HTTP Request Size Monitor



```
watch_id : "a5049f6d-aab7-481d-8702-33109b9b492a"
```

Apr 30, 2022 @ 01:06:28.14 → May 1, 2022 @ 12:57:32.16

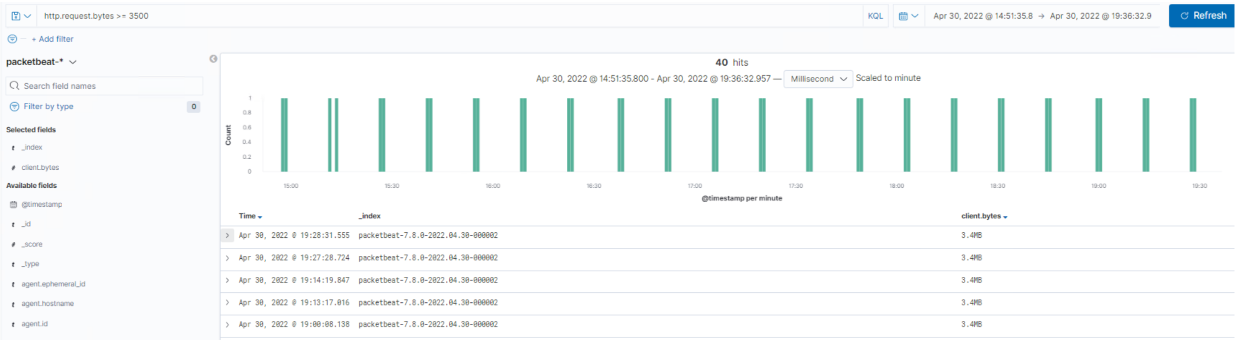
233 hits

Apr 30, 2022 @ 01:06:28.147 - May 1, 2022 @ 12:57:32.163

```
Apr 30, 2022 @ 19:31:04.157  watch_id: a5049f6d-aab7-481d-8702-33109b9b492a node: FNfCk1QkTMGDSHxIepIDug state: execution_not_needed status.state.active: true
  status.state.timestamp: 2022-04-30T15:56:28.094Z status.last_checked: 2022-04-30T19:31:04.157Z status.actions.logging_1.ack.timestamp: 2022-04-30T15:56:28.094Z
  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: Apr 30, 2022 @ 19:31:04.157 trigger_event.schedule.scheduled_time: Apr 30, 2022 @ 19:31:04.064
  input.search.request.search_type: query_then_fetch input.search.request.indices: metricbeat-* input.search.request.rest_total_hits_as_int: true
Apr 30, 2022 @ 19:30:04.110  watch_id: a5049f6d-aab7-481d-8702-33109b9b492a node: FNfCk1QkTMGDSHxIepIDug state: execution_not_needed status.state.active: true
```

Alert 1 is implemented as follows:

- **Metric**: This monitoring rule watches the http.request.bytes from metricbeat
- **Threshold**: It will fire when it reaches a sum of 3500 for the last minute
- **Syntax:** WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- **Vulnerability Mitigated**: DDoS attacks
- **Reliability**: This is a useful and highly reliable alert as it can point to suspicious activity and identify the beginning of serious DoS attacks. It's a very straight forward way of identifying large data transactions in short duration without the risk of excessive false positives



http.request.bytes >= 3500

Apr 30, 2022 @ 14:51:35.8 → Apr 30, 2022 @ 19:36:32.9

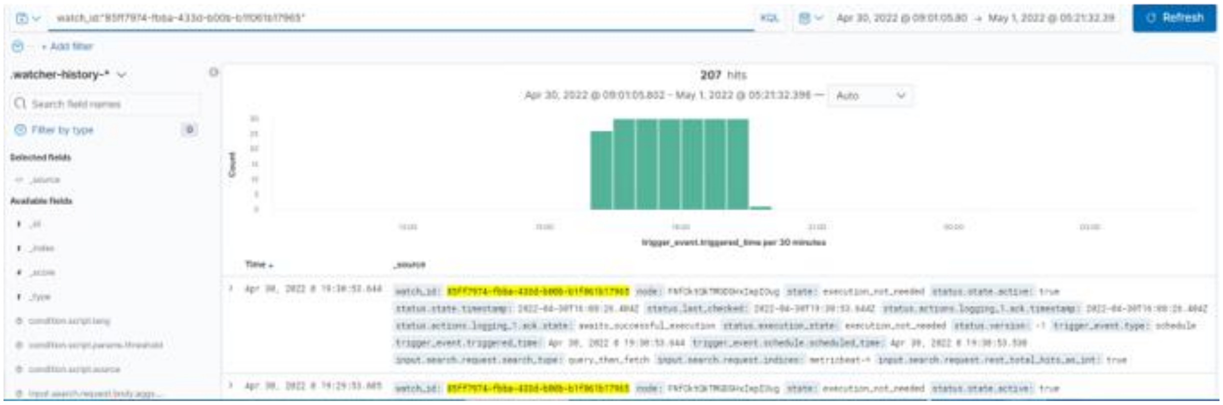
packetbeat-*

40 hits

Apr 30, 2022 @ 14:51:35.800 - Apr 30, 2022 @ 19:36:32.957 — Millisecond Scaled to minute

Time	_index	client.bytes
Apr 30, 2022 @ 19:28:31.555	packetbeat-7.8.0-2022.04.30-000002	3.4MB
Apr 30, 2022 @ 19:27:28.724	packetbeat-7.8.0-2022.04.30-000002	3.4MB
Apr 30, 2022 @ 19:14:19.847	packetbeat-7.8.0-2022.04.30-000002	3.4MB
Apr 30, 2022 @ 19:13:17.016	packetbeat-7.8.0-2022.04.30-000002	3.4MB
Apr 30, 2022 @ 19:00:08.138	packetbeat-7.8.0-2022.04.30-000002	3.4MB

Alert 2 - Excessive HTTP Errors



watch_id:"85f7974-fbba-433d-b00b-b1f061b17965"

Apr 30, 2022 @ 09:01:05.80 → May 1, 2022 @ 05:21:32.39

.watcher-history-*

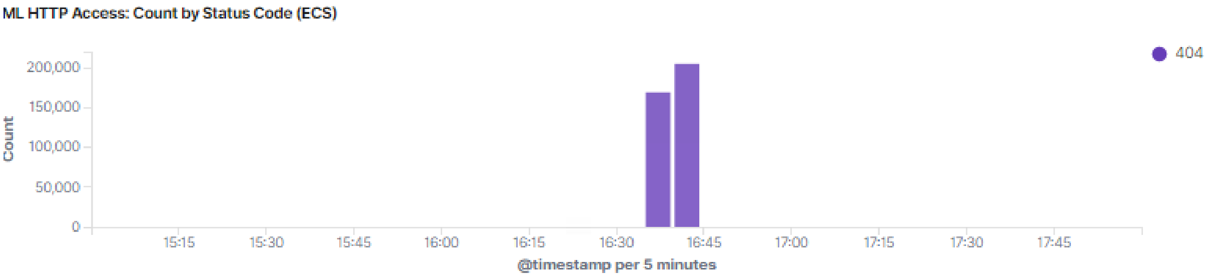
207 hits

Apr 30, 2022 @ 09:01:05.802 - May 1, 2022 @ 05:21:32.396 — Auto

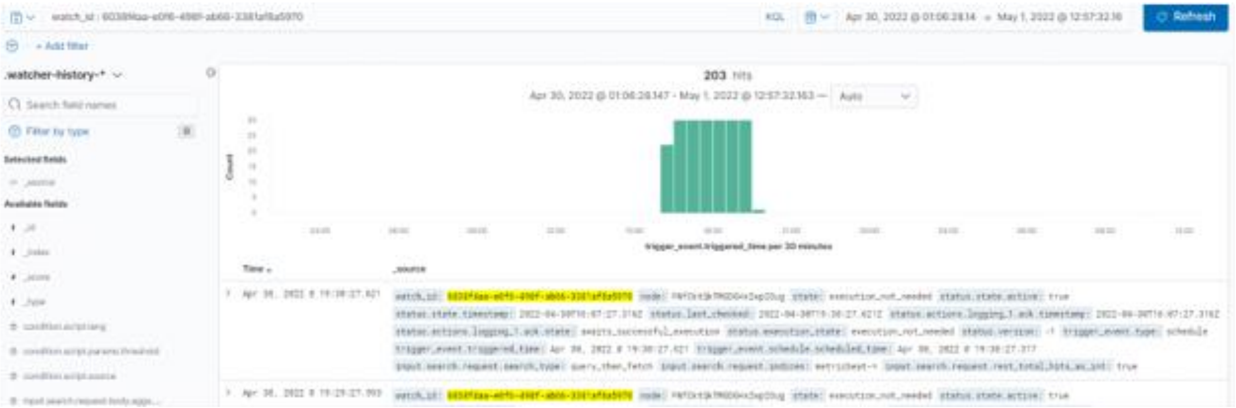
Time	_source
Apr 30, 2022 @ 19:30:53.644	watch_id: 85f7974-fbba-433d-b00b-b1f061b17965 node: FNfCkYGKTRODGHxIepDJug state: execution_not_needed status.state.active: true status.state.timestamp: 2022-04-30T16:00:26.404Z status.last_checked: 2022-04-30T19:30:53.644Z status.actions.logging_1.ack.timestamp: 2022-04-30T16:00:26.404Z 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: Apr 30, 2022 @ 19:30:53.644 trigger_event.schedule.scheduled_time: Apr 30, 2022 @ 19:30:53.530 input.search.request.search_type: query_then_fetch input.search.request.indices: metricbeat-* input.search.request.rest_total_hits_as_int: true
Apr 30, 2022 @ 19:29:53.605	watch_id: 85f7974-fbba-433d-b00b-b1f061b17965 node: FNfCkYGKTRODGHxIepDJug state: execution_not_needed status.state.active: true

Alert 2 is implemented as follows:

- **Metric**: This monitoring rule watches the http.response.status_code from metricbeat
- **Threshold**: It will fire when it reaches above a count of 400 for the last 5 minutes
- **Syntax:** WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- **Vulnerability Mitigated**: Enumeration
- **Reliability**: Alerting to the high amount of 404 errors gives a good indicator of the need for further investigation, be it enumeration or a DoS attack. This alert is reliable, without a log of false positives and highly valuable.



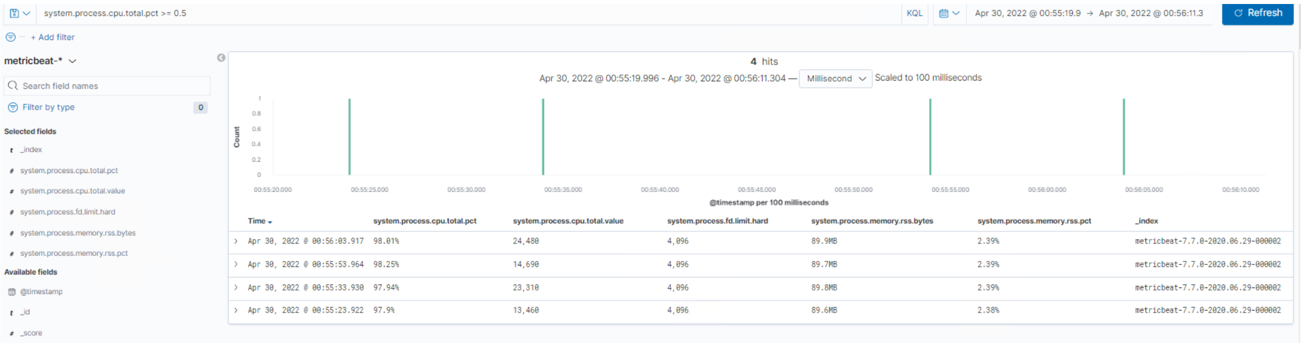
Name of Alert 3 – CPU Usage Monitor


watch_id : 6038f4aa-e0f6-498f-ab66-3381af8a5970
KQL
Apr 30, 2022 @ 01:06:28.14 → May 1, 2022 @ 12:57:32.16
Refresh
+ Add filter
.watcher-history-*
Search field names
Filter by type
Selected fields
_source
Available fields
_id
_index
_score
_type
condition.script.lang
condition.script.params.threshold
condition.script.source
input.search.request.body.aggs...
203 hits
Apr 30, 2022 @ 01:06:28.147 - May 1, 2022 @ 12:57:32.163 —
Auto
Count
trigger_event.triggered_time per 30 minutes
Time
_source

```
> Apr 30, 2022 @ 19:30:27.421  watch_id: 6038f4aa-e0f6-498f-ab66-3381af8a5970 node: PAfCrtSkTM0DGx3xpS3ug state: execution_not_needed status.state.active: true
                               status.state.timestamp: 2022-04-30T16:07:27.316Z status.last_checked: 2022-04-30T19:30:27.421Z status.actions.logging_1.ack.timestamp: 2022-04-30T16:07:27.316Z
                               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: Apr 30, 2022 @ 19:30:27.421 trigger_event.schedule.scheduled_time: Apr 30, 2022 @ 19:30:27.317
                               input.search.request.search_type: query_then_fetch input.search.request.indices: metricbeat-* input.search.request.rest_total_hits_as_int: true
> Apr 30, 2022 @ 19:29:27.393  watch_id: 6038f4aa-e0f6-498f-ab66-3381af8a5970 node: PAfCrtSkTM0DGx3xpS3ug state: execution_not_needed status.state.active: true
```

Alert 3 is implemented as follows:

- **Metric**: This monitoring rule watches the system.process.cpu.total.pct from metricbeat
- **Threshold**: It will fire when its max value remains above 0.5 over all processes for the last 5 minutes
- **Syntax:** WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- **Vulnerability Mitigated**: Denial of Service Attack, C2
- **Reliability**: This alert is low reliability because there are several things that could happen with a machine that would cause high cpu usage and be unrelated to a DDoS attack. However, in the events witnessed, it might make sense to increase the threshold to .85 or .9. It wouldn't have mattered in this example, but it might in normal operations.



system.process.cpu.total.pct >= 0.5

Apr 30, 2022 @ 00:55:19.9 → Apr 30, 2022 @ 00:56:11.3

metricbeat-*

4 hits

Apr 30, 2022 @ 00:55:19.996 - Apr 30, 2022 @ 00:56:11.304 — Millisecond Scaled to 100 milliseconds

Time	system.process.cpu.total.pct	system.process.cpu.total.value	system.process.fd.limit.hard	system.process.memory.rss.bytes	system.process.memory.rss.pct	_index
Apr 30, 2022 @ 00:56:03.917	98.01%	24,480	4,096	89.9MB	2.39%	metricbeat-7.7.0-2020.06.29-000002
Apr 30, 2022 @ 00:55:53.964	98.25%	14,690	4,096	89.7MB	2.39%	metricbeat-7.7.0-2020.06.29-000002
Apr 30, 2022 @ 00:55:33.930	97.94%	23,310	4,096	89.8MB	2.39%	metricbeat-7.7.0-2020.06.29-000002
Apr 30, 2022 @ 00:55:23.922	97.9%	13,460	4,096	89.6MB	2.38%	metricbeat-7.7.0-2020.06.29-000002

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 - Hardening Against Vulnerable Ports 22 and 80 on Target 1
  - Close port 22 and use port 443 with https instead of 80.
  - Port 22 will prevent open ssh access to the machine. Using port 443 will provide a layer of security using ssl instead of the open port.
    - Port 80 and 22 can be shut down with:
    - sudo ufw deny PORT 80
    - sudo ufw deny PORT 22
    - Sudo ufw allow PORT 443
    - Each command should be run one at a time and checked status with
    - sudo ufw status verbose

- Vulnerability 2 - Hardening Against Weak/Insecure Passwords on Target 1
  - Users should change passwords to a best practices format involving at least 16 characters, no dictionary words, special characters, numbers and symbols. 1 hour lock outs should be implemented after 5 unsuccessful attempts within 15 minutes. Multi-factor authentication should also be used.
    - Complex passwords are difficult to crack with brute force and lockouts will prevent multiple attempts. Additionally, notification alerts could be generated to further protect the accounts
    - How to install it (include commands) https://ostechnix.com/how-to-set-password-policies-in-linux/
- Vulnerability 3 - Hardening Against Python Privilege Escalation on Target 1

  - Python privileges should be removed for users vulnerable to ssh as well as users who are not authorized for root privileges.
    - Removing the python sudo privileges will eliminate the potential for circumventing access restrictions
    - vi /etc/sudoers
      - Delete this line: steven ALL=(ALL) NOPASSWD: /usr/bin/python

- Vulnerability 4 - Hardening Against Enumerate Wordpress Site on Target 1
  - Deploy the Ansible Playbook that updates the Wordpress site to a patched version with Stop User Enumeration plug-in and adjust firewall to block similar behaviors of enumerating traffic
    - Updated versions Wordpress won't allow enumeration with appropriate plugins
    - Run the ansible playbook discussed in the link (https://github.com/jacobsstarks/Final-Project-Rice-Cybersecurity/blob/main/WPandApache.yml) and make sure Stop User Enumeration plug-in is installed and enabled
    - https://wordpress.org/plugins/stop-user-enumeration/

- Vulnerability 5 - Hardening Against Apache 2.4.1 CVE-2016-4975 on Target 1
  - Regularly update Apache server to latest stable version
  - Apache tends to have significant vulnerabilities with every version. To keep ahead of these threats, it is important to maintain a consistent approach to upgrading the versions
    - Run the ansible playbook found here (https://github.com/jacobsstarks/Final-Project-Rice-Cybersecurity/blob/main/WPandApache.yml) which provides the automation for updating both the wordpress and apache files to the latest and least-known vulnerabilities
    - This should be run on a recurring basis