Blue Team: Summary of Operations

Table of Contents

Network Topology

TODO: Fill out the information below.

The following machines were identified on the network:

MS Azure (Jump Box)

- Operating System: Windows 10 pro
- Purpose: Hosts all the other Virtual Machines
- IP Address: 192.168.1.1
- MAC Address: 00:15:5d:00:04:0d

Kali

- Operating System: Linux
- Purpose: Attacking Box acting as an outside agent
- IP Address: 192.168.1.90
- MAC Address: 00:15:5D:00:04:12

Target 1

- Operating System: Linux 3.2 – 4.9
- Purpose: Apache Web Server to Host Raven Security Website
- IP Address: 192.168.1.110
- MAC Address: 00:15:5D:00:04:10

Target 2

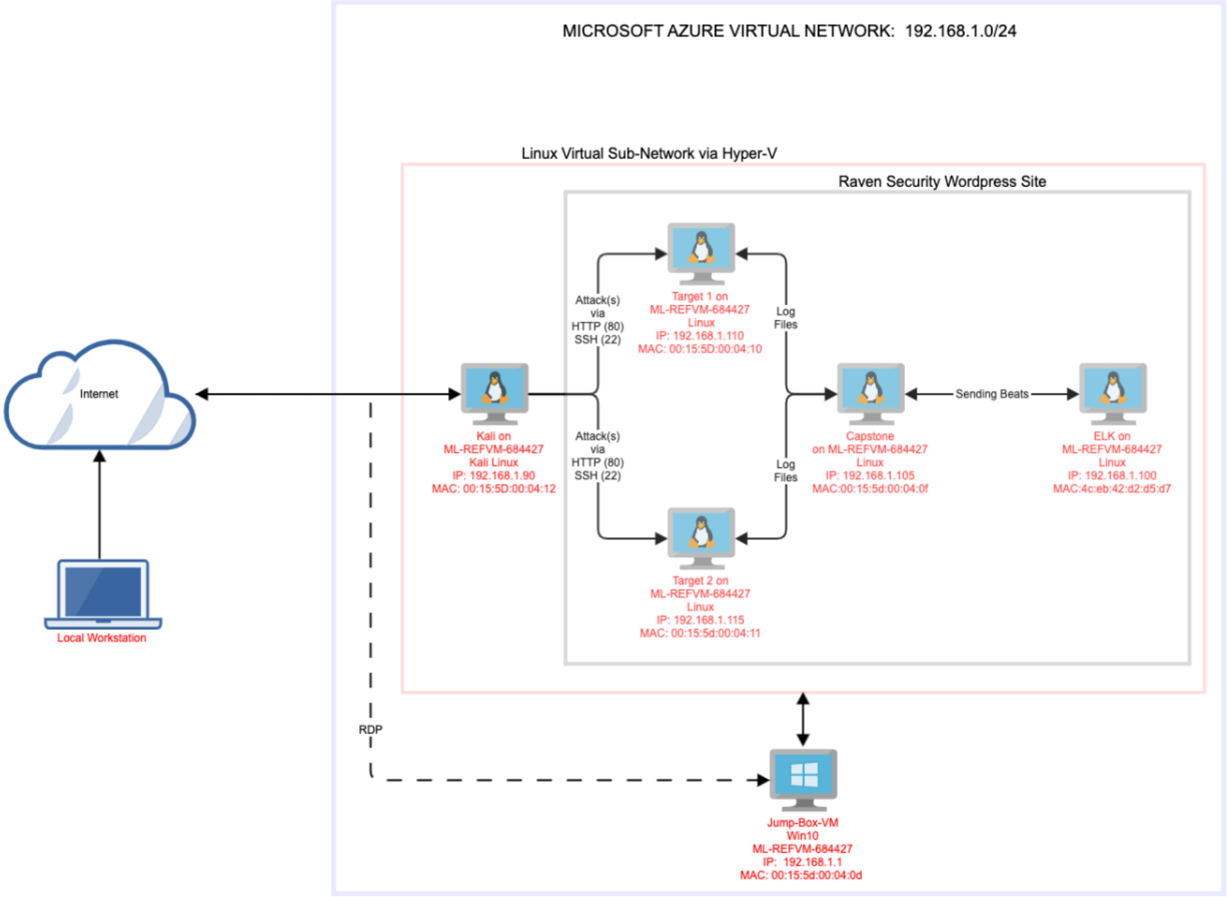
- Operating System: Linux 3.2 – 4.9
- Purpose: Apache Web Server to Host a Second Raven Security Website
- IP Address: 192.168.1.115
- MAC Address: 00:15:5d:00:04:11

Capstone

- Operating System: Linux
- Purpose: For collecting log files from the webserver and then picking select logs (i.e. beats) to send to the ELK server
- IP Address: 192.168.1.105
- MAC Address: 00:15:5d:00:04:0f

ELK

- Operating System: Linux
- Purpose: To be the database on the network that holds the selected “beats” log that inform our network alerts and host the Kibana dashboard
- IP Address: 192.168.1.100
- MAC Address: 4c:eb:42:d2:d5:d7


MICROSOFT AZURE VIRTUAL NETWORK: 192.168.1.0/24
Linux Virtual Sub-Network via Hyper-V
Raven Security Wordpress Site
Target 1 on ML-REFVM-684427 Linux IP: 192.168.1.110 MAC: 00:15:5D:00:04:10
Attack(s) via HTTP (80) SSH (22)
Log Files
Internet
Kali on ML-REFVM-684427 Kali Linux IP: 192.168.1.90 MAC: 00:15:5D:00:04:12
Attack(s) via HTTP (80) SSH (22)
Log Files
Capstone on ML-REFVM-684427 Linux IP: 192.168.1.105 MAC:00:15:5d:00:04:0f
Sending Beats
ELK on ML-REFVM-684427 Linux IP: 192.168.1.100 MAC:4c:eb:42:d2:d5:d7
Target 2 on ML-REFVM-684427 Linux IP: 192.168.1.115 MAC: 00:15:5d:00:04:11
Local Workstation
RDP
Jump-Box-VM Win10 ML-REFVM-684427 IP: 192.168.1.1 MAC: 00:15:5d:00:04:0d

Description of Targets

- We have two front end apache servers, Target #1(192.168.1.110) and Target 2 (192.168.1.115), on the Virtual Network. The purpose of these servers are to host or display the raven security websites and style them in readable formats. They are also responsible for receiving and responding to HTTP requests.
- We have one backend server (192.168.1.105) which is called our Capstone Box. The purpose of this server is to execute business logic and read/write corresponding data to and from a database. In this case it is specifically collecting "Beats" which are special-purpose data collection modules. that allow us to only collect very specific information rather than collecting all a machine's log data.
    - Filebeat collects data about the file system (very specific files, such as those generated by Apache and MySQL databases). it must be installed on the VMs you want to monitor.
    - Metricbeat collects machine metrics, such as uptime.
    - Note *ELK officially supports eight Beats*
- We have one database server (192.168.1.100) which is called ELK. ELK is an acronym. Each letter stands for an open-source technology (E: Elasticsearch / L: Logstash / K: Kibana)
    - Elasticsearch - search and analytics engine
    - Logstash - Server-side data processing pipeline that sends data Elasticsearch
    - Kibana - Tool for visualizing Elasticsearch data with charts and graphs

Monitoring the Targets

Our scan(s) identified the services below as potential points of entry:

- **Target 1 (IP Address: 192.168.1.110 / MAC Address: 00:15:5D:00:04:10)**
    - Open SSH (6.1p1 Debian, Protocol 2.0) / Open port 22 /
    - HTTP (Apache v. 2.4.10 Debian) / Open port 80
    - RPCbind (Version 2 - 4, RPC#100000) / Open port 111
    - Netbios-ssn Service (Samba version 3.x - 4.x) / Open port 139
    - Microsoft-ds Service (Samba version 3.x - 4.x) / Open port 445

- **Target 2 (IP Address: 192.168.1.115 / MAC Address: 00:15:5d:00:04:11)**
    - Open SSH (6.1p1 Debian, Protocol 2.0) / Open port 22 /
    - HTTP (Apache v. 2.4.10 Debian) / Open port 80
    - RPCbind (Version 2 - 4, RPC#100000) / Open port 111

- ○ Netbios-ssn Service (Samba version 3.x - 4.x) / Open port 139
- ○ Microsoft-ds Service (Samba version 3.x - 4.x) / Open port 445

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

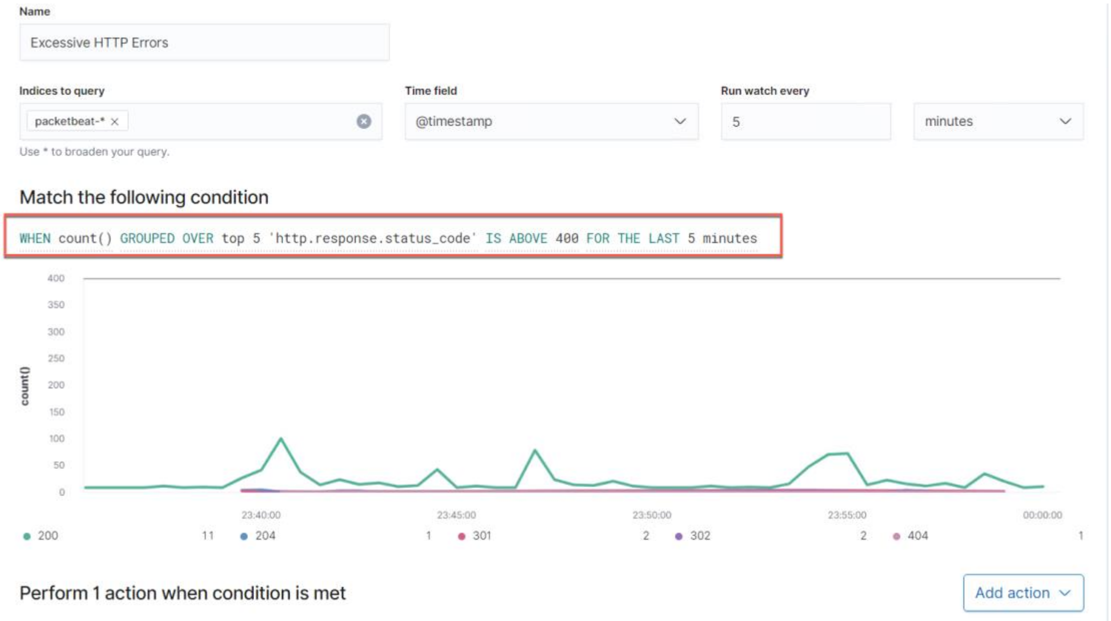
The Watcher overview page lists our watches and includes details such as state, last fired, and last triggered. A watch has one of four states:

Firing. The watch is triggered and actively performs the associated actions.
Error. The watch is not working properly.
OK. The watch is not actively firing but working properly.
Disabled. The watch will not fire under any circumstances.

Excessive HTTP Errors

Excessive HTTP Errors is implemented as follows:

- Metric: packetbeat
- Threshold: When a HTTP Status Code (i.e. 200-204-30-302-404) fires more than 400 times in a 5 minute span
- Vulnerability Mitigated: Bad Request
- Reliability: TODO: Does this alert generate lots of false positives/false negatives? Rate as low, medium, or high reliability.



Name
Excessive HTTP Errors

Indices to query
packetbeat-* ×
Use * to broaden your query.

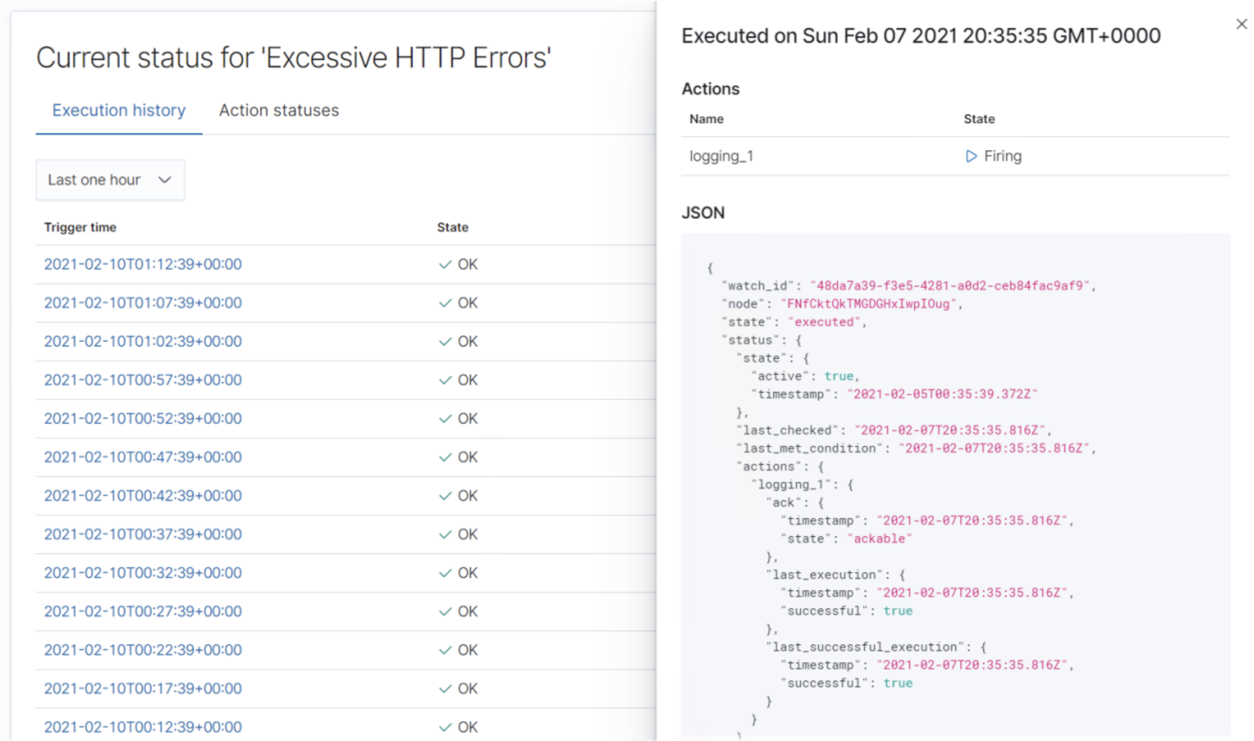
Time field
@timestamp

Run watch every
5 minutes

Match the following condition

```
WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
```

Perform 1 action when condition is met



Current status for 'Excessive HTTP Errors'

Execution history | Action statuses

Last one hour

Trigger time	State
2021-02-10T01:12:39+00:00	✓ OK
2021-02-10T01:07:39+00:00	✓ OK
2021-02-10T01:02:39+00:00	✓ OK
2021-02-10T00:57:39+00:00	✓ OK
2021-02-10T00:52:39+00:00	✓ OK
2021-02-10T00:47:39+00:00	✓ OK
2021-02-10T00:42:39+00:00	✓ OK
2021-02-10T00:37:39+00:00	✓ OK
2021-02-10T00:32:39+00:00	✓ OK
2021-02-10T00:27:39+00:00	✓ OK
2021-02-10T00:22:39+00:00	✓ OK
2021-02-10T00:17:39+00:00	✓ OK
2021-02-10T00:12:39+00:00	✓ OK

Executed on Sun Feb 07 2021 20:35:35 GMT+0000

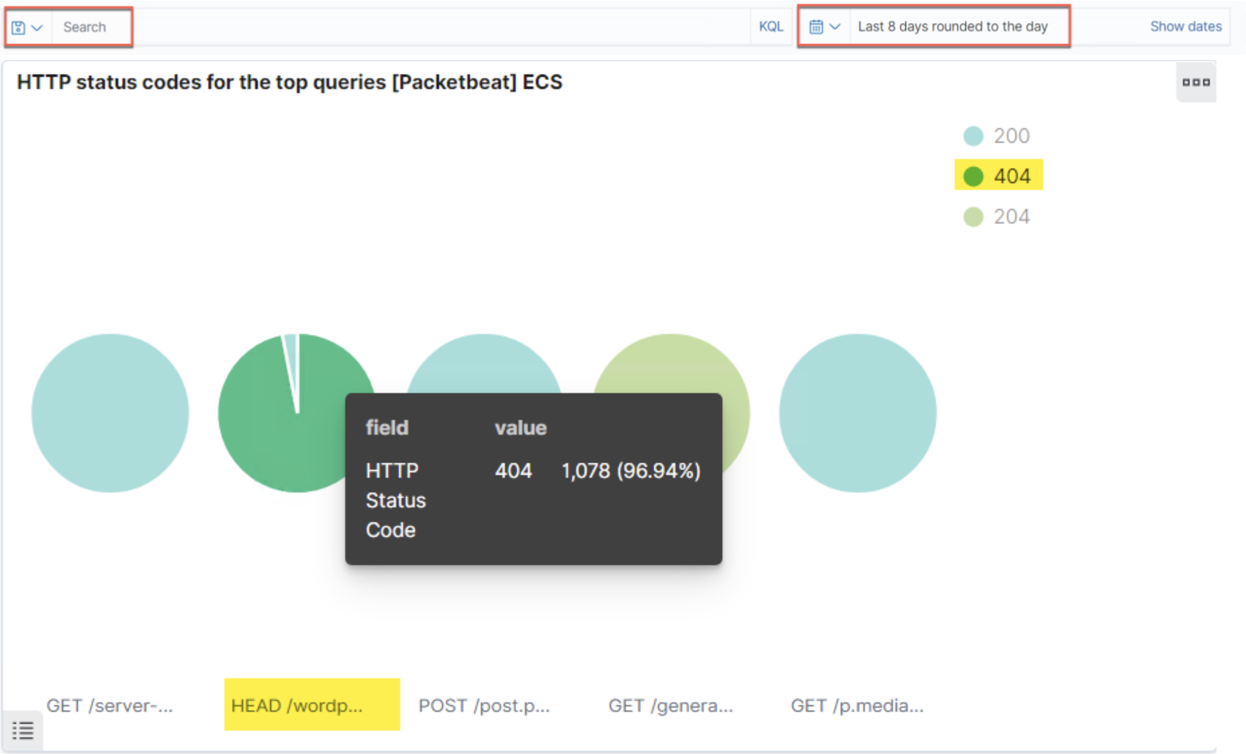
×

Actions

Name	State
logging_1	▷ Firing

JSON

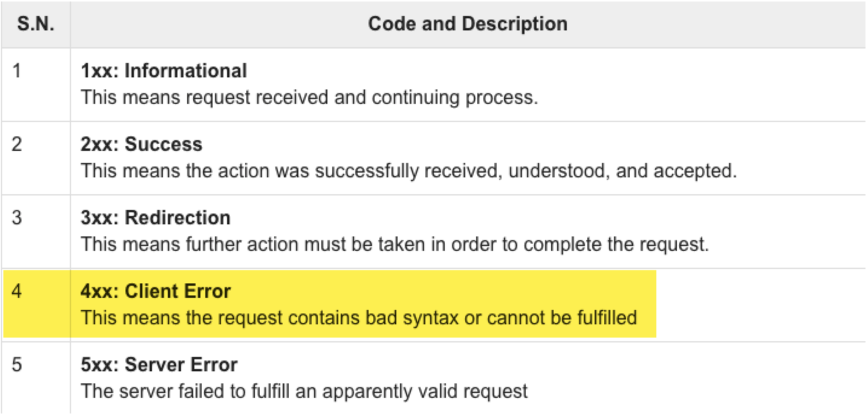
```
{
  "watch_id": "48da7a39-f3e5-4281-a0d2-ceb84fac9af9",
  "node": "FNfCktQkTMGDGHxIwpIOug",
  "state": "executed",
  "status": {
    "state": {
      "active": true,
      "timestamp": "2021-02-05T00:35:39.372Z"
    },
    "last_checked": "2021-02-07T20:35:35.816Z",
    "last_met_condition": "2021-02-07T20:35:35.816Z",
    "actions": {
      "logging_1": {
        "ack": {
          "timestamp": "2021-02-07T20:35:35.816Z",
          "state": "ackable"
        },
        "last_execution": {
          "timestamp": "2021-02-07T20:35:35.816Z",
          "successful": true
        },
        "last_successful_execution": {
          "timestamp": "2021-02-07T20:35:35.816Z",
          "successful": true
        }
      }
    }
```



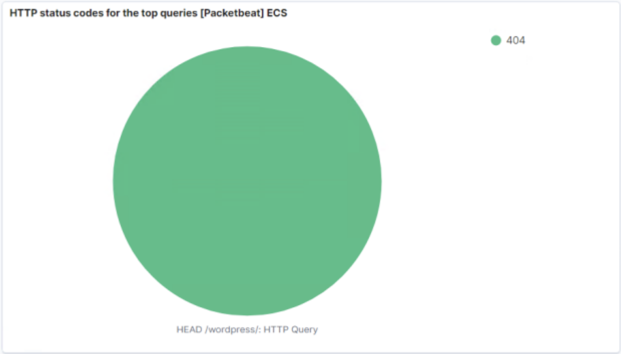
Search | KQL | Last 8 days rounded to the day | Show dates

HTTP status codes for the top queries [Packetbeat] ECS

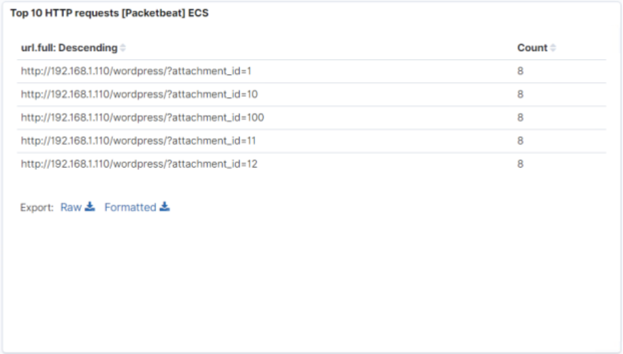
field	value	
HTTP Status Code	404	1,078 (96.94%)



S.N.	Code and Description
1	**1xx: Informational** This means request received and continuing process.
2	**2xx: Success** This means the action was successfully received, understood, and accepted.
3	**3xx: Redirection** This means further action must be taken in order to complete the request.
4	**4xx: Client Error** This means the request contains bad syntax or cannot be fulfilled
5	**5xx: Server Error** The server failed to fulfill an apparently valid request



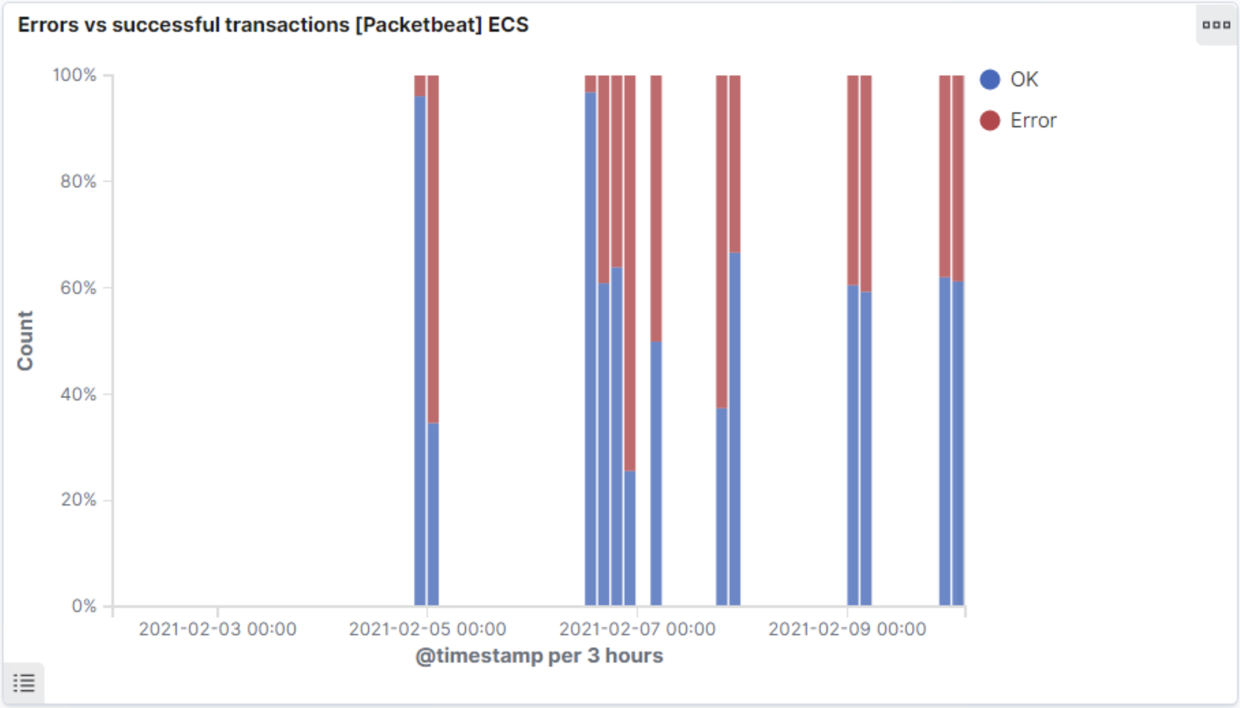
HTTP status codes for the top queries [Packetbeat] ECS



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.110/wordpress/?attachment_id=1	8
http://192.168.1.110/wordpress/?attachment_id=10	8
http://192.168.1.110/wordpress/?attachment_id=100	8
http://192.168.1.110/wordpress/?attachment_id=11	8
http://192.168.1.110/wordpress/?attachment_id=12	8

Export: Raw Formatted

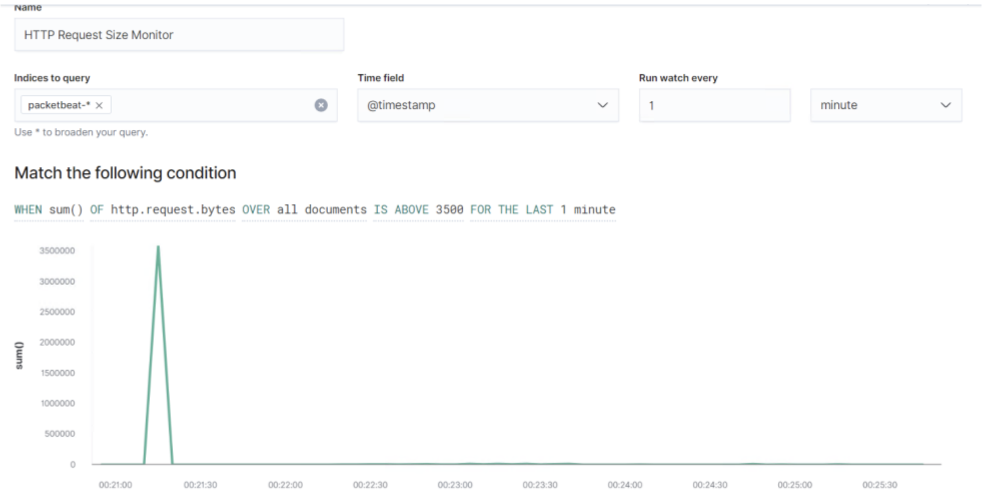


Errors vs successful transactions [Packetbeat] ECS

HTTP Request Size Monitor

HTTP Request Size Monitor is implemented as follows:

- Metric: packetbeat
- Threshold: When the sum of bytes for any document is above 3500 in the last 1 minute
- Vulnerability Mitigated: HTTP Request Smuggling
- Reliability: Does this alert generate lots of false positives/false negatives? Rate as low, medium, or high reliability. Reliability is high on this one. We also observe that our monitor is a state of “firing” for the majority of the 1 minute windows that are measured.



Name

HTTP Request Size Monitor

Indices to query

packetbeat-* ×

Use * to broaden your query.

Time field

@timestamp

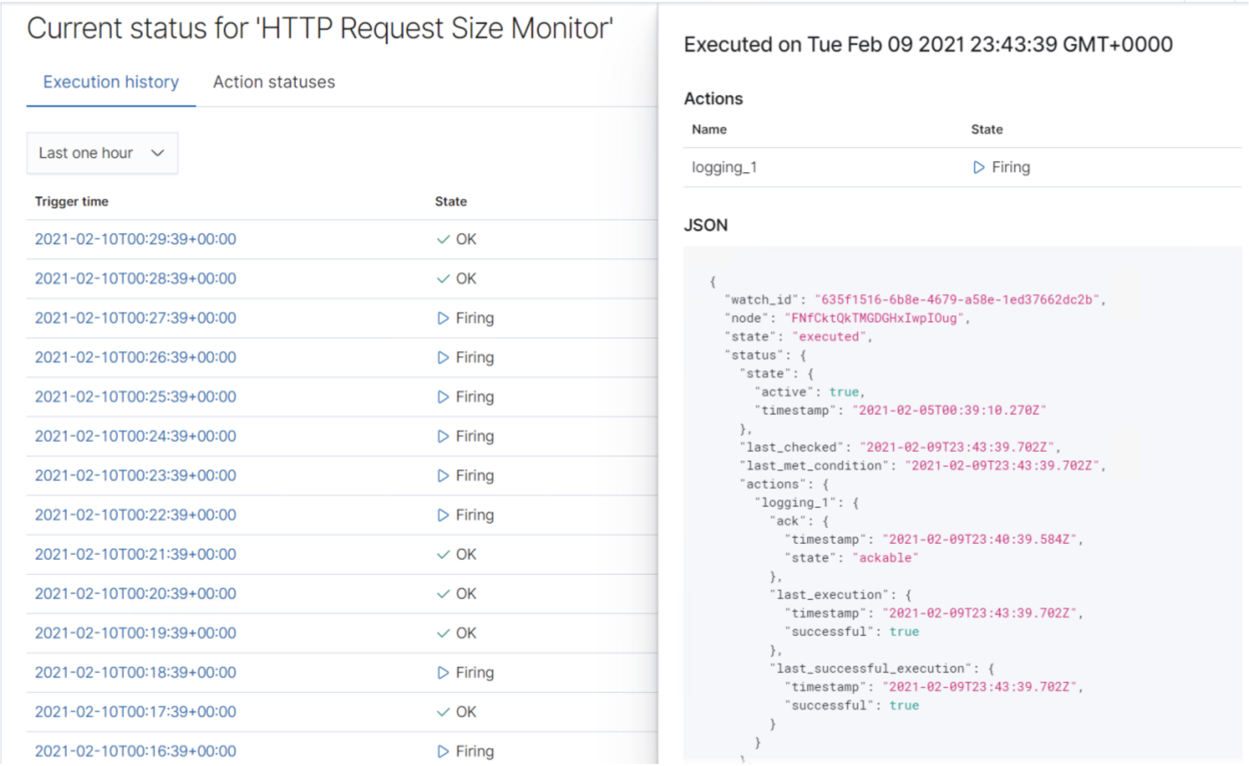
Run watch every

1

minute

Match the following condition

```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```



Current status for 'HTTP Request Size Monitor'

Trigger time	State
2021-02-10T00:29:39+00:00	✓ OK
2021-02-10T00:28:39+00:00	✓ OK
2021-02-10T00:27:39+00:00	▷ Firing
2021-02-10T00:26:39+00:00	▷ Firing
2021-02-10T00:25:39+00:00	▷ Firing
2021-02-10T00:24:39+00:00	▷ Firing
2021-02-10T00:23:39+00:00	▷ Firing
2021-02-10T00:22:39+00:00	▷ Firing
2021-02-10T00:21:39+00:00	✓ OK
2021-02-10T00:20:39+00:00	✓ OK
2021-02-10T00:19:39+00:00	✓ OK
2021-02-10T00:18:39+00:00	▷ Firing
2021-02-10T00:17:39+00:00	✓ OK
2021-02-10T00:16:39+00:00	▷ Firing

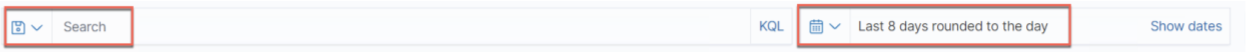
Executed on Tue Feb 09 2021 23:43:39 GMT+0000

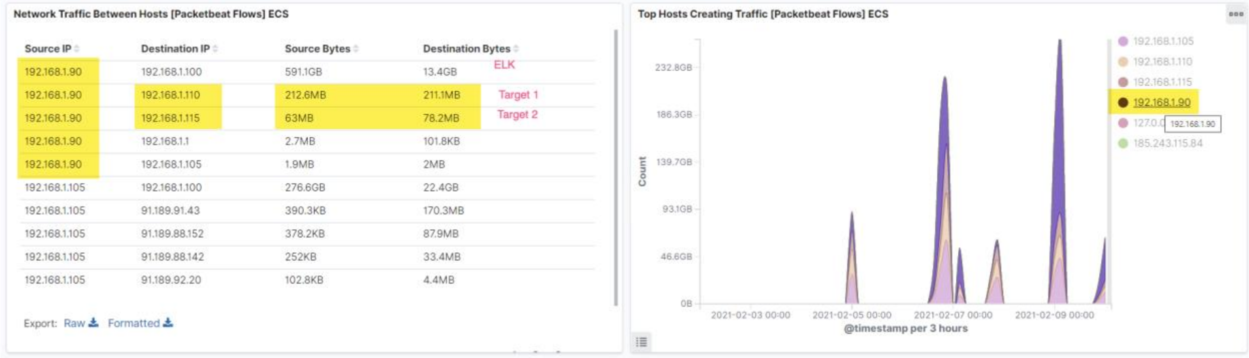
Actions

Name	State
logging_1	▷ Firing

JSON

```
{
  "watch_id": "635f1516-6b8e-4679-a58e-1ed37662dc2b",
  "node": "FNfCktQkTMGDGHxIwpIOug",
  "state": "executed",
  "status": {
    "state": {
      "active": true,
      "timestamp": "2021-02-05T00:39:10.270Z"
    },
    "last_checked": "2021-02-09T23:43:39.702Z",
    "last_met_condition": "2021-02-09T23:43:39.702Z",
    "actions": {
      "logging_1": {
        "ack": {
          "timestamp": "2021-02-09T23:40:39.584Z",
          "state": "ackable"
        },
        "last_execution": {
          "timestamp": "2021-02-09T23:43:39.702Z",
          "successful": true
        },
        "last_successful_execution": {
          "timestamp": "2021-02-09T23:43:39.702Z",
          "successful": true
        }
      }
    }
```





Network Traffic Between Hosts [Packetbeat Flows] ECS

Source IP	Destination IP	Source Bytes	Destination Bytes	
192.168.1.90	192.168.1.100	591.1GB	13.4GB	ELK
192.168.1.90	192.168.1.110	212.6MB	211.1MB	Target 1
192.168.1.90	192.168.1.115	63MB	78.2MB	Target 2
192.168.1.90	192.168.1.1	2.7MB	101.8KB	
192.168.1.90	192.168.1.105	1.9MB	2MB	
192.168.1.105	192.168.1.100	276.6GB	22.4GB	
192.168.1.105	91.189.91.43	390.3KB	170.3MB	
192.168.1.105	91.189.88.152	378.2KB	87.9MB	
192.168.1.105	91.189.88.142	252KB	33.4MB	
192.168.1.105	91.189.92.20	102.8KB	4.4MB	

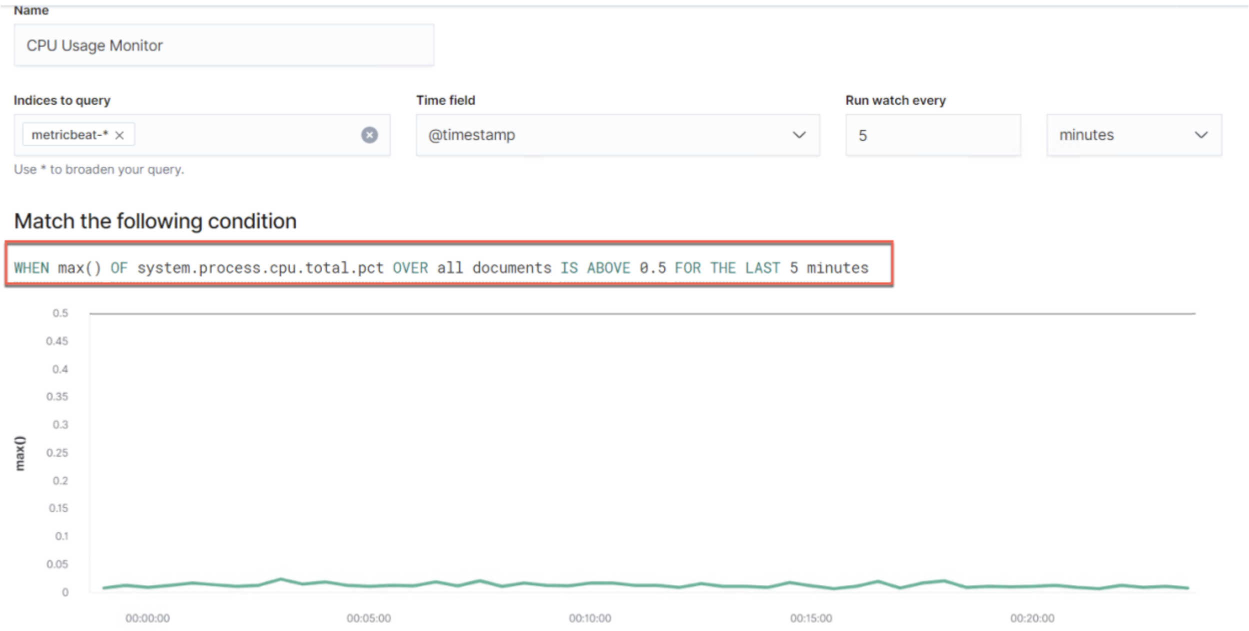
Export: Raw Formatted

Top Hosts Creating Traffic [Packetbeat Flows] ECS

CPU Usage Monitor

CPU Usage Monitor is implemented as follows:

- Metric: metricbeat
- Threshold: When the max of any system process is above 0.5 in the last 5 minutes
- Vulnerability Mitigated: Monitors the cpu usage of the webserver to determine if it is under a Distributed denial of service attack (DDos)
- Reliability: TODO: Does this alert generate lots of false positives/false negatives? Rate as low, medium, or high reliability. There’s no false positives/false negatives. The reason might be that the threshold is too high. The recommendation would be to set the threshold lower at .2 or .25



Name

CPU Usage Monitor

Indices to query: metricbeat-*

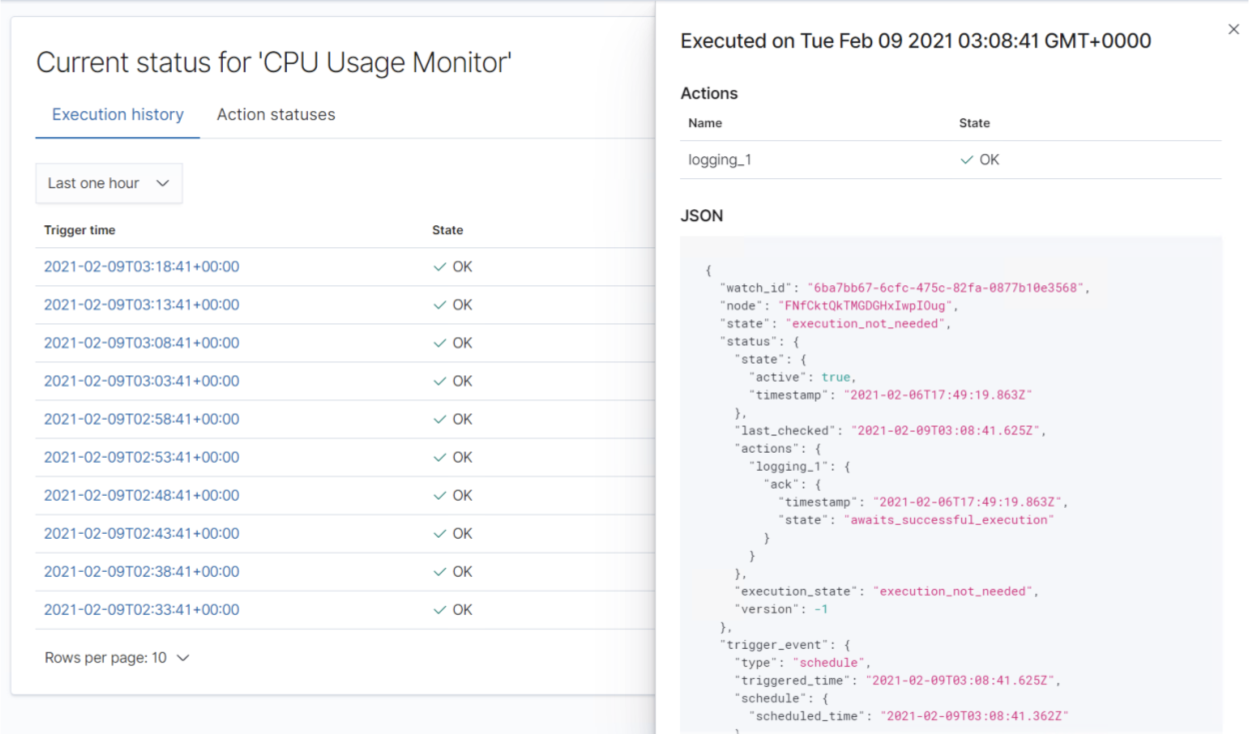
Use * to broaden your query.

Time field: @timestamp

Run watch every: 5 minutes

Match the following condition

```
WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
```



Current status for 'CPU Usage Monitor'

Trigger time	State
2021-02-09T03:18:41+00:00	✓ OK
2021-02-09T03:13:41+00:00	✓ OK
2021-02-09T03:08:41+00:00	✓ OK
2021-02-09T03:03:41+00:00	✓ OK
2021-02-09T02:58:41+00:00	✓ OK
2021-02-09T02:53:41+00:00	✓ OK
2021-02-09T02:48:41+00:00	✓ OK
2021-02-09T02:43:41+00:00	✓ OK
2021-02-09T02:38:41+00:00	✓ OK
2021-02-09T02:33:41+00:00	✓ OK

Executed on Tue Feb 09 2021 03:08:41 GMT+0000

Actions

Name	State
logging_1	✓ OK

JSON

```
{
  "watch_id": "6ba7bb67-6cfc-475c-82fa-0877b10e3568",
  "node": "FNfCktQkTMGDGHxIwpIOug",
  "state": "execution_not_needed",
  "status": {
    "state": {
      "active": true,
      "timestamp": "2021-02-06T17:49:19.863Z"
    },
    "last_checked": "2021-02-09T03:08:41.625Z",
    "actions": {
      "logging_1": {
        "ack": {
          "timestamp": "2021-02-06T17:49:19.863Z",
          "state": "awaits_successful_execution"
        }
      }
    },
    "execution_state": "execution_not_needed",
    "version": -1
  },
  "trigger_event": {
    "type": "schedule",
    "triggered_time": "2021-02-09T03:08:41.625Z",
    "schedule": {
      "scheduled_time": "2021-02-09T03:08:41.362Z"
```

Suggestions for Going Further

Suggest a patch for each vulnerability identified by the alerts above. Remember: alerts only detect malicious behavior. They do not prevent it. 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. 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 Vulnerability associated with Excessive HTTP Errors: Denial of Service

- Patch: Thorough application Backup of system and Database before attempting any fixes.
- special-security-package with apt-get Upgrade to the latest version.
- Why It Works: [For example: special-security-package scans the system for viruses every day]

Vulnerability 2 Vulnerability associated with HTTP Request Size Monitor

- Patch:
- Why It Works:

Vulnerability 3 Vulnerability associated with Excessive CPU Usage Denial of Service
CVE-2019-9517

- Patch:
- Why It Works:

Vulnerability 4 Weak Passwords

1. Make the password on each VM expire after a set time, so users will have to reset it
2. We will set the length, number and character requirements for a password on the network:
    a. The only tools we need are the command “chage” (to change when a password expires and the command “nano” (to edit the settings inside the /etc/security/pwquality.conf file)
    b. Recommendation(s)
        i. Set the maximum number of days between password changes to 90
        ii. Require passwords to have a minimum of 16 characters
        iii. Require a passwords to have at least two numbers - upper case letters - lower case letters - two special characters
        iv. Require new passwords to differ from an old password
        v. Force immediate password expiration for all users on the server

Vulnerability 5 Unauthorized SSH Access

1. We will implement a technical security control to harden SSH access by editing the sshd configuration file:
    a. **Change the default TCP port** where SSH daemon is listening. An example is changing it from port 22 to something much higher like 24596. We want to ensure we do not use a port number that is easy to guess, such as 222, 2222 or 22222
    b. **Use SSH key pairs for authentication** to enable passwordless SSH login. It is safer and also allows logging in without the need to use a password which is faster and more convenient.
    c. **Disable password-based logins** on the server. This eliminates the possibility of passwords getting cracked and used for login.
    d. **Disable root access to the server and utilize a service account** with the su - command to switch to a root user once properly authenticated and logged in.
    e. We can **restrict access to certain IP addresses or hostnames** via TCP wrappers and editing the /etc/hosts.allow and etc/hosts.deny files.

Vulnerability 6 Unfiltered ports

1. We will implement a network firewall which is a technical security control that would be placed in front of the target 1 and 2 apache web servers to block malicious internet traffic from entering the network. Thus providing the ability to:
    a. Intercept traffic before it reaches its target host or router
    b. Inspect the source / destination address/ports, TCP flags, and other features of the incoming packets.
    c. Only allow packets that come from trusted sources and deny packets that do not.
    d. Target a firewalld type because it would not require the disruption of services when implementing new services
        i. Uses zones to divide network interfaces into groups of shared trust level.
        ii. Zones are assigned set of rules depending on the needs and restrictions of each zones interface
    e. If company resources allow:
        i. Would implement a IDS or IPS system