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

TODO: Fill out the information below.

The following machines were identified on the network:

- Name of VM 1: TARGET1
  - **Operating System**: Linux 3.2 - 4.9
  - **Purpose**: WordPress Webserver
  - **IP Address**: 192.168.1.110
- Name of VM 2: TARGET2
  - **Operating System**: Linux 3.2 - 4.9
  - **Purpose**: WordPress Webserver
  - **IP Address**: 192.168.1.115
- Name of VM 3: Kibana
  - **Operating System**: Linux Ubuntu 18.4.1
  - **Purpose**: ELK Stack
  - **IP Address**: 192.168.1.100
- Name of VM 4: Capstone
  - **Operating System**: Linux 2.6
  - **Purpose**: Vulnerable Webserver
  - **IP Address**: 192.168.1.90
- Name of VM 5: Kali
  - **Operating System**: Debian Kali 5.4.0
  - **Purpose**: Penetration Tester
  - **IP Address**: 192.168.1.90

Description of Targets

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:

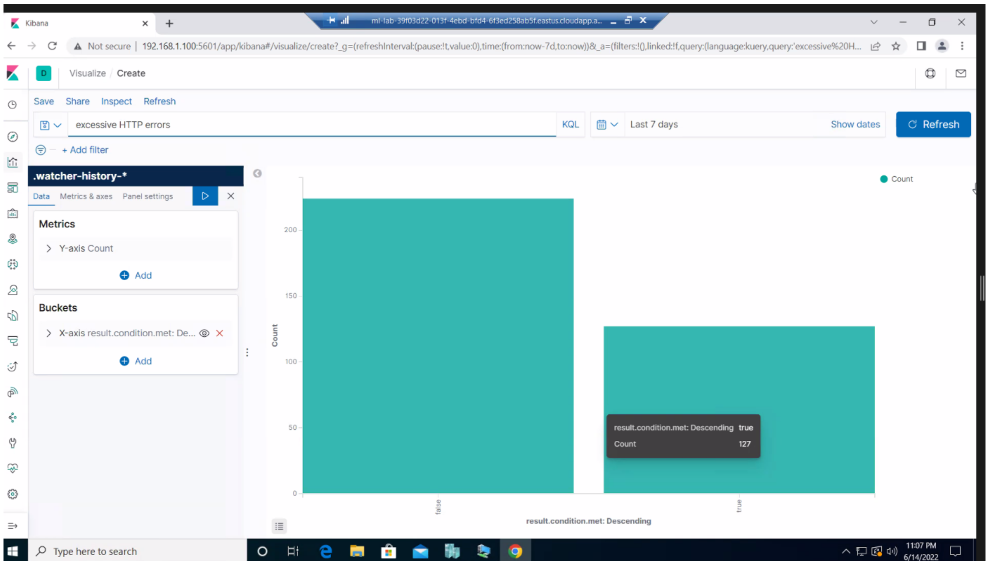
Monitoring the Target

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**: packetbeat
- **Threshold**: http.response.status_code above 400 for the last 5 minutes
- **Vulnerability Mitigated**: Brute Force/Enumeration
- **Reliability**: The alert provides high reliability. The 400+ will filter out any successful or normal responses. The 400 codes are known as client/server errors that are of concern. There will not be any false positives or negatives, as the alert is able to take a look at the status codes and log them accordingly (if they are above or below 400). However, to make it even more reliable we could trigger the alert every minute instead of every 5 minutes.



Not secure | 192.168.1.100:5601/app/kibana#/visualize/create?_g=(refreshInterval:(pause:!t,value:0),time:(from:now-7d,to:now))&_a=(filters:!(),linked:!f,query:(language:kuery,query:'excessive%20H...

excessive HTTP errors

Last 7 days

.watcher-history-*

Metrics

Y-axis Count

Buckets

X-axis result.condition.met: De...

Count

result.condition.met: Descending

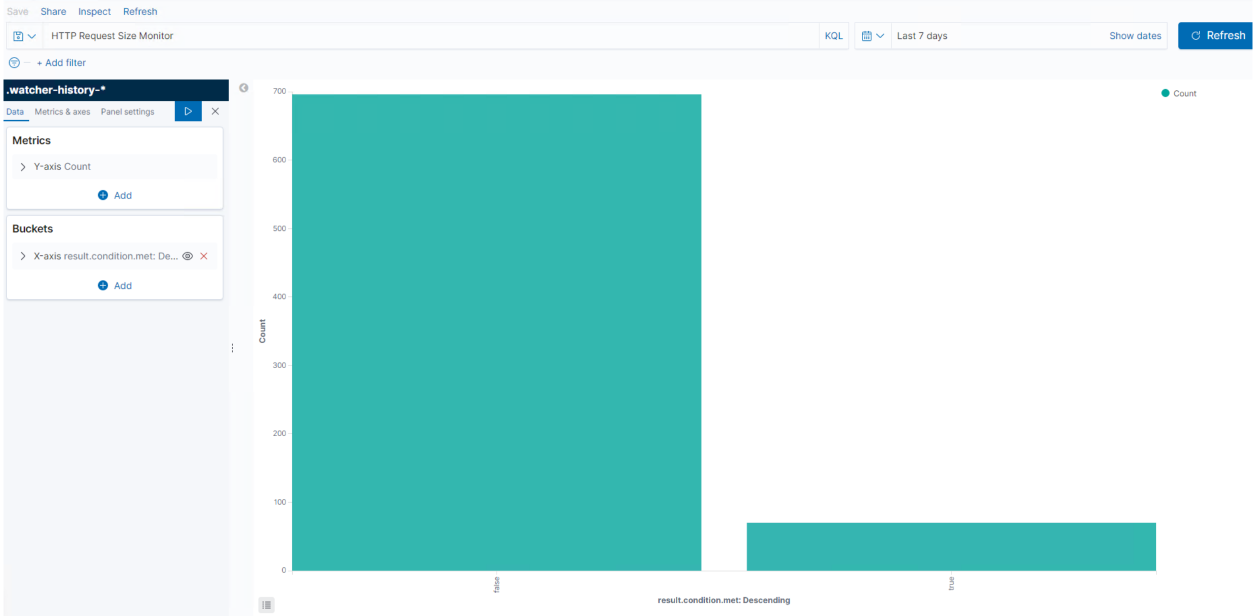
result.condition.met: Descending true
Count 127

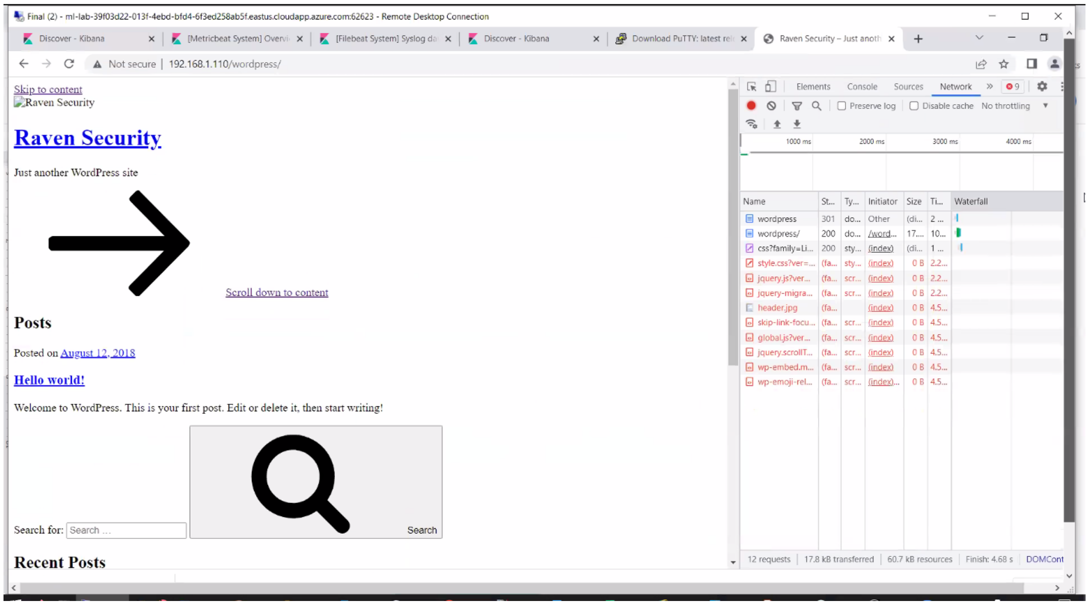
Name of Alert 2: HTTP Request Size Monitor

Alert 2 is implemented as follows:

- **Metric**: packetbeat
- **Threshold**: http.request.bytes above 3500 for the last minute
- **Vulnerability Mitigated**: DoS attack

- **Reliability**: This alert can generate lots of false positives, telling us that it has low reliability. It seems as though 3500 bytes is quite low for an http request, and it would flag a lot of “normal” traffic. Thus, it would be hard to locate when there is bad traffic (such as a DoS attack when there would be an absurdly large amount of bytes). In order to make this more reliable, I would increase the threshold to 5 MB.





Final (2) - ml-lab-39f03d22-013f-4ebd-bfd4-6f3ed258ab5f.eastus.cloudapp.azure.com:62623 - Remote Desktop Connection
Discover - Kibana
[Metricbeat System] Overvi
[Filebeat System] Syslog da
Discover - Kibana
Download PuTTY: latest rel
Raven Security – Just anoth
Not secure
192.168.1.110/wordpress/
Elements
Console
Sources
Network
9
1000 ms
2000 ms
3000 ms
4000 ms

Skip to content

Raven Security

Raven Security

Just another WordPress site

Scroll down to content

Posts

Posted on August 12, 2018

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Search for: Search ... Search

Recent Posts

□ Preserve log □ Disable cache No throttling

Name	St...	Ty...	Initiator	Size	Ti...	Waterfall
wordpress	301	do...	Other	(di...	2 ...	
wordpress/	200	do...	/word...	17....	10...	
css?family=Li...	200	sty...	(index)	(di...	1 ...	
style.css?ver=...	(fa...	sty...	(index)	0 B	2.2...	
jquery.js?ver...	(fa...	scr...	(index)	0 B	2.2...	
jquery-migra...	(fa...	scr...	(index)	0 B	2.2...	
header.jpg	(fa...		(index)	0 B	4.5...	
skip-link-focu...	(fa...	scr...	(index)	0 B	4.5...	
global.js?ver...	(fa...	scr...	(index)	0 B	4.5...	
jquery.scrollT...	(fa...	scr...	(index)	0 B	4.5...	
wp-embed.m...	(fa...	scr...	(index)	0 B	4.5...	
wp-emoji-rel...	(fa...	scr...	(index)...	0 B	4.5...	

12 requests | 17.8 kB transferred | 60.7 kB resources | Finish: 4.68 s | DOMCont

Name of Alert 3: CPU Usage Monitor

Alert 3 is implemented as follows:

- **Metric**: metricbeat

- **Threshold**: system.process.cpu.total.pct above 0.5 for the last 5 minutes
- **Vulnerability Mitigated**: Malware/Viruses using up space.
- **Reliability**: The alert provides high reliability. There should not be any inaccuracy, as the watcher is able to record the CPU usage. Thus, if a malicious program runs (that has a high CPU usage), the alert will be triggered. This alert has quite a high threshold, being that the average CPU usage was about 0.3%, so it is a reliable alert.