BIG DATA CYBERSECURITY

LAB 8 STREAMING DATA ENRICHMENT

STREAMING DATA ENRICHMENT

Lab Description: A recent study of big data cybersecurity analytics systems showed that research papers address topics of intrusion detection systems, alert correlation, forensic analysis, and detection of denial of service attacks, botnets, advanced persistent threat, malware and phishing. The study identified increase in use of streaming data processing due to the rapidly changing time-sensitive requirements of cybersecurity analytics systems [1].

This laboratory assignment demonstrates the use of Apache Metron for analysis of streaming data. This is commonly achieved by reading and streaming log files comprised of network traffic, firewall, web access, system activity, router access, database access data, etc. using tools from Apache stack such as Flink, Kafka, NiFi, Solr, Spark or Storm [1-3] Recent advances from several manufacturers, e.g. Arista, Cisco and Juniper, allow configuring network devices to stream certain telemetry data at particular intervals [4-6].

Cybersecurity streaming data are commonly produced by a sensor, which may be network devices, e.g. switches, routers or network taps, or event devices, e.g. firewalls, intrusion detection or prevention systems, or system log files [7, 8]. Certain sensors are capable of emitting streaming data whereas others require helper programs to read and stream stored data.

Sensors collect data about a network, which is used for monitoring and decision making. Network sensors collect data directly from network traffic while host-based sensors examine hosts and operating systems. Service sensors report an activity of a service, such as sending email or accessing Web resources. Logs commonly store debugging and troubleshooting information and have custom formats. Therefore, different types of log files need custom parsers capable of retrieving data of interest. Event-based sensors, e.g. intrusion detection systems (IDS) analyze data and create

events when certain conditions are met on the contrary to simple sensors, such as NetFlow, report all observed data. [8].

Bro recently renamed to Zeek is a passive network traffic analyzer capable of inspecting all network traffic on a link for suspicious activity and traffic analysis tasks not related to cybersecurity, e.g. performance measurement [9].

Metron is supplied with three sensors: metron-bro-plugin-kafka sending Bro (Zeek) log data to Kafka, fastcapa capable of capturing network data from a network interface and sending high-volume raw packet data to Kafka, and pycapa performing lightweight network packet capture [10].

Metron also comes with sensor stubs simulating a sensor behavior by sending sample data to Kafka topics. Three sensor stubs are supplied with sample data: Bro, Snort and YAF. Bro stub streams deep packet inspection (DPI) metadata—a complement for pcap packet capture [11]. Snort stub works with alerts produced by Snort—one of the most widely-used network intrusion prevention systems, which examines network traffic and produces signature-based alerts [12]. YAF stub ingests NetFlow data, which is a high-level summary of network flows contained in pcap files. Yet Another Flowmeter (YAF) tool is used to generate NetFlow data in IPFIX format from a pcap probe.

In this assignment, we will use Bro, Snort and YAF sensor stubs to send streaming data to Metron, inspect alerts and study a Metron dashboard.

Lab Files that are Needed: metron-cluster.ppk.

LAB EXERCISE/STEP 1

In the AWS cloud, create a new Metron virtual machine (VM) comprising a single-node Hadoop cluster as described in Lab 1. Establish an SSH connection with the VM as specified in Steps 1 through 4 of Lab 2. Note: it is recommended setting up a new VM to avoid misconfiguration because of the settings kept from other assignments.

Switch to superuser and change the current directory to /root.

```
sudo su
cd
```

At this step, we need to install Ansible, a tool for IT automation allowing to configure systems, deploy software and manage other more complex IT tasks [13]. We aim to execute an Ansible playbook in order to install and start sensor stubs, and start services for streaming sample data.

```
yum install ansible -y
```

Verify that Ansible installed successfully by executing the line below to print a list of Ansible commands

```
ansible -h
```

Make a screenshot and store it in the submission document.

LAB EXERCISE/STEP 2

Clone a repository with adjusted Ansible playbook for installing sensor stubs from GitHub

```
git clone https://github.com/arudniy/sensor_stubs_custom_install.git
```

List the directory contents to make sure it was cloned successfully. And change the directory.

```
ls
cd sensor_stubs_custom_install
```

Execute the Ansible playbook to copy sample data, create and start services to stream the sample data.

```
ansible-playbook install-service-centos-7.yml -v
```

Metron is supplied with Elasticsearch index templates for default sensors. On the other hand, when adding a new sensor, a corresponding index must be set up [14].

LAB EXERCISE/STEP 3

At this point, the newly created services will start streaming the sample Bro, Snort and YAF data to Kafka topics thus populating corresponding Elasticsearch indexes. Start a web browser and navigate to Ambari → Elasticsearch → Quick Links → Elasticsearch Indexes.

This screen should show one index for each of the sensor stubs: Bro, Snort and YAF. Refresh the screen by pressing the F5 button or in some other way. The documents count in each of these three indexes should increase. This means that

services are streaming data to the corresponding sensor stubs, which we have just installed.

Make a screenshot and store it in the submission document.

LAB EXERCISE/STEP 4

In a web browser, navigate to Ambari → Metron → Quick Links → Management UI. Use metron as a username and Smoothmetron2 as a password. Metron Management UI should show eight sensors in total and three running sensors: Bro, Snort and YAF. Values in the Latency and Throughput columns will be changing due to the random nature of streaming.

Make a screenshot and store it in the submission document.

LAB EXERCISE/STEP 5

In a web browser, navigate to Ambari → Metron → Quick Links → Alerts UI. This view will show a number of alerts. Some alerts will have a score value set in the Score column. A higher score value denotes a potentially higher risk and should attract analyst's attention.

Make a screenshot and store it in the submission document.

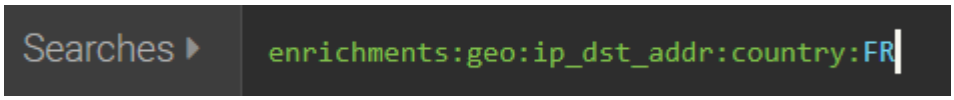
The Group By row is located above the table and contains four buttons. Click the source:type button. The table will be grouped by the source type showing the total number of alerts and the total score per source.

Click the source:type button once more to remove the filter and click the next button, ip_dst_addr. This will group the data, show the number of alerts and total score by IPv4. Click the same button once more to remove the filter and click the enrichment:country button. This action will group the data by country.

Make a screenshot and store it in the submission document.

Repeat the same action to group data by ip_src_addr. Finally, remove the filter.

Notice, that filters by value are available in the left panel. Using this panel, display the data with destination IP addresses belonging to France. Notice that the Searches bar now contains this filter:



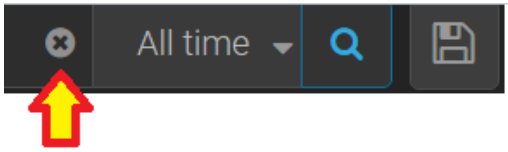
LAB EXERCISE/STEP 6

Copy one IP address and verify if it really belongs to France. To do so, type in your favorite online search engine keyword "whois *ipv4address*" substituting ipv4address with an actual IP address. A number of services are available which would provide with WhoIs data for an IP address for free. Open a web page showing the WhoIS data.

Make a screenshot and store it in the submission document.

LAB EXERCISE/STEP 7

Clean the search filter by clicking the circle with a cross on the right side of the search bar. Then click the magnifying glass.



Click the Actions button. Notice that it allows to Open, Dismiss, Escalate, Resolve or Add to Alert a record from the data table. Alerts can be grouped into a meta alert, which can be used to preserve the context of an investigation [14].

LAB EXERCISE/STEP 8

At this step, we will explore the Metron Dashboard, which needs to be installed first. To do so, in a web browser navigate to Ambari → Metron → Service Actions → Kibana Dashboard Install.

Open the dashboard by navigating to Ambari → Kibana → Quick Links → Metron UI → Dashboard → Metron-Dashboard. Read the dashboard description available at the top of the screen. The Metron dashboard is intended to facilitate in identifying, investigating and analyzing

cybersecurity data. In particular, it will display multiple visualization for the data sent to Metron sensors, which we explored in Metron Alerts UI.

The dashboard consists of several views. The top view shows the total number of events, an event count pie chart and a time series bar chart where each bar corresponds to a thirty-second interval.

Make a screenshot and store it in the submission document.

The view below shows a geographical map showing alert counts mapped using latitude and longitude. Next to it, the total number of geo-IP locations and a pie chart depicting events by country are located.

Make a screenshot and store it in the submission document.

The dashboard continues with views presenting YAF flows data, followed by Snort alerts, Web requests and DNS requests visualizations. Notice, that data tables have internal scrollbars and can be navigated page by page.

LAB EXERCISE/STEP 9

Be sure to terminate your AWS Metron VM to avoid budget depletion. Resources created in a cloud environment under your account have associated costs. AWS offers more than 60 products at a free tier with associated free tier usage limits [19].

PUZZLER

Study the Ansible playbook used in this assignment. Using its code, find out how data is streamed to Metron. Write a shell script to stream data for one of existing Metron sensors. Obtain data from one of online cybersecurity data repositories [15-18]. Start a sensor you selected and stream the data using a Kafka topic. Verify that a new Elasticsearch data is created and that alerts produced from the new data are displayed in Metron Alerts UI.

What to submit

Submit a Word (or other text editor) document with embedded screenshots made as requested in the assignment and a brief description for each screenshot.

References

[1] F. Ullah and M.A. Babar, "Architectural Tactics for Big Data Cybersecurity Analytic Systems: A Review," arXiv:1802.03178v1 [cs.CR], Feb. 2018.

[2] DZone, "What is data streaming?" [Online]. Available: https://dzone.com/articles/what-is-data-streaming. [Accessed Aug 8, 2019].

[3] P. Kanumarlapudi, "Building the Cybersecurity Data Pipeline Using Apache NiFi," [Online]. Available: https://medium.com/@pkmar437/building-the-cybersecurity-data-pipeline-using-apache-nifi-46843aec06b6. [Accessed Aug 8, 2019].

[4] Juniper Networks, "Understanding OpenConfig and gRPC on Junos Telemetry Interface," [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/concept/open-config-grpc-junos-telemetry-interface-understanding.html. [Accessed Aug 8, 2019].

[5] J. Cohoe, "Enterprise Streaming Telemetry and You: Getting Started with Model Driven Telemetry," *blogs.cisco.com*, Jul. 8, 2019. [Online]. Available: https://blogs.cisco.com/developer/getting-started-with-model-driven-telemetry. [Accessed Aug 8, 2019].

[6] Arista, "Arista NetDB Streaming Telemetry and Analytics: A High-Precision Alternative to Polling Technologies," [Online]. Available: https://www.arista.com/assets/data/pdf/EMA-Arista-0416-IB.pdf. [Accessed Aug 8, 2019].

[7] P. Stephenson, "SIEM," *SC Magazine*, vol. 25, no. 4, p. 36, 2014.

[8] M.S. Collins, *Network Security Through Data Analysis*, Sebastopol, CA: O'Reilly Media, 2014. [Ebook] Available: O'Reilly ebook.

[9] Zeek, "Docs Introduction," [Online]. Available: https://docs.zeek.org/en/stable/intro/index.html. [Accessed Aug 8, 2019].

[10] GitHub, "Metron Sensors," [Online]. Available: https://github.com/apache/metron/tree/master/metron-sensors. [Accessed Aug 8, 2019].

[11] Cloudera, "Administration. Bro." [Online]. Available: https://docs.cloudera.com/HDPDocuments/HCP1/HCP-1.4.1/bk_administration/content/bro_ingest_data_source.html. [Accessed Aug 8, 2019].

[12] Cloudera, "Administration. Snort." [Online]. Available: https://docs.cloudera.com/HDPDocuments/HCP1/HCP-

1.4.1/bk_administration/content/supported_datasources.html. [Accessed Aug 8, 2019].
[13] Ansible, "Ansible Documentation," [Online]. Available: https://docs.ansible.com/ansible/latest/index.html. [Accessed Aug 8, 2019].
[14] GitHub, "Apache Metron Indexing," [Online]. Available: https://github.com/apache/metron/blob/b9a130ca96774add38865d3934ed61ca19599b87/metron-platform/metron-indexing/metron-indexing-common/README.md. [Accessed Aug 8, 2019].
[15] SecRepo.com, "Samples of Security Related Data," [Online]. Available: https://www.secrepo.com/. [Accessed Aug 8, 2019].
[16] VizSec, "Data Sets," [Online]. Available: https://vizsec.org/data/. [Accessed Aug 8, 2019].
[17] GitHub, "Shramos / Awesome-Cybersecurity-Datasets," [Online]. Available: https://github.com/shramos/Awesome-Cybersecurity-Datasets/tree/a7284990437d0c754f50a4a4cddfe995e0bde218. [Accessed Aug 8, 2019].
[18] GitHub, "Jivoy / Awesome Machine Learning for Cybersecurity," [Online]. Available: https://github.com/jivoi/awesome-ml-for-cybersecurity. [Accessed Aug 8, 2019].
[19] AWS, "AWS Free Tier," [Online]. Available: https://aws.amazon.com/free/. [Accessed Aug 8, 2019].