BIG DATA CYBERSECURITY

LAB 3 DETECTING TYPOSQUATTING

DETECTING TYPOSQUATTING WITH APACHE METRON AND A SINGLE-NODE APACHE HADOOP CLUSTER

Lab Description:

Cybersquatting

In 1990s, a tremendous growth in internet commerce triggered a spike in new web domain registrations. A new phenomenon took place. Individuals known as "cybersquatters" began to register domain names resembling companies and brands names, which did not have online presence. The goal was to sell these domains back to companies and trademark owners at much higher prices [1].

One of the examples of cybersquatting is related to the TRUMP trademark owned by Donald Trump. In 2007, J. Taikwok Yung registered four web sites parodying Mr. Trump and TV shows with his involvement. The domain names—trumpbudhabi.com, trumpindia.com, trumpmumbai.com, and trumpbeijing.com—were registered when the Trump organization declared intentions to build condominiums and hotels in Bangalore and Mumbai, India. Attorneys representing Trump filed a suit against Yung in 2013 seeking $100,000 as a compensation for damages of federal cybersquatting for each of the domains based on the Anticybersquatting Consumer Protection Act. The court ruled that Mr. Yung must pay $32,000 damages to Mr. Trump [2, 3].

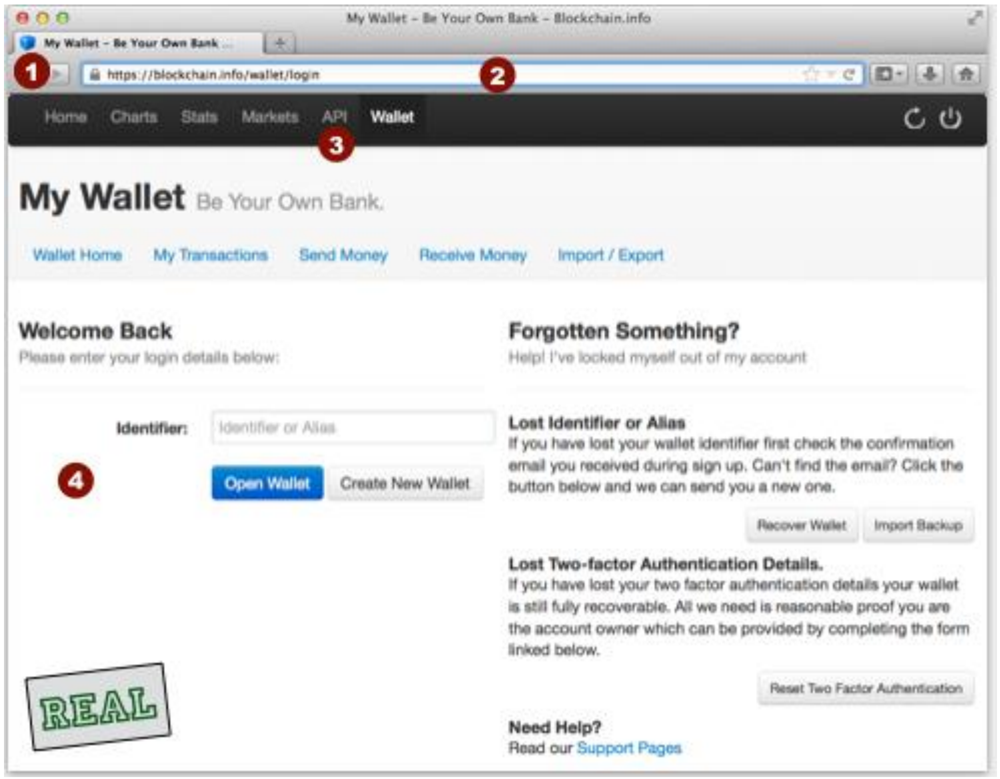
Typosquatting

Typosquatting is a use of domain names spelled closely to well-known ones in a malicious ways, such as for phishing or spear phishing attacks. A study conducted by Sophos [4], anti-malware and encryption products developer, analyzed a number of domain names generated by one-character mistakes for Facebook.com, Google.com, Twitter.com, Microsoft.com, Apple.com and Sophos.com. This resulted in 2,249 unique site names, such as pple.com, facemook.com, twitterz.com, etc. The study revealed that not all of the generated one-character-wrong domain names were registered. Several web sites from the produced list belonged to their rightful owners, e.g. racebook.com was a racing site and goole.com described a large port in England.

The study [4] revealed that approximately 3% of studied URLs were related to hacking, phishing, online fraud or spamming; 2% fell into the adult and dating category, 15% of links contained or were related to advertisement sites and popups, and 12% belonged to cybersquatting or domain parking category.

Another study produced by Sophos [5], described a case of a person who lost 16.50 of Bitcoin, which was worth $10,000 at then current prices. In a URL, instead of typing blockchain the fraud victim entered blockochain, and consequently a browser was redirected to a phishing website. Only four differences could be visually detected between the original and phishing sites: (1) Favicon was missing from the phishing site, (2) Differences in domain names and protocols in URLs, (3) Differences in menu items, and (4) The phishing site prompted a user to enter a password in the very beginning while the original site did not.

ORIGINAL WEBSITE



Welcome Back

Please enter your login details below:

Forgotten Something?

Help! I've locked myself out of my account

Lost Identifier or Alias

If you have lost your wallet identifier first check the confirmation email you received during sign up. Can't find the email? Click the button below and we can send you a new one.

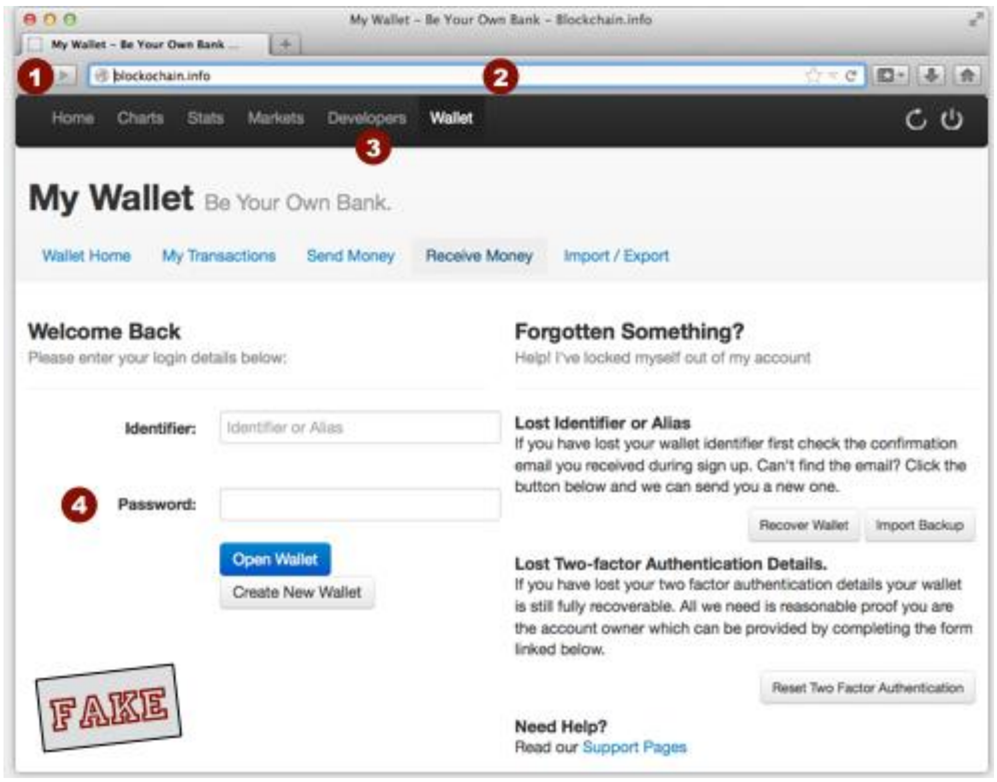
Lost Two-factor Authentication Details.

If you have lost your two factor authentication details your wallet is still fully recoverable. All we need is reasonable proof you are the account owner which can be provided by completing the form linked below.

Need Help?

Read our Support Pages

PHISHING WEBSITE


My Wallet – Be Your Own Bank – Blockchain.info
My Wallet – Be Your Own Bank ...
1
blockchain.info
2
Home
Charts
Stats
Markets
Developers
Wallet
3
4
FAKE

My Wallet Be Your Own Bank.

Wallet Home | My Transactions | Send Money | Receive Money | Import / Export

Welcome Back
Please enter your login details below:

Identifier: Identifier or Alias

Password:

Open Wallet
Create New Wallet

Forgotten Something?
Help! I've locked myself out of my account

Lost Identifier or Alias
If you have lost your wallet identifier first check the confirmation email you received during sign up. Can't find the email? Click the button below and we can send you a new one.

Recover Wallet | Import Backup

Lost Two-factor Authentication Details.
If you have lost your two factor authentication details your wallet is still fully recoverable. All we need is reasonable proof you are the account owner which can be provided by completing the form linked below.

Reset Two Factor Authentication

Need Help?
Read our Support Pages

Problem Statement

Generate summary data for domains from the Alexa Top Sites list [6] in a Bloom Filter and use it to detect potential typosquatting instances in proxy data. Apply Apache Metron to combat typosquatting using its flat file loader and typosquatting generation capabilityes. Generate a list of candidate typosquatted domain names, import it into an HBase table, and verify these candidate domain names with Metron's ENRICHMENT_EXISTS function [8].

BLOOM FILTER

Bloom Filter was introduced by Burton H. Bloom in his 1970 paper "Space/time trade-offs in hash coding with allowable errors." [7] The Bloom is commonly used to efficiently test whether an item belongs to a set with two possible outcomes: possibly in a set or definitely not in a set. Apache Metron allows to create, merge or modify bloom filters with the following functions:

- BLOOM_INIT(size, fpp) creates a Bloom filter to handle size number of elements with fpp probability of false positives (0 < fpp < 1).
- BLOOM_ADD(filter, object) adds an item to an existing Bloom filter.

- BLOOM_MERGE(filters) merges Bloom filters passed in a list data structure.

Proxy

When employed, a proxy device intercepts and sends requests from client computers on a network to web servers. Then it receives responses, such as web page data, from websites, and sends them back to the individual clients on the network.

Squid Proxy

Squid is a caching proxy for the Web supporting HTTP, HTTPS, FTP, and more. It reduces bandwidth and improves response times by caching and reusing frequently-requested web pages. Squid has extensive access controls and makes a great server accelerator. It runs on most available operating systems, including Windows and is licensed under the GNU GPL. [9]

Generating Typosquatting Domain with Metron

Apache Metron contains a capability for detecting typosquatting domain generation using strategies, such as listed below:

- Bitsquatting—when a malicious domain name is one bit different from the original one. Example [10]:

Binary representation of cnn.com:

01100011	01101110	01101110	0101110	01100011	01101111	01101101
C	n	n	.	c	o	m

Domain name with one bit difference from cnn.com:

01100011	01101111	01101110	0101110	01100011	01101111	01101101
C	o	n	.	c	o	m

- Homoglyphs. Substituting characters for ASCII or Unicode analogues which are visually similar, e.g. latimes.com vs. latlmes.com where character i is substituted with a vertical bar.

- Subdomains. Making part of the domain a subdomain, e.g. am.azon.com where the top-level domain name is azon.com and am.azon.com is a subdomain.
- Hyphenation. Inserting or removing hyphens.
- Insertion of characters, e.g. google.com vs. gooogle.com.
- Omission of characters, e.g. google.com vs. gogle.com.
- Replacement, e.g. google.com vs. gopgle.com.
- Transposition, e.g. google.com vs. gogole.com.
- In this assignment, we need to generate summary sketches from flat data and load it into HBase.

Apache HBase

Apache HBase is a NoSQL database, which operates within the Hadoop ecosystem. HBase is open-source non-relational distributed database, which originates from the Google Bigtable process. One of the main features of HBase is that it works in column-oriented fashion, grouping table columns into column families. Each column family is stored as a separate data file. More details about HBase are available in [11].

Elasticsearch

Elasticsearch is an open-source distributed search and analytics engine, which uses REST architecture. Elasticsearch is commonly used for log analytics, full-text search, and operational intelligence use cases. Elasticsearch can be connected with Kibana, a visualization tool, to provide near-real time analytics using large volumes of data. More details about Elasticsearch are available in [12].

Generating Summaries with Metron

Apache Metron has two utilities, flatfile_summarizer.sh and flatfile_loader.sh, which will be used for generating summaries [8] Useful flatfile_summarizer.sh parameters:

state_init for initializing a state object.

state_update for updating a state object.

state_merge – for merging a list of states.

Both flatfile_loader.sh and flatfile_summarizer.sh allow adjusting parallel processing settings by specifying the number of threads (-p option) and batch size (-b option), to specify an output destination (-o option) and the output mode (-om option). The list of output modes are:

LOCAL. Output to local disk (default).

CONSOLE. Write the object summarized to std out (terminal window). This is useful for visual analyzis of summary statistics about the data being imported, e.g. view the number of typosquatted domains generated for the Alexa Top Sites list.

HDFS. Write to a folder in the Hadoop Distributed File System.

Reading Summaries with Metron

Metron's built-in language, Stellar, allows to read data stored in HDFS using function OBJECT_GET(hdfs_path), which will also deserialize the data into an object for later use [8].

Example. The flatfile_summarizer.sh utility was used to write an object to /apps/metron/objects/alexa_10k_filter.ser file in HDFS. It is possible to read, deserialize this object and use the Bloom filter to determine if the domain goggle is a typosquatted domain with the following line of Stellar language:

```
BLOOM_EXISTS(
OBJECT_GET('/apps/metron/objects/alexa_10k_filter.ser'), 'goggle')
```

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 and set environment variables as specified in Steps 1 through 4 of Lab 2. Note: it is recommended setting up a new VM for each lab to avoid misconfiguration because of the settings kept from other assignments.

LAB EXERCISE/STEP 2

Pull Zookeeper Configuration

Zookeper is a coordinator within an Apache Hadoop cluster. ZooKeeper is a centralized service for maintaining configuration, naming, providing distributed synchronization, and providing group services in distributed applications. Zookeper service itself is distributed and highly reliable.

Pull the Zookeeper configuration information into a local folder using the following command (this must be typed as a single line):

```
sudo $METRON_HOME/bin/zk_load_configs.sh --mode PULL -z
$ZOOKEEPER -o $METRON_HOME/config/zookeeper/ -f
```

Make a screenshot and store it in the submission document.

LAB EXERCISE/STEP 3

Install Squid proxy.

Install Squid proxy by executing the command:

```
sudo yum install -y squid
```

Start the Squid proxy with this command:

```
sudo service squid start
```

Make a screenshot and store it in the submission document.

LAB EXERCISE/STEP 4

Retrieve Alexa Top Sites data.

Navigate to your user’s home directory:

```
cd ~
```

Download Alexa Top Sites Data:

```
wget http://s3.amazonaws.com/alexa-static/top-1m.csv.zip
```

Unzip the downloaded zip archive:

```
unzip top-1m.csv.zip
```

Copy data for top 10,000 web sites into a new file, top-10k.csv

```
head -n 10000 top-1m.csv > top-10k.csv
```

Verify the content of top-10k.csv by issuing this command:

```
vi top-10k.csv
```

Make a screenshot and store it in the submission document.

Since no changes were made to the file, exit the file using this command:

```
:q!
```

LAB EXERCISE/STEP 5

Use the Bloom filter to estimate the number of typosquatted domains.

Configuration of the Bloom filter requires us to estimates two numbers: the number of items in the Bloom filter and the probability of false positives. These numbers will be used to determine the size of the Bloom filter.

First, we will run flatfile_summarizer.sh in the CONSOLE output mode to display and count the number of typosquatted domain names across the entire document. This will provide us with the numbers for the next step.

Create a file ~/extractor_count.json, which will hold the extractor configuration:

```
vi ~/extractor_count.json
```

Press the i button on your keyboard to switch the vi editor to the Insert mode.

Copy and paste the following content:

```
{
 "config" : {
  "columns" : {
    "rank" : 0,
    "domain" : 1
  },
  "value_transform" : {
    "domain" : "DOMAIN_REMOVE_TLD(domain)"
  },
  "value_filter" : "LENGTH(domain) > 0",
  "state_init" : "0L",
  "state_update" : {
    "state" : "state + LENGTH( DOMAIN_TYPOSQUAT( domain ))"
              },
    "state_merge" : "REDUCE(states, (s, x) -> s + x, 0)",

    "separator" : ","

  },

  "extractor" : "CSV"

}
```

Press the Esc (Escape) button on your keyboard to exit the Insert mode. Then save the file using the following key combination:

```
:wq
```

Press Enter.

The extractor configuration uses the following properties:

• columns: the number of columns in the CSV data format. There are 2 columns, rank at the first position and domain at the second position.

• separator: commas are used to separate columns.

• value_transform: transform values in the domain column by removing top level domain data such as .com, .net, .org, etc.

• value_filter: discard empty values.

• state_init: initialize the state to 0. The 0L value means that 0 of the Long Integer type should be used.

• state_update: update the state with the number of typosquatted domains per domain

• state_merge: partial sums from several threads will be merged into the total.

Run the flatfile_summarizer.sh with the configuration file we created above:

```
sudo $METRON_HOME/bin/flatfile_summarizer.sh -i ~/top-10k.csv -e
~/extractor_count.json -p 5 -om CONSOLE/
```

Make a screenshot and store it in the submission document.

The second from the bottom line of the output should be "Processed 9999 - \"

And the bottom line should display 3496552, which is the total possible number of elements in the Bloom filter. We will use this number in the actual Bloom filter in the next step.

LAB EXERCISE/STEP 6

Generate the Bloom Filter

Now we can generate the Bloom filter to do the actual work. The extractor configuration file will be adapted from the previous step using the value we obtained (3496552). Then, we will use the extractor to generate the Bloom filter rather than the sum of the typosquatted domains.

The following settings are different from the previous step:

- state_init: This setting dictates to create a bloom filter initialized with two values
  - The size of 3496552 we calculated in the previous step
  - The false positive probability (0.001, which is 0.1%)
- state_update: Update the bloom filter (the state variable) with each typosquatted domain
- state_merge: Merge the bloom filters generated per thread into a final, single bloom filter.

Create a file ~/extractor_filter.json, which will hold new extractor configuration:

```
vi ~/extractor_filter.json
```

Press the i button on your keyboard to switch the vi editor to the Insert mode.

Copy and paste the following content:

```
{
 "config" : {
  "columns" : {
     "rank" : 0,
     "domain" : 1
  },
  "value_transform" : {
     "domain" : "DOMAIN_REMOVE_TLD(domain)"
  },
  "value_filter" : "LENGTH(domain) > 0",
  "state_init" : "BLOOM_INIT(3496552, 0.001)",
  "state_update" : {
     "state" : "REDUCE( DOMAIN_TYPOSQUAT( domain ), (s, x) ->
BLOOM_ADD(s, x), state)"
                  },
  "state_merge" : "BLOOM_MERGE(states)",
  "separator" : ","
 },
 "extractor" : "CSV"
}
```

Press the Esc (Escape) button on your keyboard to exit the Insert mode. Then save the file using the following key combination:

```
:wq
```

Press Enter.

Now we can use the Bloom filter in HDFS to store data in the /tmp/reference/alexa10k_filter.ser file. To do so execute this command:

```
sudo $METRON_HOME/bin/flatfile_summarizer.sh -i ~/top-10k.csv -o /tmp/reference/alexa10k_filter.ser -e ~/extractor_filter.json -p 5 -om HDFS
```

To ensure the Bloom filter functions as expected, execute commands to see if google, gogle, github, and gituub are included into the set of typosquatted domains. Note: google and github are the correct domain names and should not be present in the set of typosquatted domains.

Start the Stellar language command line interface, which helps in debugging, troubleshooting and learning Stellar. To do so, execute this command:

```
sudo $METRON_HOME/bin/stellar -z $ZOOKEEPER
```

Now, submit queries for each of the four domain names:

```
BLOOM_EXISTS(OBJECT_GET('/tmp/reference/alexa10k_filter.ser'), 'gogle')

BLOOM_EXISTS(OBJECT_GET('/tmp/reference/alexa10k_filter.ser'), 'google')

BLOOM_EXISTS(OBJECT_GET('/tmp/reference/alexa10k_filter.ser'), 'github')
BLOOM_EXISTS(OBJECT_GET('/tmp/reference/alexa10k_filter.ser'),
'gituub')
```

Make a screenshot and store it in the submission document.

There should be a bigger delay for the first query and shorter for the subsequent calls since the results for the subsequent calls will be pulled from the cache created by the first query.

Exit Stellar command line by typing:

```
quit
```

LAB EXERCISE/STEP 7

Squid Data Parser

Create the squid topic in Metron with this command:

```
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --zookeeper
$ZOOKEEPER --create --topic squid --partitions 1 --replication-factor 1
```

Start the squid parser:

```
sudo $METRON_HOME/bin/start_parser_topology.sh -z $ZOOKEEPER -s
squid
```

Make a screenshot and store it in the submission document.

LAB EXERCISE/STEP 8

Set up Enrichment, Threat Intel and Threat Triage

The Squid parser is running now. Next, we should create an enrichment to add a field is_potential_typosquat which will help to determine if a domain is potentially typosquatted. We will set up an alert for typosquatted domains and triage those alerts.

In the enrichment configuration file below, the following changes are made:

- A new field is_potential_typosquat is created to indicate whether the domain is typosquatted according to the Bloom filter of the 10,000 Alexa Top Sites list.
- The is_alert field is updated according to the is_potential_typosquat field.

- A new threat triage rule is added to provide an analyst with a context and a score.

Crete a squid enrichment configuration file:

```
sudo vi $METRON_HOME/config/zookeeper/enrichments/squid.json
```

Press the i button on your keyboard to switch the vi editor to the Insert mode. Copy and paste the following content:

```
{
  "enrichment": {
    "fieldMap": {
      "stellar" : {
        "config" : [
          "domain_without_tld :=
DOMAIN_REMOVE_TLD(domain_without_subdomains)",
          "is_potential_typosquat :=
BLOOM_EXISTS(OBJECT_GET('/tmp/reference/alexa10k_filter.ser'),
domain_without_tld)",
          "domain_without_tld := null"
        ]
      }
   }
  ,"fieldToTypeMap": { }
  },
  "threatIntel": {
    "fieldMap": {
      "stellar" : {
        "config" : [
          "is_alert := (exists(is_alert) && is_alert) || is_potential_typosquat"
        ]
      }


    },
    "fieldToTypeMap": { },
    "triageConfig" : {
      "riskLevelRules" : [
        {
          "name" : "Alexa 10k Typosquat Bloom",
          "comment" : "Inspect a bloom filter with potentially typosquatted
domains from the top Alexa 10k",
          "rule" : "is_potential_typosquat != null && is_potential_typosquat",
          "score" : 10,
          "reason" : "FORMAT('%s is a potential typosquatted domain from
the top 10k domains from alexa', domain_without_subdomains)"
        }
      ],
      "aggregator" : "MAX"
    }
  }
}
```

Press the Esc (Escape) button on your keyboard to exit the Insert mode. Then save the file using the following key combination:

```
:wq
```

Press Enter.

Push the configuration:

```
sudo $METRON_HOME/bin/zk_load_configs.sh -m PUSH -i
$METRON_HOME/config/zookeeper -z $ZOOKEEPER
```

LAB EXERCISE/STEP 9

Configure Elasticsearch Index

Elasticsearch creates a new index automatically when it received new data. We will adjust the mappings for the indx to add alert as a nested field and ensure that the rest of the fields will be assigned correct data types. To do so, we will specify a template by executing the following command (copy and paste into the command line and then press Enter):

```
curl -XPOST "http://$ES_HOST/_template/squid_index" -d '{
         "template": "squid_index*",
         "mappings": {
           "squid_doc": {
             "dynamic_templates": [
        {
          "timestamps": {
            "match": "*:ts",
            "match_mapping_type": "*",
            "mapping": {
            "type": "date",
            "format": "epoch_millis"
            }
                        }
        },
        {
          "threat_triage_score": {
            "mapping": {
              "type": "float"
            },
            "match": "threat:triage:*score",
            "match_mapping_type": "*"
          }
        },
        {
          "threat_triage_reason": {
            "mapping": {
              "type": "text",
              "fielddata": "true"
            },
            "match": "threat:triage:rules:*:reason",
            "match_mapping_type": "*"
          }
        }
      ],
      "properties" : {
        "action" : {
          "type" : "text","fielddata" : true
        },
        "bytes" : {
          "type" : "long"
        },
        "code" : {
          "type" : "long"
        },
        "domain_without_subdomains" : {
          "type" : "text","fielddata" : true
        },
        "elapsed" : {
          "type" : "long"
        },
        "full_hostname" : {
          "type" : "text","fielddata" : true
        },
        "guid" : {
          "type" : "keyword"
        },
        "ip_dst_addr" : {
          "type" : "ip"
        },
        "ip_src_addr" : {
          "type" : "ip"
        },
        "is_alert" : {
          "type" : "text","fielddata" : true
        },
        "is_potential_typosquat" : {
          "type" : "boolean"
        },
        "method" : {
          "type" : "text","fielddata" : true
```

```
            },
            "original_text" : {
              "type" : "text","fielddata" : true
            },
            "source:type" : {
              "type" : "keyword"
            },
            "timestamp" : {
              "type" : "date",
              "format": "epoch_millis"
            },
            "url" : {
              "type" : "text","fielddata" : true
            },
            "alert" : {
              "type" : "nested"
            }
          }
        }
  }
}'
```

Make a screenshot and store it in the submission document.

LAB EXERCISE/STEP 10

Generate Sample Data

We will now use squidclient to visit a regular domain and typosquatted domains, and send the data to Kafka. To do so, run this command:

```
squidclient http://www.github.com
squidclient http://gituub.com/apache/metron
```

Make a screenshot and store it in the submission document.

Feed the generated Squid access.log into Kafka:

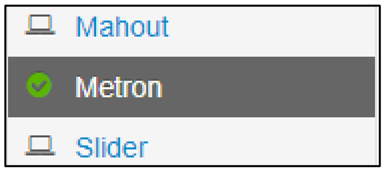
```
sudo cat /var/log/squid/access.log | /usr/hdp/current/kafka-broker/bin/kafka-console-producer.sh --broker-list $BROKERLIST --topic squid
```

Make a screenshot and store it in the submission document.

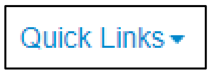
LAB EXERCISE/STEP 11

Investigate alerts with Metron Alerts UI

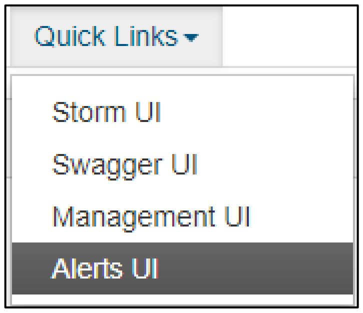
The data, which we fed into Kafka, should have reached the Elasticsearch squid index. Let's investigate triggered alerts using Metron Alerts UI. Navigate to Ambari, enter username and password. In the left-hand-side menu, click on Metron.



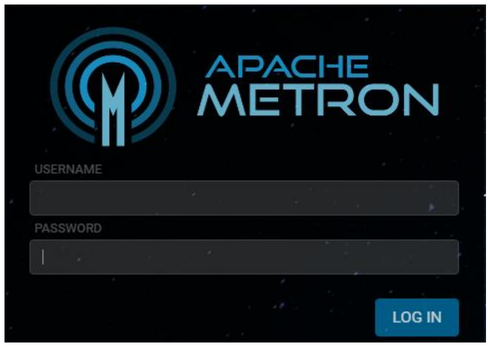
On the top of the screen, in its middle section, click the Quick Links menu.



The menu will display a drop-down list. Click Alerts UI.



A new window will open with a prompt to enter username and password.

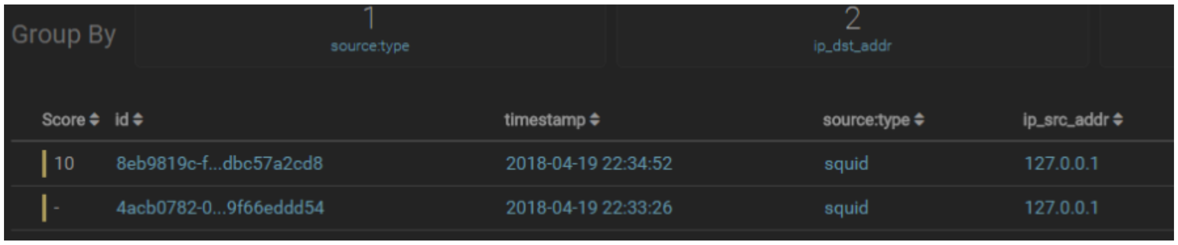


Use the following credentials:

Username: metron

Password: Smoothmetron2

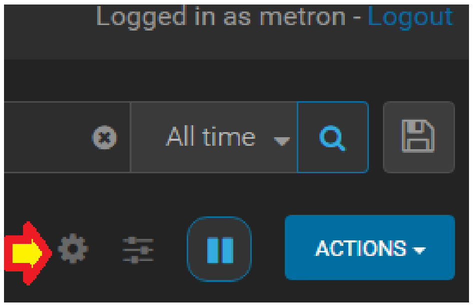
In the Metron UI, you should see the two records from Squid. One with score 10 and another one without a score.



Group By: 1 source:type, 2 ip_dst_addr

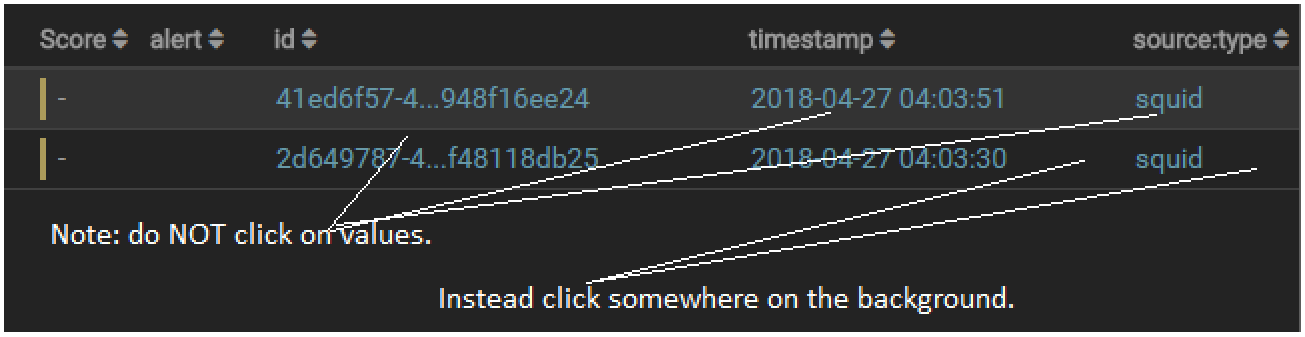
Score	id	timestamp	source:type	ip_src_addr
10	8eb9819c-f...dbc57a2cd8	2018-04-19 22:34:52	squid	127.0.0.1
-	4acb0782-0...9f66eddd54	2018-04-19 22:33:26	squid	127.0.0.1

Add the Score column to the table. To do so, click the settings button in the top right corner:



In the settings screen, scroll down until is_alert appears. Check the box next to it and click the Save button.

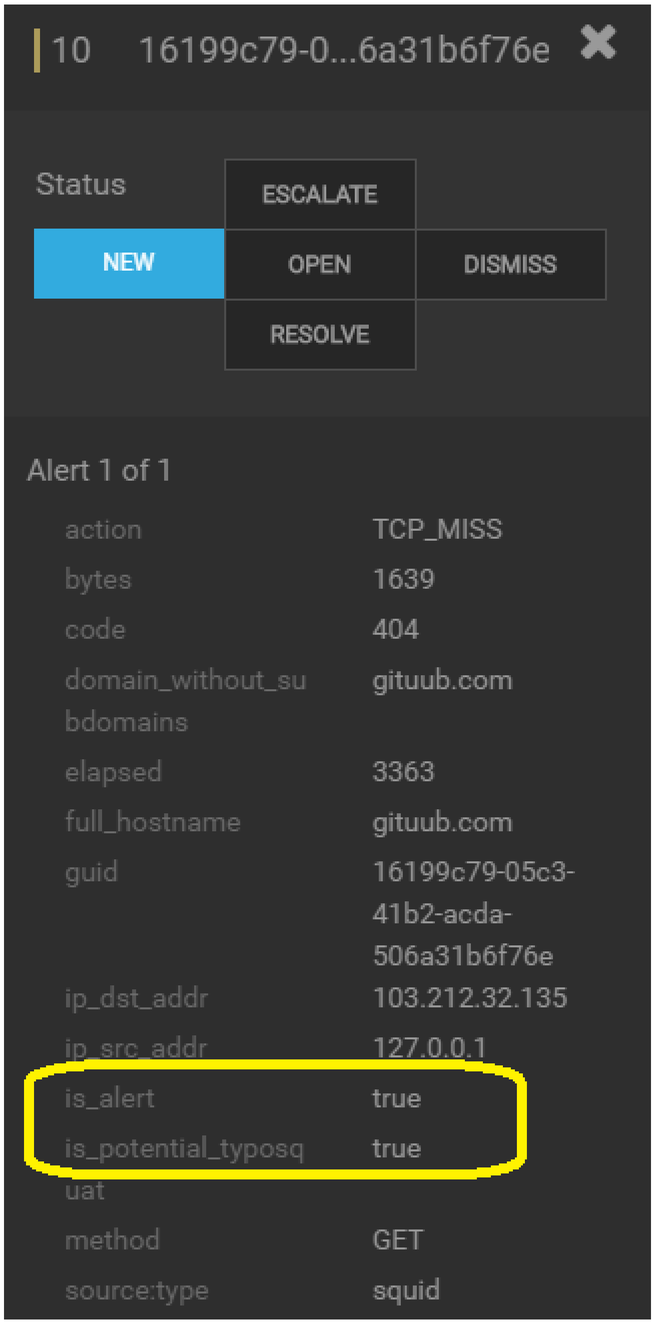
To see record details and to change its status, click somewhere within the record—do not click on any value, but click on the background within the record:



Score	alert	id	timestamp	source:type
-		41ed6f57-4...948f16ee24	2018-04-27 04:03:51	squid
-		2d649787-4...f48118db25	2018-04-27 04:03:30	squid

Note: do NOT click on values.
Instead click somewhere on the background.

Then, you will see more details and will have the ability to change the status to Escalate, Dismiss, Resolve, or Open.



Alert 1 of 1

action	TCP_MISS
bytes	1639
code	404
domain_without_subdomains	gituub.com
elapsed	3363
full_hostname	gituub.com
guid	16199c79-05c3-41b2-acda-506a31b6f76e
ip_dst_addr	103.212.32.135
ip_src_addr	127.0.0.1
is_alert	true
is_potential_typosquat	true
method	GET
source:type	squid

Make a screenshot and store it in the submission document.

LAB EXERCISE/STEP 12

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 [13].

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. D. Makovoz, "Typosquatting Detection Using Advanced Analytics," *securitycommunity.tcs.com*. [Online]. Available: https://securitycommunity.tcs.com/infosecsoapbox/articles/2018/04/23/typosquatting-detection-using-advanced-analytics-0. [Accessed Aug. 8, 2019].
2. H. Draznin, "Trump awarded damages in 'cybersquatting' case over domain names," *CNN*, March, 2014. [Online]. Available: https://edition.cnn.com/2014/03/01/studentnews/trump-cybersquatting-lawsuit/index.html. [Accessed Aug. 8, 2019].
3. Upcounsel, "Cybersquatting Examples: Everything You Need to Know," [Online]. Available: https://www.upcounsel.com/cybersquatting-examples. [Accessed Aug. 8, 2019].
4. Naked Security by Sophos, "Typosquatting – what happens when you mistype a website name?" [Online]. Available: https://nakedsecurity.sophos.com/typosquatting/. [Accessed Aug. 8, 2019].
5. Naked Security by Sophos, "Bitcoin user loses $10K to typosquatters – tips to avoid opening your wallet to imposters," [Online]. Available: https://nakedsecurity.sophos.com/2014/03/24/bitcoin-user-loses-10k-to-typosquatters/. [Accessed Aug. 8, 2019].
6. Alexa. An Amazon Company, "The top 500 sites on the web," [Online]. Available: https://www.alexa.com/topsites. [Accessed Aug. 8, 2019].
7. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422-426, 1970.

8. GitHub, "Apache Metron Use Cases. Typosquat Detection," [Online]. Available: https://github.com/apache/metron/tree/master/use-cases/typosquat_detection. [Accessed Aug. 8, 2019].
9. K. Saini, *Squid Proxy Server 3.1 beginner's guide*, Birmingham, UK: Packt Publishing, 2011.
10. A. Dinabourg, "Bitsquatting: DNS Hijacking without exploitation," [Online]. Available: http://dinaburg.org/bitsquatting.html. [Accessed Aug. 8, 2019].
11. The Apache Software Foundation, "Apache HBase Reference Guide," [Online]. Available: https://hbase.apache.org/book.html. [Accessed Aug 8, 2019].
12. Elasticsearch. https://aws.amazon.com/elasticsearch-service/what-is-elasticsearch/. [Accessed Aug 8, 2019].
13. AWS, "AWS Free Tier," [Online]. Available: https://aws.amazon.com/free/. [Accessed Aug 8, 2019].