

Splunk4Ninjas - Machine Learning for Security

Lab Guide

Overview

This lab guide contains the hands-on exercises for the Splunk4Ninjas - Machine Learning for Security workshop. Before proceeding with these exercises, please ensure that you have a copy of the workshop slide deck, which will help to put into context the tasks you are carrying out.

Download the workshop slide deck: <https://splk.it/ML-Sec-Attendee>

Prerequisites

In order to complete these exercises, you will need your own Splunk instance. Splunk's hands-on workshops are delivered via the [Splunk Show portal](#) and you will need a splunk.com account in order to access this.

If you don't already have a Splunk.com account, please create one [here](#) before proceeding with the rest of the workshop.

Troubleshooting Connectivity

If you experience connectivity issues with accessing either your workshop environment or the event page, please try the following troubleshooting steps. If you still experience issues please reach out to the team running your workshop.

- **Use Google Chrome** (if you're not already)
- If the event page (i.e. <https://show.splunk.com/event/<eventID>>) didn't load when you clicked on the link, try **refreshing the page**
- **Disconnect from VPN** (if you're using one)
- **Clear your browser cache and restart your browser** (if using Google Chrome, go to: Settings > Privacy and security > Clear browsing data)
- **Try using private browsing mode** (e.g. Incognito in Google Chrome) to rule out any cache issues
- **Try using another computer** such as your personal computer - all you need is a web browser! Cloud platforms like AWS can often be blocked on corporate laptops.

Table of Contents

Overview..... 1

Lab 1 – Detect Remote Password Spraying Attacks..... 3

 Description..... 3

 First technique: Simple stats..... 3

 Second technique: Using the Splunk AI Toolkit (MLTK)..... 3

Lab 2 – Detecting Privilege Escalation Events..... 5

 Description..... 5

 First technique: Simple stats..... 5

 Second technique: Using the Splunk AI Toolkit (MLTK)..... 5

Lab 3 – Detecting Categorical Outliers..... 7

 Description..... 7

 Steps..... 7

 Clustering..... 8

 Predict Categorical Fields..... 8

Lab 4 – Leveraging Third Party LLMs..... 9

 Description..... 9

Lab 1 – Detect Remote Password Spraying Attacks

Description

There are two techniques we could use to detect these types of attacks.

First technique: Simple stats

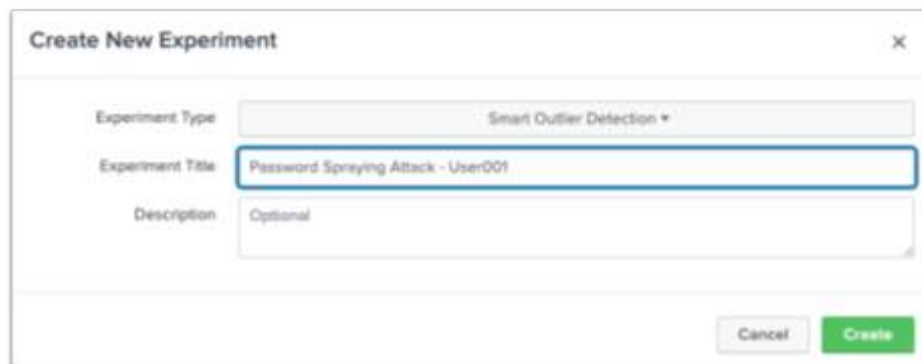
1. Login to your Splunk instance using the “UserXXX” account provided (each workshop attendee has a unique user ID, e.g. User001, User002, etc.) **Please do not use the admin account for participating in the labs.**
2. Open the search interface by either going to the **Search & Reporting** app or within Enterprise Security browsing to **Search > Search**.
3. Run the following search:

```
index="main" sourcetype="XmlWinEventLog_ws" EventCode=4625 LogonType=3
| bucket span=2m _time
| stats dc(TargetUserName) as unique_accounts values(TargetUserName) as
  tried_accounts by _time, IPAddress, LogonType, dvc
| eventstats avg(unique_accounts) as comp_avg, stdev(unique_accounts) as comp_std
  by IPAddress, LogonType, dvc
| eval upperBound=(comp_avg+comp_std*2)
| eval isOutlier=if(unique_accounts > 6 AND unique_accounts >= upperBound, 1, 0)
| search isOutlier=1
```

Second technique: Using the Splunk AI Toolkit (MLTK)

1. Navigate to the **Splunk AI Toolkit** app
2. Choose **Experiments** from the menu bar
3. Select **Smart Outlier Detection** and name your experiment “**Password Spraying Attack - UserXXX**”, where “UserXXX” is your unique username from Splunk Show, e.g. User001.

Note: This is a shared environment so please make sure you use your unique username to avoid confusion!



The screenshot shows a 'Create New Experiment' window. It has three main input fields: 'Experiment Type' (a dropdown menu showing 'Smart Outlier Detection'), 'Experiment Title' (a text box containing 'Password Spraying Attack - User001'), and 'Description' (a text box containing 'Optional'). At the bottom right, there are two buttons: 'Cancel' and 'Create'.

4. For the **Search** enter the following search:

```
index="main" sourcetype="XmlWinEventLog_ws" EventCode=4625 LogonType=3
| bucket span=2m _time
| stats dc(TargetUserName) as unique_accounts values(TargetUserName) as
  tried_accounts by _time, IPAddress, LogonType, dvc
| eventstats avg(unique_accounts) as comp_avg, stdev(unique_accounts) as comp_std
  by IPAddress, LogonType, dvc
| eval HourOfDay = strftime(_time,"%H")
| eval HourOfDay=floor(HourOfDay/4)*4
| eval DayOfWeek = strftime(_time,"%w")
| eval isWeekend=if(DayOfWeek >= 1 AND DayOfWeek <= 5, 0,1)
```

Click on **Next**.

5. Under the **Detect Outliers** section of the side menu, set the following parameters:

Field to analyze: **unique_accounts**
Split by fields: **dvc**
Distribution Type: **Auto**
Outlier Tolerance Threshold: **0.0017**

6. Click on **Detect Outliers** to run the analysis.
7. Click on **Next** to review the experiment.
8. Click on **Save and Next** and then finally on **Save**.
9. Publish your model by clicking on **Publish Outlier Models**. Name your model
“**Password_Spray_Attack_UserXXX**” where “UserXXX” is your unique username from Splunk Show,
e.g. User001.

Publish the Models

Publishing an Experiment model means the main model with any associated preprocessing models will be copied as lookup files in the user's namespace within the selected destination app.

New Main Model Title:

Model names must start with a letter or underscore and contain only letters, numbers, and underscores

Destination App:

Leave the Destination App as **Enterprise Security** and click **Submit** to publish your model.

Lab 2 – Detecting Privilege Escalation Events

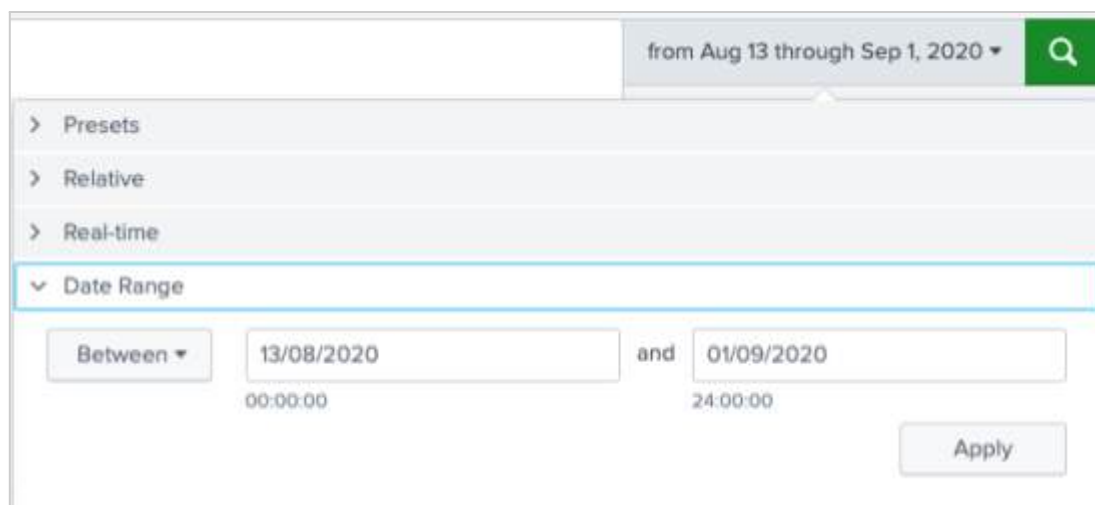
Description

There are two techniques we could use to detect these types of events.

First technique: Simple stats

1. Login to your Splunk instance using the “UserXXX” account provided (each workshop attendee has a unique user ID, e.g. User001, User002, etc.) **Please do not use the admin account for participating in the labs.**
2. Open the search interface by either going to the **Search & Reporting** app or within Enterprise Security browsing to **Search > Search**.
3. Run the following search over the time range **13th August 2020–1st September 2020**:

```
index="main" source="*WinEventLog:Security" EventCode=4648  
| timechart count by src_user
```



The screenshot shows the Splunk search interface. At the top, there is a search bar with the query: `index="main" source="*WinEventLog:Security" EventCode=4648 | timechart count by src_user`. Below the search bar, there is a time range selector. The time range is set to "from Aug 13 through Sep 1, 2020". The "Date Range" section is expanded, showing a "Between" dropdown, two date input fields: "13/08/2020 00:00:00" and "01/09/2020 24:00:00", and an "Apply" button.

4. Visualize the output with a **Line Chart**.

Second technique: Using the Splunk AI Toolkit (MLTK)

1. Navigate to the **Splunk AI Toolkit** app
2. Choose **Experiments** from the menu bar
3. Choose **Detect Numeric Outliers**
4. Click on **Create New Experiment**

5. Choose a title for your experiment (**add your username as it is a shared instance**) such as “**Escalated Privilege Windows - UserXXX**” where “UserXXX” is your unique username from Splunk Show, e.g. User001.

Click on **Create** to create the experiment.

6. In the **Enter a search** box, add the following search:

```
index="main" source="*WinEventLog:Security" EventCode=4648
| bucket _time span=1d
| stats count by src_user _time
```

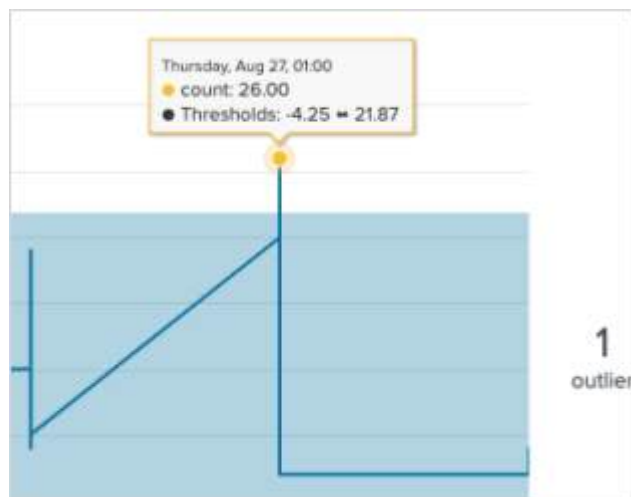
7. Under the search bar set the following parameters:

Field to analyze:	count
Threshold method:	Standard Deviation
Threshold multiplier:	2

You can tweak your Threshold multiplier based on the result that you get, e.g. too many outliers.

You can optionally specify a Sliding window and you can also split by fields.

8. Click **Detect Outliers** to run the analysis.
9. If you wish to view the SPL query that was generated for this analysis - along with a step-by-step explanation of each line in the query - click on the **Show SPL** button beneath any of the charts.
10. On the **Data and Outliers** chart, click on the yellow outlier to see who the outlier is!



Lab 3 – Detecting Categorical Outliers

Description

In this lab you will focus on detecting outliers for the HTTP user agent (http_user_agent).

Steps

1. Navigate to the **Splunk AI Toolkit** app
2. Choose **Experiments** from the menu bar
3. Choose **Detect Categorical Outliers**
4. Click on **Create New Experiment**
5. Choose a title for your experiment (**add your username as it is a shared instance**) such as “**HTTP User Agent Outliers - UserXXX**” where “UserXXX” is your unique username from Splunk Show, e.g. User001.

Click on **Create** to create the experiment.

6. In the **Enter a search** box, add the following search:

```
index=* TERM(agent) sourcetype="stream:http" src_ip="*" http_user_agent="*"
| table http_user_agent, action, dest_port, bytes_in, bytes_out
```

? What do you think is going on behind the scene? What is the method being used here to find outliers?

7. Set **Field to analyze** to **http_user_agent** and click on **Detect Outliers** to run the analysis.
8. Try to run the same search but change the **Field to analyze** (e.g. bytes_in, bytes_out...)

Q. What results do you get?

Q. Do you think it's a good approach?

Let's try a different approach and use Clustering!

Clustering

9. Staying in the **Splunk AI Toolkit** app, click on **Search** in the menu bar.

10. Apply the TFIDF algorithm to the `http_user_agent` field by running the following search:

```
index=* TERM(agent) sourcetype="stream:http" src_ip="*" http_user_agent="*"
| table http_user_agent, src_ip, dest_ip, action, dest_port, bytes_in, bytes_out
| head 3000
| fit TFIDF http_user_agent
```

11. Add clustering by applying the KMeans algorithm to your search:

```
index=* TERM(agent) sourcetype="stream:http" src_ip="*" http_user_agent="*"
| table http_user_agent, src_ip, dest_ip, action, dest_port, bytes_in, bytes_out
| head 3000
| fit TFIDF http_user_agent
| fit KMeans k=5 http_user_agent_tfidf_*
| stats values(http_user_agent) by cluster
```

Note: This search may take a few moments to run - especially in our shared workshop environment - so please be patient!

12. Now find the maximum cluster distance by running the following search:

```
index=* TERM(agent) sourcetype="stream:http" src_ip="*" http_user_agent="*"
| table http_user_agent, src_ip, dest_ip, action, dest_port, bytes_in, bytes_out
| head 3000
| fit TFIDF http_user_agent
| fit KMeans k=1 http_user_agent_tfidf_*
| fields - http_user_agent_tfidf_*
| stats max(cluster_distance) by cluster http_user_agent
| sort - max(cluster_distance)
```

Predict Categorical Fields

13. Navigate to the **Splunk AI Toolkit** app

14. Choose **Showcase** from the menu bar

15. Click on **Predict Fields** and under the **Featured Examples** section click on **Predict the Presence of Malware**

16. Check the data in the `firewall_traffic.csv` lookup file

Q. What fields are being used for our prediction?

17. Try the different algorithms and see which one is the more precise

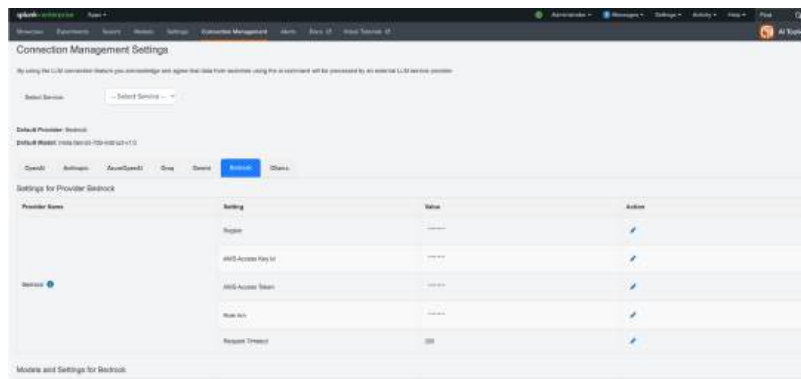
Lab 4 – Leveraging Third Party LLMs

Description

In this lab you will focus on leveraging **bedrock provider** with a specific model to get some recommendations.

Steps

1. Navigate to the **Splunk AI toolkit** app
2. Within the **Splunk AI toolkit**, navigate to the tab “**Connection Management**”
3. You should see that the Bedrock integration has already been setup for you



4. Now, navigate to the Search tab within the Splunk AI toolkit, and run the below search, to familiarize yourself with the lookup file we will be using:

```
| inputlookup http_error_dataset.csv
```

Challenge question! (Get some recommendations from LLM)

1. The command which should be used is **| ai prompt=**
2. The provider to be used is Bedrock and the model, **anthropic.claude-3-5-sonnet-20240620-v1:0**, as followed **provider=Bedrock model=anthropic.claude-3-5-sonnet-20240620-v1:0**
3. Make sure to put the interesting fields between curly braces and single quotes as shown here - **'{Error_Message}'**
4. This is the search you need to run

```
| inputlookup http_error_dataset.csv  
| head 5
```

```
| ai prompt="HTTP Error '{HTTP_Code}' occurred with message: '{Error_Message}'. Root cause: '{Root_Cause}'. What specific steps can we take to resolve this? Provide a precise but informative answer."
provider=Bedrock model=anthropic.claude-3-5-sonnet-20240620-v1:0
```