



Lab 5 - Detecting Bad Trons

CSEC 2324 - Network Security
Lab Guide v1.0

William Cox, M.S.
Visiting Assistant Professor
wcox@ualr.edu or wcox@uaptc.edu

January 2023




Contents

Lab Guide Instructions	4
Lab File Formatting	4
Markdown How-To	5
Suggested Setup	5
Lab Assignment 5 - Detecting Bad Trons	5
Overview	5
Lab Artifacts	6
Lab Software	6
Part 1: Using Zeek Bro...	6
Starting Zeek	6
Using Zeek	7
Logging	7
Suricata (The way of the Meerkat)	8
Configure	8
Signatures	9

Lab Guide Instructions

The labs included in this guide will help you understand the principles of Network Security.

When you see a  you will be required to take a screenshot of that step.

Included is the requirement for a Lab Report file.

Follow the formatting instructions for full credit.

Note: Using Markdown is only required for the first lab.

Lab File Formatting

When completing your labs please follow these instructions.

1. All Labs created using Markdown & will use the Eisvogel template. Labs files created in another word processing program will use Times New Roman, 12 Font, Double Space.
Get Eisvogel here: <https://github.com/Wandmalfarbe/pandoc-latex-template>
2. Create a title page with the following details:
 1. Title of the Lab
 2. Class Name
 3. Your name
 4. Date
3. Section 2 will have all screenshots and questions/answers for the lab.
 1. Each question must be listed with its question number.
 2. Answers will be indented on the next line and start with an "a."
 3. If answer includes a picture, make sure picture is big enough for your instructor to interpret, but not too big to distract from the quality of your work.
4. Section 3 will be labeled Reflection.
 1. This is where you add any reflections needed.
 2. Make sure to quote your sources with parenthetical citations.
 3. Do not use quotes, but instead rewrite the quote in your own words. *Remember to still give credit to the author.*
5. Section 4 will be References
 1. All references should be in alphabetical order
 2. Use either APA or IEEE formatting

Markdown How-To

Below you will find a quick reference on how to use Markdown. This is not all inclusive and you may need to research steps that are particular to your situation. IE. Operating System, IDE, etc...

Suggested Setup

(You don't need to follow if you know Markdown)

1. Download and install following programs:
 1. VSCode Download
 2. Pandoc Install Instructions **Note: make sure to install all of the required dependencies**
 3. Eisvogel Template Download
2. Once you have downloaded and installed the required applications, you will need to set up your template and environments.
 1. VSCode Setup - Install following extensions:
 1. Markdown All in One, Author: Yu Zhang
 2. Dictionary Completion, Author: Yu Zhang
 3. markdownlint, Author: David Anson
 2. Setup Pandoc template Eisvogel Install Instructions
3. Basic Markdown syntax can be found at Markdown Guide
4. Convert Markdown to PDF syntax
 1. Open terminal and navigate to your markdown location
 2. Execute the following command replacing the file names with your information.

```
1 pandoc filename.md -o filename.pdf --from markdown --template  
   eisvogel --listings
```

Lab Assignment 5 - Detecting Bad Trons

Overview

This lab will introduce you to Different Intrusion Detecting Systems. Don't just go through the motions in this lab, but try to understand what you are doing and how you could defend against these attacks.

Lab Artifacts

Build a Lab report file with the requested answers or screenshots presented in this lab. Follow the Lab file format found in the beginning of the Lab Guide Manual.

Lab Software

Programs tcpdump, Bro, Suricata **Operating System:** Linux

Terminal Emulator: bash, shell, zsh, csh

Environment Cyber Arena

Sudo Password CSEC2324_Student!


Part 1: Using Zeek Bro...

References:


<https://docs.zeek.org/en/master/quickstart.html>

<https://suricata.readthedocs.io/en/suricata-6.0.0/quickstart.html>


Starting Zeek

1. Log into the Cyber Arena
2. Select CSEC2324 Workout
3. Open the terminal
4. Change directory to /opt/zeek
5. Run the following command: 

```
1 sudo bin/zeekctl
```

6. This will open the Zeek control tool 

```
1 [ZeekControl] >
```

7. Now type the following commands in order 



1. install
2. deploy
3. exit

Zeek is now running and ready to use

Using Zeek

As a reminder here is the folder structure for Zeek:

```
1 /opt/zeek/  
2 |_ bin/  
3 |_ etc/  
4 |_ include/  
5 |_ lib/  
6 |_ logs/  
7 |_ share/  
8 |_ spool/
```

1. Change directory to `/opt/zeek/share/zeek/site`
2. List all files in this folder 
3. Now we need to edit the `local.zeek` file
4. Type: `sudo vim local.zeek`
5. Scroll to the very bottom of the file and type `@load alert-all-notices` 
6. Save and exit

We have now set up our Zeek configuration to load in this new custom module we are about to create!

7. Create a new file by typing `sudo vim alert-all-notices.zeek`
8. Type the following in the new file: 

```
1 hook Notice::policy(n: Notice::Info)  
2 {  
3     add n$actions[Notice::ACTION_LOG];  
4     add n$actions[Notice::ACTION_EMAIL];  
5 }
```

9. Save and exit vim by type `:wq`
10. Next we need to deploy our updates

```
1 sudo /opt/zeek/bin/zeekctl
```


11. Type the following commands in order

```
1 [ZeekControl] > deploy  
2 [ZeekControl] > exit
```

Logging

We will now view the log file we created in the last section

1. Change directory to `/opt/zeek/logs/current`

2. List all of the Logs in this folder 
3. Use the zeek-cut command to view our conn.log file

```
1 cat conn.log | zeek-cut id.orig_h id.orig_p id.resp_h id.resp_p
```

This command will parse the output and show Source IP, Source Port, Destination IP, and Destination Port
IE:

```
1 ...
2 192.168.0.1 12345 88.22.11.11 22
3 192.168.0.1 51029 52.226.105.214 443
4 ...
```

4. Now use your research skills. Look through the traffic and identify any abnormal traffic.
5. Write ~300 words on this traffic and why it is abnormal and add it to your reflections.


Suricata (The way of the Meerkat)

Next, you will explore how to use a signature based IDS. There are many ways to run Suricata wither manually or through a SIEM like Kabana. We are going to configure and use this IDS manually.

Configure

1. Change directory to `/etc/suricata/`
2. Suricata uses a YAML file for configurations. Use VIM to make changes to this file.

```
1 sudo vim /etc/suricata/suricata.yaml
```

3. Change/verify the following lines are configured like so: 

```
1 HOME_NET: "[192.168.0.0/23,172.16.0.0/23]"
2 EXTERNAL_NET: "!$HOME_NET"
```

4. Save and close the file when you are finished

```
1 :wq
```

5. Next, change directory to `/etc/suricata/rules/`
6. Pick a rules file to look through and find a signature that you want to analyze.
7. Explain what this signature does and what makes this signature work. What file was it located? Add this explanation to your reflections.

Signatures

1. Identify the Threat

The first step in creating a new signature is to identify the threat that you want to detect. This could be a specific type of network traffic, a specific type of payload, or anything else that you want to flag as malicious.

For the purposes of this lab, we will assume that we want to detect a specific type of ICMP that is being used to launch a fictitious DDoS attack.

2. Create the Signature

Reference: <https://suricata.readthedocs.io/en/latest/rules/index.html>

Once you have identified the threat, you can create the signature to detect it. Signatures in Suricata are written in the rules language and consist of a number of different fields, including:

Rule header: This field contains metadata about the rule, such as the rule ID, the version, and the author.

Rule options: This field contains options that control how the rule is applied, such as the action to take when the rule is triggered (e.g. alert, drop, pass) and the severity of the rule.

Rule content: This field contains the actual content of the rule, which is used to match against network traffic.

Here is an example of a simple signature that detects the ICMP packets:

```
1 alert icmp any any -> any any (msg:"its pinging"; sid:1000001;)
```

This signature consists of a rule header (which is generated automatically by Suricata), a rule option (alert) that specifies what to do, and a rule content (icmp any any <=> any any) that specifies the ICMP packets we want to detect.

3. Add the Signature to the Configuration File

Once you have created the signature, you need to add it to the Suricata configuration file (suricata.yaml) in order for it to be used. In the configuration file, you can specify which signatures are enabled and which are disabled, as well as other options such as the path to the signature file and the rule action.

Here is an example of how you might add the signature to the configuration file:

```
1 rule-files:  
2   - icmp.rules
```

This example specifies that the signature file ddos.rules should be loaded and that the default action for all rules is alert.

4. Test your signature

You can use Suricata's inline mode to pass traffic through Suricata and see if it triggers any alerts. You can do this by running Suricata with the `-r` option, followed by the path to a pcap file containing the traffic you want to test. Download the PCAP named `icmp.pcap` from www.liquidswrds.com. Got to the schools link then click on your class. The pcap is in the folder labeled Packet Capture.

For example:

```
1 suricata -r /path/to/icmp.pcap
```

If your signature is triggered, you should see an alert in the Suricata output.