



CYBER THREAT INTELLIGENCE LAB  
College of Engineering & Computer Science  
Florida Atlantic University

# HANDS-ON VLABS: ZEEK (BRO) IDS

**ELIAS BOU-HARB, Ph.D.**

Assistant Professor

**ANTONIO MANGINO**

Research Assistant



**NSF Award 1829698**

CyberTraining CIP: Cyberinfrastructure Expertise on High-throughput Networks for Big Science Data Transfers

**July 23<sup>rd</sup>, 2019**

Training Workshop for Network Engineers and Educators on Tools and  
Protocols for High-Speed Networks

# Zeek (Bro) Labs: Outline

2

Lab 1. Introduction to the Capabilities of Zeek

Lab 2. An Overview of Zeek Log Files

Lab 3. Parsing, Reading and Organizing Bro Log Files

Lab 4. Generating and Analyzing Network Scanner Traffic

Lab 5. Generating, Capturing and Analyzing DoS-centric Network Traffic

# Zeek (Bro) Labs: Outline

3

Lab 6. Introduction to Zeek Scripting

Lab 7. Advanced Zeek Scripting for Anomaly and Malicious Event Detection

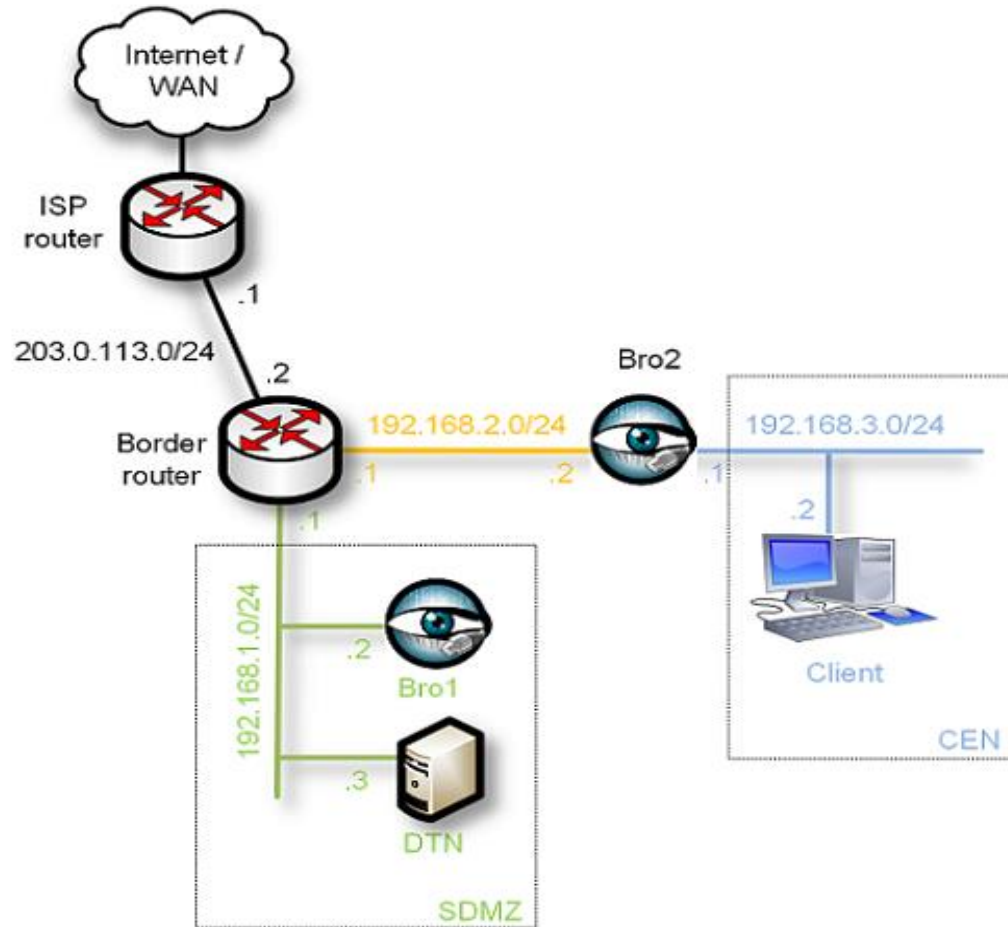
Lab 8. Preprocessing of Zeek Output Logs for Machine Learning

Lab 9. Developing Machine Learning Classifiers for Anomaly Inference and Classification

Lab 10. Profiling and Performance Metrics of Zeek

# Lab 3: Parsing, Reading and Organizing Zeek Log Files

4



# Lab 3: Motivation

5

- ❑ Zeek log files contain extensive packet and connection-related data, separated in tab-delimited columns
- ❑ Without requiring complex processing, packet data and connection results can be extracted using terminal utilities
- ❑ Extracted data can confirm the existence of malicious events or anomalies, representing the need for further analysis

# Lab 3: Objectives

6

- Use Linux tools and commands for text file processing
- Practice Linux shell scripts and the AWK scripting language
- Incorporate AWK with the zeek-cut utility to provide formatted log files

# Lab 3: The zeek-cut Utility

7

- Zeek's network traffic processing generates multiple log files, each declared within event-based policy scripts
- Log files include headers and padding – requiring popular Unix terminal utilities to format and analyze specific queries
- The zeek-cut utility is used to retrieve specific columns and data entries from Zeek log files to be processed

# Lab 3: Default Zeek Log File Format

8

- Log file formatting, path and padding options

```
#separator \x09
#set_separator ,
#empty_field (empty)
#unset_field -
#path conn
#open 2019-07-15-14-00-58
#fields ts uid id.orig_h id.orig_p id.resp_h id.resp_p proto
service duration orig_bytes resp_bytes conn_state local_orig
local_resp missed_bytes history orig_pkts orig_ip_bytes resp_pkts
resp_ip_bytes tunnel_parents
#types time string addr port addr port enum string interval count
count string bool bool count string count count count count set[string]
1295981542.708292 CYPaWH1PU5PK00Sble 192.168.3.131 55950 72.14.213.102 80
tcp http 0.058485 944 487 SF - - 0 ShADFadRf
5 1156 4 659 -
1295981543.461968 C8Cn4g1cJiS2zEfnrc 192.168.3.131 55955 207.46.148.38 80
tcp http 0.028620 448 279 SF - - 0 ShADfFa 5
660 3 407 -
```



# Lab 3: Default Zeek Log File Format

9

## □ Fields – Categories of packet features

```
#separator \x09
#set_separator ,
#empty_field (empty)
#unset_field -
#path conn
#open 2019-07-15-14-00-58
#fields ts uid id.orig_h id.orig_p id.resp_h id.resp_p proto
service duration orig_bytes resp_bytes conn_state local_orig
local_resp missed_bytes history orig_pkts orig_ip_bytes resp_pkts
resp_ip_bytes tunnel_parents
#types time string addr port addr port enum string interval count
count string bool bool count string count count count count set[string]
1295981542.708292 CYPaWH1PU5PK00SBle 192.168.3.131 55950 72.14.213.102 80
tcp http 0.058485 944 487 SF - - 0 ShADFadRf
5 1156 4 659 -
1295981543.461968 C8Cn4g1cJiS2zEfnc 192.168.3.131 55955 207.46.148.38 80
tcp http 0.028620 448 279 SF - - 0 ShADfFa 5
660 3 407 -
```

# Lab 3: Default Zeek Log File Format

10

- Types – Variable data objects of packet features

```
#separator \x09
#set_separator ,
#empty_field (empty)
#unset_field -
#path conn
#open 2019-07-15-14-00-58
#fields ts uid id.orig_h id.orig_p id.resp_h id.resp_p proto
service duration orig_bytes resp_bytes conn_state local_orig
local_resp missed_bytes history orig_pkts orig_ip_bytes resp_pkts
resp ip bytes tunnel parents
#types time string addr port addr port enum string interval count
count string bool bool count string count count count count set[string]
1295981542.708292 CYPaWH1PU5PK00SBle 192.168.3.131 55950 72.14.213.102 80
tcp http 0.058485 944 487 SF - - 0 ShADfAdRf
5 1156 4 659 -
1295981543.461968 C8Cn4g1cJiS2zEfnc 192.168.3.131 55955 207.46.148.38 80
tcp http 0.028620 448 279 SF - - 0 ShADfFa 5
660 3 407 -
```

# Lab 3: Default Zeek Log File Format

11

## □ Tab-Delimited packet data

```
#separator \x09
#set_separator ,
#empty_field (empty)
#unset_field -
#path conn
#open 2019-07-15-14-00-58
#fields ts uid id.orig_h id.orig_p id.resp_h id.resp_p proto
service duration orig_bytes resp_bytes conn_state local_orig
local_resp missed_bytes history orig_pkts orig_ip_bytes resp_pkts
resp_ip_bytes tunnel_parents
#types time string addr port addr port enum string interval count
count string bool bool count string count count count count set[string]
```

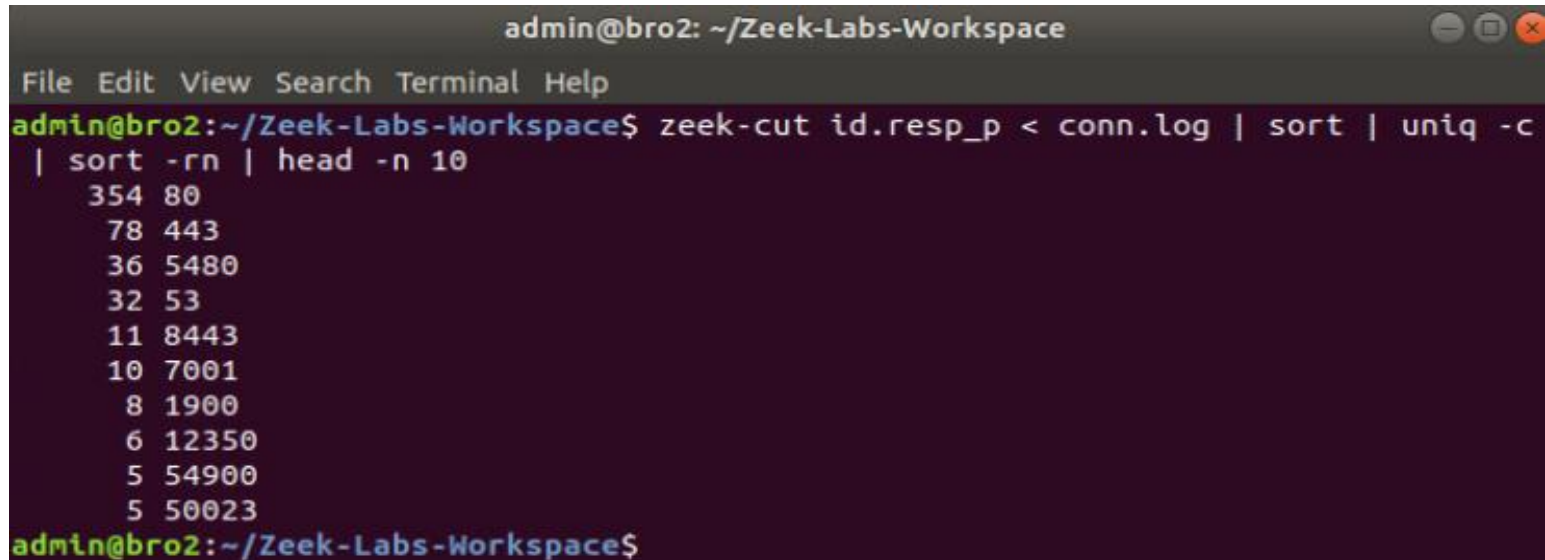
1295981542.708292			CYpaWH1PU5PK00SBle		192.168.3.131	55950	72.14.213.102	80		
tcp	http	0.058485	944	487	SF	-	-	0	ShADFadRf	
5	1156	4	659	-						
1295981543.461968			C8Cn4g1cJiS2zEfnc		192.168.3.131	55955	207.46.148.38	80		
tcp	http	0.028620	448	279	SF	-	-	0	ShADFFa	5
660	3	407	-							

# Lab 3: Example of zeek-cut Query

12

- Display the 10 destination ports that received the most network traffic, organized in descending order

```
zeek-cut id.resp_p < conn.log | sort | uniq -c | sort -rn | head -n 10
```



The screenshot shows a terminal window titled 'admin@bro2: ~/Zeek-Labs-Workspace'. The prompt is 'admin@bro2:~/Zeek-Labs-Workspace\$'. The command entered is 'zeek-cut id.resp\_p < conn.log | sort | uniq -c | sort -rn | head -n 10'. The output is a list of 10 lines, each representing a destination port and the number of connections to it, sorted in descending order of connection count. The output is:

```
354 80
78 443
36 5480
32 53
11 8443
10 7001
8 1900
6 12350
5 54900
5 50023
```

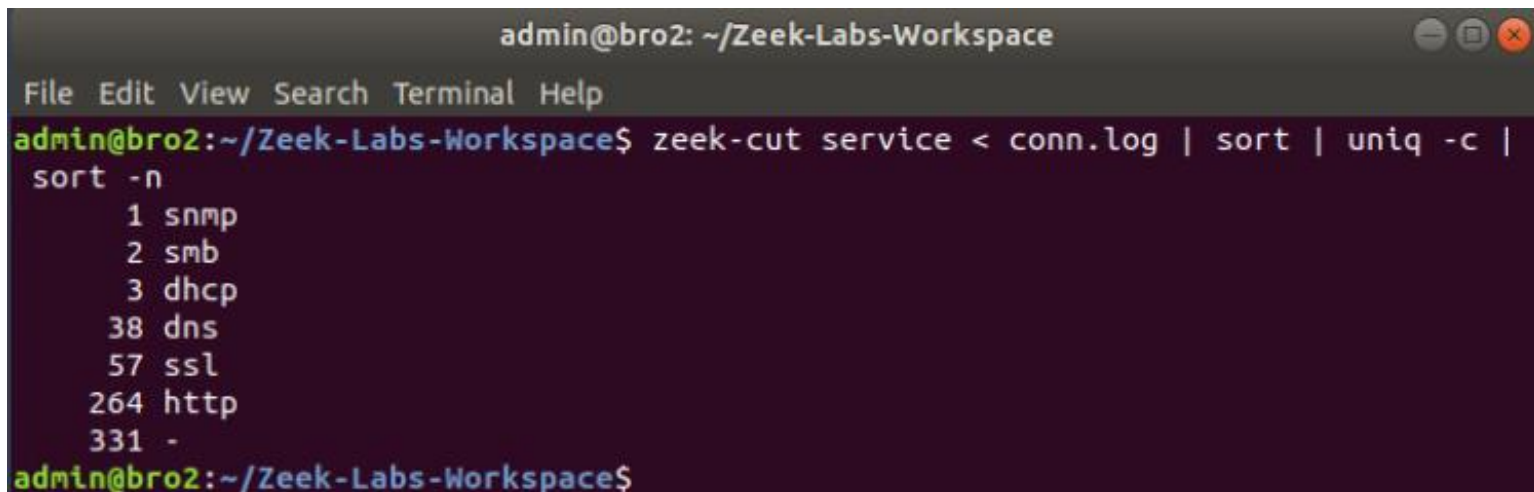
The terminal prompt is now 'admin@bro2:~/Zeek-Labs-Workspace\$'.

# Lab 3: Example zeek-cut Query

13

- Display the number of packets received per (protocol) service, organized in ascending order

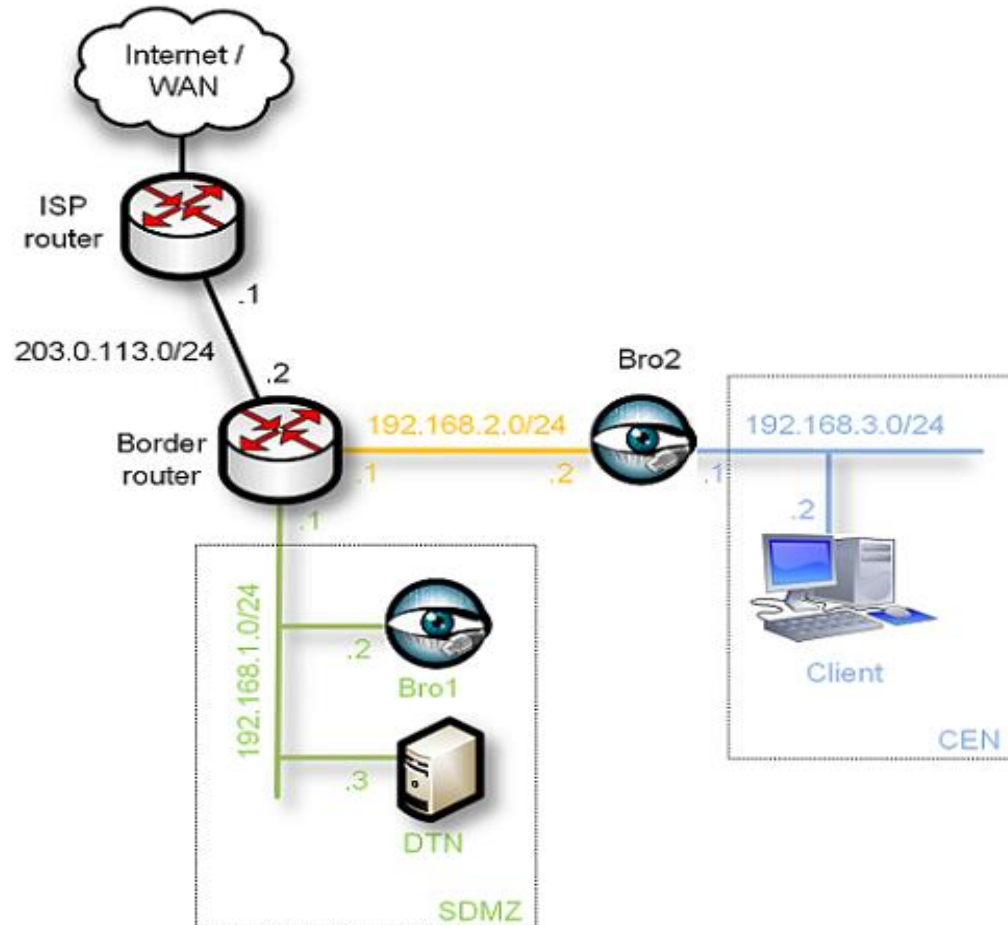
```
zeek-cut service < conn.log | sort | uniq -c | sort -n
```



```
admin@bro2: ~/Zeek-Labs-Workspace
File Edit View Search Terminal Help
admin@bro2:~/Zeek-Labs-Workspace$ zeek-cut service < conn.log | sort | uniq -c |
sort -n
    1 snmp
    2 smb
    3 dhcp
   38 dns
   57 ssl
  264 http
  331 -
admin@bro2:~/Zeek-Labs-Workspace$
```

# Lab 4: Generating and Analyzing Network Scanning Traffic

14



# Lab 4: Motivation

15

- Zeek policy scripts and filters can be employed during network traffic processing to include new event-based actions
- Malicious network traffic such as network scanning can be detected and extracted using policy scripts or filters
- Active scanning traffic targeting networks raises concerns over possible network vulnerabilities

# Lab 4: Objectives

16

- Introduce the notion of scanning events
- Introduce the nmap software
- Utilize the lab topology to generate, record and analyze scan traffic



# Lab 4: Network Scanning

17

- ❑ Network scanning is a preliminary action typically executed to find vulnerabilities or exploit hosts
- ❑ Malicious events targeting discovered vulnerabilities may cause data breaches, denial of service or monetary losses
- ❑ Discovery of network scanning traffic allows network analysts to study and mitigate possible vulnerabilities probed by scan traffic

# Lab 4: Example Live Network Capture

18

- ❑ The virtual lab workspace includes the *nmap* software
- ❑ *nmap* is used to generate various forms of scan traffic, which is then captured using the Zeek IDS
- ❑ Zeek utilizes the *tcpdump* utility for live network capture

```
sudo tcpdump -i ens33 -w packetcapture.pcap
```

# Lab 4: Example Live Network Capture

19

- Generating probing traffic:

```
admin@bro1: ~  
File Edit View Search Terminal Help  
admin@bro1:~$ sudo nmap -sT 192.168.2.2  
  
Starting Nmap 7.60 ( https://nmap.org ) at 2019-07-15 14:48 EDT  
Nmap scan report for 192.168.2.2  
Host is up (0.00024s latency).  
All 1000 scanned ports on 192.168.2.2 are closed  
  
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds  
admin@bro1:~$
```

# Lab 4: Example Live Network Capture

20

- Capturing live network traffic:

```
admin@bro2: ~/Zeek-Labs-Workspace
File Edit View Search Terminal Help
admin@bro2:~/Zeek-Labs-Workspace$ sudo tcpdump -i ens33 -w scantraffic.pcap
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
^C4020 packets captured
4020 packets received by filter
0 packets dropped by kernel
admin@bro2:~/Zeek-Labs-Workspace$
```

# Lab 4: Example Live Network Capture

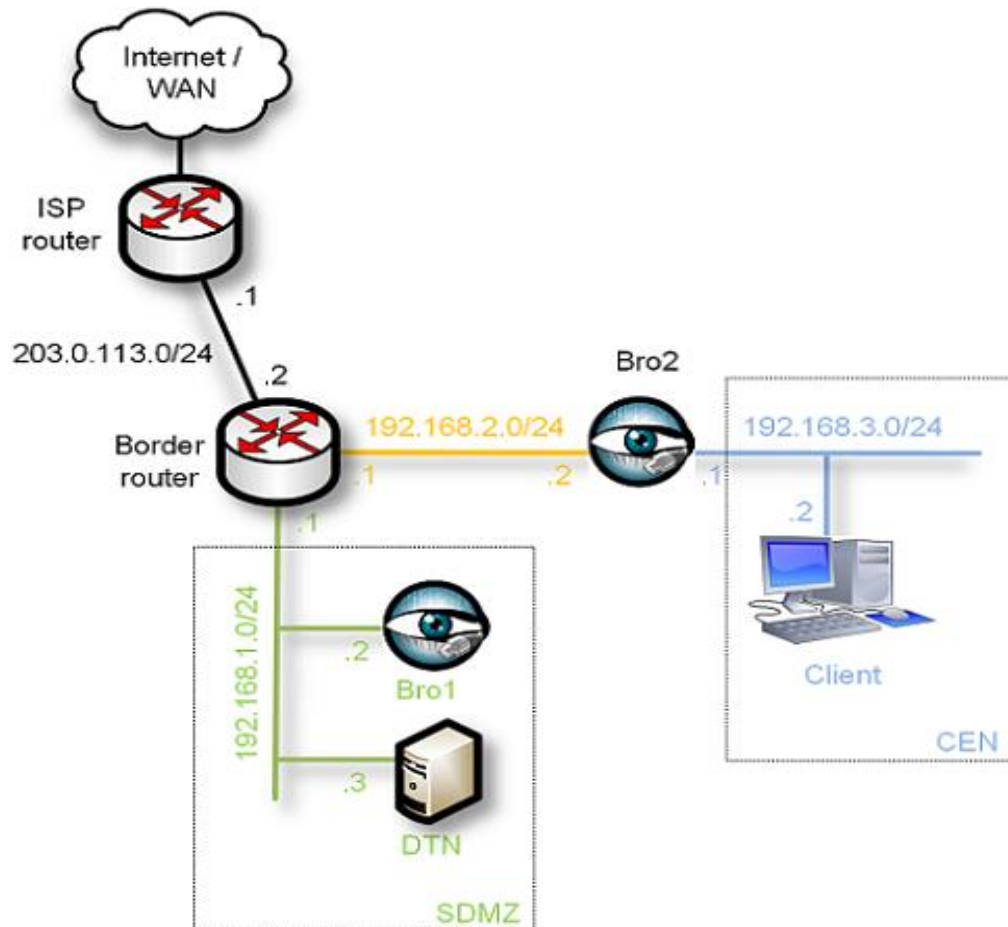
21

- Viewing the network capture file:

```
admin@bro2: ~/Zeek-Labs-Workspace
File Edit View Search Terminal Help
admin@bro2:~/Zeek-Labs-Workspace$ tcpdump -r scantraffic.pcap | grep '192.168.1.2'
reading from file scantraffic.pcap, link-type EN10MB (Ethernet)
19:27:30.092645 IP 192.168.1.2 > bro2: ICMP echo request, id 3834, seq 0, length 8
19:27:30.092675 IP bro2 > 192.168.1.2: ICMP echo reply, id 3834, seq 0, length 8
19:27:30.092704 IP 192.168.1.2.57190 > bro2.https: Flags [S], seq 1632392789, win 1024, options [mss 1460], length 0
19:27:30.092713 IP bro2.https > 192.168.1.2.57190: Flags [R.], seq 0, ack 1632392790, win 0, length 0
19:27:30.092717 IP 192.168.1.2.57190 > bro2.http: Flags [R.], seq 0, ack 1632392789, win 1024, length 0
19:27:30.092721 IP bro2.http > 192.168.1.2.57190: Flags [R], seq 1632392789, win 0, length 0
19:27:30.092734 IP 192.168.1.2 > bro2: ICMP time stamp query id 24283 seq 0, length 20
```

# Lab 9: Developing Machine Learning Classifiers for Anomaly Inference and Classification

22



# Lab 9: Motivation

23

- ❑ Malicious network attacks are continually evolving, utilizing new packet features and abusing critical protocols
- ❑ Keeping filters and scripts updated to handle newly emerging malicious techniques is a daunting task
- ❑ Machine learning classifiers can be used to predict, infer and fingerprint unlabeled traffic

# Lab 9: Objectives

24

- Introduce the advantages of leveraging machine learning for network analysis
- Develop and train a decision table to classify scan-related network traffic activities
- Test the developed models and review their classification output on test datasets



# Lab 9: Machine Learning Classifiers

25

- By preprocessing Zeek log files, they can be converted to training and test artifacts for machine learning classifiers
- If trained with a valid, comprehensive dataset, classifiers can be used to identify and predict anomalous traffic
- The virtual lab workspace includes the *Weka* software, which is used to preprocess and train machine learning classifiers

# Lab 9: Example Classifier Predictions

26

## □ Unlabeled dataset:

```
@RELEATION ntraffic
```

```
@ATTRIBUTE time NUMERIC
```

```
@ATTRIBUTE sourceip NOMINAL
```

```
@ATTRIBUTE destip NUMERIC
```

```
@ATTRIBUTE sourceport NUMERIC
```

```
@ATTRIBUTE destport NUMERIC
```

```
@ATTRIBUTE protocol {tcp, udp, icmp}
```

```
@ATTRIBUTE service NOMINAL
```

```
@ATTRIBUTE class {1, 0}
```

```
@DATA
```

```
1563396794,19216813,19216822,5110,80,tcp,http,?
```

```
1563396797,19216822,19216813,80,5510,tcp,http,?
```

# Lab 9: Example Classifier Predictions

27

## □ Classifier predicted dataset:

```
@RELEATION ntraffic
```

```
@ATTRIBUTE time NUMERIC
```

```
@ATTRIBUTE sourceip NOMINAL
```

```
@ATTRIBUTE destip NUMERIC
```

```
@ATTRIBUTE sourceport NUMERIC
```

```
@ATTRIBUTE destport NUMERIC
```

```
@ATTRIBUTE protocol {tcp, udp, icmp}
```

```
@ATTRIBUTE service NOMINAL
```

```
@ATTRIBUTE class {1, 0}
```

```
@DATA
```

```
1563396794,19216813,19216822,5110,80,tcp,http,1
```

```
1563396797,19216822,19216813,80,5510,tcp,http,0
```

# Questions

*Thank you*



**ELIAS BOU-HAB, Ph.D.**  
Assistant Professor

**ANTONIO MANGINO**  
Research Assistant

**NSF Award 1829698**

CyberTraining CIP: Cyberinfrastructure Expertise on High-throughput Networks for Big Science Data Transfers

