



Introduction to Zeek

Zeek Training 2022

May 20th

Who Are We?



Keith Lehigh

- UIISO @ Indiana University
- 10 yrs running Zeek on large .edu network
- Zeek LT Member

Fatema Bannat Wala

- Security Engineer @ ESnet/LBNL
- Big Fan of Zeek, ever since it was Bro!
- 7 yrs of Network defense
- Zeek LT Member / Zeek Training Subgroup Lead



Goals

You will learn about:

- Who uses Zeek? - Keith
- What is Zeek? - Keith
- Running Zeek - Fatema
- What does Zeek do? - Keith
- Zeek's Filesystem layout - Fatema
- Customizing Zeek - Keith
- Zeek Package Manager (zkg) - Fatema
- Log Processing Intro - Keith
- Interact with some common frameworks - Fatema / Keith



Housekeeping

- Github Link- Docker How to doc
 - <https://github.com/zeek/zeek-training/tree/master/Intro-to-Zeek22>
- Docker image
 - Zeek v4.0.6 installed on the image-
<https://hub.docker.com/repository/docker/zeekurify/zeek-training-2022>
 - `# docker pull zeekurify/zeek-training-2022`
 - Path inside the container to training resources used during this training:
`/zeek/training-res/`
- Zeek Official Document:
 - <https://docs.zeek.org/en/v4.0.6/>



Who Uses Zeek?

Zeek Users



- Malware Researchers
- Network Defenders
 - .edu / .gov / private sector
 - Forensics, Monitoring, Detection
- Network Researchers
- People building Zeek in their products



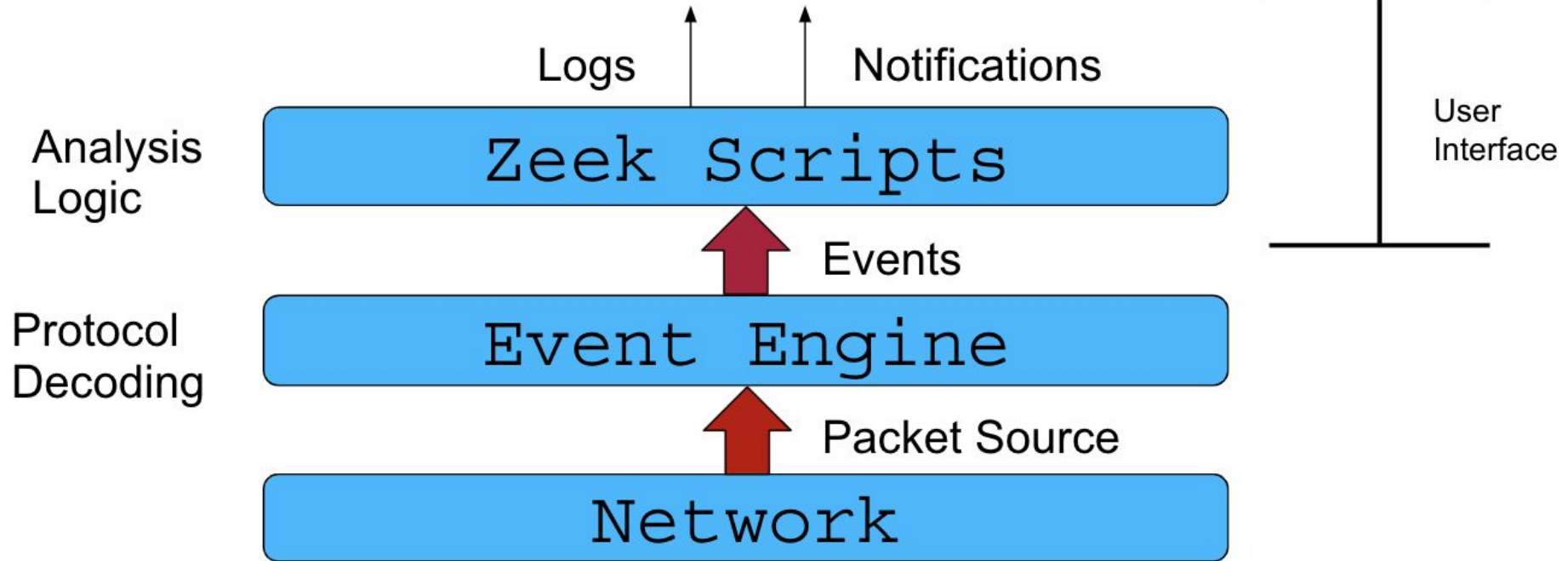
What Is Zeek?

Zeek Platform



- 25 years old, long history in academia
- Domain Specific Network Monitoring Language
- Policy Neutral
- Leave your IDS ideas behind

Zeek Architecture



Analyzers



- Protocol Analyzers

- Most popular - SMTP / HTTP / SSL / DNS / DHCP
- Authentication - SSH / KERBEROS / RADIUS
- MS protocols - RPC / NTLM / SMB / RDP
- Interesting ones - SOCKS / TUNNEL / IRC / FTP

- File Analyzers

- EXTRACT / HASH / PE / X509

- Spicy!

- C++ parser generator that makes it easy to create robust parsers for network protocols, file formats, and more..



Dynamic Protocol Detection

- Hints from known_ports
- Failure to attach or no known_ports
 - Try analyzers and signatures until success
- Successful protocol parsing reflected in:
 - conn.log proto field
 - files.log analyzers field
- DPD detects known protocols on unknown ports
 - Non-standard port doesn't mean it is bad
 - But it sure is interesting
 - known_services.log can be a goldmine

HAND OFF TO FATEMA





Running Zeek



Docker / Zeek Usage

Docker Setup

- Github Link- Docker How to doc
 - <https://github.com/zeek/zeek-training/tree/master/Intro-to-Zeek22>
- Docker image
 - Zeek v4.0.6 installed on the image-
<https://hub.docker.com/repository/docker/zeekurify/zeek-training-2022>
 - `# docker pull zeekurify/zeek-training-2022`
 - `# docker run -it zeekurify/zeek-training-2022 /bin/bash`
 - Path inside the container to training resources used during this training: `/zeek/training-res/`

Verifying if Zeek installed correctly:

```
# /zeek/bin/zeek -h
```

```
# export PATH=/zeek/bin/:$PATH
```



Running Zeek : CLI

Different Zeek CLI running options:

1. `zeek -r capture.pcap`
2. `zeek -C -r capture.pcap`
3. `zeek -r capture.pcap my-script.zeek`
4. `zeek -r capture.pcap local`
5. `zeek -r capture.pcap local -e 'redef Site::local_nets += { 192.168.125.0/24 };'`
6. `zeek -r modbus.pcap protocols/modbus/track-memmap.zeek`



Running Zeek : CLI

Run the command lines below and compare output. 10 Minutes!

```
# zeek -C -r /zeek/training-res/capture.pcap
```

```
# zeek -C -r /zeek/training-res/capture.pcap local
```

WARNING: No Site::local_nets have been defined. It's usually a good idea to define your local networks.

```
# zeek -C -r /zeek/training-res/capture.pcap local -e 'redef Site::local_nets += {  
192.168.125.0/24 }';
```

<no warning this time>

```
# zeek -r /zeek/training-res/modbus.pcap protocols/modbus/track-memmap.zeek
```




Running Zeek online: try.zeek.org

Running Zeek online - Get familiar with try.zeek website and running some simple scripts on the website

Go to your web-browser and type “try.zeek.org” website.

Try out running zeek on scripts as well as provided pcaps on the server!

Can you share your custom #tryzeek script with others?



Running Zeek : only with scripts

Running Zeek with just scripts without any pcaps. Printing hello world like examples. This will show another way of running Zeek

```
# zeek /zeek/training-res/hello-world.zeek
```



dump-events

Handiest tool in the Zeek toolbox :

```
# zeek -r <pcap> misc/dump-events
```

For Ex, Run:

```
# zeek -C -r /zeek/training-res/capture.pcap misc/dump-events
```

HAND OFF TO KEITH



Take five minute break to stretch or play with dump-events



What Does Zeek Do?

Logs



Logs



```
[root@t                current]# ls
bhr.log                conn_s0.log
capture_loss.log       dhcp.log
conn-2.log             dns.log
conn_bulk.log          dpd.log
conn.log               files.log
conn_long.log          ftp.log
[root@t                current]#
```

http.log	notice.log	smtp.log	stderr.log
intel.log	ntp.log	snmp.log	stdout.log
known_certs.log	owamp.log	software.log	traceroute.log
known_hosts.log	react.log	ssh.log	tunnel.log
known_services.log	reporter.log	ssl.log	weird.log
notice_alarm.log	sip.log	stats.log	x509.log

- All the logs are written in ASCII log files (tsv format)
- Zeek generates the log files for the protocols it sees in your network traffic (more than 50 protocols currently parsed)
- Apart from **conventional protocol log** files, **interesting logs** pertaining to noticeable/statistical activity (weird.log, notice.log etc.)



Logs

- Pop to shell

```
# rm *.log
# zeek -C -r /zeek/training-res/ssl.pcap
```
- Show log header/footer
- Show conn.log
- Show files.log
- Show ssl.log

Alerting



Alerting



- ACTION_LOG - log to notice.log
- ACTION_EMAIL - email notices



Notices

Intel::Notice

Heartbleed::SSL_Heartbeat_Attack

Scan::Address_Scan

Scan::Port_Scan

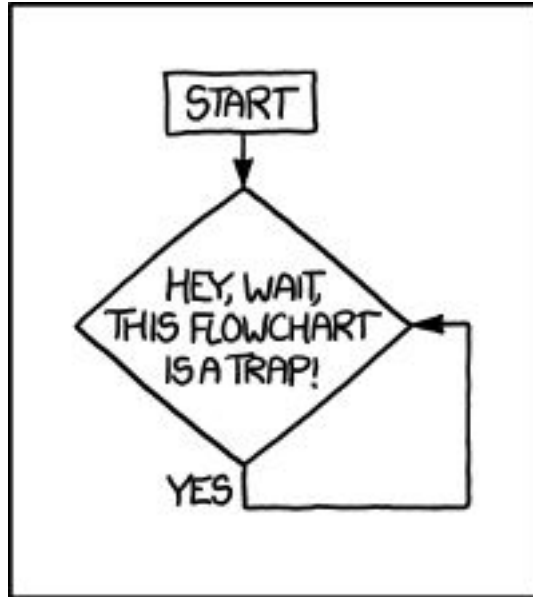
SSL::Certificate_Expired

SSH::Interesting_Hostname_Login

SSH::Watched_Country_Login

SSH::Password_Guessing

Custom Logic



HAND OFF TO FATEMA





Filesystem Layout

- Not an exhaustive tour!
- Default Path : `/opt/zeek/` or `/usr/local/zeek`
 - `bin/`
 - `etc/`
 - `include/` - ignoring today
 - `lib/` - ignoring today
 - `logs/`
 - `share/`
 - `spool/`



Filesystem Layout

- bin/
 - **zeek-cut**
 - Extract columns from zeek logs (non-JSON)
 - Convert Unix epoch
 - **zeek-wrapper**
 - Deprecation wrapper
 - ZEEK_IS_BRO environment variable
 - Define to rid yourself of deprecation warnings
 - **zeekctl**
 - Cluster management tool
 - **Zeek**
 - The zeek binary
 - **Zkg**
 - Zeek package manager



Filesystem Layout

- etc/
 - network.cfg
 - Define your local networks
 - node.cfg
 - Configure a cluster
 - zeekctl.cfg
 - Configure cluster management tool
 - zkg/config
 - Zkg config file



Filesystem Layout

- spool/
 - state.db
 - cluster worker state
 - zeekctl-config.sh
 - Cluster management variables
 - worker-1
 - .pid
 - .cmdline
 - .env_vars
 - logger
 - logs (current is just a symlink)



Filesystem Layout

- share/
 - zeek/base/
 - Base scripts
 - zeek/policy/
 - Additional scripts
 - zeek/site/
 - Site specific scripts



Filesystem Layout

- current directory
 - Default Log Path (CLI mode, logs overwritten each run)
- logs/
 - current
 - Default Log Path (cluster-mode)
 - YYYY-MM-DD
 - Logs archived to this dir, rotated hourly by default

Zeek Script Example



Break to look at policies in zeek/share

```
# cd /zeek/share/zeek  
# find . -name '*.zeek' | less
```



10 Minute Break

HAND OFF TO KEITH





Anatomy of a Zeek Script

Pop out & take a look at

```
# less /zeek/share/zeek/base/protocols/http/main.zeek
```



Customizing Zeek

- `/zeek/share/zeek/site/local.zeek`
 - local customizations
 - Can be done inline or separate script
 - enable or disable scripts
 - change variables



Customizing Zeek

Loading additional scripts:

```
# vi /opt/zeek/share/zeek/site/local.zeek
```

```
@load packet-fragments
```

```
zeek -h ($ZEEKPATH)
```

Zeek search path order :

- . (current directory)

- /opt/zeek/share/zeek

- /opt/zeek/share/zeek/policy

- /opt/zeek/share/zeek/site



Customizing Zeek

Loading additional scripts:

```
# nano /zeek/share/zeek/site/local.zeek
```

```
@load policy/protocols/http/header-names
```

```
zeek -h  
$ZEEKPATH
```

Zeek file search path order :

- . (current directory)
- /opt/zeek/share/zeek
- /opt/zeek/share/zeek/policy
- /opt/zeek/share/zeek/site



Customizing Zeek

- Zeek scripts have an export section.
 - Redefinable variables
 - main.zeek is best place to start looking

```
# less /zeek/share/zeek/base/protocols/http/main.zeek
```

```
export {
```

```
    # This setting changes if passwords used in  
    # Basic-Auth are captured or not.  
    option default_capture_password = F;
```

```
}
```



Customizing Zeek

Value

```
---  
redef SSH::password_guesses_limit = 15;
```

Table

```
---  
redef SSH::ignore_guessers += {  
    [192.168.1.2/32] = 192.168.1.20/32 };
```

Boolean

```
---  
redef HTTP::log_server_header_names = T;
```



Config Framework

Change values on the fly

```
module Test;
```

```
export {  
    option enable_feature = F;  
}
```

```
@redef Test::enable_feature = T;
```

Define config option file

```
---
```

```
redef Config::config_files += { "/path/to/config.dat" };  
[192.168.1.2/32] = 192.168.1.20/32 };
```

Format

```
---
```

```
[option name][tab/space][new value]
```

```
Test::enable_feature    T
```



Customizing Zeek

10 Minute Exercise!

```
# cp /zeek/share/zeek/site/local.zeek .
```

Log server header names in HTTP
`@load policy/protocols/http/header-names`

Log http usernames and passwords
`redef HTTP::default_capture_password = T;`

Start with : `/zeek/training-res/http-auth.pcap`

```
# zeek -C -r /zeek/training-res/http-auth.pcap local.zeek
```

HAND OFF TO FATEMA





Package Manager (zkg)

- Package Manager
 - zkg
 - <https://packages.zeek.org>
- Can add new scripts or analyzers
 - Analyzer examples :
 - QUIC
 - PerfSonar
 - Scripts :
 - BZAR
 - DHCP OUI data



Let's Use zkg!

- List all zeek packages and install a few
 - # zkg refresh
 - # zkg list all
 - # zkg install cve-2020-0601
- Check if they got installed
 - # zkg list installed
 - # ls /zeek/share/zeek/site
- Let's load the packages
 - Add *@load packages* in site/local.zeek



Let's Use zkg!

- zkg load/unload and info
 - # zkg unload cve-2020-0601
 - # zkg info cve-2020-0601
- Remove and uninstall all the packages
 - # zkg purge
 - # zkg list installed
- Other useful zkg commands
 - remove, pin, unpin, upgrade, refresh



Zkg Exercise

10 minutes..

Running Zeek with zkg installed packages, specifically with JA3.

```
# zkg install ja3
```

```
# zeek -C -r /zeek/training-res/ssl.pcap local
```

```
# less ssl.log
```



Zkg Exercise

Extract a file from a file transfer connection using file-extract [zkg package](#)

```
# zkg install file-extraction
```

Add all file types extraction to the config of the package:

```
# nano /zeek/share/zeek/site/packages/file-extraction/config.zeek
@load ./plugins/extract-all-files.zeek
```

```
# zeek -C -r extract.pcap local
```

What all files got extracted from the pcap?

```
# ls extract_files/
```



10 Minute Break

HAND OFF TO KEITH





Quick Log Processing Intro



General Approaches

Scope your searches.

- Less logs, less confusion.

- Also respect your user's privacy.

Baseline your network, if you can.

Iterative process.

- Build your command lines one command at a time.

Use CUID & FUID to pivot around logs.



Log Processing Tools

zeek-cut - Extracts columns from an ASCII zeek log on standard input.

```
$ zeek-cut ts id.orig_h host uri < http.log
```

1320279600.688672	192.168.2.76	www.zeek.org	/
1320279600.921091	192.168.2.76	www.zeek.org	/css/pygments.css
1320279600.924479	192.168.2.76	www.zeek.org	/css/print.css

```
$ zeek-cut -d ts id.orig_h host uri < http.log
```

2011-11-02T20:19:37-0400	192.168.2.76	slashdot.org	/
2011-11-02T20:19:37-0400	192.168.2.76	a.fsdn.com	/sd/topic1
2011-11-02T20:19:37-0400	192.168.2.76	a.fsdn.com	/sd/topic2

Log Processing Tools



sort - sort lines of text files

```
$ zeek-cut id.orig_h resp_bytes <conn.log |sort -rnk2 |head
```

```
172.16.238.131 13871
172.16.238.1 2887
172.16.238.131 1724
172.16.238.1 714
172.16.238.131 459
172.16.238.131 456
172.16.238.131 451
172.16.238.131 448
172.16.238.131 447
```

Log Processing Tools



uniq - report or filter out repeated lines in a file

```
$ zeek-cut id.orig_h id.resp_p <conn.log | sort | uniq -c | sort -rnk1
```

```
14 172.16.238.131 53
11 172.16.238.131 53
1 172.16.238.131 80
1 172.16.238.131 5353
1 172.16.238.131 53
1 172.16.238.131 53
1 172.16.238.131 22
1 172.16.238.131 123
```

Log Processing Tools



awk - pattern-directed scanning and processing language

Find hosts with http not on port 80

```
$ awk '$8 == "http" && $6 != 80 { print $0 }' conn.log
```

Total resp_bytes for host :

```
$ awk '{ if ( $3 ~ /192.168.1.1$/ ) { sum+=$10 } END { print sum } }'
```



Log Processing Exercise - 10 min

```
# zeek -C -r /zeek/training-res/capture.pcap local
```

- 1) Show the top 5 destination ports in descending order.
- 2) List the connections by increasing order of duration, i.e., the longest conns at end.
- 3) Find all connections that are longer than 5 seconds.
- 4) Find all IP addresses of web servers that send more than 1 KB back to a client.
- 5) Show a breakdown of the number of connections by service.

HAND OFF TO FATEMA





Notice Framework

- Notices are “interesting events” (as opposed to normal protocol logs)
- Notices have built-in suppression (to avoid repeated events)
 - Generally controlled/defined by the script writer
- Hooks allow modification of notices as they are passed through the framework
 - Use “break” to suppress a given notice, based on some characteristic (such as src IP)



Notice Exercise

- Detecting scan: Uncomment the following line in local.zeek:
nano /zeek/share/zeek/site/local.zeek
@load misc/scan
- Run against mal.pcap
zeek -C -r /zeek/training-res/mal.pcap local
- Look at newly generated notice.log file

HAND OFF TO KEITH





Intel Framework

Automatically watch traffic for malicious indicators

What can Zeek watch for?

ADDR

URL

SOFTWARE

EMAIL

DOMAIN

USER_NAME

FILE_HASH

FILE_NAME

CERT_HASH



Intel Framework

Where does it look?

Conn::IN_ORIG

Conn::IN_RESP

Files::IN_HASH

Files::IN_NAME

DNS::IN_REQUEST

DNS::IN_RESPONSE

SMTP::IN_RCPT_TO

SMTP::IN_FROM

SMTP::IN_TO

SMTP::IN_RECEIVED_HEADER

SMTP::IN_REPLY_TO

HTTP::IN_HOST_HEADER



Intel Framework

file changes trigger reload

ex. : `$ mv intel.new intel.dat`

but

Old entries are **not** removed without restart

but

`policy/frameworks/intel/removal`

Boolean (`meta.remove`) to remove entries on file update

Just deleting entry from file won't remove it from worker store

`policy/frameworks/intel/do_notice`

raise notices when Intel is seen



Intel Framework

1. Enable Intel FW in local.zeek:

```
@load frameworks/intel/seen  
@load frameworks/intel/do_notice
```

2. Intel.dat - define the intelligence data to match

```
#fields indicator indicator_type meta.source meta.do_notice  
zeek.org Intel::DOMAIN source1 T
```

3. Tell zeek about the intel file - load an intelligence file:

```
redef Intel::read_files += {  
    "/some/path/intel.dat",  
};
```



Intel Framework Exercise

Edit local.zEEK to enable intel FW:

```
# nano /zeek/share/zeek/site/local.zEEK  
  
@load frameworks/intel/seen  
redef Intel::read_files += {"/zeek/training-res/intel.dat",};
```

Run zeek against mal.pcap:

```
# zeek -C -r /zeek/training-res/mal.pcap local
```

Observe the type of indicator and seen.where fields:

```
# less intel.log
```

Need hints on Intel types? See : `share/zeek/base/frameworks/intel/main.zEEK`



More Zeek Resources

Zeek Documentation:

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

Zeek packages:

<https://packages.zeek.org/packages>

Zeek btest pcaps:

<https://github.com/zeek/zeek/tree/master/testing/btest/Traces>

Zeek Github source:

<https://github.com/zeek/zeek>

Zeek online:

<https://try.zeek.org/#/?example=hello>



Thanks for attending!
Enjoy Zeeking

Stay Connected



Website - [Zeek.org](https://zeek.org)

Mailing List - zeek@lists.zeek.org

Slack - <http://bit.ly/ZeekOrgSlackInvite>

Find out more ways to connect at: <https://zeek.org/community/>

