**Synopsis of Part-1 Goals:**

1. Installation: of Zeek IDS.
2. Configuration: Going through the important config files and options.
3. Running Zeek: in tap mode and reading from a pcap as well.
4. Log File analysis: Looking through the various log files generated.
5. Zeek signatures: Introduction to Signature framework.
6. Zeek scripting: Going through the Zeek policy and script land tree structure

**Some useful links**

[**https://www.zeek.org/**](https://www.google.com/url?q=https%3A%2F%2Fwww.bro.org%2F&sa=D&sntz=1&usg=AFQjCNG1g1mv63OvvNW2UPgysOV3IM-b-A)

[**https://www.zeek.org/documentation/index.html**](https://www.google.com/url?q=https%3A%2F%2Fwww.bro.org%2Fdocumentation%2Findex.html&sa=D&sntz=1&usg=AFQjCNEaAlbSAO59Zj-PgegvNzq_N2pAwA)

[**https://www.zeek.org/download/packages.html**](https://www.google.com/url?q=https%3A%2F%2Fwww.bro.org%2Fdownload%2Fpackages.html&sa=D&sntz=1&usg=AFQjCNEO3y4E6urWyQp44rfAKavxqq0QbQ)

[**https://www.zeek.org/sphinx/install/install.html**](https://www.google.com/url?q=https%3A%2F%2Fwww.bro.org%2Fsphinx%2Finstall%2Finstall.html&sa=D&sntz=1&usg=AFQjCNG6W9jpyrmU_wo-UdcbtKHrdNqVVQ)

[**https://www.zeek.org/sphinx/quickstart/index.html#id3**](https://www.google.com/url?q=https%3A%2F%2Fwww.bro.org%2Fsphinx%2Fquickstart%2Findex.html%23id3&sa=D&sntz=1&usg=AFQjCNETQy4Owx_RRc0csu7AesAE4rwgsQ)

1. **Installation of Zeek IDS:**

* Unzip the vm1-16-04 from the BSidesDE18 folder you just copied off the USB.
* Open it in Workstation or Fusion
* Start the VM.
* Login with username: zeek, and password: zeek####

In the vm1 VM install the zeek IDS,

* sudo sh -c "echo 'deb http://download.opensuse.org/repositories/security:/zeek/xUbuntu\_18.04/ /' > /etc/apt/sources.list.d/security:zeek.list"
* wget -nv https://download.opensuse.org/repositories/security:zeek/xUbuntu\_18.04/Release.key -O Release.key
* sudo apt-key add - < Release.key
* sudo apt-get update
* sudo apt-get install zeek

**2. Configuration of Zeek IDS:**

Configure the Run-Time Environment

* # export PATH=/opt/zeek/bin/:$PATH

A Minimal Starting Configuration:

These are the basic configuration changes to make for a minimal ZeekControl installation that will manage a single Zeek instance on the localhost:

In /opt/zeek/etc/node.cfg, set the right interface to monitor.

* # vi /opt/zeek/etc/node.cfg

[zeek]

type=standalone

host=localhost

interface=eth0        # change this according to your listening interface, find it by doing ifconfig.

In /opt/zeek/etc/networks.cfg, comment out the default settings and add the networks that Zeek will consider local to the monitored environment.

* # vi /opt/zeek/etc/networks.cfg

10.0.0.0/8          Private IP space

192.168.0.0/16      Private IP space

#128.175.0.0/16  If not behind nat, mention your local net

If not behind NAT, mention your local net

Since this is the first-time use of the shell, perform an initial installation of the ZeekControl configuration:

* # ethtool -K ens33 rx off tx off sg off tso off gso off gro off

**3. Running Zeek IDS:**

***Running Zeek - Tap Mode:***

* # zeekctl install
* # zeekctl start

While having Zeek running, bring up another shell to watch the logs it generated:

In another shell, do this:

* $ sudo -s
* # cd /opt/zeek/logs/current
* # ls

Note that while running zeek in tap mode, the default log dir always is /opt/zeek/logs/current/ dir.

You should see all the log files getting generated by zeek in corresponding protocol named log files: like http.log, dns.log etc

If there are errors while trying to start the Zeek instance, you can can view the details with the diag command.

If started successfully, the Zeek instance will begin analyzing traffic according to a default policy and output the results in /opt/zeek/logs/current.

Now start surfing the internet and visit web pages, so that ZEEK can see the traffic and start generating some log files.

While running Zeek, clone the github repo for getting the test-pcaps and other files:

(We will clone the repo in Zeek installation path:)

* # cd /opt/zeek/
* # git clone https://github.com/fatemabw/BSidesDE17-Bro.git

After some time, stop Zeek:

* # zeekctl stop

The log files will get rotated to the dir named with current date:

* # cd /opt/zeek/logs
* # ls

Take a look at the different log files that get generated, when we ran Zeek in tap mode.

***Running Zeek - Offline Mode (Reading from a pcap):***

Getting the pcaps:

* # cd /opt/zeek/logs
* # zeek -r /opt/zeek/BSidesDE17-Bro/test-pcaps/http.cap local
* # ls

 Note that when run in offline mode, the logs are generated in the local dir where you run zeek command.

You can later delete those log files in the local dir (in our case we ran zeek in /opt/zeek/logs/ dir).

**4. ZEEK Log files analysis:**

Zeek Log Files:

By default, logs are written out in human-readable (ASCII) format and data is organized into columns (tab-delimited). Logs that are part of the current rotation interval are accumulated in /opt/zeek/logs/current/ (if Zeek is not running, the directory will be empty). For example, the http.log contains the results of Zeek HTTP protocol analysis.   
  
Some logs are worth explicit mention:  
  
**conn.log:** Contains an entry for every connection seen on the wire, with basic properties such as time and duration, originator and responder IP addresses, services and ports, payload size, and much more. This log provides a comprehensive record of the network’s activity.  
  
**notice.log:** Identifies specific activity that Zeek recognizes as potentially interesting, odd, or bad. In Zeek-speak, such activity is called a “notice”. Might not be generating when you are on NAT, as Zeek won't be seeing anything worth noticing. ut a really good log file to search for scanners, sql-injection attempts, brute-force attempts etc.  
  
**software.log:** Zeek tries to identify which software versions other clients talking on the network are running.  
  
**known\_services.log:** Zeek also tries to figure out the client services running on the systems that are trying to connect.

**weird.log:** Another worth mentioning log file, as the name suggests, zeek tries to find out the network activity that doesn't comply with RFCs and other standards for different protocols. Examples include: bad\_TCP\_checksum, data\_before\_established etc.  
  
By default, ZeekControl regularly takes all the logs from /opt/zeek/logs/current and archives/compresses them to a directory named by date, e.g. /opt/zeek/logs/2011-10-06. The frequency at which this is done can be configured via the LogRotationInterval option in $/opt/zeek/etc/zeekctl.cfg.  
  
**Zeek logs Cheatsheet**: https://github.com/corelight/zeek-cheatsheets/blob/master/Corelight-Zeek-Cheatsheets-2.5.pdf  
**Look at some logs on:** [**http://try.zeek.org**](http://try.bro.org)

**5. ZEEK Signature Example:**

Simple rules to define:

<Conditions to match>

<Action to perform upon match>

**Syntax:**  
signature <sig-name>{  
<header field> <comparison op> <value>  
event <msg>  
}  
  
**Header fields can be:**  
Source/Dest. IP address (single, list, CIDR notation)  
Source/Dest. port (single, list)  
IP Protocol: TCP, UDP, ICMP  
  
**Comparison op can be:** ==, !=, <, <=, >, >=  
  
**Value list can be:** Single or multiple values.  
  
**Steps to create your first signature in ZEEK:**

Create a file called **proto-port.sig** in **/opt/zeek/share/zeek/site** dir:

* # vi /opt/zeek/share/zeek/site/proto-port.sig

signature proto-port{

ip-proto == tcp

dst-ip == 192.168.0.0/16

event "CnC to bad IP range"

}

Edit /opt/zeek/share/zeek/site/local.zeek

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

@load-sigs site/proto-port.sig

Tell Zeek to reload in zeekctl:

* # zeekctl install
* # zeekctl start

Generate some traffic to trigger the signature. (surf to internet from your machine and zeek should log the event)

Go to /opt/zeek/logs/current and look for the logs generated by the signature script, i.e. look for signatures.log

If no signatures.log is generated, try to do this:

* # ssh root@192.168.1.10

and see if there are logs for signatures.log

After some time stop Zeek:

* # zeekctl stop

Learn more about signatures: [http://www.zeek.org/sphinx-git/frameworks/signatures.html](http://www.google.com/url?q=http%3A%2F%2Fwww.bro.org%2Fsphinx-git%2Fframeworks%2Fsignatures.html&sa=D&sntz=1&usg=AFQjCNHv79UBaZycviykVs2SDny1kK0Baw)

**6. ZEEK Script Example:**

A good reference:  [https://www.zeek.org/sphinx/scripting/index.html](https://www.google.com/url?q=https%3A%2F%2Fwww.bro.org%2Fsphinx%2Fscripting%2Findex.html&sa=D&sntz=1&usg=AFQjCNHbkPUcEZuSh910UFBOI1l8h9nrzQ)

Next let's play with some of the Zeek scripting Framework. It allows you to develop code in Zeek's robust scripting language,

Let's take our above signature detection forward. Let's say we want to raise a NOTICE (i.e we want to log an entry in notice.log) whenever our signature gets triggered.

Create a file called **proto-port.zeek** in **/opt/zeek/share/zeek/site** dir

* # vi /opt/zeek/share/zeek/site/proto-port.zeek

@load base/frameworks/notice

@load base/frameworks/signatures/main

redef Signatures::ignored\_ids += /proto-port/;

module ProtoPort;

export {

        redef enum Notice::Type += {

                ## Raised when the condition in proto-port sig is true.

                Testing\_Signature,

        };

  }

event signature\_match(state: signature\_state, msg: string, data: string) &priority=-5

  {

        if ( /proto-port/ !in state$sig\_id ) return;

        NOTICE([$note=ProtoPort::Testing\_Signature,

                        $msg=fmt("Testing the signature raising the notice "),

                        $sub=data,

                        $conn=state$conn,

                        $identifier=fmt("%s%s", state$conn$id$orig\_h,

                                        state$conn$id$resp\_h)]);

  }

Edit /opt/zeek/share/zeek/site/local.zeek

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

@load site/proto-port.zeek

Tell Zeek to reload in zeekctl

* # zeekctl install
* # zeekctl start

Generate some traffic to trigger the signature. (surf to internet from your machine and zeek should log the event)

* # ssh root@192.168.1.10

Go to /opt/zeek/logs/current and look for the logs generated by the signature script, i.e. look for notice.log

After some time stop Zeek:

* # zeekctl stop

Go to : [http://try.zeek.org/#/tryzeek/saved/162245](http://www.google.com/url?q=http%3A%2F%2Ftry.bro.org%2F%23%2Ftrybro%2Fsaved%2F162245&sa=D&sntz=1&usg=AFQjCNGRETjprv67YzLPLi5A-U5giCFsig)  
And try out some examples, that will make you familiar with Zeek's scripting Framework ...  
You can also, copy and paste those scripts in your local Zeek instance to play around with it on your machine and network traffic ...