**Synopsis of Part-2 Goals:**

1. ZeroAccess example of Signature/Script framework
2. NOTICE FW: Generating various types of notices - Scan detection,
3. NOTICE FW: Generating various types of notices SSH bruteforce detection)
4. INTEL FW: Generating intel logs based on the IOCs traced on the network

**Some useful links**

Zeek Frameworks: [**https://www.zeek.org/sphinx/frameworks/index.html**](https://www.google.com/url?q=https%3A%2F%2Fwww.bro.org%2Fsphinx%2Fframeworks%2Findex.html&sa=D&sntz=1&usg=AFQjCNGg0wV9ftVaWVSSCTqKBRhdLFhGRg)

Notice FW: [**https://www.zeek.org/sphinx/frameworks/notice.html**](https://www.google.com/url?q=https%3A%2F%2Fwww.bro.org%2Fsphinx%2Fframeworks%2Fnotice.html&sa=D&sntz=1&usg=AFQjCNFjQCmqjtWBviRei0d-meES-Ds_eA)

Intel FW: [**https://www.zeek.org/sphinx/frameworks/intel.html**](https://www.google.com/url?q=https%3A%2F%2Fwww.bro.org%2Fsphinx%2Fframeworks%2Fintel.html&sa=D&sntz=1&usg=AFQjCNHsZPn_sDN4zZexfUwYjc0xXnL2cQ)

1. **Detecting ZeroAccess using Signature and Script FW:**

Zeroaccess is a dangerous rootkit and has been circulating for quite some time.  
You can read more about the malware here: [http://nakedsecurity.sophos.com/zeroaccess2/](http://www.google.com/url?q=http%3A%2F%2Fnakedsecurity.sophos.com%2Fzeroaccess2%2F&sa=D&sntz=1&usg=AFQjCNHHjWeJdvB_ZhIUP01rSshzw5NfQA)  
The zeroaccess C2 Protocol has some static elements we can examine:

Let's go ahead and replay the traffic using our detection script:

* # cd /opt/zeek/logs
* # rm \*.log
* # zeek -r /opt/zeek/BSidesDE17-Bro/test-pcaps/BIN\_ZeroAccess\_3169969E91F5FE5446909BBAB6E14D5D\_2012-10.pcap   /opt/zeek/BSidesDE17-Bro/ZeroAccess  local
* # less notice.log

1. **Detecting Scanners using NOTICE FW:**

* # rm \*.log
* # zeek -r /opt/zeek/BSidesDE17-Bro/test-pcaps/nmap\_standard\_scan local

Enable the scan script in local.zeek if no results get logged in notice.log regarding the scan.

Ignore these warning:

WARNING: No Site::local\_nets have been defined.  It's usually a good idea to define your local networks.  
1391767837.328537 warning in /opt/zeek/share/zeek/base/misc/find-filtered-trace.zeek, line 48: The analyzed trace file was determined to contain only TCP control packets, which may indicate it's been pre-filtered.  By default, Zeek reports the missing segments for this type of trace, but the 'detect\_filtered\_trace' option may be toggled if that's not desired.

* # less notice.log
* # less conn.log
* # less conn.log | zeek-cut id.orig\_h | sort -u
* # less conn.log | zeek-cut id.orig\_p | sort -u
* # less conn.log | zeek-cut id.resp\_h | sort –u
* # less conn.log | zeek-cut id.resp\_p | sort -u | wc -l

Detecting live:

Run an nmap TCP scan from your system

* # zeekctl deploy
* # nmap -Pn -sT 192.168.0.0/16
* # less notice.log | grep scan

Install nmap using “apt-get install nmap” if nmap isn’t currently installed.

1. **Detecting SSH BruteForcing using NOTICE FW:**

* # rm \*.log
* # zeek -r /opt/zeek/BSidesDE17-Bro/test-pcaps/ssh.pcap local

Ignore if getting a WARNING: No Site::local\_nets have been defined.  It's usually a good idea to define your local networks.

It can be ignored, Zeek is saying yelling about defining local subnets, which is a good practice in prod environment.

* # less notice.log
* # less ssh.log
* # less conn.log

Incident triaging:

Alert from notice.log: “SSH::Password\_Guessing”

Get the Offending IP: 172.16.238.1

Look up in ssh.log to see if any successful attempts from the IP:

* # less ssh.log | grep 172.16.238.1 | grep T

Contact the admins/owners of the machines using the 127.16.238.168, 238.129 and 238.136 IPs to ask them to verify if the connections were successful.

1. **Detecting IOCs in traffic using INTEL FW:**

Enable the Intel FW in Zeek:

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

# Enabling BIF

@load frameworks/intel/seen

@load frameworks/intel/do\_notice

@load base/frameworks/intel/files.zeek

# Enabling ZEEK integration with CIF

@load policy/integration/collective-intel

# Custom script - Input for Intel FW

@load site/intel-input-read.zeek

A small script to read the IOCs intel files:

* # vi /opt/zeek/share/zeek/site/intel-input-read.zeek

@load frameworks/files/hash-all-files

                      redef Intel::read\_files += {

                          "/opt/zeek/bro-IOC.intel"

                     };

Copy the zeek-IOCs.intel file, containing Zeek formatted intel to the following location, and run zeek:

* # cp /opt/zeek/BSidesDE17-Bro/bro-IOC.intel  /opt/zeek/
* # vi /opt/zeek/bro-IOC.intel

\* delete the spaces between the fields, and make them tab separated\*

* # cd /opt/zeek/logs
* # rm \*.log
* # zeek -r /opt/zeek/BSidesDE17-Bro/test-pcaps/http.cap local
* # less intel.log