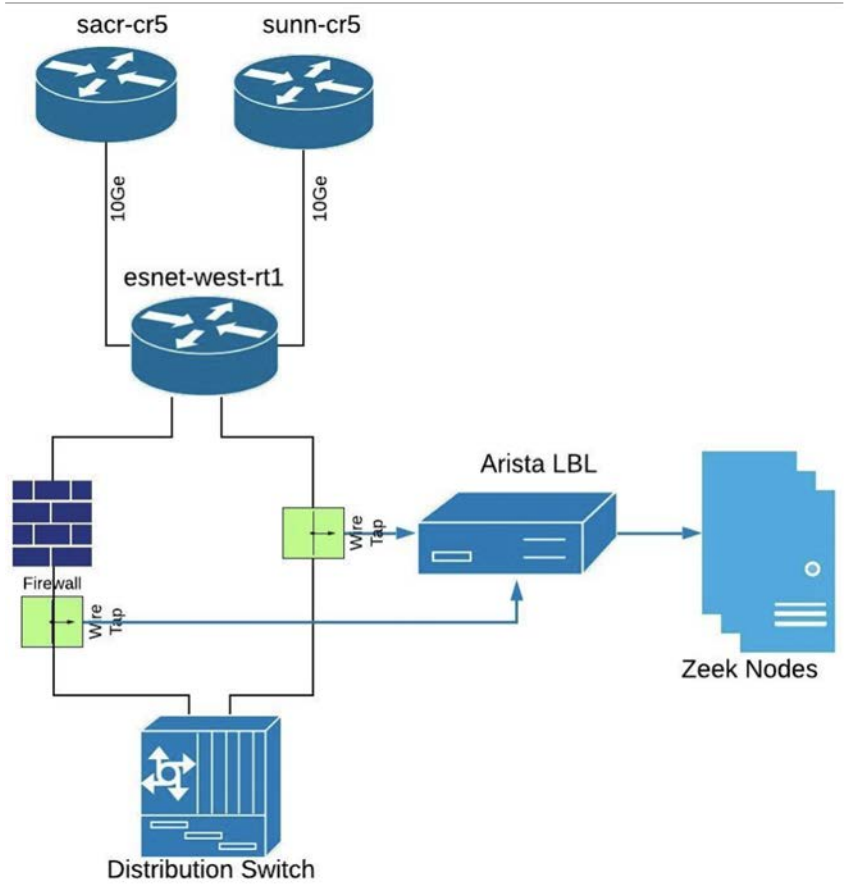
Zeek known services classification - ZTA edition

Fatema Bannat Wala
Security Engineer @ ESnet/LBNL

Esnet Architecture



What are known services in Zeek?

- An active service is defined as an IP address and port of a server for which
  - A TCP handshake (SYN+ACK) is observed, Or
  - assumed to have been done in the past (started seeing packets mid-connection, but the server is actively sending data), Or
  - sent at least one UDP packet.
- If a protocol name is found/known for service, that will be logged, but services whose names can’t be determined are also still logged.

What are known services in Zeek?

- Zeek generates known_services.log file based on the pre-loaded script policy/protocols/conn/known-services.zeek

Ex:

$cat known_services.log

#fields	ts	host	port_num	port_proto	service	
#types	time	addr	port	enum	set[string]	
1665718175.791134		10.20.0.111		80	tcp	HTTP
1665718175.880135		10.20.0.130		443	tcp	SSL
1665718175.880154		10.20.0.130		22	tcp	SSH
1665718175.880198		10.20.0.130		123	udp	NTP

How known services detected in Zeek?

```
## The hosts whose services should be tracked and logged.
## See :zeek:type:`Host` for possible choices.
option service_tracking = LOCAL_HOSTS;
```

```
function known_services_done(c: connection)
    {
    local id = c$id;

    if ( ! addr_matches_host(id$resp_h, service_tracking) )
        return;
```

Checks if the Dest IP is in LOCAL_HOSTS

known services for east-west traffic

- Problem assessing the attack surface
  - Open to the internet? Or
  - Open to the internal network?

conn.log:

```
1665713319.241928 CW2WfF3Mz3XSBIbdy1  198.129.x.x 35470 198.128.y.y 22 tcp ssh
0.05614495277404785 1461 1250 SF  0 ShADTdtaFf 16 3770 14 3244 zeek-west-w7
```

known_services.log:

```
1665713319.128558 198.128.y.y 22 tcp SSH
```
-> open to only internal net

Custom Known Services

- Add a flag is_orig_local to the known_services.log

```
## The hosts whose services should be tracked and logged.
## See :zeek:type:`Host` for possible choices.
option service_tracking = LOCAL_HOSTS;
```

```
function known_services_done(c: connection)
    {
    local id = c$id;

    if ( ! addr_matches_host(id$resp_h, service_tracking) )
        return;
```

```
local local_orig: bool;

if ( c$id$orig_h in Site::local_nets )
    local_orig = T;
else
    local_orig = F;

local info = ServicesInfo($ts = network_time(), $host = id$resp_h,
                          $port_num = id$resp_p,
                          $port_proto = get_port_transport_proto(id$resp_p),
                          $service = tempservs,
                          $is_local_orig = local_orig );
```

known services for east-west traffic

- Now we can get a terse list of services that are “only” open to the internet..

Known_services.log stats

```
Is_orig_local =>

Values   Count     %
True     2,240    96.76%
False    75       3.24%   -> open to the internet
```

Egress traffic filtering - Zero Trust

- So far, filtering the inbound connections based on the services doesn’t need to be open to the internet
- Egress traffic filtering - Restrict the outbound access to the internet based on what is needed and what is not
- How? - Figure out what services are required access, block rest on a network firewall
- Solution? - Use Zeek to detect known outbound services

Known services outbound detection - Zeek

- Known services outbound
  - checks for id.resp_h NOT to be in Local_hosts

```
function KnownOut::known_services_done(c: connection)
    {
    local id = c$id;

    if ( addr_matches_host(id$resp_h, Known::service_tracking) )
        return;
```

Custom known services, but flipped!

Known services detection - Zeek

Use-cases:

Case	Orig IP	Resp IP	IS_ORIG_LOCAL	Logging	service
1	LOCAL	LOCAL	TRUE	known_services.log	LOCAL/INBOUND
2	INTERNET	LOCAL	FALSE	known_services.log	INTERNET/INBOUND
3	LOCAL	INTERNET	TRUE	known_services_outbound.log	LOCAL/OUTBOUND
4	INTERNET	INTERNET	FALSE	known_services_outbound.log	INTERNET/OUTBOUND

Local hosts/services open to the internet

Internet hosts/services accessed by the local hosts

case no. 4 should never happen, but if does, then it will be logged.

Interesting Investigations - Egress traffic

- Statistical summary
  - Only ~12-15 services detected outbound
  - Investigated those services, resulted in interesting findings!

service{}	count	percent	is_local_orig
DNS	157404	73.384555	T
SSL	35870	16.723234	T
NTP	1776	0.828003	T
HTTP	1534	0.715178	T
SSH	250	0.116554	T
SMTP	160	0.074595	T
AYIYA	90	0.04196	T
OWAMP	78	0.036365	T
FTP	7	0.003264	T
IRC	1	0.002331	T

Investigation #1 - Outbound HTTP connections

- Seen in the traffic: Most of our ubuntu servers were connecting to “security.ubuntu.com” for updates

- Cause: The source lists running had defaults debian repos enabled that pointed to the security.ubuntu.com for updates

Investigation #1 - Outbound HTTP connections

- Reason: Turns out a config error in ansible that deployed the repo settings on those servers.

- Resolution: A ticket to the INF team to fix the typo in the ansible code and point them to linux.mirrors.es.net.

Investigation #2 - Outbound IRC connection

- Seen in the traffic: One of our servers seen connecting to some IP in China on port 6669.

conn.log:

```
1662952332.859229   CotCBF3ujyxiin97U8  198.129.224.35        80        118.78.68.8
          6669      tcp        irc       23.647002 392804    147       OTH       -         -
          0         HadADTT    49        66412     137       7715      -
```

- Cause: Zeek missed initial syn of the TCP connection hence the connection was detected as outbound.
- But is it really IRC?

Investigation #2 - Outbound IRC connection

- weirds to the rescue!!!
- weird.log is all about invalid content in IRC, which is true bcoz the connection isn't actually IRC:

```
#types    time  string  addr  port  addr  port  string      string  bool string  string
198.129.224.35 80   118.78.68.8 6669    connection_originator_SYN_ack -  F zeek TCP
198.129.224.35 80   118.78.68.8 6669    irc_line_too_short  -          F zeek IRC
198.129.224.35 80   118.78.68.8 6669    irc_invalid_reply_number -     F zeek IRC
198.129.224.35 80   118.78.68.8 6669    irc_invalid_command -          F zeek IRC
198.129.224.35 80   118.78.68.8 6669    irc_line_size_exceeded -       F zeek IRC
```

<It was actually a inbound HTTP request to linux.mirrors.es.net to get CentOS 7 iso>

Investigation #2 - Outbound IRC connection

Resolution: There was a PR by Vern to actually flip the connection if the initial syn is lost but the connection looks legit:

v5.0.2 and older:

```
1662952332.859229   CotCBF3ujyxiin97U8 198.129.224.35  80  118.78.68.8 6669    tcp
irc        23.647002 392804     147        OTH        -           -           0           HadADTT
           49         66412      137        7715       -
```

v5.1.0-rc1:

```
1662952332.859229   CZin6InXYMITfpPVj  118.78.68.8 6669    198.129.224.35  80  tcp
irc        23.647002 147        392804     OTH        -           -           0           ^hADadtt
           138        7783       48         66344      -
```

But, the weirds reported were same, as it was still detected as IRC..

Investigation #2 - Outbound IRC connection

Resolution: Thanks to JAzoff for helping troubleshoot.. :-)

Submitted a bug report to fix the analyzer_confirmation once the connections are flipped.

The bug has been fixed and now the application protocol is correctly detected.

Zeek v5.2:

```
1662952332.859229  CZhhjInXYytKjyhSd  118.78.68.8 6669   198.129.224.35  80  tcp
http      23.647002 147       392804    OTH       -         -         0    ^hADadtt
          138       7783      48        66344     -
```

Summary

- Still investigating some potential miss-configurations with the network tapping.
- A decent idea of the internet services our systems are using.
- Nice to verify SSH/SMTP and other services work as expected.

Where to find the scripts?

Available via zkg install:

zkg install Zeek-Known-Services-With-OrigFlag

zkg install zeek-outbound-known-services-with-origflag

OR

Scripts:

https://github.com/esnet-security/Zeek-Known-Services-With-OrigFlag

https://github.com/esnet-security/zeek-outbound-known-services-with-origflag

Thanks for attending!
Questions?