Welcome to

Suricata, Zeek and DNS Capture

An Evening with Packets

Henrik Lund Kramshøj hlk@zencurity.com

Slides are available as PDF, kramse@Github
suricatazeek-workshop.tex in the repo security-courses

Goal

Don’t Panic!

Spend an evening using automated packet dissecting tools, multiple tools:

Try different ways to parse packets

Try The Zeek Network Security Monitor

Try Suricata a network threat detection engine

How to get started using packet capture tools in an enterprise

We try to do a lot, feel free to focus on specific parts

Packet sniffing tools

Tcpdump for capturing packets

Wireshark for dissecting packets manually with GUI

Zeek Network Security Monitor

Snort, old timer Intrusion Detection Engine (IDS)

Suricata, modern robust capable of IDS and IPS (prevention)

ntopng High-speed web-based traffic analysis

Maltrail Malicious traffic detection system https://github.com/stamparm/MalTrail

Often a combination of tools and methods used in practice

Full packet capture big data tools also exist

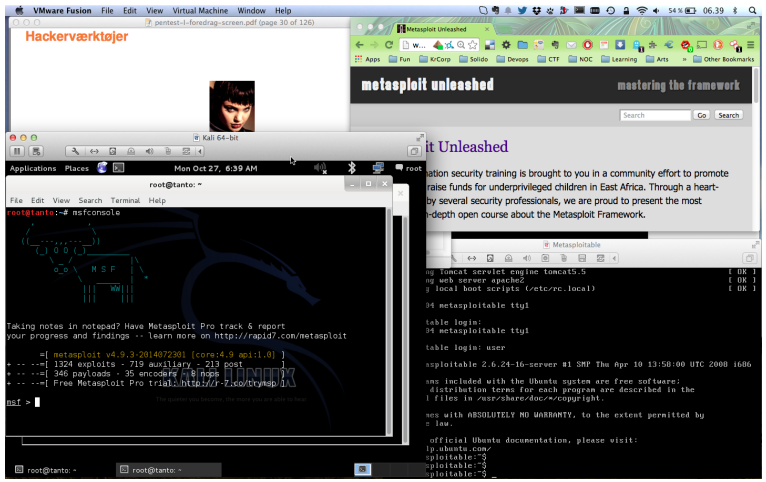
Kali Linux the pentest toolbox

Kali `http://www.kali.org/` brings together 100s of tools

100.000s of videos on YouTube alone, searching for kali and $TOOL

Use this to generate bad traffic

Hackerlab setup



- Hardware: most modern laptop CPUs have virtualization support
  May need to enable it in BIOS
- Software: use your favorite operating system, Windows, Mac, Linux

- Virtualization software: VMware, Virtual box, choose your poison
- Hackersoftware: Kali as a Virtual Machine `https://www.kali.org/`
- Install sniffing VM - put into *bridge mode*

Your lab setup

- Go to GitHub, Find user Kramse, click through security-courses, courses, suricatazeek and download the PDF files for the slides and exercises:

  `https://github.com/kramse/security-courses/tree/master/courses/networking/suricatazeek-workshop`

- Get the lab instructions, from

  `https://github.com/kramse/kramse-labs/tree/master/suricatazeek`

Exercise

Now lets do the exercise

Check your VM, run Ansible

which is number 1 in the exercise PDF.

What happens today?

Think like a blue team member find hacker traffic

Get basic tools running

Improve situation

- See where the data end up
- What kind of data and metadata can we extract
- How can we collect and make use of it
- Databases and web interfaces, examples shown
- Consider what your deployment could be

Today focus on the lower parts, but user interfaces are important too

Security devops

We need devops skillz in security

automate, security is also big data

integrate tools, transfer, sort, search, pattern matching, statistics, ...

tools, languages, databases, protocols, data formats

Use GitHub! So many libraries and programs that can help, maybe solve 90% of your problem, and you can glue the

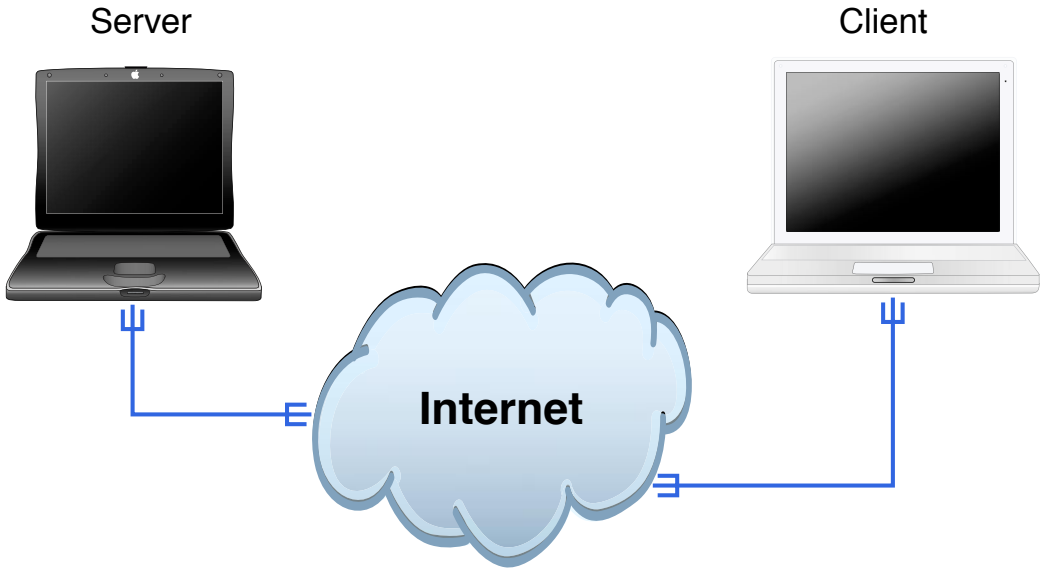
rest together

Example introductions:

- Seven languages/database/web frameworks in Seven Weeks
- Elasticsearch the definitive guide

We are all Devops now, even security people!

Internet today



Clients and servers

Roots in academia

Protocols more than 20 years old

HTTP is becoming encrypted, but a lot other traffic is not

OSI and Internet models

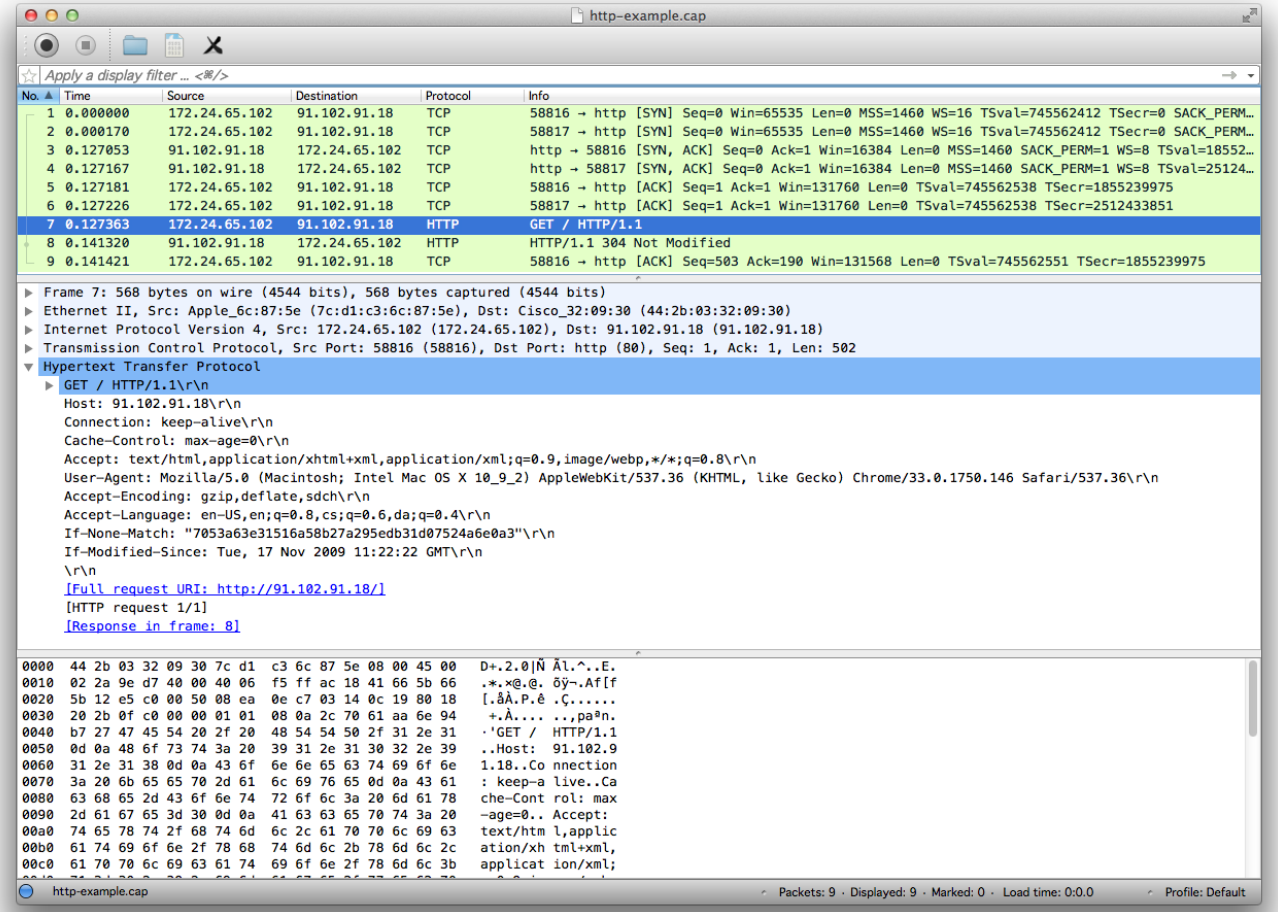
OSI Reference Model

Application
Presentation
Session
Transport
Network
Link
Physical

Internet protocol suite

Applications HTTP, SMTP, FTP, SNMP,	NFS
	XDR
	RPC
TCP UDP	
IPv4 IPv6 ICMPv6 ICMP	
ARP RARP MAC	
Ethernet token-ring ATM ...	

Using Wireshark


http-example.cap
Apply a display filter ... <⌘/>
No. Time Source Destination Protocol Info
1 0.000000 172.24.65.102 91.102.91.18 TCP 58816 → http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=745562412 TSecr=0 SACK_PERM…
2 0.000170 172.24.65.102 91.102.91.18 TCP 58817 → http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=745562412 TSecr=0 SACK_PERM…
3 0.127053 91.102.91.18 172.24.65.102 TCP http → 58816 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=8 TSval=18552…
4 0.127167 91.102.91.18 172.24.65.102 TCP http → 58817 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=8 TSval=25124…
5 0.127181 172.24.65.102 91.102.91.18 TCP 58816 → http [ACK] Seq=1 Ack=1 Win=131760 Len=0 TSval=745562538 TSecr=1855239975
6 0.127226 172.24.65.102 91.102.91.18 TCP 58817 → http [ACK] Seq=1 Ack=1 Win=131760 Len=0 TSval=745562538 TSecr=2512433851
7 0.127363 172.24.65.102 91.102.91.18 HTTP GET / HTTP/1.1
8 0.141320 91.102.91.18 172.24.65.102 HTTP HTTP/1.1 304 Not Modified
9 0.141421 172.24.65.102 91.102.91.18 TCP 58816 → http [ACK] Seq=503 Ack=190 Win=131568 Len=0 TSval=745562551 TSecr=1855239975
Frame 7: 568 bytes on wire (4544 bits), 568 bytes captured (4544 bits)
Ethernet II, Src: Apple_6c:87:5e (7c:d1:c3:6c:87:5e), Dst: Cisco_32:09:30 (44:2b:03:32:09:30)
Internet Protocol Version 4, Src: 172.24.65.102 (172.24.65.102), Dst: 91.102.91.18 (91.102.91.18)
Transmission Control Protocol, Src Port: 58816 (58816), Dst Port: http (80), Seq: 1, Ack: 1, Len: 502
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
Host: 91.102.91.18\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.146 Safari/537.36\r\n
Accept-Encoding: gzip,deflate,sdch\r\n
Accept-Language: en-US,en;q=0.8,cs;q=0.6,da;q=0.4\r\n
If-None-Match: "7053a63e31516a58b27a295edb31d07524a6e0a3"\r\n
If-Modified-Since: Tue, 17 Nov 2009 11:22:22 GMT\r\n
\r\n
[Full request URI: http://91.102.91.18/]
[HTTP request 1/1]
[Response in frame: 8]
0000 44 2b 03 32 09 30 7c d1 c3 6c 87 5e 08 00 45 00 D+.2.0|Ñ Ãl.^..E.
0010 02 2a 9e d7 40 00 40 06 f5 ff ac 18 41 66 5b 66 .*.x@.@. õÿ¬.Af[f
0020 5b 12 e5 c0 00 50 08 ea 0e c7 03 14 0c 19 80 18 [.åÀ.P.è .Ç.....
0030 20 2b 0f c0 00 00 01 01 08 0a 2c 70 61 aa 6e 94 +.À.... ..,paªn.
0040 b7 27 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 ·'GET / HTTP/1.1
0050 0d 0a 48 6f 73 74 3a 20 39 31 2e 31 30 32 2e 39 ..Host: 91.102.9
0060 31 2e 31 38 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 1.18..Co nnection
0070 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 43 61 : keep-a live..Ca
0080 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6d 61 78 che-Cont rol: max
0090 2d 61 67 65 3d 30 0d 0a 41 63 63 65 70 74 3a 20 -age=0.. Accept:
00a0 74 65 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 text/htm l,applic
00b0 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c ation/xh tml+xml,
00c0 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b applicat ion/xml;
http-example.cap
Packets: 9 · Displayed: 9 · Marked: 0 · Load time: 0:0.0
Profile: Default

https://www.wireshark.org

Exercise

Now lets do the exercise

Wireshark and tcpdump

which is number 2 in the exercise PDF.

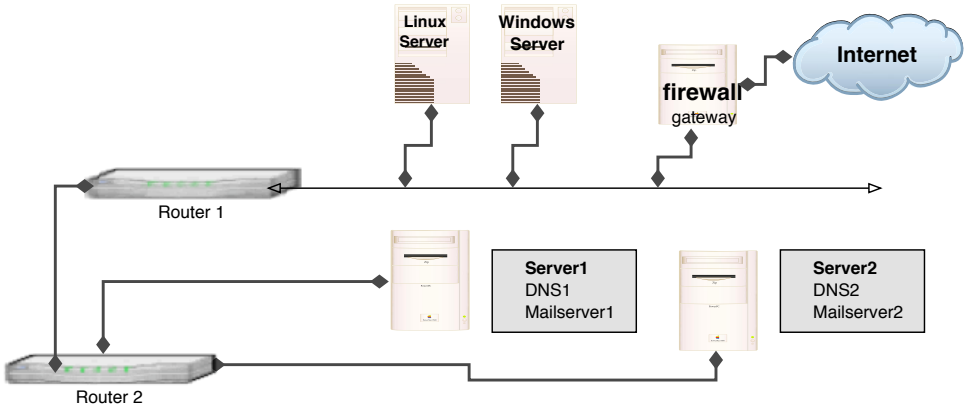
Exercise

Now lets do the exercise

Capturing network packets

which is number 3 in the exercise PDF.

Network mapping



Using traceroute and similar programs it is often possible to make educated guess to network topology

Time to live (TTL) for packets are decreased when crossing

a router

when it reaches zero the packet is timed out, and ICMP mes-sage sent back to source

Default Unix traceroute uses UDP, Windows tracert use ICMP

traceroute – UDP

```
 #  tcpdump -i en0 host 10.20.20.129 or host 10.0.0.11
tcpdump: listening on en0
23:23:30.426342 10.0.0.200.33849 > router.33435: udp 12  [ttl 1]
23:23:30.426742 safri > 10.0.0.200:  icmp: time exceeded in-transit
23:23:30.436069 10.0.0.200.33849 > router.33436: udp 12  [ttl 1]
23:23:30.436357 safri > 10.0.0.200:  icmp: time exceeded in-transit
23:23:30.437117 10.0.0.200.33849 > router.33437: udp 12  [ttl 1]
23:23:30.437383 safri > 10.0.0.200:  icmp: time exceeded in-transit
23:23:30.437574 10.0.0.200.33849 > router.33438: udp 12
23:23:30.438946 router > 10.0.0.200: icmp: router  udp port 33438 unreachable
23:23:30.451319 10.0.0.200.33849 > router.33439: udp 12
23:23:30.452569 router > 10.0.0.200: icmp: router  udp port 33439 unreachable
23:23:30.452813 10.0.0.200.33849 > router.33440: udp 12
23:23:30.454023 router > 10.0.0.200: icmp: router  udp port 33440 unreachable
23:23:31.379102 10.0.0.200.49214 > safri.domain:  6646+ PTR?
200.0.0.10.in-addr.arpa. (41)
23:23:31.380410 safri.domain > 10.0.0.200.49214:  6646 NXDomain* 0/1/0 (93)
14 packets received by filter
```

```
0 packets dropped by kernel
```

Low TTL, UDP, high ports above 33000 = Unix traceroute signature

Experiences gathered

Lots of information

Reveals a lot about the network, operating systems, services etc.

I use a template when getting data

- Respond to ICMP: □ echo, □ mask, □ time
- Respond to traceroute: □ ICMP, □ UDP
- Open ports TCP og UDP:
- Operating system:

- ... (banner information)

Beware when doing scans it is possible to make routers, fire-walls and devices perform badly or even crash!

The Zeek Network Security Monitor

Why Choose Zeek? Zeek is a powerful network analysis framework that is much different from the typical IDS you may know.

Adaptable
Zeek's domain-specific scripting language enables site-specific monitoring policies.

Efficient
Zeek targets high-performance networks and is used operationally at a variety of large sites.

Flexible
Zeek is not restricted to any particular detection approach and does not rely on traditional signatures.

Forensics
Zeek comprehensively logs what it sees and provides a high-level archive of a network's activity.

In-depth Analysis
Zeek comes with analyzers for many protocols, enabling high-level semantic analysis at the application layer.

Highly Stateful
Zeek keeps extensive application-layer state about the network it monitors.

Open Interfaces
Zeek interfaces with other applications for real-time exchange of information.

Open Source
Zeek comes with a BSD license, allowing for free use with virtually no restrictions.

The Zeek Network Security Monitor is not a single tool, more of a powerful network analysis framework (former name Bro)

Source `https://www.zeek.org/`, redirects to `https://www.bro.org/zeek.html`

Zeek scripts

```
global dns_A_reply_count=0;
global dns_AAAA_reply_count=0;
...
event dns_A_reply(c: connection, msg: dns_msg, ans: dns_answer, a: addr)
        {
        ++dns_A_reply_count;
        }

event dns_AAAA_reply(c: connection, msg: dns_msg, ans: dns_answer, a: addr)
        {
        ++dns_AAAA_reply_count;
        }
```

source: dns-fire-count.bro from

https://github.com/LiamRandall/bro-scripts/tree/master/fire-scripts https://www.bro.org/sphinx-git/script-reference/scripts.html

Exercise

Now lets do the exercise

Zeek on the web

which is number 4 in the exercise PDF.

Exercise setup

We will use a combination of your virtual servers, my switch hardware and my virtual systems.

There will be sniffing done on traffic!
Don't abuse information gathered

We try to mimic what you would do in your own networks during the exercises.

Another way of running exercises might be:
`https://github.com/jonschipp/ISLET`

Recommended and used by Zeek and Suricata projects.

Get Started with Zeek

To run in “base” mode: `bro -r traffic.pcap`

To run in a “near broctl” mode: `bro -r traffic.pcap local`

To add extra scripts: `bro -r traffic.pcap myscript.bro`

Note: the project was renamed from Bro to Zeek in Oct 2018

Bro demo: Run

```
// install zeek first
kunoichi:~ root# broctl
Hint: Run the broctl "deploy" command to get started.

Welcome to BroControl 1.5
Type "help" for help.

[BroControl] > install
creating policy directories ...
installing site policies ...
generating standalone-layout.bro ...
generating local-networks.bro ...
generating broctl-config.bro ...
generating broctl-config.sh ...
```

```
...
kunoichi:etc root# grep eth0 node.cfg
interface=eth0
```

Zeek demo: Run Zeek

```
// back to Broctl and start it
[BroControl] > start
starting bro
// and then
kunoichi:bro root# pwd
/usr/local/var/spool/bro
kunoichi:bro root# tail -f dns.log
```

More examples at:

https://www.bro.org/sphinx/script-reference/log-files.html

Exercise

Now lets do the exercise

Zeek DNS capturing domain names

which is number 5 in the exercise PDF.

Exercise

Now lets do the exercise

Zeek TLS capturing certificates

which is number 6 in the exercise PDF.

DNS is important

Another tool that provides a basic SQL-frontend to PCAP-files

`https://www.dns-oarc.net/tools/packetq`

`https://github.com/DNS-OARC/PacketQ`

Going back in time and finding systems that visited a specific domain can explain when and where an infection started.

Deciding on which tool to use, Zeek or PacketQ depends on the situation.

Suricata IDS/IPS/NSM

Suricata is a high performance Network IDS, IPS and Network Security Monitoring engine. Open Source and owned by a community run non-profit foundation, the Open Information Security Founda-

tion (OISF). Suricata is developed by th e OISF and its supporting vendors.

http://suricata-ids.org/ http://openinfosecfoundation.org

Exercise

Now lets do the exercise

Suricata Basic Operation

which is number 7 in the exercise PDF.

Exercise

Now lets do the exercise

Basic Suricata rule configuration

which is number 8 in the exercise PDF.

Exercise

Now lets do the exercise

Configure Mirror Port

which is number 9 in the exercise PDF.

Exercise

Now lets do the exercise

Save Suricata JSON Output in Database

which is number 10 in the exercise PDF.

NetFlow

NetFlow is getting more important, more data share the same links

Accounting is important

Detecting DoS/DDoS and problems is essential

NetFlow sampling is vital information - 123Mbit, but what kind of traffic

Currently also investigating sFlow - hopefully more fine

grained

Exercise

Now lets do the exercise

Suricata Netflow

which is number 11 in the exercise PDF.

Zeek JSON

As shown with Suricata, it is also possible to insert Bro output into Elasticsearch.

One example is shown in the blog posts: `https://logz.io/blog/bro-elk-part-1/`

Commercial Support

You can and should use updated rulesets for Suricata.

I Recommend the Emerging Threats ET Pro ruleset, which is about USD 900 per year per sensor. So two sites with Suricata running in both, 2x USD 900

Summary

We started from a basic Ubuntu/Debian server, and using Zeek and Suricata we now know more about our network.

We can collect logs about network traffic based on generic NetFlow, specific types DNS/TLS/protocols, and traffic matching advanced rulesets.

We know it is possible to create dashboards and visualizing the data.

What are the next steps?

Questions?

Henrik Lund Kramshøj hlk@zencurity.com

Need help with infrastructure security or pentesting, ask me!

You are always welcome to send me questions later via email