Welcome to

Suricata, Bro og DNS Capture

An Evening with Packets

Henrik Lund Kramshøj hlk@zencurity.dk

Slides are available as PDF, kramse@Github
suricatabro-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 Bro 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

Bro 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

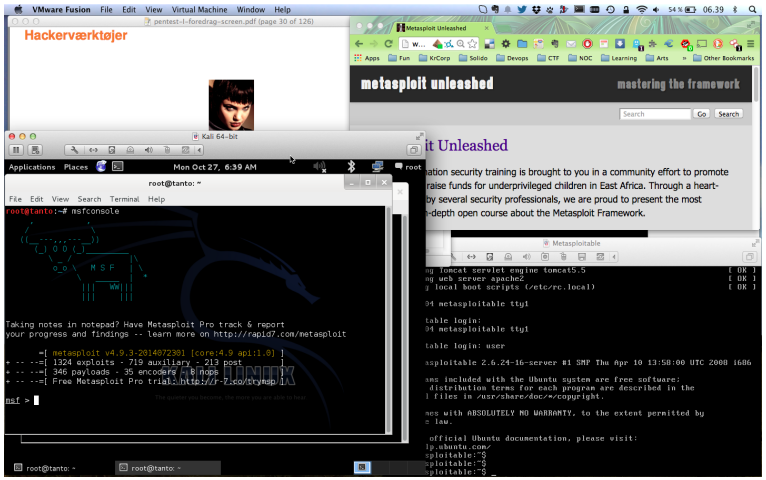
Kali Linux the pentest toolbox

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 laptops has CPU with virtualization
  May need to enale 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*

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

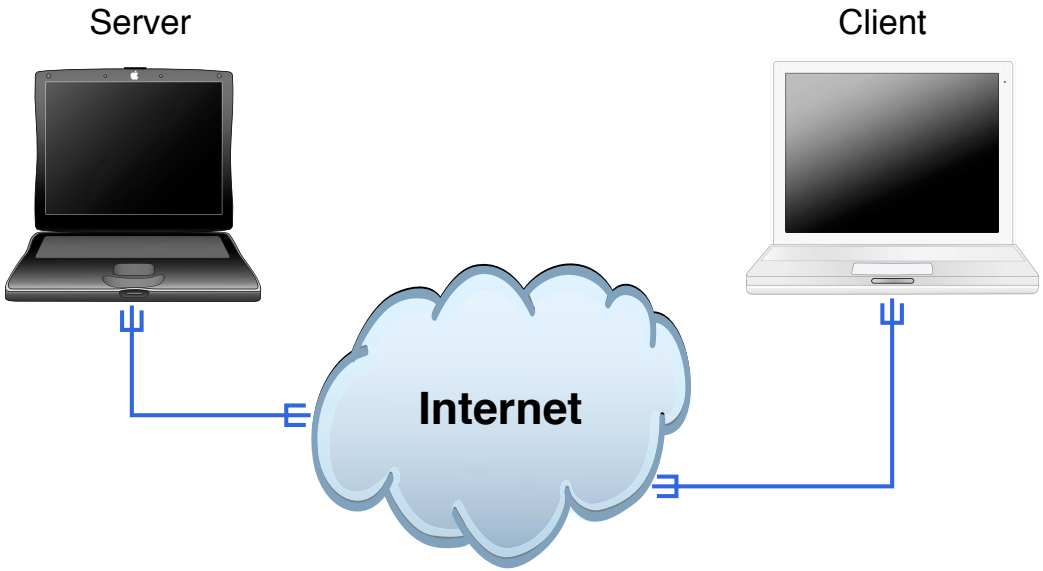
Use Github! So many libraries and programs that can help, maybe solve 90% of your problem, and you can glue the rest together

Example introductions:

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

We are all Devops now, even security people!

Internet today



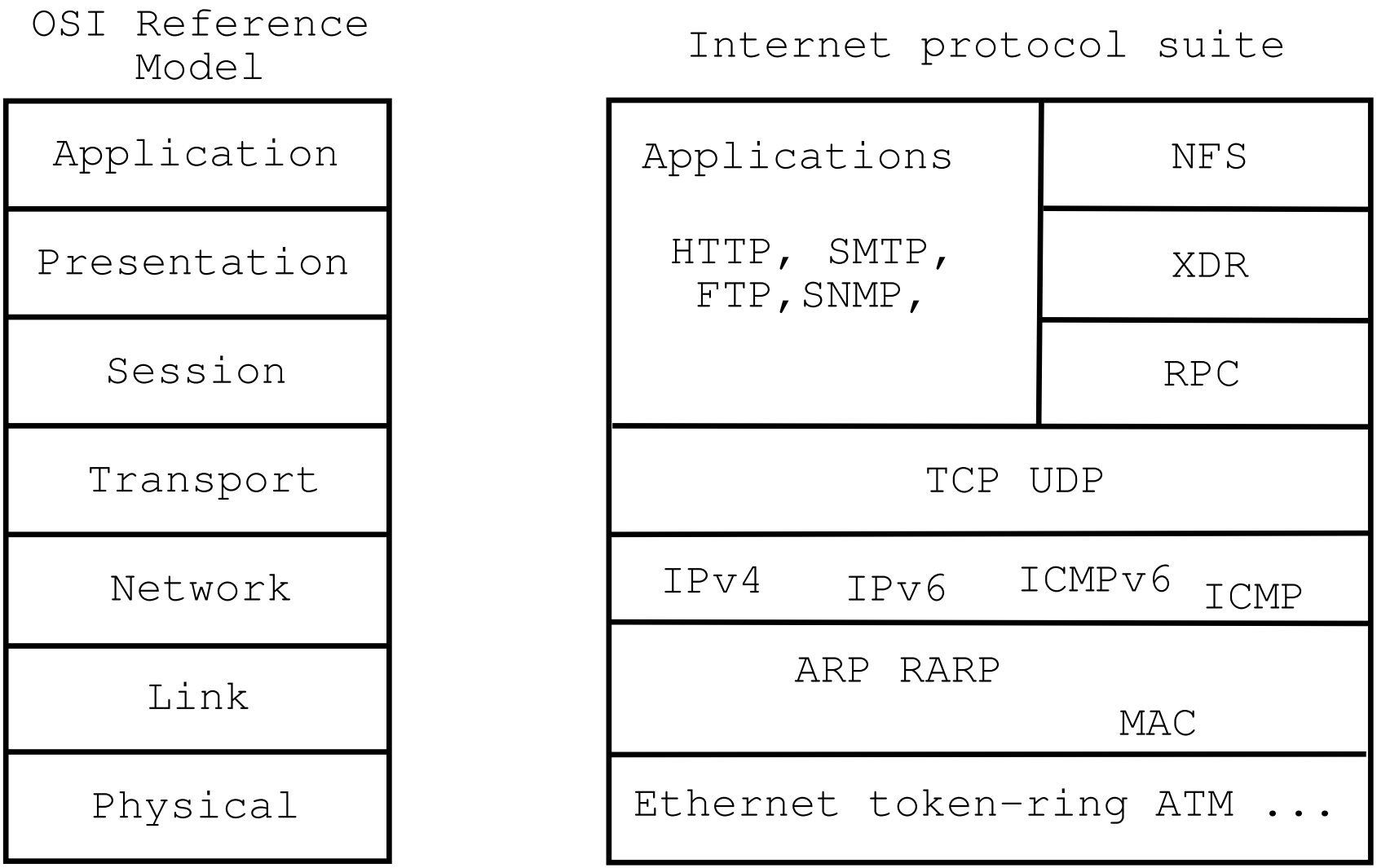
Clients and servers

Roots in academia

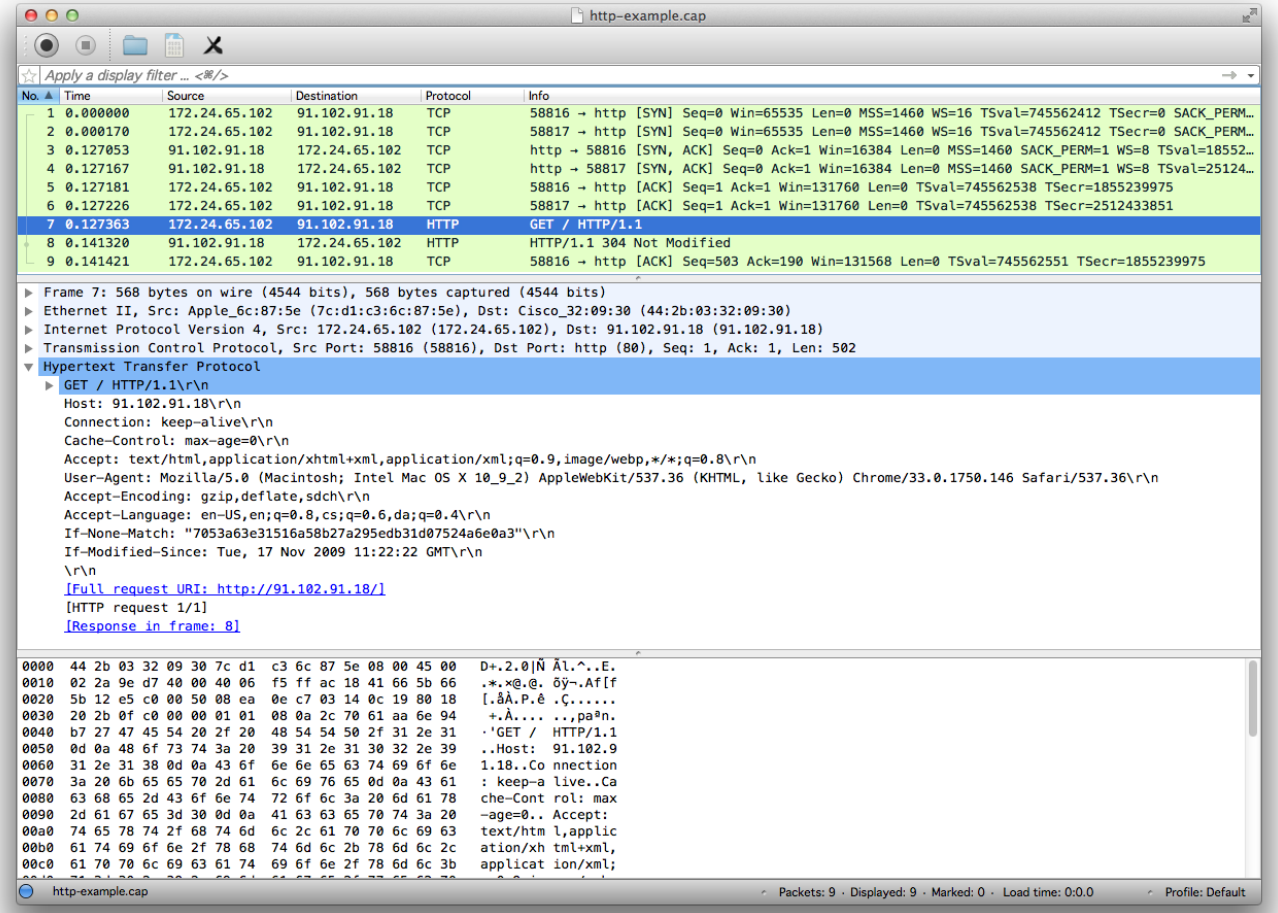
Protocols more than 20 years old

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

OSI og Internet modellerne



Using Wireshark



https://www.wireshark.org

Exercise

Now lets do the exercise

Wireshark and tcpdump

which is number 1 in the exercise PDF.

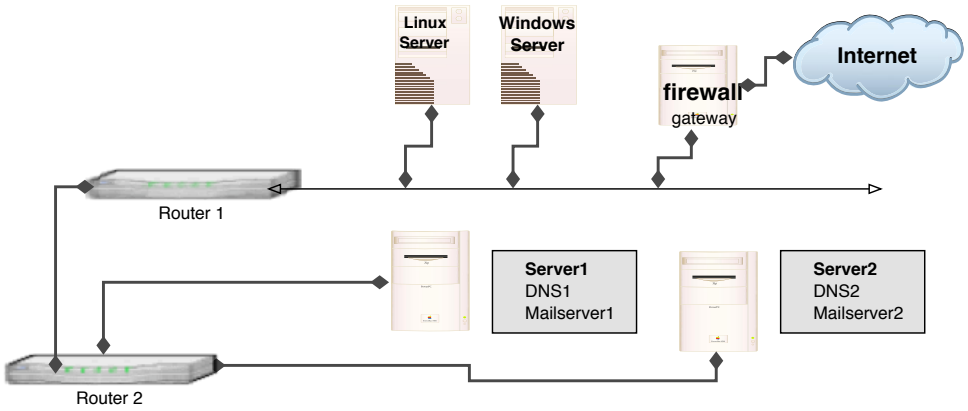
Exercise

Now lets do the exercise

Capturing network packets

which is number 2 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 message 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, firewalls and devices perform badly or even crash!

The Bro Network Security Monitor

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

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

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

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

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

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

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

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

While focusing on network security monitoring, Bro provides a comprehensive platform for more general network traffic analysis as well. Well grounded in more than 15 years of research, Bro has successfully bridged the traditional gap between academia and operations since its inception. Today, it is relied upon operationally in particular by many scientific environments for securing their cyberinfrastructure. Bro's user community includes major universities, research labs, supercomputing centers, and open-science communities.

The Bro Network Security Monitor is not a single tool, but more of a powerful network analysis framework

Source `https://www.bro.org/`

Bro 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

Bro on the web

which is number 3 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 Bro and Suricata projects.

Get Started with Bro

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`

Bro demo: Run

```
// install bro 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
```

Bro demo: Run bro

```
// 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

Bro DNS capturing domain names

which is number 4 in the exercise PDF.

Exercise

Now lets do the exercise

Bro TLS capturing certificates

which is number 5 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, Bro 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 Foundation (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 6 in the exercise PDF.

Exercise

Now lets do the exercise

Basic Suricata rule configuration

which is number 7 in the exercise PDF.

Exercise

Now lets do the exercise

Configure Mirror Port

which is number 8 in the exercise PDF.

Exercise

Now lets do the exercise

Save Suricata JSON Output in Database

which is number 9 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 10 in the exercise PDF.

Bro 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 Bro 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.dk

Need DDoS testing or pentest, ask me!

You are always welcome to send me questions later via email

Did you notice how a lot of the links in this presentation use HTTPS - encrypted