

Network Tapping

PROSA SURVEIL_IT

Optional exercises

Henrik Lund Kramshoej
hlk@zencurity.com

April 2, 2019



Contents

1	Wireshark and tcpdump 15min	3
2	Capturing network packets 15min	5
3	Zeek on the web 15min	7
4	Check your VM, run Ansible up to 30min	8
5	Zeek DNS capturing domain names up to 30min	9
6	Zeek TLS capturing certificates 15min	11
7	Configure Mirror Port - Junos Demo 10min	12
8	Mirror port with TP-Link - 30min	13
A	Host information	15

Preface

This material is prepared for use in *Network Tapping PROSA SURVEIL_IT* workshop and was prepared by Henrik Lund Kramshøj, <http://www.zencurity.com>. It describes the networking setup and applications for trainings and workshops where hands-on exercises are needed.

Further a presentation is used which is available as PDF from kramse@Github. Look for `surveil_it-workshop-exercises` in the repo `security-courses`.

These exercises are expected to be performed in a training setting with network connected systems. The exercises use a number of tools which can be copied and reused after training. A lot is described about setting up your workstation in the repo

<https://github.com/kramse/kramse-labs>

Prerequisites

This material expects that participants have a working knowledge of TCP/IP from a user perspective. Basic concepts such as web site addresses and email should be known as well as IP-addresses and common protocols like DHCP.

Have fun and learn

Introduction to networking

IP - Internet protocol suite

It is extremely important to have a working knowledge about IP to implement secure and robust infrastructures. Knowing about the alternatives while doing implementation will allow the selection of the best features.

ISO/OSI reference model

A very famous model used for describing networking is the ISO/OSI model of networking which describes layering of network protocols in stacks.

This model divides the problem of communicating into layers which can then solve the problem as smaller individual problems and the solution later combined to provide networking.

Having layering has proven also in real life to be helpful, for instance replacing older hardware technologies with new and more efficient technologies without changing the upper layers.

In the picture the OSI reference model is shown along side with the Internet Protocol suite model which can also be considered to have different layers.

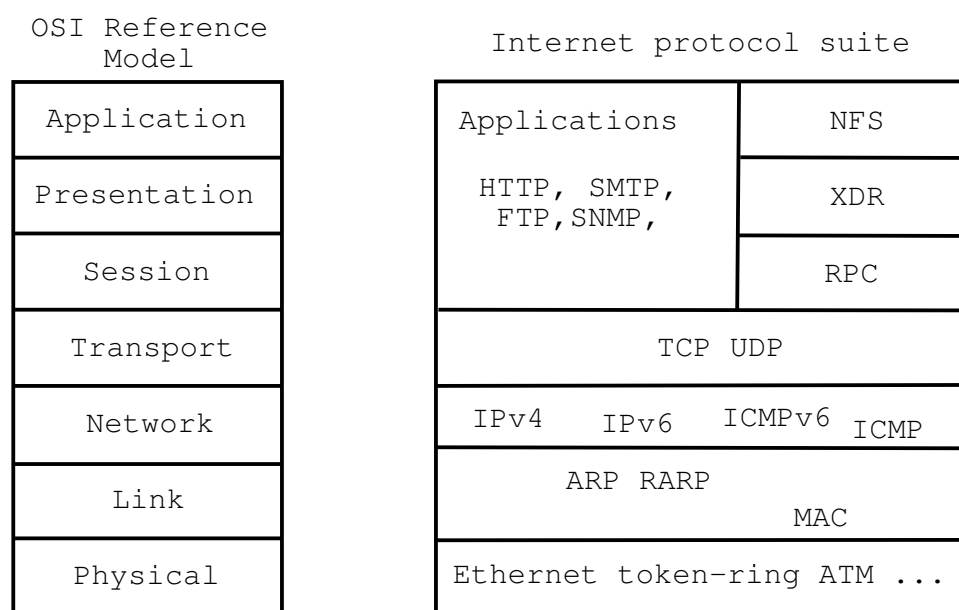


Figure 1: OSI og Internet Protocol suite

Exercise content

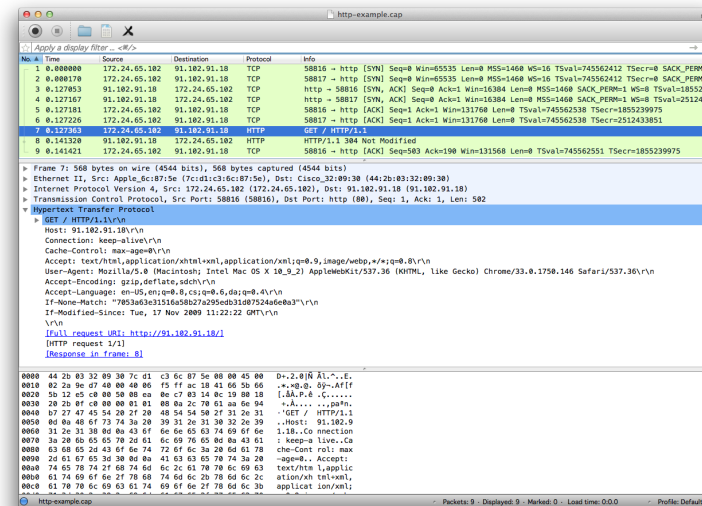
Most exercises follow the same procedure and has the following content:

- **Objective:** What is the exercise about, the objective
- **Purpose:** What is to be the expected outcome and goal of doing this exercise
- **Suggested method:** suggest a way to get started
- **Hints:** one or more hints and tips or even description how to do the actual exercises
- **Solution:** one possible solution is specified
- **Discussion:** Further things to note about the exercises, things to remember and discuss

Please note that the method and contents are similar to real life scenarios and does not detail every step of doing the exercises. Entering commands directly from a book only teaches typing, while the exercises are designed to help you become able to learn and actually research solutions.

Exercise 1

Wireshark and tcpdump 15min



Objective:

Try the program Wireshark locally your workstation, or tcpdump

You can run Wireshark on your host too, if you want.

Purpose:

Installing Wireshark will allow you to analyse packets and protocols

Tcpdump is a feature included in many operating systems and devices to allow packet capture and saving network traffic into files.

Suggested method:

Download and install the program, either download from web server locally or from <http://www.wireshark.org>

Wireshark requires a packet capture library to be installed

Hints:

PCAP is a packet capture library allowing you to read packets from the network. Tcpdump uses libpcap library to read packet from the network cards and save them. Wireshark is a graphical application to allow you to browse through traffic, packets and protocols.

Solution:

When Wireshark is installed sniff some packets. We will be working with both live

traffic and saved packets from files in this course.

If you want to capture packets as a non-root user on Debian, then use the command to add a Wireshark group:

```
sudo dpkg-reconfigure wireshark-common
```

and add your user to this:

```
sudo gpasswd -a $USER wireshark
```

Dont forget to logout/login to pick up this new group.

Discussion:

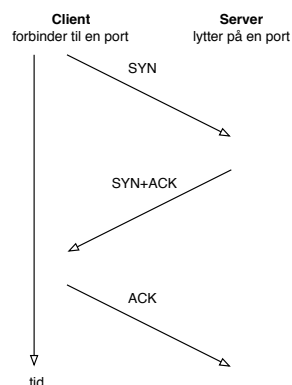
Wireshark is just an example other packet analyzers exist, some commercial and some open source like Wireshark

We can download a lot of packet traces from around the internet, we might use examples from

<https://www.bro.org/community/traces.html>

Exercise 2

Capturing network packets 15min



Objective:

Sniff packets and dissect them using Wireshark

Purpose:

See real network traffic, also know that a lot of information is available and not encrypted.

Note the three way handshake between hosts running TCP. You can either use a browser or command line tools like cURL

```
curl http://www.zencurity.com
```

Suggested method:

Open Wireshark and start a capture

Then in another window execute the ping program while sniffing

or perform a Telnet connection while capturing data

Hints:

When running on Linux the network cards are usually named `eth0` for the first Ethernet and `wlan0` for the first Wireless network card. In Windows the names of the network cards are long and if you cannot see which cards to use then try them one by one.

Solution:

When you have collected some packets you are done.

Discussion: Is it ethical to collect packets from an open wireless network?

Also note the TTL values in packets from different operating systems

Exercise 3

Zeek on the web 15min

Objective:

Try Zeek Network Security Monitor - without installing it.

Purpose:

Show a couple of examples of Zeek scripting, the built-in language found in Zeek Network Security Monitor

Suggested method:

Go to <http://try.bro.org/#/?example=hello>

Hints:

The exercise *The Summary Statistics Framework* can be run with a specific PCAP.

192.168.1.201 did 402 total and 2 unique DNS requests in the last 6 hours.

Solution:

You should read the example *Raising a Notice*. Getting output for certain events may be interesting to you.

Discussion:

Zeek Network Security Monitor is an old/mature tool, but can still be hard to get started using. I would suggest that you always start out using the packages available in your Ubuntu/Debian package repositories.

They work, and will give a first impression of Zeek. If you later want specific features not configured into the binary packet, then install from source.

Also Zeek uses a broctl program to start/stop the tool, and a few config files which we should look at. From a Debian system they can be found in /etc/bro :

```
root@NMS-VM:/etc/bro# ls -la
drwxr-xr-x  3 root root  4096 Oct  8 08:36 .
drwxr-xr-x 138 root root 12288 Oct  8 08:36 ..
-rw-r--r--  1 root root  2606 Oct 30 2015 broctl.cfg
-rw-r--r--  1 root root   225 Oct 30 2015 networks.cfg
-rw-r--r--  1 root root   644 Oct 30 2015 node.cfg
drwxr-xr-x  2 root root  4096 Oct  8 08:35 site
```

Exercise 4

Check your VM, run Ansible up to 30min

Objective:

Make sure your virtual machine is in working order.

We need a Linux server for running the tools.

Purpose:

If your VM is not updated we will run into trouble later.

Suggested method:

Go to <https://github.com/kramse/kramse-labs/tree/master/suricatazeek>

Read the instructions for the setup of a VM.

Hints:

Ansible is great for automating stuff, so by running the playbooks we can get a whole lot of programs installed, files modified - avoiding the Vi editor ☺

Example playbook content

```
apt:
  name: " packages "
vars:
  packages:
    - nmap
    - curl
    - iperf
    ...
```

Solution:

When you have a updated VM and Ansible running, then we are good.

Discussion:

Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Exercise 5

Zeek DNS capturing domain names up to 30min

Objective:

We will now start using Zeek on our systems.

Purpose:

Try Zeek with example traffic, and see what happens.

Suggested method packet capture file:

Note: a dollar sign is the Linux prompt, showing the command after

```
$ cd
$ wget http://downloads.digitalcorpora.org/corpora/network-packet-dumps/2008-nitroba/nitroba.pcap
$ mkdir $HOME/bro; cd $HOME/bro; bro -r ../nitroba.pcap
... bro reads the packets
~/bro$ ls
conn.log  dns.log  dpd.log  files.log  http.log  packet_filter.log
sip.log  ssl.log  weird.log  x509.log
$ less *
```

Use :n to jump to the next file in less, go through all of them.

Suggested method Live traffic:

Make sure Zeek is configured as a standalone probe and configured for the right interface. Linux used to use eth0 as the first ethernet interface, but now can use others, like ens192 or enx00249b1b2991.

```
root@NMS-VM:/etc/bro# cat node.cfg
# Example BroControl node configuration.
#
# This example has a standalone node ready to go except for possibly changing
# the sniffing interface.

# This is a complete standalone configuration.  Most likely you will
# only need to change the interface.
[bro]
type=standalone
host=localhost
interface=eth0
...
```

Hints:

There are multiple commands for showing the interfaces and IP addresses on Linux. The old way is using `ifconfig` - a newer systems would use `ip a`

Note: if your system has a dedicated interface for capturing, you need to turn it on, make it available. This can be done manually using `ifconfig eth0 up` **Solution:** When you either run Zeek using a packet capture or using live traffic

Running with a capture can be done using a command line such as: `bro -r traffic.pcap`

Using `broctl` to start it would be like this:

```
// 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 ...
...
```

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

You should be able to spot entries like this:

```
#fields ts      uid      id.orig_h      id.orig_p      id.resp_h      id.resp_p      proto  trans_i
      query  qclass qclass_name    qtype  qtype_name    rcode  rcode_name    AA    TC    RD
1538982372.416180 CD12Dc1SpQm42QW4G3 10.xxx.0.145 57476 10.x.y.141 53 udp 20383 0.045021 www.dr.dk
1 C_INTERNET 1 A 0 NOERROR F F T T 0 www.dr.dk-v1.edgekey.net,e16198.b.akamaiedge.net,2.17.212.93
60.000000,20409.000000,20.000000 F
```

Note: this show ALL the fields captured and dissected by Zeek, there is a nice utility program `bro-cut` which can select specific fields:

```
root@NMS-VM:/var/spool/bro/bro# cat dns.log | bro-cut -d ts query answers | grep dr.dk
2018-10-08T09:06:12+0200 www.dr.dk www.dr.dk-v1.edgekey.net,e16198.b.akamaiedge.net,2.17.212.93
```

Discussion:

Why is DNS interesting?

Exercise 6

Zeek TLS capturing certificates 15min

Objective:

Run more traffic through Zeek, see the various files.

Purpose:

See that even though HTTPS and TLS traffic is encrypted it often show names and other values from the certificates and servers.

Suggested method:

Run Zeek capturing live traffic, start https towards some sites. A lot of common sites today has shifted to HTTPS/TLS.

Hints:

use broctl start and watch the output directory

```
root@NMS-VM:/var/spool/bro/bro# ls *.log
communication.log  dhcp.log  files.log  known_services.log  packet_filter.log  stats.log
stdout.log  x509.log  conn.log  dns.log  known_hosts.log  loaded_scripts.log  ssl.log
stderr.log  weird.log
```

We already looked at dns.log, now check ssl.log and x509.log

Solution:

When you have multiple log files with data from Zeek, and have looked into some of them. You are welcome to ask questions and look into more files.

Discussion:

How can you hide that you are going to HTTPS sites?

Hint: VPN

Exercise 7

Configure Mirror Port - Junos Demo 10min

Objective:

Mirror ports are a way to copy traffic from devices - for analyzing it. We will go through the steps on a Juniper switch to show how. Most switches which are configurable have this possibility.

Purpose:

We want to capture traffic for multiple systems, so we select an appropriate port and copy the traffic. In our setup, we select the uplink port to the internet/router.

It is also possible to buy passive taps, like a fiber splitter, which then takes part of the signal, and is only observable if you look for signal strength on the physical layer.

Suggested method:

We will configure a mirror port on a Juniper EX2200-C running Junos.

```
root@ex2200-c# show ethernet-switching-options | display set
set ethernet-switching-options analyzer mirror01 input ingress interface ge-0/1/1.0
set ethernet-switching-options analyzer mirror01 input egress interface ge-0/1/1.0
set ethernet-switching-options analyzer mirror01 output interface ge-0/1/0.0
set ethernet-switching-options storm-control interface all
```

Hints:

When checking your own devices this is often called SPAN ports, Mirror ports or similar.

https://en.wikipedia.org/wiki/Port_mirroring

Cisco has called this Switched Port Analyzer (SPAN) or Remote Switched Port Analyzer (RSPAN), so many will refer to them as SPAN-ports.

Solution:

When we can see the traffic from the network, we have the port configured - and can run any tool we like. Note: specialized capture cards can often be configured to spread the load of incoming packets onto separate CPU cores for performance. Capturing 100G and more can also be done using switches like the example found on the Zeek web site using an Arista switch 7150.

Discussion:

When is it ethical to capture traffic?

Exercise 8

Mirror port with TP-Link - 30min

Objective:

Configure mirror port, use it to monitor your traffic.

Purpose:

Show how real networks often sniff.

Suggested method:

Configure a mirror port - select a high numbered port not in use.

Then configure source ports, the ones in current use and destination to the selected high port.

If using the TP-Link T1500G-10PS then this link should describe the process:

https://www.tp-link.com/en/configuration-guides/mirroring_traffic/?configurationId=18210

Which describe:

1. Choose the menu MAINTENANCE > Mirroring
2. Select Edit for the Mirror Session 1
3. In the Destination Port Config section, specify a destination port for the mirroring session, and click Apply
4. In the Source Interfaces Config section, specify the source interfaces and click Apply

Using the command line would be similar to this:

```
Switch#configure
Switch(config)#monitor session 1 destination interface gigabitEthernet 1/0/7
Switch(config)#monitor session 1 source interface gigabitEthernet 1/0/1-4 both
Switch(config)#monitor session 1 source cpu 1 both
```

Hints:

Selecting a port away from the existing ones allow easy configuration of the source to be like, Source ports 1-4 and destination 7 - and easily expanded when port 5 and 6 are activated.

Solution:

When your team has configured a mirror port and seen traffic using your IDS, Wireshark or just tcpdump you are done.

Discussion:

How would you monitor a larger network?

Cisco has a feature named RSPAN

Remote SPAN (RSPAN): An extension of SPAN called remote SPAN or RSPAN. RSPAN allows you to monitor traffic from source ports distributed over multiple switches, which means that you can centralize your network capture devices. RSPAN works by mirroring the traffic from the source ports of an RSPAN session onto a VLAN that is dedicated for the RSPAN session. This VLAN is then trunked to other switches, allowing the RSPAN session traffic to be transported across multiple switches. On the switch that contains the destination port for the session, traffic from the RSPAN session VLAN is simply mirrored out the destination port.

Source: <https://community.cisco.com/t5/networking-documents/understanding-span-rspan-and-erspan/ta-p/3144951>

Appendix A

Host information

- You should note the IP-addresses used for servers and devices
- The web server for installing programs:
`http://10.10.10.10/public/windows/`
- Server used for team login: 10.10.10.10
Available usernames: team1, team2, ... team10 password: team
- You can obtain root access using: `sudo -s`

Available servers and devices:

- IP: 10.10.10.10 - OpenBSD router
- IP: 10.10.10.11 - Your laptop
- IP: 10.10.10.12 - Your laptop VM
- IP: 10.10.10.13 -
- IP: 10.10.10.14 -