FSI
LEI

2025/2026

Practical class #4

- Intrusion detection with Suricata

Intrusion Detection

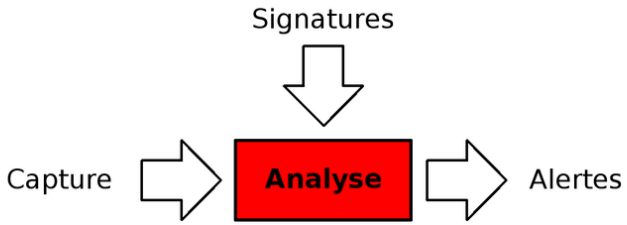
Intrusion detection system (IDS)
•Device or application that monitors the network or system for malicious activities and produces reports

Types of intrusion detection systems: NIDS, HIDS, Stack-based

NIDS: (Network-based intrusion detection)
•Network adapter in “promiscuous mode”, traffic is analysed in real-time
•Placed on key areas of network infrastructure and monitors the traffic as it flows
•Identifies heuristics and patterns (signatures) of common computer attacks

HIDS: (Host-based intrusion detection)
•Monitor system logs, file system checksums, ..
•Monitor the inbound and outbound packets from the device/host

Stack-based intrusion detection
•Integrated closely with the TCP/IP stack
•Packets are watched as they traverse their way up the network layers and discarded before reaching the application



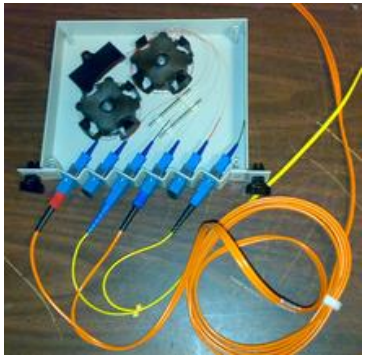
Accessing the network traffic

Mirroring ports

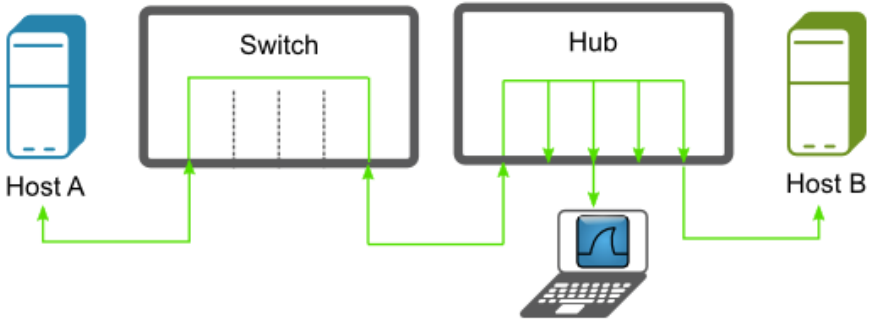
- Device (e.g. switch) sends a copy of network packets from one port (or an entire VLAN) to another (monitoring port).
- Example on Cisco: switched Port ANalyzer (SPAN) or Remote Switched Port ANalyzer (RSPAN) ports

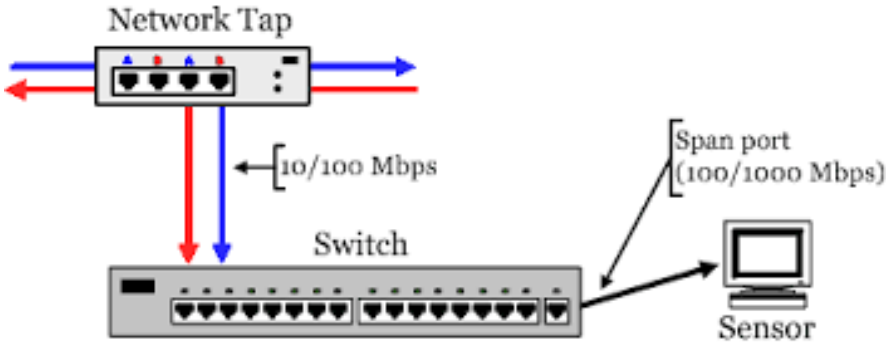
Networking TAPS

- Inserted between network devices to copy data continuously without compromising network integrity
- Available with a variety of features for both copper and fiber networks



Forced packet repetition using a non-switched Hub





http://en.wikipedia.org/wiki/Network_tap
http://www.infosecwriters.com/hhworld/hh9/roc/node4.html

Intrusion detection (Suricata)

Suricata is a “Network IDS, IPS and Network Security Monitoring engine”, https://suricata.io/

Supports different runmodes (***suricata --list-runmodes***):

- Single thread
- Autofp – multiple threads
- Workers – performs load balancing between threads, offers best performance in some cases
- More info available in official documentation

Is compatible with snort rules

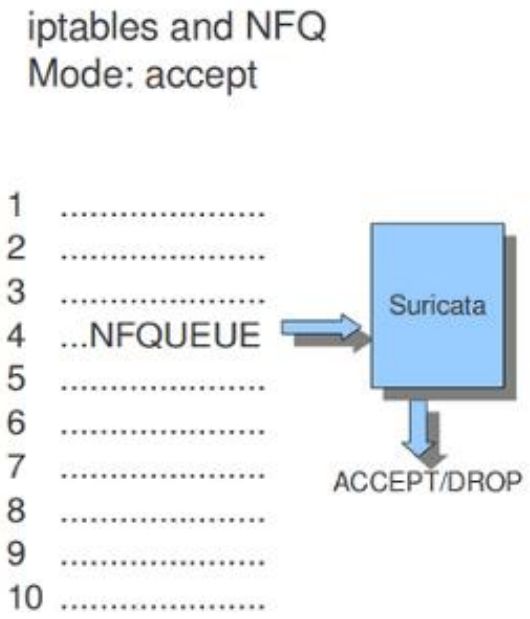
Includes a tool to update rules (***suricata-update***)

Intrusion detection (Suricata in IPS mode)

- Also supports to analyse packets sent by iptables/nftables through **NFQUEUE,** also labelled as inline mode in Linux

IPS mode of Suricata can be configured in different mode:

- Accept or Drop
- Repeat (reinject packet)
- Router to other queue
- More info available in official documentation



Suricata needs to be connected with queue
the connection can be performed at command line:
sudo suricata -c /etc/suricata/suricata.yaml -q 0

Running Suricata

In live capture mode

sudo suricata -c /etc/suricata/suricata.yaml -i eth0 -vvvv

Log Information available on: **/var/log/suricata**

- **Eve.json** – Extensible Event Format, with information on Json format, useful to integrate with other systems (more info)
- **Fast.log** – Line based alerts (more info)
- **Stats.log** – Statistics information (more info)
- **Suricata.log** – generic information regarding suricata
- **Pcap-log** – (needs to be activated), but logs all the packets captured (more info)

General Configuring of Suricata

File: ***/etc/suricata/suricata.yaml***

YAML format (be careful with spaces!!)

To activate packet logging

```
- pcap_log
    enabled: yes
    filename: log.pcap
    mode: sguil     # to create directories based on date
    dir: /var/log/suricata/pcaps/
```

Rules files and directories (more info):

```
default-rule-path: /etc/suricata/rules/
rule-files:
  - suricata.rules
```

Rules in Suricata (compatible with Snort)

Rules format (more info):

action protocol source (ports) -> destination (ports) options

Example:

alert ssh any any -> any any (msg:"SURICATA SSH"; sid:2228000; rev:1;)

Action can be: Drop, alert, pass, …
Protocol: tcp, udp ,…
Source and destination: any, IP addresses
Source and destination ports

Rules in Suricata (compatible with Snort)

Rules can be tested/validated with:

suricata –T <name_of_rules_file>

Recommended configurations in Suricata

In the configuration file:

1. Configure the information of Home network: ***HOME_NET*** (beginning of file)
2. Configure the appropriate interface for capture in ***af-packet*** group (middle of file). For multiple interfaces place a new group for ***af-packet*** with a different cluster-id.
3. Rules for alert.

Snort

Snort is:

- A lightweight network IDS
- Real-time traffic logging, content searching/matching and alerting
- Logs in tcpdump binary format and ASCII
- Integrates with IPTables for intrusion detection and prevention (inline mode)

Snort may work in three modes:

- Packet sniffer (as tcpdump or tethereal)
- Packet logging mode (useful for network traffic debugging)
- Network intrusion detection (from rules defined on a configuration file)

Snort Help

snort --help

Snort in **packet sniffer mode**, examples:

Print TCP/IP packet headers on the screen
snort -v

Print TCP/IP packet headers and also the application data
snort -vd

A more descriptive description of the packets (including data link layer headers)
snort -vde

Snort

Snort in **packet logger mode**, examples:

With a directory Snort goes in packet logger mode
snort -vde -K ascii -l ./log

Log in binary (tcpdump) format
snort -b -l ./log

Snort in “playback mode” from log file
snort -vd -r snort.log

Reading the packet log file using tcpdump
tcpdump -r snort.log

Snort in **network intrusion detection system mode**, examples:

With a directory Snort goes in packet logger mode
snort -dev -l log -c snort.conf

Detection rules, examples:

alert tcp any any –> 10.254.0.0/24 80 (msg:”HTTP packet”;)

var MY_NETS [10.254.0.0/24,10.1.0.0/24]
log tcp any any –> $MY_NETS any (flags:S; msg:”SYN packet”;)

alert tcp any any -> any 80 (content:”GET”;)

Snort

Snort in **packet logger mode**, examples:

```
root@debian:/home/sti# snort -vde -i lo -b -l /var/log/snort
Running in packet logging mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Log directory = /var/log/snort
pcap DAQ configured to passive.
Acquiring network traffic from "lo".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.19 GRE (Build 85)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2021 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.0 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=54586)
```

Snort

Snort in **network intrusion detection system mode**, examples:

With a directory Snort goes in packet logger mode
snort -dev -l log -c snort.conf

Consider the snort manual:
Section 3.5 · Payload Detection Rule
Section 3.7 – Payload

Detection rules, examples:

1. Based on content:

   alert tcp any any -> any 80 (content:"GET";)

2. Based on IPs and variables

   alert tcp any any –> 10.254.0.0/24 80 (msg:"HTTP packet";)

   var MY_NETS [10.254.0.0/24,10.1.0.0/24]
   log tcp any any –> $MY_NETS any (flags:S; msg:"SYN packet";)

Snort

Snort in **IDS mode**:

```
root@coimbra:/etc/snort# snort -vd -l /var/log/snort -c /etc/snort/stiExercise6.conf -K ascii -k none -i lo
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/stiExercise6.conf"
Tagged Packet Limit: 256
Log directory = /var/log/snort

+++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
1 Snort rules read
    1 detection rules
    0 decoder rules
    0 preprocessor rules
1 Option Chains linked into 1 Chain Headers
+++++++++++++++++++++++++++++++++++++++++++++++++++

+-------------------[Rule Port Counts]---------------------------------------
|             tcp     udp    icmp      ip
|     src       0       0       0       0
|     dst       0       0       0       0
|     any       0       0       1       0
|      nc       0       0       1       0
|     s+d       0       0       0       0
+------------------------------------------------------------------------------
```

Snort

Snort in **inline mode (with support for DAQ)**:

snort -Q --daq nfq --daq-var queue=0 -c /etc/snort/snort.conf -v -l /var/log/snort/

```
nfq DAQ configured to inline.
Reload thread starting...
Reload thread started, thread 0x7fa037a27700 (64112)

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.19 GRE (Build 85)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2021 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.0 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.2  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_S7COMMPLUS  Version 1.0  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
Commencing packet processing (pid=64111)
Decoding Raw IP4
```