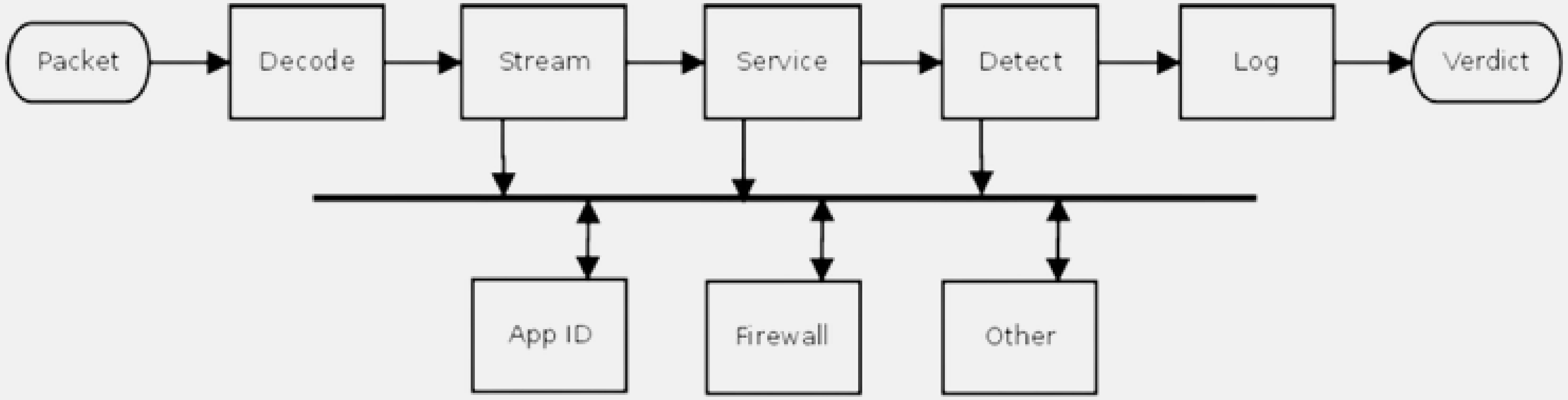
SNORT v3

Intrusion Detection & Prevention System

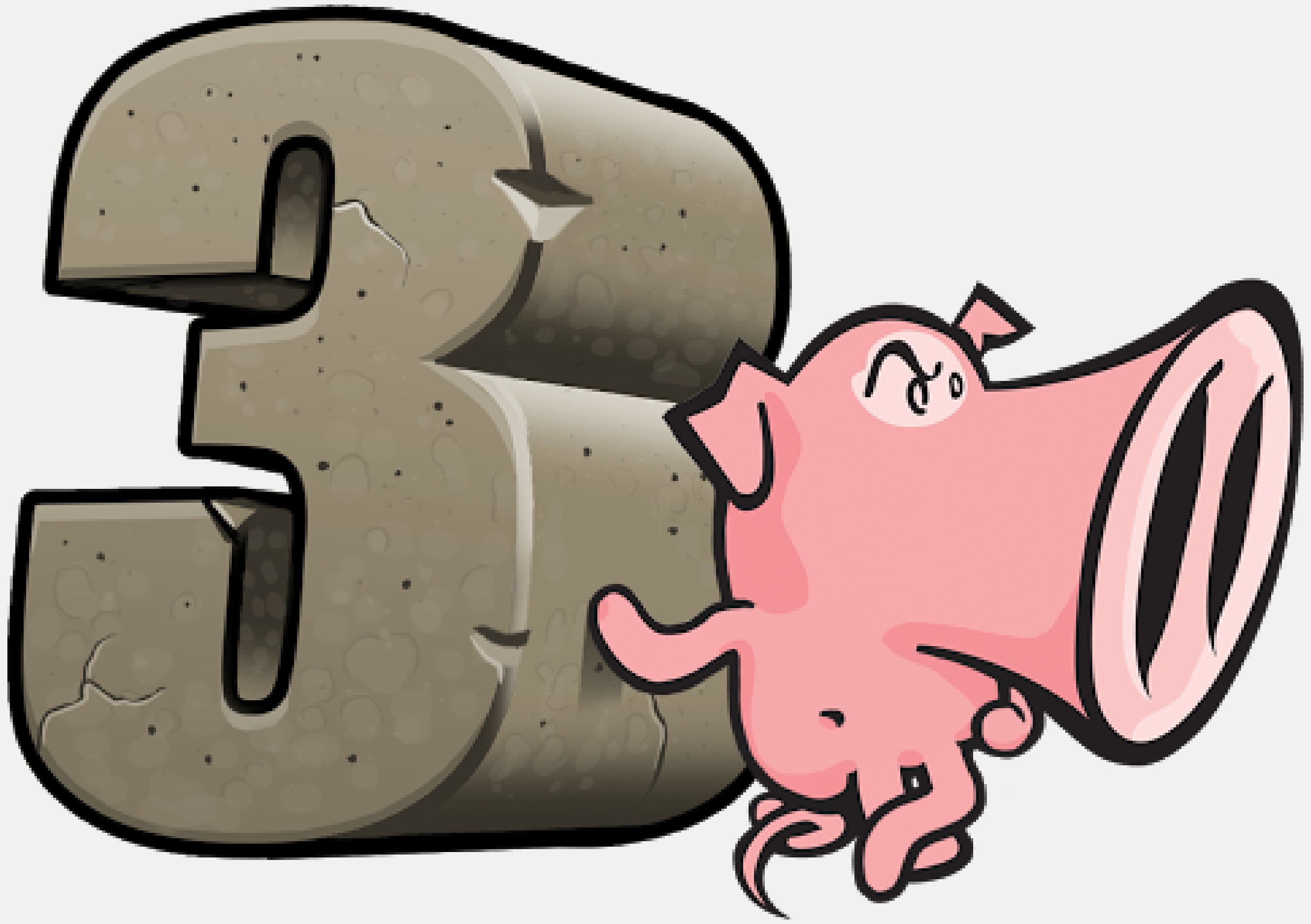
Edoardo Mich
Davide Moletta
Tefera Addis Sisay

Network Security Lab 22/23

Snort


Packet
Decode
Stream
Service
Detect
Log
Verdict
App ID
Firewall
Other

Snort2 vs Snort3



- New rule parser and rule syntax.
- Support for multiple packet-processing threads, which frees up more memory for packet processing.
- Use of a shared configuration and attribute table.
- Access to more than 200 plugins.
- Rewritten TCP handling.
- Improved shared object rules, including the ability to add rules for zero-day vulnerabilities.
- New performance monitor.
- New rule remarks and comments that are inside of the rule itself.

https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/004/341/original/snort3_information.pdf?1612471639

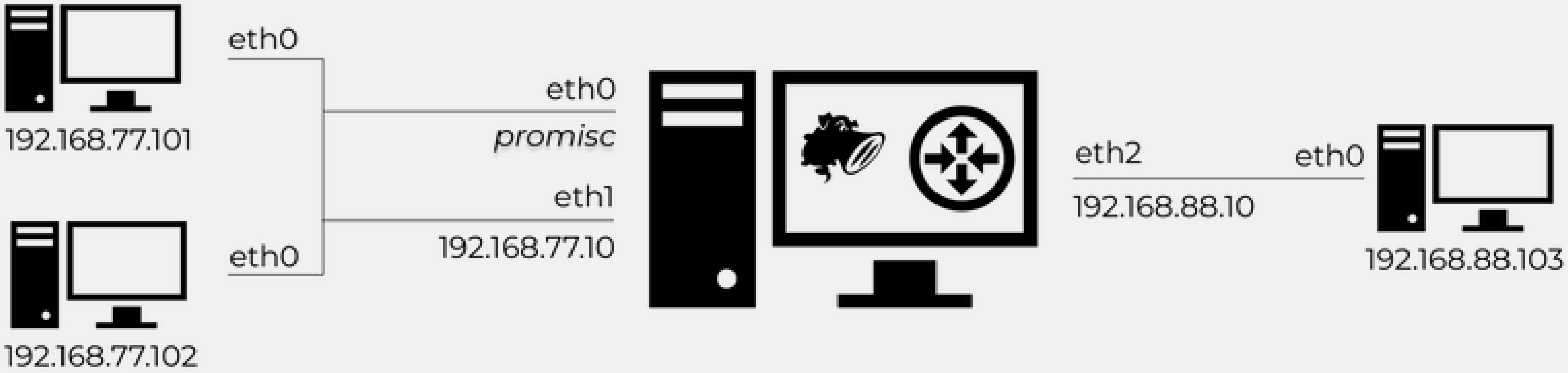
Snort3 installation

snort3/libdaq

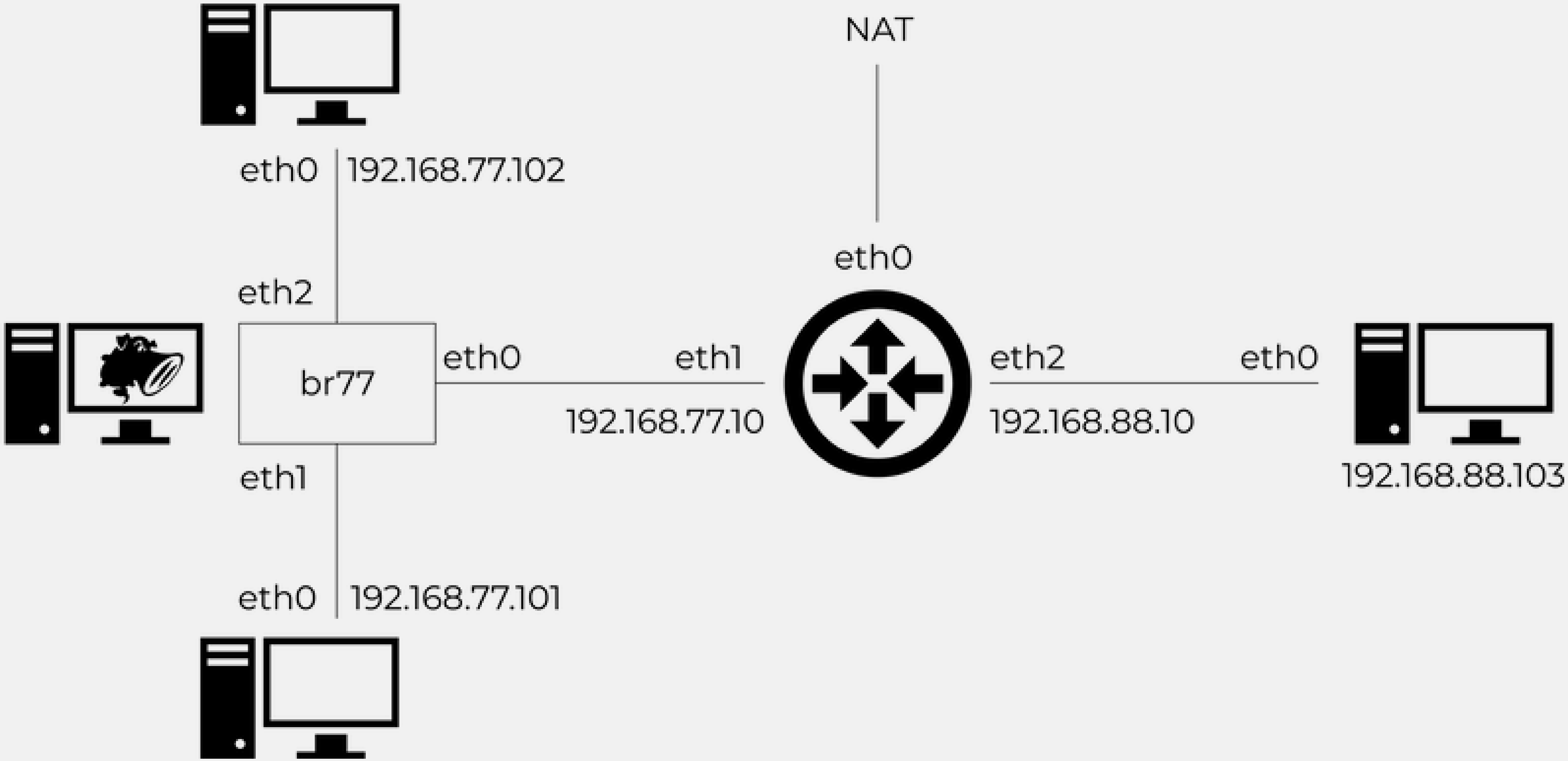
snort3/snort3

```
pkgname=snort3
pkgver=3.1.61.0
pkgrel=0
pkgdesc="Snort 3 is the next generation Snort IPS (Intrusion Prevention System)"
url="https://github.com/snort3/snort3"
arch="all"
license="GPL V2"
depends="daq"
makedepends="cmake build-base autoconf automake cpputest flex-dev libuuid
libtool hwloc-dev libnet-dev libdnet-dev libpcap-dev libpcre32 libtirpc-dev luajit-dev
openssl-dev libssl3 libnetfilter_queue-dev libmnl-dev libunwind-dev zlib-dev pcre-dev
xz-dev gperftools-dev"
builddir="$srcdir/$pkgname-$pkgver"

build() {
   ./configure_cmake.sh --prefix=/usr/local --enable-tcmalloc
}

package() {
   cd ./build
   make
   make DESTDIR="$pkgdir" install
}
```

Old Network setup



Network setup



Rules syntax

```
action protocol IPaddr port# -> IPaddr port#
(options)
```

OH YOU WOULD HAVE WRITTEN A SNORT RULE BY NOW?
TELL ME MORE ABOUT THAT
SNORT
2023 CALENDAR
Illustration by Jessie Hall
From the 2023 Snort parody calendar. Snort, and the Snort pig are registered trademarks of Cisco. All rights reserved.

Rules syntax

SNORT actions

```
action protocol IPaddr port# -> IPaddr port#
(options)
```

Basic actions:

- *Alert*: generate an alert on the single packet
- *Block*: block the current packet and all the subsequent packets in this flow
- *Drop*: drop the current packet
- *Log*: log the current packet
- *Pass*: mark the current packet as passed

Active responses:

- *React*: send response to client and terminate session.
- *Reject*: terminate session with TCP reset or ICMP unreachable
- *Rewrite*: enables overwrite packet contents based on a "replace" option in the rules

Rules syntax

SNORT Protocol

```
action protocol IPaddr port# -> IPaddr port#
(options)
```

Protocols:

- IP
- ICMP
- TCP
- UDP

Services:

- SSL
- HTTP
- SMTP
- POP3

Rules syntax

IP addresses

```
action protocol IPaddr port# -> IPaddr port# (options)
```

IP addresses on which the rule should be applied, can be defined in four ways:

- Numeric IP address: 192.168.77.101, 192.168.77.0/24
- Variable (using $) defined in the Snort config: $HOME_NET
- The keyword *any*, meaning any IP address
- List of IP addresses or subnets: [192.168.77.0/24,10.1.1.0/24]

Rules syntax

Ports

```
action protocol IPaddr port# -> IPaddr port#
(options)
```

Ports on which the traffic will be filtered can be defined in five different ways:

- The keyword any, meaning any port
- Static port: 8080
- Variable defined in the Snort config: $HTTP_PORTS
- Port ranges indicated with the range operator: 1:1024
- List of static ports: [1:1024,4444,5555]

Rules syntax

Direction operators

```
action protocol IPaddr port# -> IPaddr port#
(options)
```

There are two valid direction operators that indicates the direction of the traffic that the rule should apply to:

- ->: The first IP is the source and the second is the destination
- <>: Both addresses are source and destination

Rules syntax

Snort options

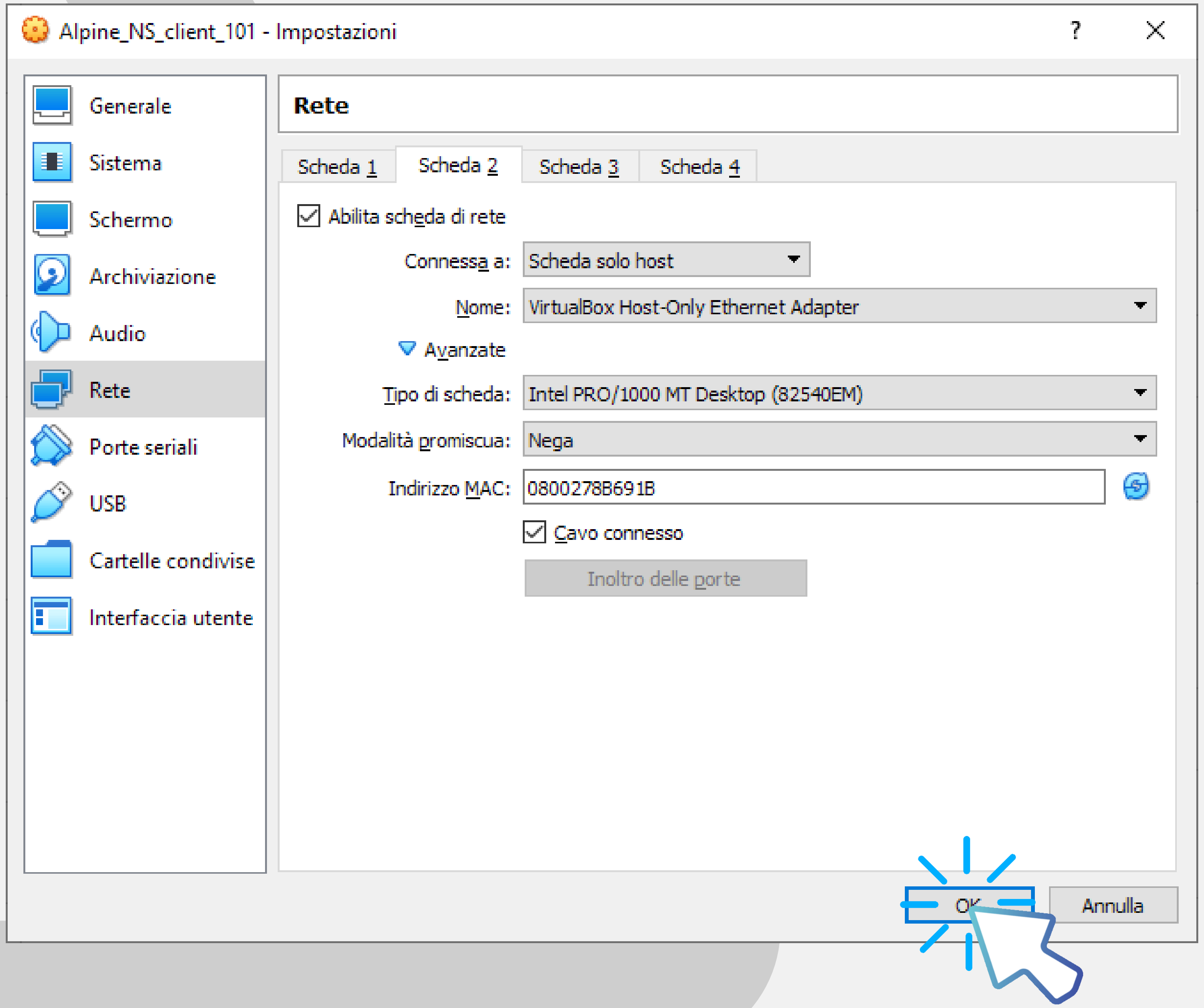
```
action protocol IPaddr port# -> IPaddr port#
(options)
```

All rule options are enclosed in parentheses after the end of the rule.
Each rule option is declared as key- value: <name>: <specific criteria>;
Some examples of rule options are:

- *msg*: set a message to be displayed when the rule fires
- *sid*: set a numeric identifier to the rule
- *priority*: set a priority to the rule
- *content*: used to perform basic pattern matching against packet data

DISCALMER

```
>_

ssh root@<ip>
```



Alpine_NS_client_101 - Impostazioni

Rete

☑ Abilita scheda di rete

Connessa a: Scheda solo host

Nome: VirtualBox Host-Only Ethernet Adapter

▽ Avanzate

Tipo di scheda: Intel PRO/1000 MT Desktop (82540EM)

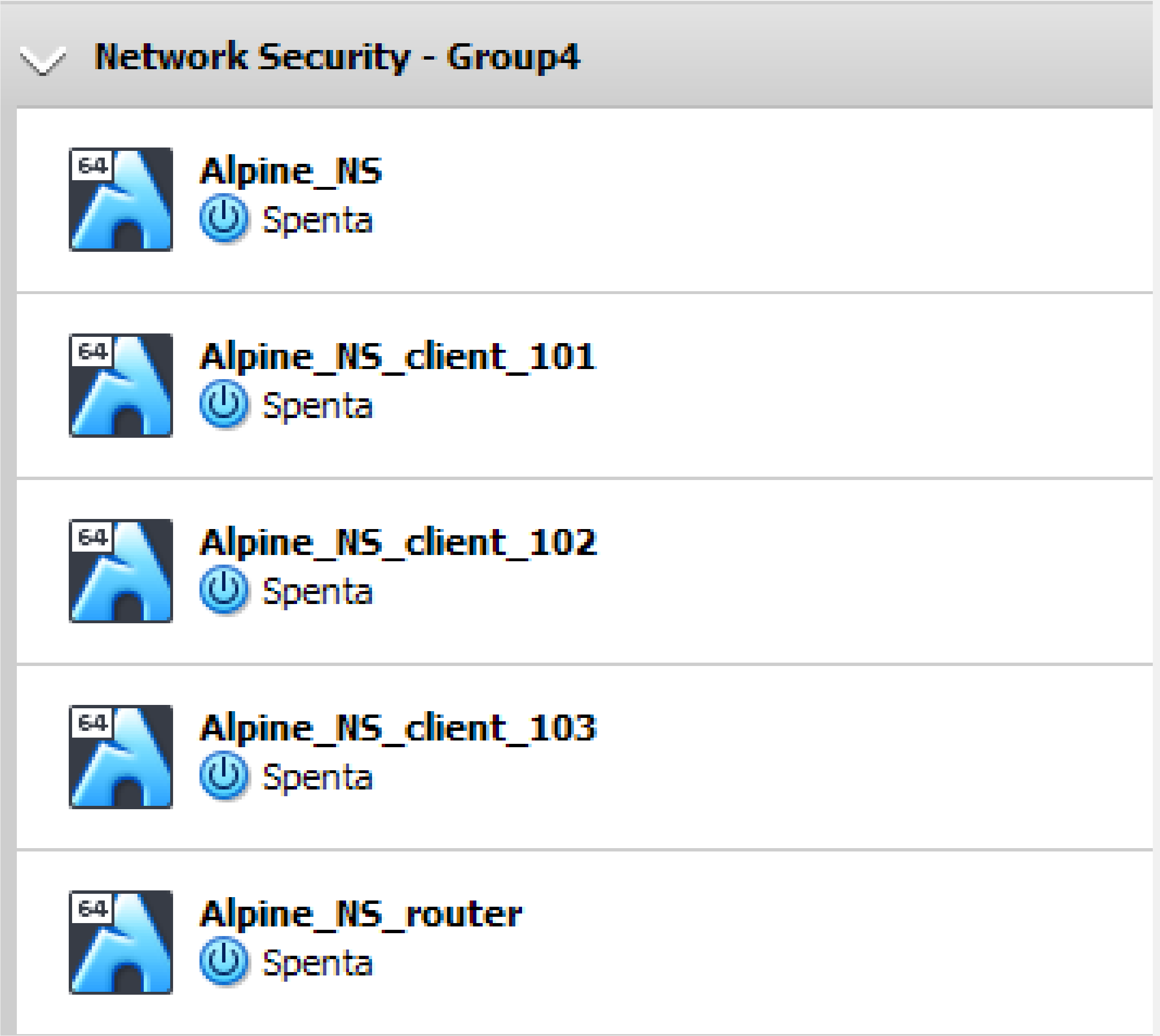
Modalità promiscua: Nega

Indirizzo MAC: 0800278B691B

☑ Cavo connesso

Virtual Machines

Where to find things



Network Security - Group4

Alpine_NS
Spenta

Alpine_NS_client_101
Spenta

Alpine_NS_client_102
Spenta

Alpine_NS_client_103
Spenta

Alpine_NS_router
Spenta

root

admin

Directories

Where to find things

```
snort
    ├──── binaries
    ├──── confs
    │        ├──── snort_ids.lua
    │        └──── snort_ips_nfq.lua
    ├──── dumps
    │        ├──── IDS_exercise
    │        └──── Qakbot.zip
    ├──── enableNFQ.sh
    └──── rules
             └──── snort3-community-rules
```

Aliases

```
snort3="snort --daq afpacket -A alert_full"

snort3_ips="snort --daq-dir /usr/local/lib/daq
            -c /root/snort/confs/snort_ips_nfq.lua -Q -A alert_full"

snort3_pcap="snort --daq pcap -A alert_fast -k none"

snort3_qakbot="snort -c /usr/local/etc/snort/snort.lua -A alert_full
               --lua \"alert_full = { file=true }\""

snort3_portscan="snort --daq afpacket -A alert_fast -i br77
                -c /usr/local/etc/snort/snort.lua
                --lua "port_scan = default_low_port_scan
                ips = {enable_builtin_rules = true, variables = default_variables}""
```

Exercise 1

Write a rule to detect ping from any IP to any IP

Run snort with the following command

```
snort3 -i br77 -R snort/rules/exercise01.rules
```

Exercise 1 solution

Write a rule to detect ping from any IP to any IP

```
alert icmp any any -> any any
(
    sid:1
)
```

Exercise 2

Write a rule to detect ping from .77.101 to .77.102 and write a message when the rule fires

Run snort with the following command

```
snort3 -i br77 -R snort/rules/exercise02.rules
```

Exercise 2 test

Execute those commands:

```
On client 101 (alerted)
$ ping 192.168.77.102

On client 102
$ ping 192.168.77.101
```

Exercise 2 solution

Write a rule to detect ping from .77.101 to .77.102 and write a message when the rule fires

```
alert icmp 192.168.77.101 any -> 192.168.77.102 any
(
   sid:2;
   msg:"ping from client 101 to client 102"
)
```

Exercise 3

Write a rule to detect ping replies from .77.101 to .77.102

Run snort with the following command

```
snort3 -i br77 -R snort/rules/exercise03.rules
```

Hint: use Wiresharsk to inpect the packets and
use the snort option itype: to write the rule

Exercise 3 test

Execute those commands:

```
On client 101 (not alerted)
$ ping 192.168.77.102

On client 102 (alerted only replies from client 101)
$ ping 192.168.77.101
```

Exercise 3 solution

Write a rule to detect ping replies
from .77.101 to .77.102

```
alert icmp 192.168.77.101 any -> 192.168.77.102 any
(
    itype:0;
    sid:3;
    msg:"ping reply from client 101 to client 102"
)
```

Exercise 4

Write a rule to detect TCP packets from .77.101 to .88.103

Run snort with the following command

```
snort3 -i br77 -R snort/rules/exercise04.rules
```

Hint: every client machine has a python script that sends TCP packets to the specified IP address. Use python3 packet.py on client 101 and provide a valid IP to send the packet

Exercise 4 test

Execute those commands:

```
On client 101 (alerted)
$ python3 packet.py (input: 192.168.88.103)

On client 102
$ python3 packet.py (input: 192.168.88.103/.77.101)

On client 103 (reply from client 101 alerted)
$ python3 packet.py (input: 192.168.77.101)
```

Exercise 4 solution

Write a rule to detect TCP packets
from .77.101 to .88.103

```
alert tcp 192.168.77.101 any -> 192.168.88.103 any
(
   sid:4;
   msg:"A TCP packet from client 101 to client 103"
)
```

Exercise 5

Write a rule to detect TCP packets with ACK flag from .77.101 to .88.103

Run snort with the following command

```
snort3 -i br77 -R snort/rules/exercise05.rules
```

Hint: use

client 103: nc -l -p 666
client 101: nc 192.168.88.103 666

Exercise 5 test

Execute those commands:

```
On client 103
$ nc -l -p <port#>

On client 101 (alerted)
$ nc 192.168.88.103 <port#>

On client 102
$ nc 192.168.88.103 <port#>
```

Exercise 5 solution

Write a rule to detect TCP packets with ACK flag from .77.101 to .88.103

```
alert tcp 192.168.77.101 any -> 192.168.88.103 any
(
   flags:*A;
   sid:5;
   msg:"ACK tcp packet from client 101 to client 103"
)
```

Exercise 6

Write a rule to detect DNS requests from home network to anything but the company DNS server

Run snort with the following command

```
snort3 -i br77 -R snort/rules/exercise06.rules
-c snort/confs/snort_ids.lua
```

Hint: rememeber that you can use variables instead of static IPs, check the configuration file snort_ids.lua to get further hints

Exercise 6 test

Execute those commands:

```
On client 101/102
$ nslookup google.com 192.168.88.103

On client 101/102 (alerted)
$ nslookup google.com 1.1.1.1
```

Exercise 6 solution

Write a rule to detect DNS requests from home network to anything but the company DNS server

```
alert udp $HOME_NET any -> !$DNS_SERVER 53
(
    sid:6;
    msg:"DNS request to a not standard DNS server"
)
```

Exercise 7

Write a rule to detect HTTP requests with unallowed user-agents from HOME_NET to .88.103

Run snort with the following command

```
snort3 -i br77 -R snort/rules/exercise07.rules
-c snort/confs/snort_ids.lua
```

Hint: the only user agent we want to accept is wget family

Exercise 7 test

Execute those commands:

```
On client 103
$ python3 -m http.server

On client 101/102
$ wget 192.168.88.103:<port#>

On client 101/102 (alerted)
$ curl 192.168.88.103:<port#>
```

Exercise 7 solution

Write a rule to detect HTTP requests with unallowed user-agents from HOME_NET to .88.103

```
alert http $HOME_NET any -> any any
(
   http_header:field user-agent;
   content:!"wget", nocase;
   sid:7;
   msg:"Unallowed user-agent in HTTP request from HOME_NET"
)
```

Exercise 8

Write a rule to detect SQLi on the pcap file

Run snort with the following command

```
snort3_pcap -c snort/confs/snort_ids.lua
-r snort/dumps/sqlinj.pcap -R snort/rules/exercise08.rules
```

Exercise 8 solution

Write a rule to detect SQLi on the pcap file

```
alert http any any -> any 8080
(
   http_uri:query;
   content:"search=",nocase;
   pcre:"/(.*[\"\']\;)+.*/";
   sid:8;
   msg:"SQLi command in search parameter"
)
```

Exercise 9

An example of Snort3 portscan module

Run snort with the following command

```
snort --daq afpacket -A alert_fast -i br77
-c /usr/local/etc/snort/snort.lua
--lua "port_scan = default_low_port_scan
ips = {enable_builtin_rules = true, variables = default_variables}"
```

Hint: use the alias *snort3_portscan*

Exercise 10

Write a rule to detect QAKbot on the pcap file

Unzip the Quakbot artifact ZIP:

```
unzip snort/dumps/Qakbot.zip -d snort/dumps/Qakbot
```

Password is 'infected'

Exercise 10

Write a rule to detect QAKbot on the pcap file

Run snort with the following command

```
snort3_qakbot -r snort/dumps/Qakbot/2023-05-24-obama264-Qakbot-infection.pcap -R snort/rules/exercise10.rules
```

You can find the logs of snort in the file "alert_full" on the current directory

Exercise 10 solution

Write a rule to detect QAKbot on the pcap file

You can find the rules in the file /opt/snort_rules/exercise10.rules

IPS

Intrusion Prevention System

Run the following command to enable the IPS capabilities for snort

```
snort/enableNFQ.sh
```

Exercise 11

Write a rule to block ping from any IP to any IP

Run snort with the following command

```
snort3_ips -R snort/rules/exercise11.rules
```

Exercise 11 solution

Write a rule to block ping from any IP to any IP

```
block icmp any any -> any any
(
    sid:11
)
```

Exercise 12

Write a rule to block TCP packets with ACK flags from .77.101 to .88.103

Run snort with the following command

```
snort3_ips -R snort/rules/exercise12.rules
```

Exercise 12 solution

Write a rule to block TCP packets with ACK flags from .77.101 to .88.103

```
block tcp 192.168.77.101 any -> 192.168.88.103 any
(
   flags:*A;
   sid:12;
   msg:"Blocked an ACK packet from client 101 to client 103"
)
```

Exercise 13

Write a rule to drop DNS requests for unitn.it domains to not company DNS server

Run snort with the following command

```
snort3_ips -R snort/rules/exercise13.rules
```

Hint: use Wireshark to inspect payload packets and use the snort option *content*

Exercise 13 solution

Write a rule to drop DNS requests for unitn.it domains to not company DNS server

```
drop udp $HOME_NET any -> !$DNS_SERVER 53
(
    content:"|05 75 6e 69 74 6e 02 69 74 00|";
    sid:13;
    msg:"Blocked DNS request to a not standard DNS server"
)
```

Exercise 14

Write a rule to block SQLi on the pcap file

Run snort with the following command

```
snort3_pcap -c snort/confs/snort_ids.lua
-r snort/dumps/sqlinj.pcap -R snort/rules/exercise14.rules
```

Exercise 14 solution

Write a rule to block SQLi on the pcap file

```
block http any any -> any 8080
(
    http_uri:query;
    content:"search=",nocase;
    pcre:"/(.*[\"\']\;)+.*(DROP|INSERT)+/";
    sid:14;
    msg:"Blocked SQL command in search parameter"
)
```

Thank you for your kind attention

Edoardo Mich
Davide Moletta
Tefera Addis Sisay

Network Security Lab 22/23