Ασφάλεια Δικτύων και Επικοινωνιών

(Network and Communication Security)

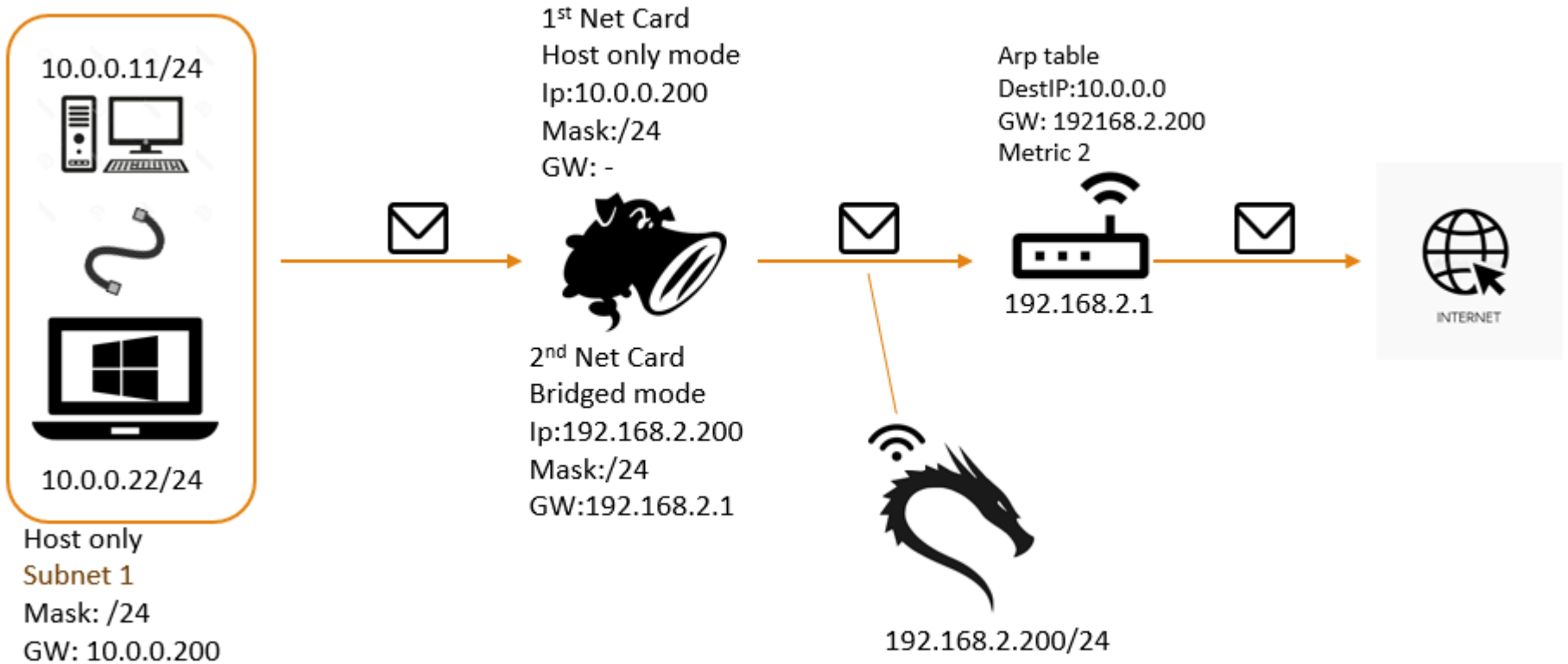
snort NIDS/NIPS lab

Αν. Καθ. Παναγιώτης Κοτζανικολάου

Υπ.Διδ. Δημήτρης Κούτρας

ΠΜΣ Κυβερνοασφάλεια και Επιστήμη Δεδομένων

Τοπολογία


10.0.0.11/24
10.0.0.22/24
Host only
Subnet 1
Mask: /24
GW: 10.0.0.200
1st Net Card
Host only mode
Ip:10.0.0.200
Mask:/24
GW: -
2nd Net Card
Bridged mode
Ip:192.168.2.200
Mask:/24
GW:192.168.2.1
Arp table
DestIP:10.0.0.0
GW: 192168.2.200
Metric 2
192.168.2.1
INTERNET
192.168.2.200/24

Snort - Ubuntu

Installation 2.9.* Version

We need a)pcap - Packet capture b)PCRE - Perl Compatible Regular Expressions c)Libdnet - network functions D)DAQ - Data AcQuisition modules

BHMA 1. Commands

- sudo apt install -y gcc libpcre3-dev zlib1g-dev libluajit-5.1-dev libpcap-dev openssl libssl-dev libnghttp2-dev libdumbnet-dev bison flex libdnet autoconf libtool
- mkdir ~/snort_src && cd ~/snort_src
- wget https://www.snort.org/downloads/snort/daq-2.0.7.tar.gz
- tar -xvzf daq-2.0.7.tar.gz && cd daq-2.0.7
- autoreconf -f -i (need to have autoconf and libtool installed)may not necessary, depends on system
- ./configure && make && sudo make install
- cd ~/snort_src
- wget https://www.snort.org/downloads/snort/snort-2.9.17.tar.gz
- tar -xvzf snort-2.9.16.tar.gz && cd snort-2.9.16
- ./configure --enable-sourcefire && make && sudo make install

Snort - Ubuntu

Installation 2.9.* Version

BHMA 2. Folders

Create the Snort directories:

- sudo mkdir /etc/snort
- sudo mkdir /etc/snort/rules
- sudo mkdir /etc/snort/rules/iplists
- sudo mkdir /etc/snort/preproc_rules
- sudo mkdir /usr/local/lib/snort_dynamicrules
- sudo mkdir /etc/snort/so_rules

BHMA 2. Folders

Create some files that stores rules and ip lists

- sudo touch /etc/snort/rules/iplists/black_list.rules
- sudo touch /etc/snort/rules/iplists/white_list.rules
- sudo touch /etc/snort/rules/local.rules
- sudo touch /etc/snort/sid-msg.map

Snort - Ubuntu

Installation 2.9.* Version

BHMA 2. Folders

#Creating our logging directories:

- sudo mkdir /var/log/snort
- sudo mkdir /var/log/snort/archived_logs

Copy files

- cd ~/snort_src/snort-2.9.8.0/etc/
- sudo cp *.conf* /etc/snort
- sudo cp *.map /etc/snort
- sudo cp *.dtd /etc/snort

BHMA 2. Folders

#Copy rules from snort.org to our files:

Issue:

misconfiguration

Solution:

cd ~/snort_src/snort-2.9.8.0/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicpreprocessor/

sudo cp * /usr/local/lib/snort_dynamicpreprocessor/

Snort - Ubuntu

Installation 3 Version

- Installation with docker
  - https://www.snort.org/snort3
  - https://www.youtube.com/watch?v=PYP0YH2PVuo&t=22s

- Installation by the book (page 3-6)
  - https://www.snort.org/documents
  - Then open the file Snort 3.1.17.0 on Ubuntu 18 & 20

Snort - Ubuntu

BHMA 3. Configuration

sudo gedit /etc/snort/snort.conf

Setup the network addresses you are protecting: ipvar HOME_NET server_public_ip/32

Set up the external network addresses. Leave as "any" in most situations: ipvar EXTERNAL_NET !$HOME_NET

Path to your rules files (this can be a relative path):

- var RULE_PATH /etc/snort/rules
- var SO_RULE_PATH /etc/snort/so_rules
- var PREPROC_RULE_PATH /etc/snort/preproc_rules

Set the absolute path appropriately:

- var WHITE_LIST_PATH /etc/snort/rules/iplists
- var BLACK_LIST_PATH /etc/snort/rules/iplists

line521: output unified2: filename snort.log, limit 128, nostamp, mpls_event_types, vlan_event_types

#comment all default rules: sed –i “s/include \$RULE_PATH/#include \$RULE_PATH/” /etc/snort/snort.conf

Snort - Ubuntu

BHMA 4.Testing

- sudo snort -T -c /etc/snort/snort.conf -i ens33 (ip addr)

BHMA 5. Example rule

- sudo gedit /etc/snort/rules/local.rules
- alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;)

BHMA 6. Log View

- sudo snort -A console -i eth0 -c /etc/snort/snort.conf
- snort -r /var/log/snort/snort.log.(numberOfLog)

Tip for the topology Στο ubuntu μηχάνημα εκτελούμε

Sudo gedit /etc/sysctl.conf : uncomment the line: net.ipv4.ip_forwarding=1 -> reboot

SNORT MODES

- Sniffer mode: Διαβάζει τα πακέτα και τα εμφανίζει με συνεχή ροή στην κονσόλα.
- Packet Logger mode: Δημιουργεί τα log αρχεία.
- Network Intrusion Detection System (NIDS) mode: Πραγματοποιεί ανίχνευση και ανάλυση στην κίνηση του δικτύου.

Sniffer Mode

Snort –v

Εμφανίζει απλώς τις κεφαλίδες IP και TCP/UDP/ICMP

Snort –v –d

Εμφανίζει τα δεδομένα πακέτων καθώς και τις κεφαλίδες

Snort –v –d –e

Εμφανίζει τις κεφαλίδες στο data link επίπεδο με περισσότερη λεπτομέρεια

Packet Logger Mode

Default logging path : /var/log/snort/snort.log.(numberOfLog)

Custom log directories:

snort -dev -l ./log : specify a logging directory.

snort -dev -l ./log -h 192.168.1.0/24 : Αυτός ο κανόνας λέει στο Snort ότι θέλετε να εκτυπώσετε όλη την πληροφορία στο κατάλογο ./log, σε σχέση με το υποδίκτυο (C)192.168.1.0.

snort -l ./log –b : Μια πιο συμπαγή μορφή των πακέτων σε ένα αρχείο για μελλοντική ανάλυση.

snort -dvr packet.log icmp : Διαβάζει τα binary αρχεία φιλτράρει σύμφωνα με το sniffer mode τι μέρος του πακέτου θέλει να δει και διαλέγει μόνο όσα αφορούν icmp.

Network Intrusion Detection System Mode

-c snort.conf : Για την ενεργοποίηση αυτού του mode πρέπει απλά να συνδεθούμε με το configuration file

Alert mode :
`-A fast` : Fast alert mode. Writes the alert in a simple format with a timestamp, alert message, source and destination IPs/ports.
`-A full` : Full alert mode. This is the default alert mode and will be used automatically if you do not specify a mode.
`-A unsock` : Sends alerts to a UNIX socket that another program can listen on.
`-A none` : Turns off alerting.
`-A console` : Sends "fast-style" alerts to the console (screen).
`-A cmg` : Generates "cmg style" alerts.(no reports)

https://linuxhint.com/snort_alerts/

READ THE OUTPUT

The Packet Wire Totals and Action Stats sections of Snort's output include additional fields:

- `Filtered` count of packets filtered out and not handed to Snort for analysis.
- `Injected` packets Snort generated and sent, e.g. TCP resets.
- `Allow` packets Snort analyzed and did not take action on.
- `Block` packets Snort did not forward, e.g. due to a block rule.
- `Replace` packets Snort modified.
- `Whitelist` packets that caused Snort to allow a flow to pass w/o inspection by any analysis program.
- `Blacklist` packets that caused Snort to block a flow from passing.
- `Ignore` packets that caused Snort to allow a flow to pass w/o inspection by this instance of Snort.

The action stats show "blocked" packets instead of "dropped" packets to avoid confusion between dropped packets
(those Snort didn't actually see) and blocked packets (those Snort did not allow to pass)

Inline-InlineTest-Passive Mode

Inline mode, Λειτουργεί ως IPS που επιτρέπει την ενεργοποίηση των κανόνων drop. Το Snort μπορεί να ρυθμιστεί ώστε να εκτελείται σε λειτουργία inline χρησιμοποιώντας το όρισμα -Q και τη λειτουργία ρύθμισης `policy mode` snort ως εξής :

- `snort -Q`
- `config policy_mode:inline`

Σε αυτό το mode το snort επηρεάζεται άμεσα από τα daq modules … τα οποία είναι συσκευές που δρουν σαν interfaces ανάμεσα στο computer και τον αισθητήρα.

Το snort υποστηρίζει διάφορα modules όπως pcap-winpcap-**afpacket-Ipq-Nfq-Ipfw**-Dump. Τα bold υποστηρίζουν inline mode.

Λειτουργούν σαν sniffers και πιάνουν πακέτα, έτσι ώστε το Snort να αποφασίσει τί θα γίνουν.

Inline-InlineTest-Passive Mode

Passive mode, λειτουργεί ως IDS. Οι κανόνες drop δεν φορτώνονται. Το Snort μπορεί να ρυθμιστεί σε παθητική λειτουργία χρησιμοποιώντας τη λειτουργία `policy mode` snort ως εξής:

- config policy_mode:tap

Inline-Test mode προσομοιώνει την inline mode λειτουργία του Snort, επιτρέποντας την αξιολόγηση της inline mode χωρίς να επηρεάζει την ΚΙΝΗΣΗ των πακέτων. Οι κανόνες drop θα φορτωθούν και θα ενεργοποιηθούν ως ειδοποίηση Wdrop (Would Drop). Το Snort μπορεί να είναι έχει ρυθμιστεί ώστε να εκτελείται inline-test mode χρησιμοποιώντας την επιλογή γραμμής εντολών (–enable-inline-test) ή χρησιμοποιώντας το snort λειτουργία `policy mode` ως εξής:

- snort --enable-inline-test
- config policy_mode:inline_test

Packet Acquisition

snort --daq-list :DAQ or Data Acquisition library

Ο τύπος, η λειτουργία, η μεταβλητή και ο κατάλογος DAQ μπορούν να καθοριστούν είτε μέσω της γραμμής εντολών είτε στο αρχείο conf. Η γραμμή εντολών έχει προτεραιότητα

Εάν η λειτουργία δεν έχει οριστεί ρητά, το -Q θα το αναγκάσει σε inline mode και αν δεν έχει οριστεί, το -r θα το αναγκάσει να διαβάσει το αρχείο και εάν αυτό δεν έχει οριστεί, η προεπιλεγμένη λειτουργία είναι η παθητική.

Επίσης, επιτρέπονται -Q και --daq-mode inline, αφού δεν υπάρχει conflict, αλλά το -Q και οποιαδήποτε άλλη λειτουργία DAQ θα προκαλέσει μοιραίο σφάλμα κατά την εκκίνηση.

pcap

Είναι το default module που έχει το DAQ

Δεν μετράει τα φιλτραρισμένα πακέτα

- snort -i <device>
- snort -r <file>
- snort --daq pcap --daq-mode passive -i <device>
- snort --daq pcap --daq-mode read-file -r <file>

You can specify the buffer size pcap uses with:

- snort --daq pcap --daq-var buffer_size=<#bytes>

*pcap on data link layer (OSI layer 2).

AFPACKET

afpacket λειτουργεί παρόμοια με το memory mapped pcap DAQ χωρίς να χρειάζεται κάποια external βιβλιοθήκη:

- ./snort --daq afpacket -i <device>

Εάν θέλετε να εκτελέσετε το afpacket σε λειτουργία inline, πρέπει να ορίσετε τη συσκευή σε ένα ή περισσότερα ζεύγη διεπαφής :

- eth0:eth1
- eth0:eth1::eth2:eth3

Από προεπιλογή, το afpacket DAQ εκχωρεί 128 MB για τη μνήμη πακέτων. Μπορείτε να το αλλάξετε αυτό με :

--daq-var buffer_size_mb=<#MB>

*Raw packet works on IP level (OSI layer 3),

NFQ-IPQ-IPFW

Το **NFQ** είναι ο νέος και βελτιωμένος τρόπος επεξεργασίας πακέτων iptables

Το **IPQ** είναι ο παλιός τρόπος επεξεργασίας πακέτων iptables. (2.9 και κάτω)

IPFW is available for BSD systems (Unix systems but not linux)

- IPFW only supports ip4 traffic

Dump

Το dump DAQ επιτρέπει να δοκιμάσετε τις διάφορες λειτουργίες inline mode που είναι διαθέσιμες στο 2.9 Snort

Το dump χρησιμοποιεί το pcap daq για την απόκτηση πακέτων. Επομένως, δεν μετράει τα φιλτραρισμένα πακέτα

Το dump module δεν αφορά το inline mode.

- snort -r <pcap> -Q --daq dump --daq-var load-mode=read-file
- snort -i <συσκευή> -Q --daq dump --daq-var load-mode=passive

Snort - Rules

include: <include file path/name> var: <name> <value>

Rule structure

- action for traffic matching the rule, alert in this case
  - alert - generate an alert using the selected alert method, and then log the packet
  - log - log the packet
  - pass - ignore the packet
  - activate - alert and then turn on another dynamic rule
  - dynamic - remain idle until activated by an activate rule, then act as a log rule
- traffic protocol like TCP, UDP or ICMP
- the source address and port, simply marked as any to include all addresses and ports
- the destination address and port, `$HOME_NET` as declared in the configuration and any for port
- some additional bits
  - log message
  - unique rule identifier (sid) which for local rules needs to be 1000001 or higher
  - rule version number.

INLINE MODE: DROP, SDROP(no alert), REPLACE

- drop tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"EXPLOIT gobbles SSH exploit attempt"; flow:to_server,established; content:"GOBBLES"; reference:bugtraq,5093; classtype:misc-attack; sid:1812; rev:2;)
- sdrop udp $EXTERNAL_NET any -> $HOME_NET 1434 (msg:"MS-SQL ping attempt"; content:"|02|"; offset:0; depth:1; reference:nessus,10674; classtype:misc-activity; sid:2049; rev:1;)
- replace tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP ~root attempt"; flow:to_server,established; content:"~root"; replace:"~ userroot";nocase;)

COMMENTS

- IP Addresses: Εδώ μπορούμε να χρησιμοποιήσουμε και τον χαρακτήρα ! σαν «not»
- Port numbers: 1:1024 - :6000 - 500: , Χρησιμοποιούμε το ! για να εξαιρέσουμε ports (!2:1025)
- The Direction Operator: -> , <>

HELPFUL LINK

https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm#flags

Running Snort in the background

sudo gedit /lib/systemd/system/snort.service

SCRIPT

[Unit]
Description=Snort NIDS Daemon
After=syslog.target network.target
[Service]
Type=simple
ExecStart=/usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
[Install]
WantedBy=multi-user.target

sudo systemctl daemon-reload

sudo systemctl start snort (stop, restart, and status)

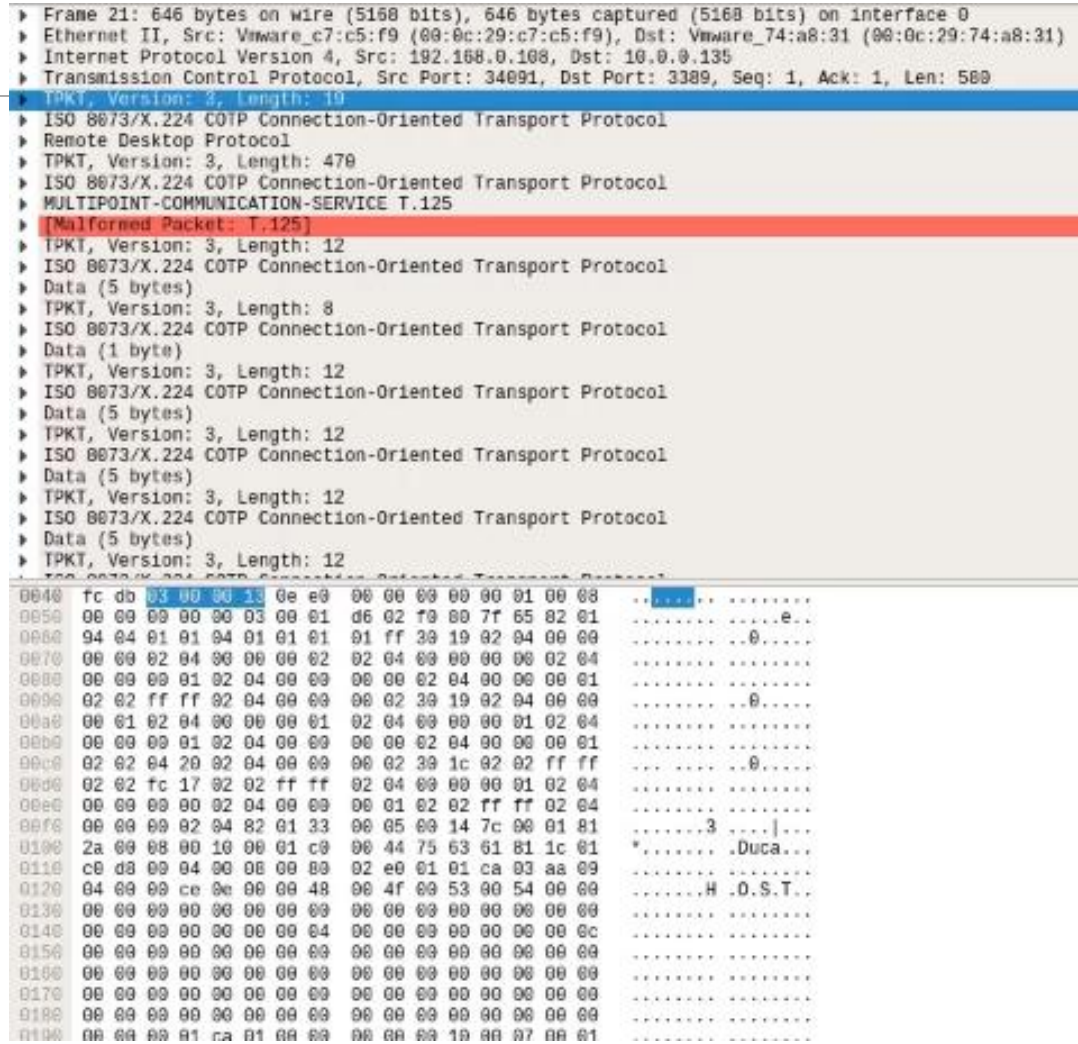
Παραδείγματα Εφαρμογής Κανόνων

Payload Example

- Win 7 Exploit **ms12-020** link: https://pastebin.com/jzQxvnpj
- Useful Link: https://resources.infosecinstitute.com/topic/snort-rules-workshop-part-one/#article
- https://blog.joelesler.net/2010/03/offset-depth-distance-and-within.html
- https://www.computersecuritystudent.com/SECURITY_TOOLS/Metasploit/lesson19/index.html
- We need ips mode in snort(-Q) and a rule with drop.
- Snort Rule against Exploit /etc/snort/rules/local.rules: **drop tcp $EXTERNAL_NET any -> $HOME_NET 3389 (msg:"DOS Microsoft remote desktop protocol RDP "; flow:to_server, established; content:"|03 00|"; depth:2; content:"|e0|";distance:3; within:1; content:"|03 00|"; distance:0; content:"|f0|"; distance:3; within:1; content:"|7f 65|"; distance:1; within:2; content:"|04 01 01 04 01 01 01 01 ff 30|"; distance:3; within:10; content:"|02|"; distance:1; within:1; sid:80000008;rev:1;)**

Payload Example

- Βγάζουμε το default firewall από τα windows. Όλα τα μηχανήματα τρέχουν Python
- Στέλνουμε το exploit από το Kali μηχάνημα στα Windows 7.
  - #python exploit.py 10.0.0.22
- Το Windows 7 μηχάνημα εμφανίζει blue screen
- Ανοίγουμε το Wireshark στο Snort μηχάνημα και βλέπουμε το payload. Κοιτάμε από πού ξεκινάει (δίπλα εικόνα) και φτιάχνουμε ανάλογα τον κανόνα.
- Αφού φτιάξουμε τον κανόνα και ενεργοποιήσουμε το Snort (IDS mode) ξαναδοκιμάζουμε να στείλουμε το κακόβουλο αρχείο.
- Το αρχείο γίνεται drop και τα Windows λειτουργούν κανονικά.



Simple kali lab

U-K: ifconfig

U: gedit /etc/snort/rules/local.rules

snort -T -c /etc/snort/snort.conf -i ens33

snort -A console -q -c /etc/snort/snort.conf -i ens33

K: open wireshark

Nmap –A 192.168.233.139 - ping 192.168.233.139

Open legion -> run a scan to snort

H/W

dictionary bruteforce attack with legion to snort

Simple kali lab 2

K : Create a random txt file (not so many bytes)

Open wireshark

Send this file to snort

sudo hping3 -p 80 -d 5 -E /root/Desktop/5bytes 192.168.233.139 -c 3

U: Create a rule to local.rules to alert to this payload

Open wireshark

Rule be like -> alert tcp any any -> $HOME_NET any (msg:”Custom rule for random payload”;flow:stateless;content:”|54 56 59|”;depth:3; content:”|12 12|”;distance:1;within:2;sid:70000003;rev:1;)

afpacket

sudo gedit /etc/snort/snort.conf

line:157

config daq: afpacket

config daq_dir: /usr/local/lib/daq

config daq_mode: inline

config daq_var: buffer_size_mb=512

Line:265-269(under step 5) uncomment

Σε αυτό το mode πρέπει να έχουμε και drop κανόνες … για αυτό πηγαίνουμε sudo gedit /etc/snort/rules/local.rules και αλλάζουμε το alert σε drop.

Σε αυτό το σημείο το μηχάνημά μας δουλεύει ως ένας διαμεσολαβητής που από την μία κάρτα δικτύου παίρνει πληροφορία και την δίνει φιλτραρισμένη στην άλλη . Για αυτό βάζουμε και δύο interface στην εντολή παρακάτω. Αρα το module δουλεύει με bridged λογική.

-Q inline mode

Sudo snort -c /etc/snort/snort.conf -I ens33:ens34 -Q -A console

Question Τι θα συμβεί εάν βάλουμε 2048 buffer_size_mb σε μηχάνημα με 2 ραμ

nfq

sudo gedit /etc/snort/snort.conf

line:157

config daq: nfq

config daq_dir: /usr/local/lib/daq

config daq_mode: inline

config daq_var: queue=0

Εδώ έχουμε σύνδεση με ένα εργαλείο που λέγεται netfilter και που ουσιαστικά μας δίνει την δυνατότητα να συνδέσουμε τα iptables, arptables με το snort. Γενικότερα μεταφέρονται τα πακέτα από το firewall στην ουρά του snort.

Εάν η ουρά είναι γεμάτη τα πακέτα γίνονται drop.

Εάν το snort δεν τρέχει τα πακέτα μπλοκάρονται.

Libs: sudo apt-get -y install libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev libnfnetlink0

Iptables command: iptables -A FORWARD -j NFQUEUE

nfq install

Snort --daq-list

Cd snort_src/daq-2.0./

sudo apt-get -y install libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev libnfnetlink0

./configure --enable-nfq-module=yes

Make && make install

Snort --daq-list && Cd /usr/local/lib/daq/ && ls

iptables -A FORWARD -j NFQUEUE (-F[delete] –L[show])

iptables -A FORWARD -j NFQUEUE –queue-num=0

Snort –A console –q –c /etc/snort/snort.conf -Q

Questions: what can we do if we lost packets when both iptables and snort are activated?

Problem - Issue 1

Οι παρακάτω εντολές λύνουν ένα θέμα που δημιουργείται με την χρήση κανονικών μηχανημάτων και όχι εικονικών. Το θέμα αφορά το μέγεθος των πακέτων γιατί το snort κόβει τα πακέτα μεγαλύτερα των 1518 bytes (Large Receive Offload and Generic Receive offload)

sudo apt-get install -y ethtool

post-up ethtool -K eth0 gro off

post-up ethtool -K eth0 lro off

https://askubuntu.com/questions/579231/whats-the-difference-between-prerouting-and-forward-in-iptables

Problem - Issue 2

Τα ICMP πακέτα δρομολογούνται κατευθείαν στην τοπική IP χωρίς να περάσουν από την ουρά σε κάποιες περιπτώσεις. Αυτό λύνεται με κανόνα που προλαμβάνει την δρομολόγηση στην τοπική IP(10,0,0,8)

sudo iptables -t nat -I PREROUTING -j NFQUEUE --queue-num=0 OR

sudo iptables -t nat -A PREROUTING -j NFQUEUE --queue-num=0

installation:

snort --daq-list

cd /snort_src/daq-2.0.6/

./configure --enable-nfq=yes

make

cd /usr/local/lib/daq/ then ls and we find nfq

go to iptables: Insert – Append – jump

iptables -F (delete iptables)

iptables -L

iptables -A FORWARD -j NFQUEUE --queue-num=0

sudo snort -A console -c /etc/snort/snort.conf –Q -q: Δημιουργεί αυτόματα το bridge και δεν χρειάζεται interface. Το αποτέλεσμα είναι να βλέπουμε στο μηχάνημά μας την κίνηση όπως τη βλέπει αυτός που την στέλνει.

Snort - Windows

- Εύκολη εγκατάσταση
- 7zip, winrar
- winpcap (network module that used in wireshark and windows packet capture library)
- copy **preproc** , **rules** from official snort website

Βήματα μετά την εγκατάσταση

- Verify snort installation : Snort -V in the c:\Snort\bin>
- Άνοιγμα αρχείου snort.conf

Καθορισμός ipvar HOME_NET 192.168.233.0/24
Set up external address -> !$HOME_NET

- HTTP ports – IANA 0-1023(defaults) 1024-49151(commonly used) 49152-65535(free)

Snort - Windows

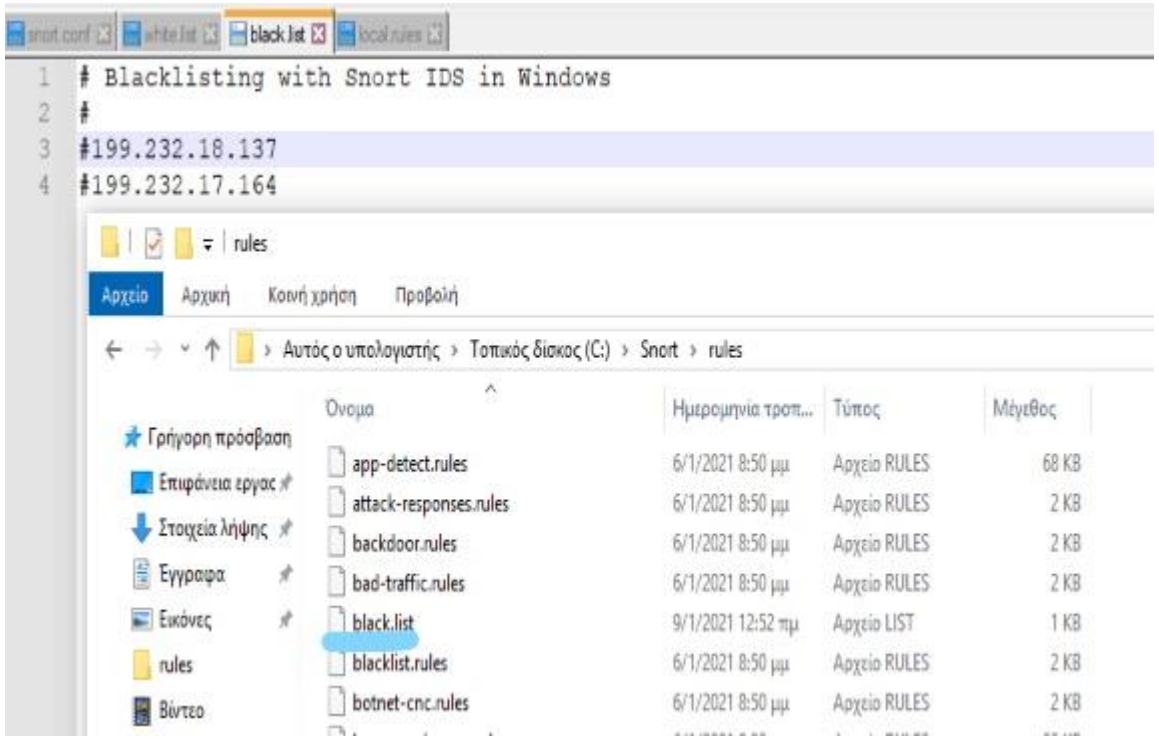
Βήματα μετά την εγκατάσταση

- Line 104: change var rule path c:\Snort\rules
- Comment the shared object path because it not used in windows
- Change the preproc rule path variable c:\Snort\preproc_rules
- Change **white** and **black list** path variable with c:\Snort\rules
- Line 186: uncomment logdir and we put the path c:\Snort\log
- Line 243: comment the 3rd lib
  - We fix the pathing for the other two line 247,250 (the one can target to the file and the other to the folder)

Snort - Windows

Βήματα μετά την εγκατάσταση

```
506 # Reputation preprocessor. For more information see README.reputation
507 preprocessor reputation: \
508    memcap 500, \
509    priority whitelist, \
510    nested_ip inner, \
511    whitelist $WHITE_LIST_PATH\white.list, \
512    blacklist $BLACK_LIST_PATH\black.list
513
```



Snort - Windows

Βήματα μετά την εγκατάσταση

- Line 265-269: we comment the preprocessor tasks because there is not inline mode in windows
- Line 335: comment back orifice detection …. a computer program designed for remote system administration … it was used for win 95
- Line 418: enable portscan (sfportscan)
- Line 507: Δημιουργούμε δύο αρχεία .list με σκοπό να βάλουμε λίστες από IP(μπορούμε να αφήσουμε και τα αρχεία .rules αλλά θα πρέπει το περιεχόμενο να έχει την μορφή κανόνων)(Δείτε την προηγούμενη διαφάνεια)
- Line 528 : we set the server with the plug in for the output of the logs
- Step 7(line:539): replace / with \
- Step 8(line:654): replace / with \ and enable the preproc rules

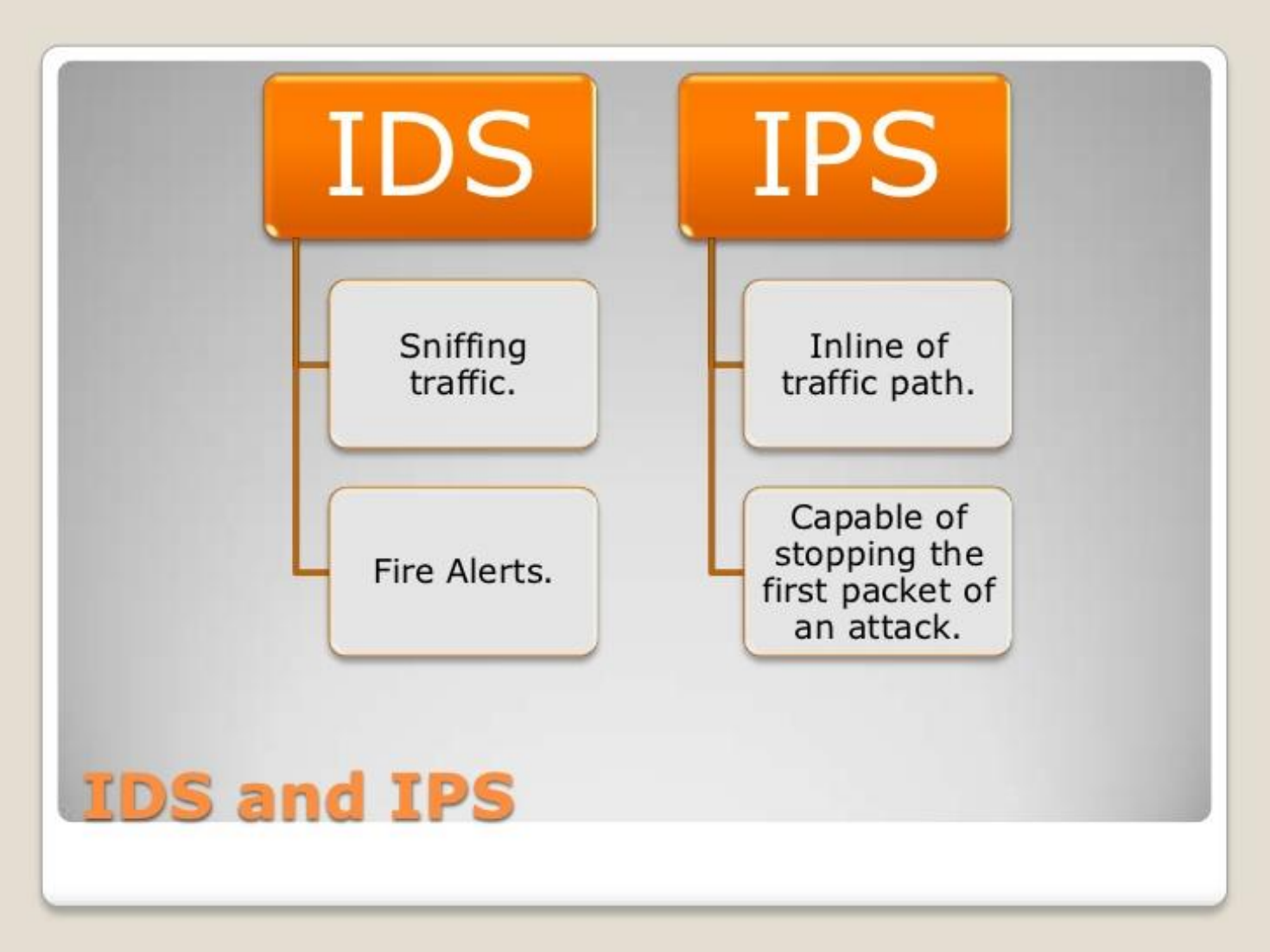
Snort - Windows

TESTING

- *(We have to move on the Snort\bin)*
- *Find the interface:*
  - **snort -W**
  - **wmic nic get NetConnectionID**
  - **netsh int show int**
  - **ip addr**
- ***snort -i Ethernet0 -c c:\Snort\etc\snort.conf -T***

RULES

- *(DAQ(-1) ERROR: remove from bin the wpcap.dll and the Packet.dll and after we reinstall the winpcap)*
- Simple rule: Into the local.rules:
  - ***alert tcp any any -> any any (msg:"mpla mpla"; sid:1000001;)***
- Test rule: **snort -i 2 -c c:\Snort\etc\snort.conf -A console**


IDS
Sniffing traffic.
Fire Alerts.
IPS
Inline of traffic path.
Capable of stopping the first packet of an attack.
IDS and IPS