



Lab setup for Cybersecurity students (Defensive Security)

Author: Yagub Hajiyev
Contact: yagub.hajiyev.995@gmail.com



Table of Contents

Summary	3
Pre-requirements	3
1. OPNsense setup and general configuration	7
1.1 Installing OPNsense firewall	7
1.2 Initial configuration	8
1.3 Firewall access configuration	15
1.4 Configuration backup	18
1.5 Aliases	18
1.6 Writing initial rules	20
1.6.1 CLIENTS rules	20
1.6.2 JUMPSERVERS rules	25
1.6.3 GUESTS rules	27
1.7 NAT configuration	29
2. OPNsense plugins	31
2.1 Nginx plugin.....	31
2.2 Squid Web Proxy plugin	37
2.3 ClamAV and C-ICAP plugins	47
2.4 Zenarmor plugin	51
3. OPNsense additional features	56
3.1 GeoIP blocking.....	56
3.2 Captive Portal	58
3.3 IDS/IPS configuration.....	64
Conclusion	75



Summary

This lab is designed for understanding the basic network security concept in **Cybersecurity**. The main component of the network security is firewall, and we will learn how to install, configure and manage a firewall. Before we start this lab, you need to have a fundamental knowledge of firewalls. I passed the most of theoretical information and focused on practice to keep the documentation short. You also need to deploy some machines on your own, and should have fundamental **Linux** and **Active Directory** skills for performing it. If you face any trouble while setup the lab, please contact with me.

This lab covers some key security methods that demonstrated according to **TCP/IP** model in the table,

As you see, most of the security preventions will be performed in **Application** layer. In following captures, we will examine and test all these methods step by step.

At the end of the lab, you will be able to perform,

- Firewall setup and general configuration
- Firewall policy management
- Installing **OPNsense** firewall plugins and configuration
- **IDS/IPS** setup and configuration in firewall
- Monitoring activities

Pre-requirements

In order to setup this lab you have to prepare several machines in virtual environment, which are listed below,

- **DC:** Create a domain to use in your lab environment (which I used **cs301.local** domain in this lab). Use this machine as your **Domain Controller** and **DNS** server, its **IP** address will be also **DNS** address for all machines in our lab (except **Guest** machine because we will isolate it from our networks). I highly recommend to use the same parameters as I used in this lab for better troubleshooting.

IP: 172.16.10.10

Subnet: 255.255.255.0



Gateway: 172.16.10.2

DNS: 172.16.10.10 (or 127.0.0.1)

Domain: cs301.local

Hostname: DC

- **WEB:** This machine will serve as a web application server in our environment. You need to deploy the web application which named **DVWA** (Damn Vulnerable Web Application) on it. This application is vulnerable to cyberattacks and our task is to protect it from threats.

IP: 172.16.10.11

Subnet: 255.255.255.0

Gateway: 172.16.10.2

DNS: 172.16.10.10

Domain: cs301.local

Hostname: WEB

DVWA setup: <https://www.howtoforge.com/how-to-install-damn-vulnerable-web-application-on-centos-8/>

Join Oracle machine to the AD domain: <https://blogs.oracle.com/cloud-infrastructure/post/joining-an-oracle-linux-client-to-an-active-directory-domain>

- **Jump:** We will talk about jump server logic in following captures, now just deploy a **Windows** server machine and give the parameters listed below,

IP: 172.16.30.10

Subnet: 255.255.255.0

Gateway: 172.16.30.2

DNS: 172.16.10.10

Domain: cs301.local

Hostname: Jump

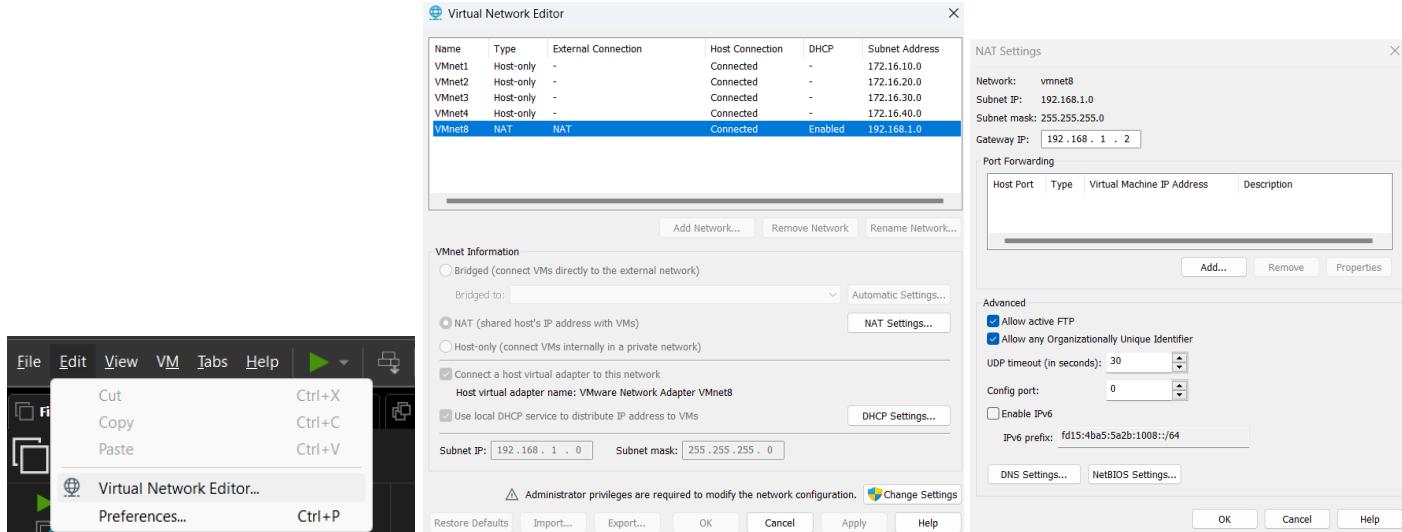
- **Client:** For simplicity, we will use only one client machine but you can use more. Join this machine to the domain and name it. We will deploy the **DHCP** service for **CLIENTS** interface in our firewall and all machines will get an **IP** from this service.

- **Guest:** It is depend on you to use whatever **OS** you want in guest machine, it also will obtain an **IP** address from the **DHCP** service deployed for **GUESTS** interface in firewall.

- **VMware:** **VMware** comes with 3 adapters (**VMNET0**, **VMNET1** and **VMNET8**) in default configuration but we need 5 network adapters in our lab. In order to prepare your **VM** (Virtual Machine) open **VMware** machine go to the **Virtual Network Editor** window (**Edit => Virtual Network Editor**) and make



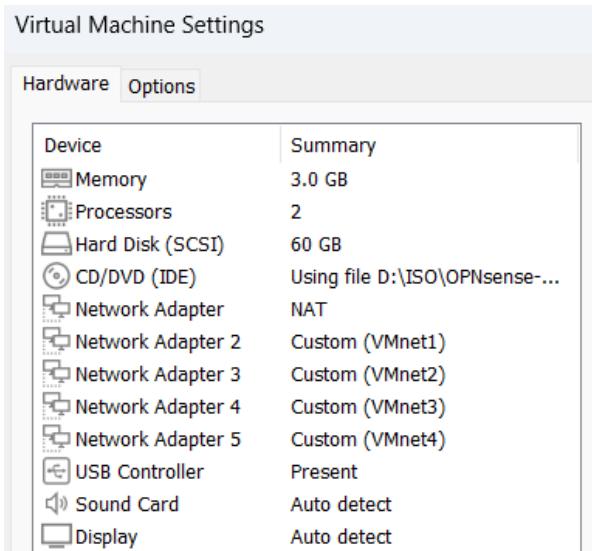
sure networks parameters are set as required. In our lab, we will use one of the adapters (**VMNET8**) as **NAT** (**DHCP** enabled in **VMware** for other machines) and others as host only (**DHCP** disabled, because we need to give static IPs for our interfaces and machines in order to control and monitor their network traffic).



The upstream gateway address (**192.168.1.2**) will let us connect our machines to the Internet (don't confuse it with other gateway addresses).

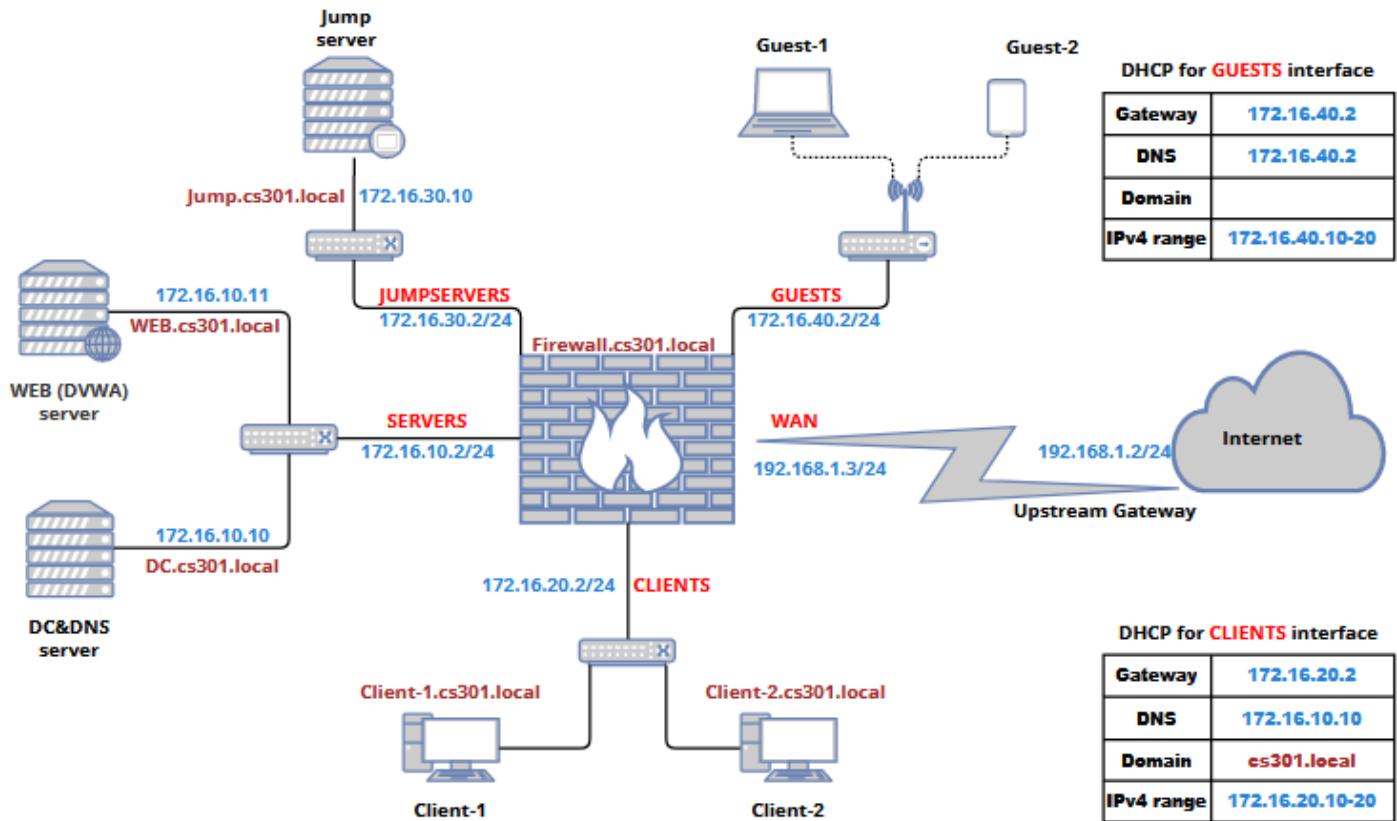
- **Firewall:** We will use an open source firewall, **OPNsense**, in our lab environment. **OPNsense** has many features which enterprise firewalls offer us. You can download its **ISO** file from **OPNsense**'s official web site. Once you downloaded the **ISO** file, create new machine in **VMware** and add all adapters required. The figure displays the example to the configuration in **VMware** (firewall) **Settings** window.

Link: <https://mirror.ams1.nl.leaseweb.net/opnsense/releases/24.1/OPNsense-24.1-dvd-amd64.iso.bz2>





The general scheme of our project is demonstrated below, for better understanding, I added 2 guests and 2 clients but we will use only one in each networks.



Adapter name	Interface name	Subnet IP	Host IP	Firewall IP
NAT	WAN	192.168.1.0/24	192.168.1.1	192.168.1.3
VMNET1	SERVERS	172.16.10.0/24	172.16.10.1	172.16.10.2
VMNET2	CLIENTS	172.16.20.0/24	172.16.20.1	172.16.20.2
VMNET3	JUMPSERVERS	172.16.30.0/24	172.16.30.1	172.16.30.2
VMNET4	GUESTS	172.16.40.0/24	172.16.40.1	172.16.40.2



OPNsense setup and general configuration

1.1 Installing OPNsense firewall

Once we start installing process, the following window will be opened, we need to login with the **installer** user by entering the default password **opnsense**.

```
Service 'sysctl' has been restarted.
>>> Invoking start script 'beep'
Root file system: /dev/iso9660:OPNSENSE_INSTALL
Sat Mar 23 12:14:23 UTC 2024

*** OPNsense.localdomain: OPNsense 24.1 ***

LAN (em0)      -> v4: 192.168.1.1/24
WAN (em1)      ->

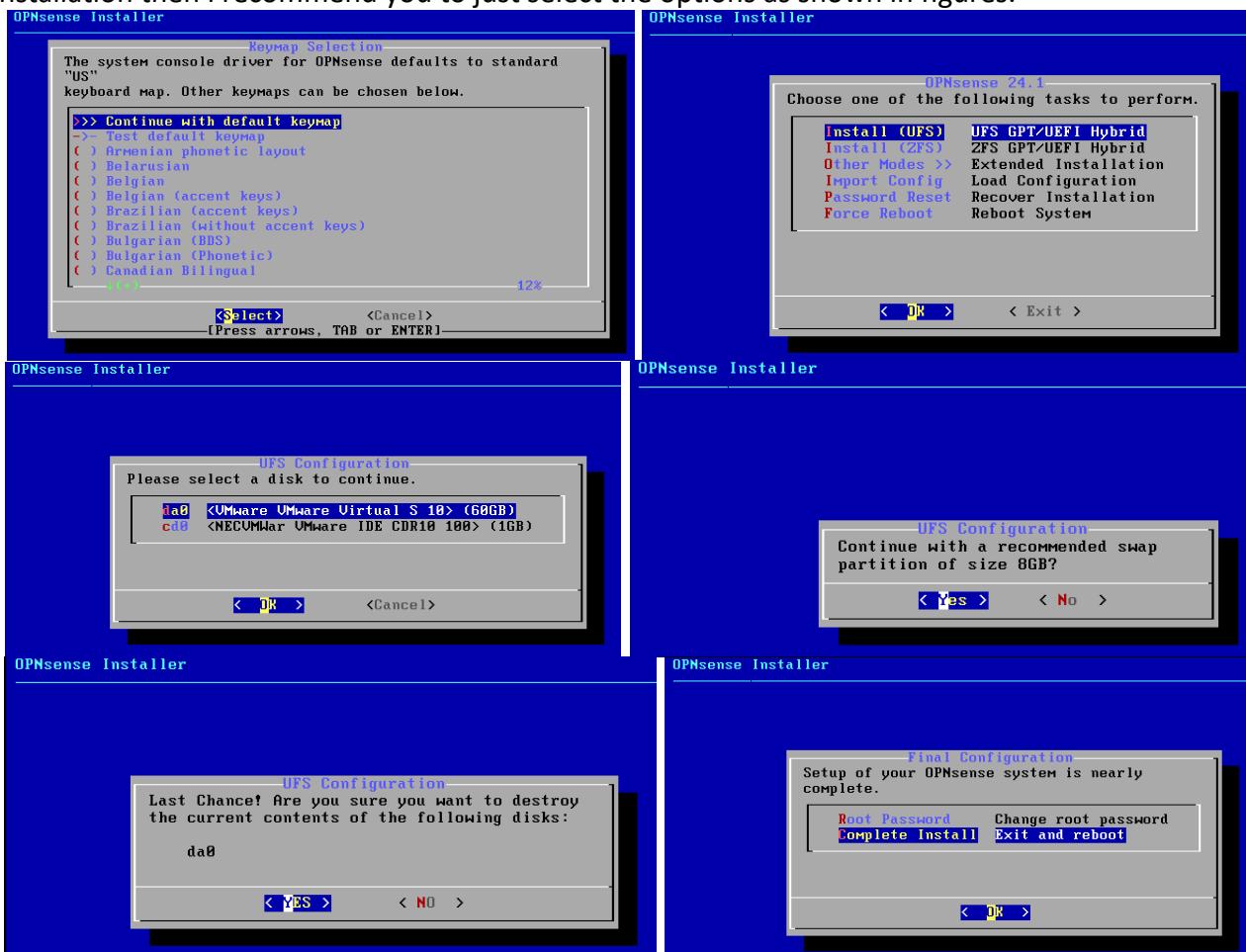
HTTPS: SHA256 CE B0 C3 C7 C0 F8 6F F3 A6 B8 C5 EA 09 30 F0 02
       88 B4 06 DC 0C 48 9A EA D3 6A 0D 8D 00 CD 6B D7
SSH:  SHA256 hBr5qH8nyI72sUdZFpQsFtktE3byh4shkYT/EP6cyHY (ECDSA)
SSH:  SHA256 I0fEix0bbsp9LgU630bOSvt8ru1CU/MoT0qD5z0U/Ac (ED25519)
SSH:  SHA256 wIB39Uq609xxUR23gp+s6F221lujLC6mJtEXL3MLFI (RSA)

Welcome! OPNsense is running in live mode from install media. Please
login as 'root' to continue in live mode, or as 'installer' to start the
installation. Use the default or previously-imported root password for
both accounts. Remote login via SSH is also enabled.

FreeBSD/amd64 (OPNsense.localdomain) (ttyv0)

login: installer
Password:■
```

After successful login into the firewall the following windows will popped up, if you are not familiar with this installation then I recommend you to just select the options as shown in figures.





1.2 Initial configuration

Rebooting the system will complete the installation and our firewall will be ready for configuration. Before we can use the firewall on **GUI** (Graphical User Interface), we need to login into **CLI** (Command Line Interface) with default credentials (username: **root** | password: **opnsense**) and configure at least two interfaces parameters, **NAT** and **LAN**, we will use **LAN** interface IP to open the **GUI**.

```
>>> Invoking start script 'newwanip'
>>> Invoking start script 'freebsd'
>>> Invoking start script 'syslog'
>>> Invoking start script 'carp'
>>> Invoking start script 'cron'
Starting Cron: OK
>>> Invoking start script 'openvpn'
>>> Invoking start script 'sysctl'
Service 'sysctl' has been restarted.
>>> Invoking start script 'beep'
Root file system: /dev/gpt/rootfs
Sat Mar 23 12:28:07 UTC 2024

*** OPNsense.localdomain: OPNsense 24.1 ***
LAN (em0)      -> v4: 192.168.1.1/24
WAN (em1)      ->

HTTPS: SHA256 CE B0 C3 C7 C0 F8 6F F3 A6 B8 C5 EA 09 30 F0 02
       88 B4 06 DC 0C 48 9A EA D3 6A 0D 8D 00 CD 6B D7

FreeBSD/amd64 (OPNsense.localdomain) (ttyv0)

login: root
Password: ■
```

Website: https://opnsense.org/ | @@@@\\ //@@@@
Handbook: https://docs.opnsense.org/ |)))))))) (((((((
Forums: https://forum.opnsense.org/ | @@@@// \\@@@@
Code: https://github.com/opnsense | @@@@@ @@@@
Twitter: https://twitter.com/opnsense | @@@@@@@@@@@@@@@@@@

*** OPNsense.localdomain: OPNsense 24.1 ***

LAN (em0) -> v4: 192.168.1.1/24
WAN (em1) ->

HTTPS: SHA256 CE B0 C3 C7 C0 F8 6F F3 A6 B8 C5 EA 09 30 F0 02
 88 B4 06 DC 0C 48 9A EA D3 6A 0D 8D 00 CD 6B D7

0) Logout
1) Assign interfaces
2) Set interface IP address
3) Reset the root password
4) Reset to factory defaults
5) Power off system
6) Reboot system
7) Ping host
8) Shell
9) pfTop
10) Firewall log
11) Reload all services
12) Update from console
13) Restore a backup

Enter an option: ■

After login into firewall, we will start with assigning interfaces. In the command line type **1** and press **Enter** to open interfaces page, for the **LAGGs** and **WLANS** sections choose **No**. As we see in the figure, it will display valid interfaces for our firewall, the names (**em0**, **em1**, **em2** and **em3**) are default assigned names by **OPNsense**, and they are arranged from up to down as we entered in **VMware** firewall **VM** settings (right click on **VM** and go to **settings** in popped window), **em0=NAT**, **em1=VMNET1**, **em2=VMNET2**, **em3=VMNET3** and **em4=VMNET4**.

NOTE: Make sure that the **MAC** addresses of interfaces are same with adapters' **MAC** addresses in **VM settings**.

```
1) Assign interfaces          8) Shell
2) Set interface IP address  9) pfTop
3) Reset the root password   10) Firewall log
4) Reset to factory defaults 11) Reload all services
5) Power off system         12) Update from console
6) Reboot system            13) Restore a backup

Enter an option: 1

Do you want to configure LAGGs now? [y/N]: n
Do you want to configure VLANs now? [y/N]: n

Valid interfaces are:

em0          00:0c:29:e8:dc:b2 Intel(R) Legacy PRO/1000 MT 82545EM (Copper)
em1          00:0c:29:e8:dc:bc Intel(R) Legacy PRO/1000 MT 82545EM (Copper)
em2          00:0c:29:e8:dc:c6 Intel(R) Legacy PRO/1000 MT 82545EM (Copper)
em3          00:0c:29:e8:dc:d0 Intel(R) Legacy PRO/1000 MT 82545EM (Copper)
em4          00:0c:29:e8:dc:da Intel(R) Legacy PRO/1000 MT 82545EM (Copper)

If you do not know the names of your interfaces, you may choose to use
auto-detection. In that case, disconnect all interfaces now before
hitting 'a' to initiate auto detection.

Enter the WAN interface name or 'a' for auto-detection: █
```

We will be asked to enter the name for **WAN** interface and later for **LAN**, **OPT1** and **OPT2**, add all names and leave empty when it asks **OPT3** name.



em0	00:0c:29:e8:dc:b2	Intel(R)	Legacy	PRO/1000 MT	82545EM	(Copper)
em1	00:0c:29:e8:dc:b3	Intel(R)	Legacy	PRO/1000 MT	82545EM	(Copper)
em2	00:0c:29:e8:dc:c6	Intel(R)	Legacy	PRO/1000 MT	82545EM	(Copper)
em3	00:0c:29:e8:dc:d0	Intel(R)	Legacy	PRO/1000 MT	82545EM	(Copper)
em4	00:0c:29:e8:dc:da	Intel(R)	Legacy	PRO/1000 MT	82545EM	(Copper)

If you do not know the names of your interfaces, you may choose to use auto-detection. In that case, disconnect all interfaces now before hitting 'a' to initiate auto detection.

Enter the WAN interface name or 'a' for auto-detection: em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(or nothing if finished): em1

Enter the Optional interface 1 name or 'a' for auto-detection
(or nothing if finished): em2

Enter the Optional interface 2 name or 'a' for auto-detection
(or nothing if finished): em3

Enter the Optional interface 3 name or 'a' for auto-detection
(or nothing if finished): em4

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(or nothing if finished): em1

Enter the Optional interface 1 name or 'a' for auto-detection
(or nothing if finished): em2

Enter the Optional interface 2 name or 'a' for auto-detection
(or nothing if finished): em3

Enter the optional interface's name or a for auto-detection
(or nothing if finished): em4

(or nothing if finished):

```
The interface configuration:  
WAN  -> em0  
LAN  -> em1  
OPT1 -> em2  
OPT2 -> em3  
OPT3 -> em4
```

Do you want to proceed? [y/N]: y

After defining interfaces, we will assign **IPv4** addresses for them. In the command line type **2** and press **Enter** to start assign IPs. In our lab we will configure only two interfaces, **WAN** and **LAN**, in **CLI** and others in **GUI**. Lets begin with **WAN** interface by choosing option **4** and enter the parameters as shown in figures below.

```
*** OPNsense.localdomain: OPNsense 24.1.4 ***

LAN (em1)      -> v4: 192.168.1.1/24
OPT1 (em2)     ->
OPT2 (em3)     ->
OPT3 (em4)     ->
WAN (em0)      -> v4/DHCP4: 192.168.1.129/24

HTTPS: SHA256 68 A0 29 B8 19 D2 29 8E 0C C6 24 14 F1 14 95 60
       33 71 1E 9A 31 7A 4A C3 8D 4E BE 21 08 F7 82 B3

  0) Logout                      7) Ping host
  1) Assign interfaces           8) Shell
  2) Set interface IP address   9) pfTop
  3) Reset the root password    10) Firewall log
  4) Reset to factory defaults  11) Reload all services
  5) Power off system            12) Update from console
  6) Reboot system                13) Restore a backup

Enter an option: 2
```

```
Enter an option: 2

Available interfaces:

1 - LAN (em1 - static, track6)
2 - OPT1 (em2)
3 - OPT2 (em3)
4 - OPT3 (em4)
5 - WAN (em0 - dhcp, dhcp6)

Enter the number of the interface to configure: 5

Configure IPv4 address WAN interface via DHCP? [Y/n] n

Enter the new WAN IPv4 address. Press <ENTER> for none:
> 192.168.1.3

Subnet masks are entered as bit counts (like CIDR notation)
e.g. 255.255.255.0 = 24
      255.255.0.0   = 16
      255.0.0.0     = 8

Enter the new WAN IPv4 subnet bit count (1 to 32):
> 24 ■
```

The upstream gateway address (which our firewall will use for connecting to the **Internet**) have to be the same with **NAT** adapter's gateway address as we assigned in **VMware, Virtual Network Editor** section above. In the last second question we will be asked to generate self-signed certificate for using **GUI** in our web browser and choose **Yes** option to generate it.

```
Choose the option to generate it.

Enter the new WAN IPv4 subnet bit count (1 to 32):
> 24

For a WAN, enter the new WAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
> 192.168.1.2

Do you want to use the gateway as the IPv4 name server, too? [Y/n] y

Configure IPv6 address WAN interface via DHCP6? [Y/n] n

Enter the new WAN IPv6 address. Press <ENTER> for none:
>

Do you want to change the web GUI protocol from HTTPS to HTTP? [y/N]
Do you want to generate a new self-signed web GUI certificate? [y/N]
Restore web GUI access defaults? [y/N] y
```



NOTE: in **VMware**, first **IP** address always automatically is assigned to the host itself, so we have to begin to use from the second address for our interfaces, but in **NAT** adapter the second **IP (192.168.1.2)** is given to the upstream gateway. This is why we used **192.168.1.3** **IP** for **WAN** interface. In other interfaces will use the second **IPs** for our interfaces because in our firewall environment there will be only one upstream gateway address.

After configuring the **WAN**, we will assign the **LAN**'s parameters as shown in following figures.

```
*** OPNsense.localdomain: OPNsense 24.1.4 ***
LAN (em1)      -> v4: 192.168.1.100/24
OPT1 (em2)     ->
OPT2 (em3)     ->
OPT3 (em4)     ->
WAN (em0)      -> v4: 192.168.1.3/24

HTTPS: SHA256 B1 2E 3A E7 2C CD 6F B1 B8 AF 74 09 E4 33 88 E4
       0B 6C 20 F8 DF F3 B5 C8 AC 47 D0 B5 CC 46 A1 58

0) Logout          7) Ping host
1) Assign interfaces 8) Shell
2) Set interface IP address 9) pfTop
3) Reset the root password 10) Firewall log
4) Reset to factory defaults 11) Reload all services
5) Power off system 12) Update from console
6) Reboot system 13) Restore a backup

Enter an option: 2 ■
```

Enter an option: 2
 Available interfaces:
 1 - LAN (em1 - static, track6)
 2 - OPT1 (em2)
 3 - OPT2 (em3)
 4 - OPT3 (em4)
 5 - WAN (em0 - static)
 Enter the number of the interface to configure: 1
 Configure IPv4 address LAN interface via DHCP? [y/N] n
 Enter the new LAN IPv4 address. Press <ENTER> for none:
 > 172.16.10.2
 Subnet masks are entered as bit counts (like CIDR notation).
 e.g. 255.255.255.0 = 24
 255.255.0.0 = 16
 255.0.0.0 = 8
 Enter the new LAN IPv4 subnet bit count (1 to 32):
 > 24 ■

```
Enter the new LAN IPv4 subnet bit count (1 to 32):
> 24

For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Configure IPv6 address LAN interface via DHCP? [y/N] n

Enter the new LAN IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? [y/N] n

Do you want to change the web GUI protocol from HTTPS to HTTP? [y/N] n
Do you want to generate a new self-signed web GUI certificate? [y/N] y
Restore web GUI access defaults? [y/N] y ■
```

Now we are ready to use **GUI**, open your browser and go to the <https://172.16.10.2> address. In the opening windows select **Advanced** and “Accept the Risk and Continue”. Later enter the default credentials (username: **root** | password: **opnsense**) and login.

Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **172.16.10.2**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

[Learn more...](#)

[Go Back \(Recommended\)](#) [Advanced...](#)

172.16.10.2 uses an invalid security certificate.
The certificate is not trusted because it is self-signed.
Error code: [MOZILLA_PRIX_ERROR_SELF_SIGNED_CERT](#)

[View Certificate](#)

[Go Back \(Recommended\)](#) [Accept the Risk and Continue](#)

OPNsense

Username:

Password:

[Login](#)

OPNsense (c) 2014-2024 Deciso B.V.



Once we enter the **GUI**, the early step should be updating system. First go to the **System => Firmware => Status** and “**Check for updates**”.

Type
Version
Architecture
Commit
Mirror
Repositories
Updated on
Checked on

Check for updates

If there are available updates it will be displayed in **System => Firmware => Updates** section. Press **Update** button and confirm the action in popped windows. By the completion of update process, the firewall will reboot itself. After the reboot we will continue our firewall’s configuration.

py39-aioquic	0.9.24	0.9.25	upgrade	OPNsense
py39-anyio	4.2.0	4.3.0	upgrade	OPNsense
py39-attrs	23.1.0	23.2.0	upgrade	OPNsense
py39-bottleneck	1.3.7_1	1.3.8	upgrade	OPNsense
py39-certifi	2023.11.17	2024.2.2	upgrade	OPNsense
py39-cryptography	41.0.7_2,1	42.0.5,1	upgrade	OPNsense
py39-dateutil	2.8.2	2.9.0	upgrade	OPNsense
py39-driftphon	2.5.0,1	2.6.1,1	upgrade	OPNsense
py39-httcore	1.0.2	1.0.4	upgrade	OPNsense
py39-htpx	0.2.6	0.2.7,0	upgrade	OPNsense
py39-markupsafe	2.1.3	2.1.5	upgrade	OPNsense
py39-numexpr	2.8.8	2.9.0	upgrade	OPNsense
py39-numpy	1.25.0_5,1	1.25.0_6,1	upgrade	OPNsense
py39-packaging	N/A	23.2	new	OPNsense
py39-pytz	2023.3,1	2024.1,1	upgrade	OPNsense
py39-sniffo	1.3.0	1.3.1	upgrade	OPNsense
py39-typing-extensions	4.9.0	4.10.0	upgrade	OPNsense
py39-tzdata	2023.4	2024.1	upgrade	OPNsense
radvd	2.19_2	2.19_3	upgrade	OPNsense
readline	8.2.7_1	8.2.10	upgrade	OPNsense
rstool	1.8_0,3	1.8_0,4	upgrade	OPNsense
sqlite3	3.45.0_1,1	3.45.1,1	upgrade	OPNsense
strongswan	5.9.13	5.9.13_1	upgrade	OPNsense
sudo	1.9.15p5_3	1.9.15p5_4	upgrade	OPNsense
suricata	7.0.2_3	7.0.4	upgrade	OPNsense
syslog-ng	4.4.0	4.6.0_2	upgrade	OPNsense
unbound	1.19.0	1.19.3	upgrade	OPNsense

Update **Cancel**

There are 83 updates available, total download size is 215.6MB. This update requires a reboot.

As we see in **Summary** capture, we choose the name **Firewall** for our **OPNsense** firewall and our domain is **cs301.local**, the time zone parameter has to be the same as we use in our other machines and it is also essential for logs because when there is an incident in our network, we have to know the exact time to take an action against it. We used our **DC** machine as our **DNS** server in the domain, so in this section we have to add **DC**’s **IPv4** address to synchronize our firewall with **DNS** service.



The next configuration will be assigning descriptive names and IP addresses (we didn't assign IPs to **OPT1**, **OPT2** and **OPT3** interfaces yet) to our firewall interfaces.

For the **WAN** interface we add **WAN** into the **Description** section, for the **LAN** interface we assign **SERVERS** name, which will be used for our servers in the following captures and leave rest of configurations as default.

For **OPT1** type **CLIENTS** which will be used for our client machines and choose "**Static IPv4**" in "**IPv4 Configuration Type**" section, scroll down and in the "**IPv4 address**" section assign **172.16.20.2** IP and set subnet **24** for **CLIENTS** interface. Later we will activate a **DHCP** service for this interface.



Repeat the same configuration for **OPT2** and **OPT3** interfaces, as we did for **OPT1**, by using the parameters which displayed in figures below. This interface will be used for controlling remotely our **DC** machine, which is located in **SERVERS** interface.

Enable	<input checked="" type="checkbox"/> Enable Interface
Lock	<input type="checkbox"/> Prevent interface
Identifier	opt2
Device	em3
Description	JUMPERS
Generic configuration	
Block private networks	<input type="checkbox"/>
Block bogon networks	<input type="checkbox"/>
IPv4 Configuration Type	Static IPv4
Interfaces [CLIENTS] [OPT2] [OPT3] [SERVERS] [WAN]	
Assignments Overview Settings Neighbors Virtual IPs Wireless	
Overwrite global settings <input type="checkbox"/>	
Static IPv4 configuration IPv4 address : 172.16.30.2	

Enable	<input checked="" type="checkbox"/> Enable Interface
Lock	<input type="checkbox"/> Prevent interface
Identifier	opt3
Device	em4
Description	GUESTS
Generic configuration	
Block private networks	<input type="checkbox"/>
Block bogon networks	<input type="checkbox"/>
IPv4 Configuration Type	Static IPv4
Interfaces [CLIENTS] [JUMPERS] [OPT3] [SERVERS] [WAN]	
Assignments Overview Settings Neighbors Virtual IPs Wireless	
Overwrite global settings <input type="checkbox"/>	
Static IPv4 configuration IPv4 address : 172.16.40.2	

After successfully configuring all interfaces, we are ready to **DHCP** service for **CLIENTS** interface. This service will run in lab environment, so it is enough to give range short as I did, **172.16.20.10-20**, and **DNS** address have to be same in whole network, which is the **IPv4** address of **DC** server in our lab. The gateway address is the address of **CLIENTS** interface, but apart from upstream gateway (which we assigned for **WAN** interface), this gateway will serve only for local communication between subnets (**WAN**, **SERVERS** and **JUMPERS**). Last we add our domain, **cs301.local**, and finish our configuration for the **DHCP** service. We can monitor and control our services on **Lobby => Dashboard => Services** section.



ISC DHCPv4: [GUESTS]

<input type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable DHCP server on the CLIENTS interface		
<input type="checkbox"/> Deny unknown clients	<input type="checkbox"/>		
<input type="checkbox"/> Ignore Client UIDs	<input type="checkbox"/>		
<input type="checkbox"/> Subnet	172.16.20.0		
<input type="checkbox"/> Subnet mask	255.255.255.0		
<input type="checkbox"/> Available range	172.16.20.1 - 172.16.20.254		
<input type="checkbox"/> Range	from	to	
	172.16.20.10		172.16.20.20
<input type="checkbox"/> Additional Pools	Pool Start	Pool End	Description
<input type="checkbox"/> WINS servers			
<input type="checkbox"/> DNS servers	172.16.10.10		
<input type="checkbox"/> Gateway	172.16.20.2		
<input type="checkbox"/> Domain name	cs301.local		

DHCP settings for GUESTS network.

Services: ISC DHCPv4: [GUESTS]

<input type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable DHCP server on the GUESTS interface		
<input type="checkbox"/> Deny unknown clients	<input type="checkbox"/>		
<input type="checkbox"/> Ignore Client UIDs	<input type="checkbox"/>		
<input type="checkbox"/> Subnet	172.16.40.0		
<input type="checkbox"/> Subnet mask	255.255.255.0		
<input type="checkbox"/> Available range	172.16.40.1 - 172.16.40.254		
<input type="checkbox"/> Range	from	to	
	172.16.40.10		172.16.40.100
<input type="checkbox"/> Additional Pools	Pool Start	Pool End	Description
<input type="checkbox"/> WINS servers			
<input type="checkbox"/> DNS servers	172.16.40.2		
<input type="checkbox"/> Gateway	172.16.40.2		



1.3 Firewall access configuration

For the security purposes it is not recommended to use default users, credentials and ports. In this capture we will create a new user, disable the default **root** user and change access ports to make our firewall environment much more secure.

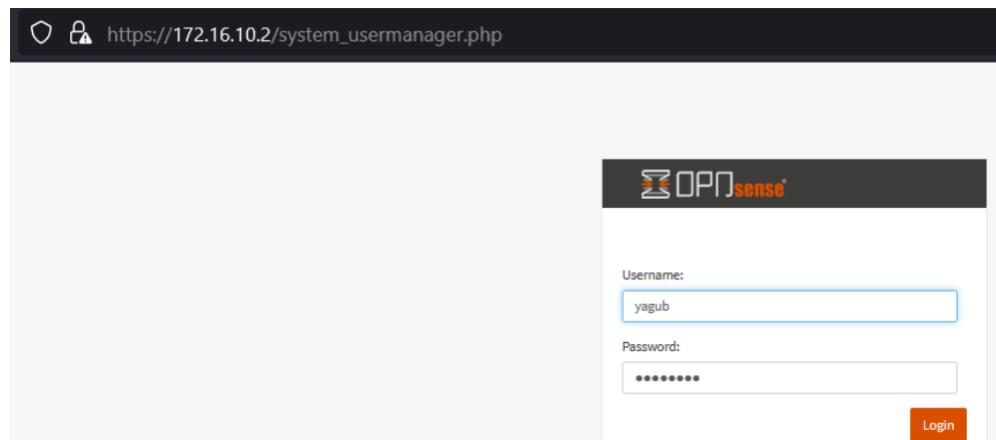
In order to create a new user, go to the **System => Access => Users** section and press plus (+) button to add new user.

Username	Full name	Groups
root	System Administrator	admins

In the opening window, add your credentials, choose your “**Login shell**” and add your user to the **admins** group which will allow this new user to control the firewall with **root** privileges.

After creating the user logout from **root** user and login with the new user.

Username	Full name	Groups
root	System Administrator	admins
yagub	Yagub Hajiyev	admins



Before we disable the **root** user, we need to activate **wheel** group in our firewall, otherwise we will not be able to use **root** privileges. Open the console (**CLI**) and open the **sudoers** file which is located under **/usr/local/etc/sudoers** path and take the comment hash (#) in front of “**%wheel ALL=(ALL:ALL) ALL**” line. (you can install **nano** tool with “**pkg install nano**” command)

```
root@Firewall:~ # nano /usr/local/etc/sudoers
```

```
GNU nano 7.2          /usr/local/etc/sudoers

## Uncomment to allow members of group wheel to execute any command
# %wheel ALL=(ALL:ALL) ALL

## Same thing without a password
# %wheel ALL=(ALL:ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
# %sudo ALL=(ALL:ALL) ALL

## Uncomment to allow any user to run sudo if they know the password
## of the user they are running the command as (root by default).
# Defaults targetpw # Ask for the password of the target user
# ALL ALL=(ALL:ALL) ALL # WARNING: only use this together with 'Defaults targetpw'

## Uncomment to show on password prompt which users' password is being expected
# Defaults passprompt="%p's password"

## Read drop-in files from /usr/local/etc/sudoers.d
@includedir /usr/local/etc/sudoers.d

^G Help      ^O Write Out  ^W Where Is   ^X Cut       ^T Execute   ^C Location
^X Exit      ^R Read File  ^M Replace    ^U Paste     ^J Justify   ^Y Go To Line
```

Go back to the **System => Access => Users** section and press **Edit** button in **root** user's line.

	Username	Full name	Groups	
	root	System Administrator	admins	
	yagub	Yagub Hajiyev	admins	

In the opened window, disable (put the tick in **Disabled** box) the root user and save. Now you can't use the **root** user login to the firewall until you enable it again.



Lobby

Reporting

System

- Access
- Users
- Groups
- Servers
- Tester

Configuration

System: Access: Users

Defined by	SYSTEM
Disabled	<input checked="" type="checkbox"/>
Username	root

In the next step we have to change the **HTTPS** and **SSH** access ports. Go to the **System => Setting => Administration** section and change the default “**TCP port**” from **443** to whatever number you want for **HTTPS** access, I used **30443** in this lab, and set the **SERVERS** interface in the “**Listen Interfaces**” section, which will allow only **SERVERS** and block other interfaces for entering to the **GUI** interface.

Reporting

System

- Access
- Configuration
- Firmware
- Gateways
- High Availability
- Routes
- Settings

Administration

- Cron
- General
- Logging
- Miscellaneous
- Tunables
- Trust
- Wizard
- Log Files
- Diagnostics

Interfaces

- Firewall
- VPN

Protocol: HTTP HTTPS

SSL Certificate: Web GUI TLS certificate

SSL Ciphers: System defaults

HTTP Strict Transport Security: Enable HTTP Strict Transport Security

TCP port: 30443

HTTP Redirect: Disable web GUI redirect rule

Login Messages: Disable logging of web GUI successful logins

Session Timeout: 240

DNS Rebind Check: Disable DNS Rebinding Checks

Alternate Hostnames: Alternate Hostnames for DNS Rebinding and HTTP_REFERER

HTTP Compression: Off

Access log: Enable access log

Server Log: Log server errors

Listen Interfaces: SERVERS

After changing **HTTPS** port, scroll down, enable **Secure Shell**, change the “**SSH port**” and set **SERVERS** in the “**Listen interfaces**” and save configuration changes.

Lobby

Reporting

System

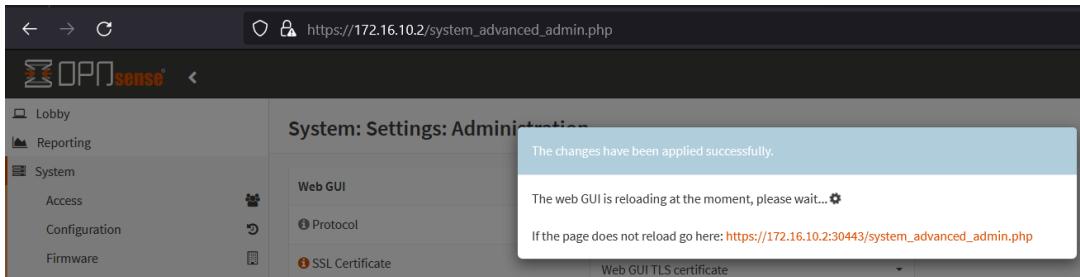
- Access
- Configuration
- Firmware
- Gateways
- High Availability
- Routes
- Settings

Secure Shell

Secure Shell Server	<input checked="" type="checkbox"/> Enable Secure Shell
Login Group	wheel, admins
Root Login	<input type="checkbox"/> Permit root user login
Authentication Method	<input checked="" type="checkbox"/> Permit password login
SSH port	3022
Listen Interfaces	SERVERS

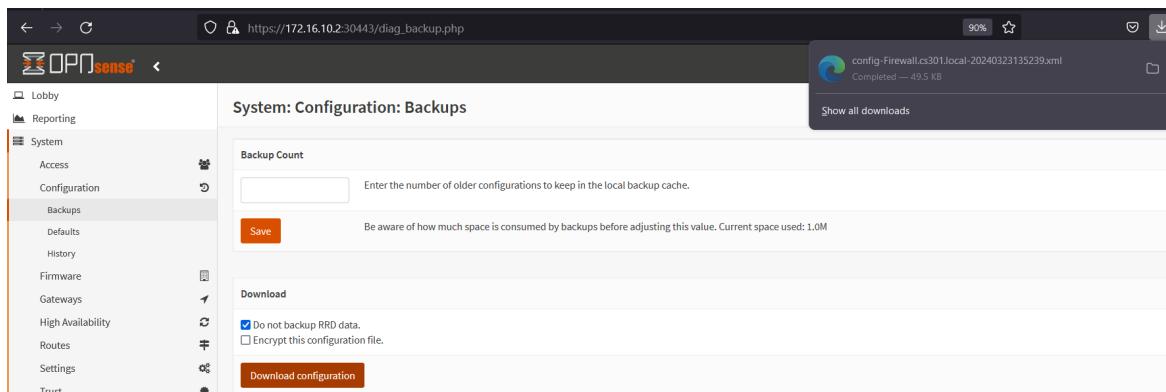


Once you hit the Save button, the system will pop up the window displayed in the figure, and will require to go new address, just click the **URL** and go to the new page. This is happened because we changed the default **HTTPS** port number and this will add extra security layer to our firewall, which means if an attacker wants to gain access our GUI interface then he/she has to find the correct port number to load login page.



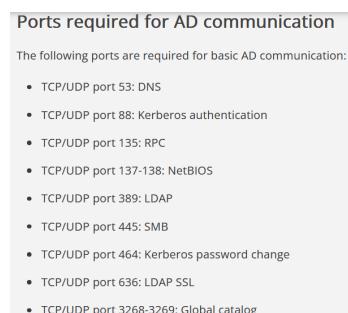
1.4 Configuration backup

When we have to make a critical configuration changes or updates the system, it sometimes causes system crash problems. One of the recommended step is to always taking backups of the system just before doing any operation. Go to the **System => Configuration => Backups** section and download the current configurations, later you can use this **XML** file to restore your configuration in this section.



1.5 Aliases

In firewall environment, aliases help us to list ports, hosts and networks under a single name, which we can use just calling the alias's name in various entities of firewall, such as firewall rule. Below in the figures, we will create the alias of **Active Directory** ports for using it in different firewall rules. This will reduce the effort we spend on each rule, because we will call just the alias name in different rules, instead of adding ports into the rules each time.





There are a few default aliases in **OPNsense** firewall, which can be found in **Firewall => Aliases** path. Press the **Add** button to add new alias.

Enabled	Name	Type	Description	Content	Loaded#	Last updated	Commands
<input type="checkbox"/>	bogons	External (advanced)	bogon networks (internal)		10		
<input type="checkbox"/>	bogonsv6	External (advanced)	bogon networks IPv6 (internal)		76		
<input type="checkbox"/>	virusprot	External (advanced)	overload table for rate limiting (int...)		0		
<input type="checkbox"/>	sshlockout	External (advanced)	abuse lockout table (internal)		0		
<input type="checkbox"/>	_wan_network	Internal (automatic)	WAN net		1		
<input type="checkbox"/>	_lan_network	Internal (automatic)	SERVERS net		1		
<input type="checkbox"/>	_lo0_network	Internal (automatic)	Loopback net		2		

In the “**Edit Alias**” window, type the name of alias in the **Name**, select **Port(s)** option in the **Type**, add the ports to the **Content** section and type a short description whatever you want.

Enabled:

Name: AD_ports

Type: Port(s)

Content: 53, 88, 135, 137, 138, 389, 445, 464, 636, 3268, 3269

Description: Active Directory ports

Buttons: Cancel, Save

Once you **Save** and close the popped up window, your new alias will be displayed in **Firewall => Aliases** section, then press the **Apply** button to make it available in firewall environment. And it is done, we are ready to use it wherever we want in our firewall.

Enabled	Name	Type	Description	Content	Loaded#	Last updated	Commands
<input type="checkbox"/>	AD_ports	Port(s)	Active Directory ports	53,88,137,138,389,445,464,636,32...	10		
<input type="checkbox"/>	bogons	External (advanced)	bogon networks (internal)		10		
<input type="checkbox"/>	bogonsv6	External (advanced)	bogon networks IPv6 (internal)		76		
<input type="checkbox"/>	virusprot	External (advanced)	overload table for rate limiting (int...)		0		
<input type="checkbox"/>	sshlockout	External (advanced)	abuse lockout table (internal)		0		
<input type="checkbox"/>	_wan_network	Internal (automatic)	WAN net		1		
<input type="checkbox"/>	_lan_network	Internal (automatic)	SERVERS net		1		



1.6 Writing initial rules

1.6.1 CLIENTS rules

In this capture we will write a few rules for **CLIENTS** interface to allow and block some actions. The short summary will be like this: for **CLIENTS** interface machines, allow **AD** ports in **DC** and block others, allow **RDP** port in **JUMPSERVERS** interface and block others. Lets begin with allowing **AD** ports in **DC**.

Go to the **Firewall => Rules => CLIENTS**, add new rule, in the rule page, select **Pass** for **Action**, select **CLIENTS** in **Interface**, select **TCP/UDP** for **Protocol**, select “**CLIENTS net**” as **Source**, select “**Single host or Network**” in **Destination** and type the **DC** address with **CIDR 32**, select your alias name for **AD** ports (which is **AD_ports** in this lab) in “**Destination port range**” section and other options as default.

Edit Firewall rule

- Action:** Pass
- Disabled:** Disable this rule
- Quick:** Apply the action immediately on match.
- Interface:** CLIENTS
- Direction:** in
- TCP/IP Version:** IPv4
- Protocol:** TCP/UDP
- Source / Invert:** Use this option to invert the sense of the match.
- Source:** CLIENTS net
- Source:** Advanced
- Destination / Invert:** Use this option to invert the sense of the match.
- Destination:** Single host or Network
 - 172.16.10.10
 - 32
- Destination port range:** from: AD_ports to: AD_ports

Our rule will be added in **Firewall => Rules => CLIENTS** section at the bottom of existing rule. Every new created rule locates at the bottom of all other existing rules, we will arrange them in this at the end of this capture.

Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description	Actions
IPv4 TCP/UDP	CLIENTS net	*	172.16.10.10	AD_ports	*	*	Automatically generated rules	[Edit]

Now we need to add new rule to block all other connections with **SERVERS** network. Again add new rule



and in the rule window, select **Action**, select **CLIENTS** for Interface, select “**CLIENTS net**” as source, select “**SERVERS net**” option as the **Destination** and leave all other options as default. This rule will block all traffic goes from **CLIENTS** network to the **SERVEFRS** network.

Firewall: Rules: CLIENTS

Edit Firewall rule

Action	<input type="button" value="Block"/>
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	<input type="button" value="CLIENTS"/>
Direction	<input type="button" value="in"/>
TCP/IP Version	<input type="button" value="IPv4"/>
Protocol	<input type="button" value="any"/>
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	<input type="button" value="CLIENTS net"/>
Source	<input type="button" value="Advanced"/>
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	<input type="button" value="SERVERS net"/>

Firewall: Rules: CLIENTS

The changes have been applied successfully.

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Actions	Description	Buttons			
<i>Automatically generated rules</i>													
<input type="checkbox"/>	IPv4 TCP/UDP	CLIENTS net	*	172.16.10.10	AD_ports	*	*		Automatically generated rules				
<input type="checkbox"/>	IPv4 *	CLIENTS net	*	SERVERS net	*	*	*						



In the next rule we will allow only RDP port (**TCP 3389**) access from **CLIENTS** to **JUMPSERVERS**. In the rule, select **Pass** for **Action**, select **CLIENTS** for interface, select **TCP** as **Protocol**, select “**CLIENTS net**” as **Source**, select **JUMPSERVERS** as **Destination**, select “**(other)**” option in “**Destination port range**” and enter the RDP port in both, **from** and **to** section, then leave all other options as default and **Save** the rule.

Firewall: Rules: CLIENTS

The changes have been applied successfully.

	Protocol	Source	Port	Destination	Port	Gateway	Schedule		Description				
<i>Automatically generated rules</i>													
	IPv4 TCP/UDP	CLIENTS net	*	172.16.10.10	AD_ports	*	*						
	IPv4 *	CLIENTS net	*	SERVERS net	*	*	*						
	IPv4 TCP	CLIENTS net	*	JUMPSERVERS net	3389 (MS RDP)	*	*						

Now we need to block all other connections with **JUMPSERVERS** network. Add new rule, select **Block** for **Action**, select **CLIENTS** for **Interface**, select **CLIENTS** as **Source**, select **JUMPSERVERS** as **Destination** and leave all other options as default.



Firewall: Rules: CLIENTS

Edit Firewall rule

Action	Block
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	CLIENTS
Direction	in
TCP/IP Version	IPv4
Protocol	any
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	CLIENTS net
Source	Advanced
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	JUMPSERVERS net

Firewall: Rules: CLIENTS

Select category

The changes have been applied successfully.

	Protocol	Source	Port	Destination	Port	Gateway	Schedule		Description				
Automatically generated rules													
<input type="checkbox"/>	IPv4 TCP/UDP	CLIENTS net	*	172.16.10.10	AD_ports	*	*						
<input type="checkbox"/>	IPv4 *	CLIENTS net	*	SERVERS net	*	*	*						
<input type="checkbox"/>	IPv4 TCP	CLIENTS net	*	JUMPSERVERS net	3389 (MS RDP)	*	*						
<input type="checkbox"/>	IPv4 *	CLIENTS net	*	JUMPSERVERS net	*	*	*						

In the next rule we will block all traffic with the **GUESTS** network, add new rule, select **Block** for **Action**, select “**CLIENTS net**” as **Source**, select “**GUESTS net**” as **Destination** and leave all other options default.



Firewall: Rules: CLIENTS

Edit Firewall rule

i Action	<input type="button" value="Block"/>
i Disabled	<input type="checkbox"/> Disable this rule
i Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
i Interface	<input type="button" value="CLIENTS"/>
i Direction	<input type="button" value="in"/>
i TCP/IP Version	<input type="button" value="IPv4"/>
i Protocol	<input type="button" value="any"/>
i Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
i Source	<input type="button" value="CLIENTS net"/>
Source	<input type="button" value="Advanced"/>
i Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
i Destination	<input type="button" value="GUESTS net"/>

Firewall: Rules: CLIENTS

The changes have been applied successfully.										Select category	Ir
Protocol	Source	Port	Destination	Port	Gateway	Schedule	Actions	Description	Buttons		
<i>Automatically generated rules</i>											
IPv4 TCP/UDP	CLIENTS net	*	172.16.10.10	AD_ports	*	*					
IPv4 *	CLIENTS net	*	SERVERS net	*	*	*					
IPv4 TCP	CLIENTS net	*	JUMPSERVERS net	3389 (MS RDP)	*	*					
IPv4 *	CLIENTS net	*	JUMPSERVERS net	*	*	*					
IPv4 *	CLIENTS net	*	GUESTS net	*	*	*					



In the last rule, we give full access for **CLIEANTS** network.

Firewall: Rules: CLIENTS

Edit Firewall rule	
Action	<input type="text" value="Pass"/>
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	<input type="text" value="CLIENTS"/>
Direction	<input type="text" value="in"/>
TCP/IP Version	<input type="text" value="IPv4"/>
Protocol	<input type="text" value="any"/>
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	<input type="text" value="any"/>
Source	<input type="button" value="Advanced"/>
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	<input type="text" value="any"/>

Firewall: Rules: CLIENTS

Select category

The changes have been applied successfully.								
	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description
Automatically generated rules								
<input type="checkbox"/>	IPv4 TCP/UDP	CLIENTS net	*	172.16.10.10	AD_ports	*	*	<input type="button" value="+"/> <input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/>
<input type="checkbox"/>	IPv4 *	CLIENTS net	*	SERVERS net	*	*	*	<input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/>
<input type="checkbox"/>	IPv4 TCP	CLIENTS net	*	JUMPSERVERS net	3389 (MS RDP)	*	*	<input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/>
<input type="checkbox"/>	IPv4 *	CLIENTS net	*	JUMPSERVERS net	*	*	*	<input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/>
<input type="checkbox"/>	IPv4 *	CLIENTS net	*	GUESTS net	*	*	*	<input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/>
<input type="checkbox"/>	IPv4 *	*	*	*	*	*	*	<input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/>

As you see we have 6 rules in **CLIENTS** interface, and they have to be arranged like shown in the figure below. In order to understand the logic of firewall rules, lets read them from left to right and from top to bottom.

1-Allow **TCP** connection from **CLIENTS** network to **JUMPSERVERS** network's port **3389**

2-Block any connection from **CLIENTS** network to **JUMPSERVERS** network

3-Allow **TCP/UDP** connections from **CLIENTS** network to **172.16.10.10 (DC)** host's **AD** ports

4-Block any connection from **CLIENTS** network to **SERVERS** network

5-Block any connection from **CLIENTS** network to **GUESTS** network

6-Allow any connection from **CLIENTS** network to any network



1.6.2 JUMPSERVERS rules

In order to keep the instruction short, I will not write all steps for **JUMPSERVERS** interface because it is similar with the **CLIENTS** rules. So it will be enough to understand its logic just by analyzing the following figures.

Firewall: Rules: JUMPSERVERS

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description	Automatically generated rules	Actions
No JUMPSERVERS rules are currently defined. All incoming connections on this interface will be blocked until you add a pass rule. Exceptions for automatically generated rules may apply.										
<input type="checkbox"/>	pass			block				reject		
<input checked="" type="checkbox"/>	pass (disabled)			block (disabled)				reject (disabled)		

Firewall: Rules: JUMPSERVERS

Edit Firewall rule

Action	Pass
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	JUMPSERVERS
Direction	in
TCP/IP Version	IPv4
Protocol	TCP/UDP
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	JUMPSERVERS net
Source	Advanced
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	Single host or Network
Destination	172.16.10.10
Destination port range	from: AD_ports to: AD_ports

Firewall: Rules: JUMPSERVERS

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description	Automatically generated rules	Actions
The firewall rule configuration has been changed. You must apply the changes in order for them to take effect.										
<input type="checkbox"/>	IPv4 TCP/UDP	JUMPSERVERS net	*	172.16.10.10	AD_ports	*	*			
<input checked="" type="checkbox"/>	pass			block				reject		
<input checked="" type="checkbox"/>	pass (disabled)			block (disabled)				reject (disabled)		



Firewall: Rules: JUMPSERVERS

Edit Firewall rule

Action	Pass
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	JUMPSERVERS
Direction	in
TCP/IP Version	IPv4
Protocol	TCP
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	JUMPSERVERS net
Source	Advanced
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	SERVERS net
Destination port range	from: (other) 3389 to: (other) 3389

Firewall: Rules: JUMPSERVERS

Select category

The changes have been applied successfully.

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Actions
<small>Automatically generated rules</small>								
<input type="checkbox"/>	IPv4 TCP/UDP	JUMPSERVERS net	*	172.16.10.10	AD_ports	*	*	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Check"/> <input type="button" value="Copy"/>
<input type="checkbox"/>	IPv4 TCP	JUMPSERVERS net	*	SERVERS net	3389 (MS RDP)	*	*	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Check"/> <input type="button" value="Copy"/>



Firewall: Rules: JUMPSERVERS

Edit Firewall rule

Action	Block
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	JUMPSERVERS
Direction	in
TCP/IP Version	IPv4
Protocol	any
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	JUMPSERVERS net
Source	Advanced
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	any
Destination port range	from: any to: any

Firewall: Rules: JUMPSERVERS

The changes have been applied successfully.

	Protocol	Source	Port	Destination	Port	Gateway	Schedule		Description					
Automatically generated rules														
<input type="checkbox"/>	IPv4 TCP/UDP	JUMPSERVERS net	*	172.16.10.10	AD_ports	*	*							
<input type="checkbox"/>	IPv4 TCP	JUMPSERVERS net	*	SERVERS net	3389 (MS RDP)	*	*							
<input type="checkbox"/>	IPv4 *	JUMPSERVERS net	*	*	*	*	*							

The reading of rules for **JUMPSERVERS** interface is like this,

- 1-Allow **TCP/UDP** connections from **JUMPSERVERS** network to 172.16.10.10 (**DC**) host's **AD** ports
- 2-Allow **TCP** connections from **JUMPSERVERS** network to **SERVERS** network's **RDP** ports
- 3-Block any connection from **JUMPSERVERS** network to any network



1.6.3 GUESTS rules

In our project we consider **GUESTS** network for unauthorized persons, so there is no need **GUESTS** to communicate with **SERVERS**, **JUMPSERVERS** and **CLIENTS** networks, it will be enough for them being able to connect to the **Internet** under our control. In order to realize it, first we have to write rules for each of this interfaces to block network traffic outgoing from **GUESTS** network, but instead of writing same rules separately for each interface, we can gather all of these 3 interfaces under one group and write just one rule for this group. Rules are read from left to right and applied from top to bottom. Group rules' priorities are higher than individual rules always. Lets create a group firstly.

Go to the **Firewall => Groups** and click the **Add** button to create a group.

Name	Members	Sequence	Description	Commands
openvpn	SERVERS, CLIENTS, JUMPSERVERS	10	all OpenVPN Interfaces	
enc0	SERVERS, CLIENTS, JUMPSERVERS	10	IPsec	
S_C_J				

In the “**Edit Group**” window, give a name for the group and select the interfaces.

Edit Group

full help

i Name	<input type="text" value="S_C_J"/>
i Members	<input type="text" value="SERVERS, CLIENTS, JUMPSERVERS"/>
i (no) GUI groups	<input type="checkbox"/>
i Sequence	<input type="text" value="0"/>
i Description	<input type="text" value="SERVERS, CLIENTS and JUMPSERVERS"/>

Cancel **Save**

Name	Members	Sequence	Description	Commands
openvpn	SERVERS, CLIENTS, JUMPSERVERS	10	all OpenVPN Interfaces	
enc0	SERVERS, CLIENTS, JUMPSERVERS	10	IPsec	
S_C_J	SERVERS, CLIENTS, JUMPSERVERS	0	SERVERS, CLIENTS and JUMPSERVERS	

After creating a group, go to the **Firewall => Rules => GUESTS** and click **Add** button.



Firewall: Rules: GUESTS

Select category

Protocol	Source	Port	Destination	Port	Gateway	Schedule		Description					
<i>Automatically generated rules</i>													
pass		block		reject		log		in		first match			
pass (disabled)		block (disabled)		reject (disabled)		log (disabled)		out		last match			

Inside the rule, select **Action**, select “**GUESTS net**” as **Source** and select the group name, “**S_C_J net**” as **Destination** which we just created above, then **Save and Apply**.

Firewall: Rules: GUESTS

Edit Firewall rule

Action	<input type="button" value="Block"/>
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	<input type="button" value="GUESTS"/>
Direction	<input type="button" value="in"/>
TCP/IP Version	<input type="button" value="IPv4"/>
Protocol	<input type="button" value="any"/>
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	<input type="button" value="GUESTS net"/>
Source	<input type="button" value="Advanced"/>
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	<input type="button" value="S_C_J net"/>

Firewall: Rules: GUESTS

Select category

The changes have been applied successfully.

Protocol	Source	Port	Destination	Port	Gateway	Schedule		Description					
<i>Automatically generated rules</i>													
	IPv4 *	<input type="button" value="GUESTS net"/>	*	<input type="button" value="S_C_J net"/>	*	*		in		first match			



After blocking outgoing traffic to **SERVERS**, **CLIENTS** and **JUMPSERVERS** networks, we need to write new rule to allow **GUESTS** interface to connect **Internet**.

Firewall: Rules: GUESTS

Edit Firewall rule	
Action	Pass
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	GUESTS
Direction	in
TCP/IP Version	IPv4
Protocol	any
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	any
Source	Advanced
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	any

Firewall: Rules: GUESTS									Select category	Inspect				
The changes have been applied successfully.														
	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Actions	Description					
	Automatically generated rules													
<input type="checkbox"/>	IPv4 *	GUESTS net	*	S_C_J net	*	*	*							
<input type="checkbox"/>	IPv4 *	*	*	*	*	*	*							

The reading of rules for **GUESTS** is like this,

- 1-Block any connection from **GUESTS** network to **S_C_J** group interfaces
- 2-Allow any connection from **GUESTS** network to any network

1.7 NAT configuration

In this capture we are going to block Internet access for **JUMPSERVERS** interface and allow all other interfaces. Default, firewall comes in **Automatic** mode which allow all interfaces to access to the internet over the **WAN** interface, but we have to choose **Hybrid** mode to control permissions manually ourselves. This is similar with firewall rules, and the logic behind the process is same.

Go to the **Firewall => NAT => Outbound** section and hit the add button to open new rule sheet.



In opened window, put tic in the “**Do not NAT**” box to disable **NAT** function and in the “**Source address**” choose **JUMPSERVERS** network. Without **NAT**, it is not possible to communicate with other devices on Internet because **NAT** allows us to map our **private IP** addresses with **public IP** addresses.



OPNsense plugins

2.1 Nginx

Interacting directly with the web server itself can be dangerous for the security reason, in order to prevent this and add an additional security layer mostly **RP** (Reverse Proxy) logic is used in practice. **RP** server stands between web servers and clients and forwards requests from clients to web servers. Another important security feature is **WAF** (Web Application Firewall) for web servers, which allow us to filter the **HTTP** requests between web applications and the **Internet**. The **OPNsense Nginx** plugin combines these two features in it.

First we need to download **Nginx** plugin, go to the **System => Firmware => Plugins**, type **nginx** and search for the plugin. Once you confirm that it is right plugin, hit install button as displayed in the figure.

	Version	Size	Tier	Repository	Comment
os-nginx	1.32.2	920KIB	3	OPNsense	Nginx HTTP server and reverse proxy

After completing installation go to the **Services => Nginx => Configuration** and enable the service.

In this section choose the “**Upstream Server**” and hit the add button.

Description	Server	Port	Priority	Comments
No results found!				



In the “Edit Upstream” window give a descriptive name to the server in the **Description** section, assign your web server’s **IPv4** address (172.16.10.11), **Port** number (our web server runs on port **80**) and the “**Server Priority**” value (if there are a few running servers we can define their priorities manually) which is **1** in this lab.

Edit Upstream

advanced mode

Description	DVWA server
Server	172.16.10.11
Port	80
Server Priority	1
Maximum Connections	
Maximum Failures	
Fail Timeout	

Later go to the **Upstream** section and add configuration as shown in the figure.

Services: Nginx: Configuration

General Settings ▾ HTTP(S) ▾ Data Streams ▾ **Upstream** ▾ Access ▾ Other ▾

Upstream Server

Upstream

Description	Servers	TLS Enabled	Commands
	No results found!		<input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

In popped window type new descriptive name for **Upstream** and choose the server we created just above in “**Server Entries**”.

Edit Upstream

advanced mode full help

Description	DVWA upstream
Server Entries	DVWA server
Load Balancing Algorithm	Weighted Round Robin
Enable TLS (HTTPS)	<input type="checkbox"/>
TLS: Servername override	
TLS: Supported Versions	Nothing selected
TLS: Session Reuse	<input checked="" type="checkbox"/>
TLS: Trusted Certificate	Nothing selected



In the next step we have to define the location for our **Upstream Server**.

Services: Nginx: Configuration

General Settings	HTTP(S)	Data Streams	Upstream	Access	Other	
Description	Location					<input type="button" value="Search"/> <input type="button" value="7"/>
	Credential					
	User List					
	HTTP Server					
	URL Rewriting					
	Naxsi WAF Policy					

No results found!

Add **+ C**

In the opened window add **Description** for our location, in the “**URL Patters**” section add the “/”, and last in the “**Upstream Servers**” section choose the **Upstream** which we created above.

Edit Location

advanced mode

Description: DVWA location

URL Pattern: /

Match Type: None

URL Rewriting: Nothing selected

Enable Security Rules:

Learning Mode:

Block XSS Score:

Block SQL Injection Score:

Custom Security Policy: Nothing selected

Upstream Servers: DVWA upstream

Path Prefix:

Cache: Directory: None

Back to the configuration page and add new configuration in **HTTP(S) => HTTP Server**.

Services: Nginx: Configuration

General Settings	HTTP(S)	Data Streams	Upstream	Access	Other	
Servername	Location					<input type="button" value="Search"/> <input type="button" value="7"/>
	Credential					
	User List					
	HTTP Server					
	URL Rewriting					
	Naxsi WAF Policy					

No results found!

Add **+ C**



In the “HTTP Listen Address” type the web application port number (it is **80** in our lab) and in the “Server Name” section type your domain name for proxy (we have to add new **DNS** record for this domain name in **DC** server).

Later back to the “General Settings” in configuration and press **Apply** button to save configuration changes.

Now we are ready for **WAF** configuration, firstly go to the **HTTP(S) => Naxsi WAF Policy** and press **Download** button to download policies for **WAF**.

When downloading is finished go back to the **HTTP(S) => Location** and edit the configuration we created above.



Services: Nginx: Configuration

Description	URL Path Prefix	Match Type	WAF Status	Force HTTPS
DVWA location	/	None	disabled	<input checked="" type="checkbox"/>

In the “**Costume Security Policy**” section you will see the all downloaded policies for **WAF**, choose whichever you want to apply your **WAF**.

Edit Location

(advanced mode)

Description	DVWA location
URL Pattern	/
Match Type	None
URL Rewriting	Nothing selected
<input type="button" value="Clear All"/>	
Enable Security Rules	<input checked="" type="checkbox"/>
Learning Mode	<input type="checkbox"/>
Block XSS Score	
Block SQL Injection Score	
Custom Security Policy	Cross Site Scripting IDs:1300-1399, Directory traversal IDs:1200-1299
Upstream Servers	<ul style="list-style-type: none"> Cross Site Scripting IDs:1300-1399 ✓ Directory traversal IDs:1200-1299 ✓ Evading tricks IDs: 1400-1500 ✓ File uploads: 1500-1600 ✓ OBVIOUS RFI IDs:1100-1199 ✓ SQL Injections IDs:1000-1099 ✓
Path Prefix	
Cache: Directory	
File System Root	

We successfully installed **Nginx** plugin for **OPNsense** firewall, now we can test it. As you remember we installed **DVWA** (Damn Vulnerable Web Application) in our **Kali Linux** machine and add this machines IP address in our **Nginx** configuration (in **HTTP(S) => HTTP server => Server** section) for redirecting requests and generated a **DNS** record for our proxy. Open your browser, go to the <http://dvwa.cs301.local> address and login with default credentials (username: **admin** | password: **password**) for **DVWA**. After login to the application select any vulnerable section, for better understanding the **WAF** logic I choose **XSS (Reflected)**



section and typed the piece of script, '`<script>alert("hi")</script>`', and submit, which should pop up a window says "hi" if our WAF doesn't block the request.

The top screenshot shows a browser window with the address bar showing 'Login : Damn Vulnerable Web App' and 'Not secure | dvwa.cs301.local/login.php'. The DVWA logo is at the top. Below it is a login form with 'Username' set to 'admin' and 'Password' set to 'password'. The bottom screenshot shows a browser window with the address bar showing 'Vulnerability: Reflected Cross Site Scripting (XSS)' and 'Not secure | dvwa.cs301.local/vulnerabilities/xss_r/'. The DVWA logo is at the top. Below it is a form with the question 'What's your name?' and a text input field containing the script '<script>alert("hi")</script>'. To the left of the form is a sidebar with various vulnerability options, and 'XSS (Reflected)' is highlighted.

But when I pressed the submit button firewall blocked my request and displayed the default error page. This means that our firewall works properly and we can modify it by adding new strict policies to make it much more secure.

The screenshot shows a browser window with the address bar showing 'Request Denied' and 'Not secure | dvwa.cs301.local/vulnerabilities/xss_r/?name=<script>alert%28"hi"%...'. The main content is a large red banner with the text 'Request Denied For Security Reasons'. Below the banner, there is a message: 'The request has been denied by the web application firewall due to a security policy violation.' and 'Request information have been logged to investigate the incident.' At the bottom of the page, it says 'If you think this is an error on our side, please contact us.' and 'Web Application Protection by OPNsense'.



2.2 Squid Web Proxy plugin

The **Squid** is a forward caching (transparent) proxy plugin available in **OPNsense** which allows us to control web requests by apply whitelist-blacklist logic. We can block unwanted web traffic with the help of this web filtering feature.

Before setup the **Squid** plugin, we have to create a **self-signed certificate** for allowing **HTTPS** traffic to flow over our proxy service. To generate a new certificate, go to the **System => Trust => Authorities** and hit the add button on the right corner of the page.

Name	Internal	Issuer	Certificates	Distinguished Name

In opened window, give a “**Descriptive name**” for your certificate, choose the **Method** as “**Create an internal Certificate Authority**” and define the “**Country Code**” where you are located, leave other parameters default and **Save**.

Descriptive name: CS301-CA

Method: Create an internal Certificate Authority

Internal Certificate Authority:

Key Type: RSA

Key length (bits): 2048

Digest Algorithm: SHA256

Lifetime (days): 825

Distinguished name:

Country Code: AZ (Azerbaijan)

Now download the certificate into your machine as shown in the figure below and copy it to your machine



whichever you want to connect to the **Internet** over the **Squid Web Proxy**. (I copied it to my **Windows 10** client machine in **VMware**, because I applied web filtering only to my **CLIENTS** interface)

System: Trust: Authorities

Name	Internal	Issuer	Certificates	Distinguished Name
CS301-CA	YES	self-signed	0	CN=cs301-ca, C=AZ Valid From: Mon, 25 Mar 2024 18:45:57 +0400 Valid Until: Sun, 28 Jun 2028 18:45:57 +0400

export CA cert

Double click on the certificate and install it to your machine where you want to use web filtering feature. Below, the figures demonstrate whole installation process step by step, just follow them if you are not familiar with certificates.

NOTE: we couldn't use the **HTTPS** websites without this certificate if we want to apply web filtering feature to our machine, because in following steps we will block default port (**443**) for **HTTPS** traffic in order to prevent proxy bypassing, so don't forget to install.

Now we are ready to install the Squid proxy, as other plugins, also the Squid have to be installed from



System => Firmware => Plugins section.

Status	Settings	Changelog	Updates	Plugins	Packages	Version	Size	Tier	Repository	Comment
				squid		1.0	293KiB	2	OPNsense	Squid is a caching proxy for the web
				os-squid						

After installing the **Squid** plugin, we can find it in the **Services** section. Here enter to the **Administration** settings of **Squid** service and fill the required boxes as displayed in the figure below.

Firstly choose the interface you want to apply web filtering (as I choose **CLIENTS** interface), put ticks in “**Enable Transparent HTTP proxy**” and “**Enable SSL inspection**”, and choose the certificate we created above for “**CA to use**”.

After defining proxy ports and enabling proxy methods (for HTTP or HTTPS traffic), in this step we can apply checklist to our proxy for web filtering, go to the **Services => Squid Web Proxy => Administration => Remote Access Control Lists** section and add new blacklist by pressing add (**plus**) button at the right of page.

Enabled	Filename	URL	Description	Edit Delete
No results found!				



In the configuration page enable it, define the **Filename** whatever you want, paste the source address for blacklist into the **URL** section (feel free to use other source, although I used ftp://ftp.ut-capitole.fr/pub/reseau/cache/squidguard_contrib/blacklists.tar.gz) and type the **Description** for it. The another important section is the “**categories (if available)**” in this configuration. But it will be available when we **Save** this current configuration and download in the next step.

Edit blacklist

full help	
i enabled	<input checked="" type="checkbox"/>
i Filename	UT1
i URL	ftp://ftp.ut-capitole.fr/pub/reseau/cache/squidguard...
i username (optional)	
i password (optional)	
i categories (if available)	Nothing selected
Clear All	
i ssl ignore cert	<input type="checkbox"/>
i Description	Université Toulouse 1 Capitole blacklist
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

Once we **Save** the configuration, it will appear in **Services => Squid Web Proxy => Administration => Remote Access Control Lists**. Now we have to download this blacklist from the source **URL**, for this press the “**Download ACLs & Apply**” button and wait for a few minutes until it is downloaded into your firewall.

Services: Squid Web Proxy: Administration

General Proxy Settings	Forward Proxy	Proxy Auto-Config	Remote Access Control Lists	Support	full help										
i Remote Blacklist															
<table border="1"> <thead> <tr> <th>Enabled</th> <th>Filename</th> <th>URL</th> <th>Description</th> <th>Edit Delete</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/></td> <td>UT1</td> <td>ftp://ftp.ut-capitole.fr/pub/reseau/cache/squidguard_contrib/blacklists.tar.gz</td> <td>black list</td> <td> </td> </tr> </tbody> </table>					Enabled	Filename	URL	Description	Edit Delete	<input checked="" type="checkbox"/>	UT1	ftp://ftp.ut-capitole.fr/pub/reseau/cache/squidguard_contrib/blacklists.tar.gz	black list	 	+ Add
Enabled	Filename	URL	Description	Edit Delete											
<input checked="" type="checkbox"/>	UT1	ftp://ftp.ut-capitole.fr/pub/reseau/cache/squidguard_contrib/blacklists.tar.gz	black list	 											
 1 					Showing 1 to 1 of 1 entries										
<input type="button" value="Apply"/> <input type="button" value="Download ACLs & Apply"/> <input type="button" value="Download ACLs"/> <input type="button" value="Schedule with Cron"/>															

When the downloading and applying are finished, edit the configuration we created above. Now we can see that the “**categories (if available)**” section is available with multiple choices. This kind of blacklist is called category based where we are free to use whatever category we want for blocking in our selected machine. In this lab I leaved this section as default to block all included categories.



Edit blacklist

full help [CI](#)

enabled	<input checked="" type="checkbox"/>
filename	UT1
URL	ftp://ftp.ut-capitole.fr/pub/reseau/cache/squidguard...
username (optional)	
password (optional)	
categories (if available)	hacking, malware, phishing, social_networks
Clear All	
ssl ignore cert	<input type="checkbox"/>
Description	Université Toulouse 1 Capitole blacklist

[Cancel](#) [Save](#)

After choosing categories, go to the **Services => Squid Web Proxy => Administration => Forward Proxy => Access Control List** settings and activate the “**advanced mode**” on the left corner of configuration window. Then add our web application domain address (**dvwa.cs301.local**) into the Whitelist and type **HTTPS port (443)** number into “**Allowed destination TCP port**” and “**Allowed SSL ports**” sections.

Services: Squid Web Proxy: Administration

General Proxy Settings [▼](#) Forward Proxy [▼](#) Proxy Auto-Config [▼](#) Remote Access Control Lists Support

advanced mode

Allowed Subnets		Clear All Copy Text
Unrestricted IP addresses		Clear All Copy Text
Banned host IP addresses		Clear All Copy Text
Whitelist	dvwa.cs301.local	Clear All Copy Text
Blacklist		Clear All Copy Text
Block browser/user-agents		Clear All Copy Text
Block specific MIME type reply		Clear All Copy Text
Google GSuite restricted		
YouTube Filter	None	
Allowed destination TCP port	443	Clear All Copy Text
Allowed SSL ports	443	

After finishing the whole configuration in **Squid** service, go to the **Services => Squid Web Proxy => Administration => General Proxy Settings** enable the service, choose **OPNsense** in the “**User error pages**” for displaying default error page to the client when any action is blocked by firewall (**Squid** service) because of web filtering and **Apply** changes.



- Reporting
- System
- Interfaces
- Firewall
- VPN
- Services
 - Captive Portal
 - Dnsmasq DNS
 - Intrusion Detection

Services: Squid Web Proxy: Administration

General Proxy Settings	Forward Proxy	Proxy Auto-Config	Remote Access Control Lists	Support
<input type="checkbox"/> advanced mode				
<input checked="" type="checkbox"/> Enable proxy				
<input type="checkbox"/> User error pages				
OPNsense				
<input type="button" value="Apply"/>				

As you remember we choose two interfaces, **CLIENTS** and **SERVERS**, for web filtering, now we have to write a few separate rules for both networks. In a short way, we can add these both interfaces into the one group and write common rules for that group.

First we need to create a group, go to the **Firewall => Groups** section and add new group.

Firewall: Groups

<input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>				
Name	Members	Sequence	Description	Commands
<input type="checkbox"/> S_C_J	SERVERS,CLIENTS,JUMPSERVERS	0	SERVERS, CLIENTS and JUMPSERVERS	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
<input type="checkbox"/> openvpn		10	all OpenVPN interfaces	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
<input type="checkbox"/> enc0		10	IPsec	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Enter the name for group and select the interfaces in the “**Edit Group**” page.

Edit Group

<small>full help Q</small>	
<input type="checkbox"/> Name	<input type="text" value="SERVERS_CLIENTS"/>
<input type="checkbox"/> Members	<input type="text" value="SERVERS, CLIENTS"/>
<input type="checkbox"/> (no) GUI groups	<input type="checkbox"/>
<input type="checkbox"/> Sequence	<input type="text" value="0"/>
<input type="checkbox"/> Description	<input type="text"/>
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

Firewall: Groups

<input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>				
Name	Members	Sequence	Description	Commands
<input type="checkbox"/> S_C_J	SERVERS,CLIENTS,JUMPSERVERS	0	SERVERS, CLIENTS and JUMPSERVERS	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
<input type="checkbox"/> SERVERS_CLIENTS	SERVERS,CLIENTS	0		<input type="button" value="Edit"/> <input type="button" value="Delete"/>
<input type="checkbox"/> openvpn		10	all OpenVPN interfaces	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
<input type="checkbox"/> enc0		10	IPsec	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

That was the basic configuration of the **Squid Web Proxy** service which will serve us as a web filer. But as you remember, in the first step, we defined two ports (**3128** for **HTTP** and **3129** for **HTTPS**) to flow traffic over our proxy service, and now we have to forward the port **80** to **3128** for **HTTP** requests and port **443** to **3129** for **HTTPS** requests. In order to doing it we have to write two new rules in firewall **NAT** configuration,



one for **HTTP** and the other one for **HTTPS**.

Go to the **Firewall => NAT => Port Forward** section and here, choose the **Interface** which you want to apply port forwarding (**SERVERS_CLIENTS** group for this lab, because I enabled web filtering only for this interfaces), define **Source** (again **SERVERS_CLIENTS** group for this lab), “**Destination port range**” as **HTTP**, “**Redirect target IP**” choose the option “**Single host or Network**” and type your firewall’s IP address (or just loopback address as shown in the figure), “**Redirect target port**” choose the “**(other)**” option and enter the port address for **HTTP** which we defined in the earliest step of the **Squid** configuration and enter the redirection port **3128**.

Firewall: NAT: Port Forward

Edit Redirect entry		
<input checked="" type="checkbox"/> Disabled	<input type="checkbox"/> Disable this rule	
<input checked="" type="checkbox"/> No RDR (NOT)	<input type="checkbox"/>	
<input checked="" type="checkbox"/> Interface	SERVERS_CLIENTS	
<input checked="" type="checkbox"/> TCP/IP Version	IPv4	
<input checked="" type="checkbox"/> Protocol	TCP/UDP	
<input checked="" type="checkbox"/> Source / Invert	<input type="checkbox"/>	
<input checked="" type="checkbox"/> Source	SERVERS_CLIENTS net	
<input checked="" type="checkbox"/> Source port range	from: any	to: any
<input checked="" type="checkbox"/> Destination / Invert	<input type="checkbox"/>	
<input checked="" type="checkbox"/> Destination	any	
<input checked="" type="checkbox"/> Destination port range	from: HTTP	to: HTTP
<input checked="" type="checkbox"/> Redirect target IP	Single host or Network 127.0.0.1	
<input checked="" type="checkbox"/> Redirect target port	(other) 3128	

Firewall: NAT: Port Forward

The changes have been applied successfully.

	Source	Destination	NAT											
□	Interface	Proto	Address	Ports	Address	Ports	IP	Ports	Description					
	SERVERS	TCP	*	*	SERVERS address	3022,30443	*	*	Anti-Lockout Rule					
	SERVERS_CLIENTS	TCP/UDP	SERVERS_CLIENTS net	*	*	80 (HTTP)	127.0.0.1	3128						

In the second rule we have to follow all steps we did above, but the only difference will be the “**Destination port range**” (**HTTPS** instead of **HTTP**) and the “**Redirect target port**” value (**3129** instead of **3128**).



Firewall: NAT: Port Forward

Edit Redirect entry

<input checked="" type="checkbox"/> Disabled	<input type="checkbox"/> Disable this rule	
<input checked="" type="checkbox"/> No RDR (NOT)	<input type="checkbox"/>	
<input checked="" type="checkbox"/> Interface	SERVERS_CLIENTS	
<input checked="" type="checkbox"/> TCP/IP Version	IPv4	
<input checked="" type="checkbox"/> Protocol	TCP/UDP	
<input checked="" type="checkbox"/> Source / Invert	<input type="checkbox"/>	
<input checked="" type="checkbox"/> Source	SERVERS_CLIENTS net	
<input checked="" type="checkbox"/> Source port range	from: any	to: any
<input checked="" type="checkbox"/> Destination / Invert	<input type="checkbox"/>	
<input checked="" type="checkbox"/> Destination	any	
<input checked="" type="checkbox"/> Destination port range	from: HTTPS	to: HTTPS
<input checked="" type="checkbox"/> Redirect target IP	Single host or Network 127.0.0.1	
<input checked="" type="checkbox"/> Redirect target port	(other) 3129	

Both rules can be seen in following figure.

Firewall: NAT: Port Forward

The changes have been applied successfully.

	Source	Destination		NAT				
	Interface	Proto	Address	Ports	IP	Ports	Description	
<input checked="" type="checkbox"/>	SERVERS	TCP	*	*	SERVERS address	3022,30443	*	Anti-Lockout Rule
<input type="checkbox"/>	SERVERS_CLIENTS	TCP/UDP	SERVERS_CLIENTS net	*	80 (HTTP)	127.0.0.1	3128	
<input type="checkbox"/>	SERVERS_CLIENTS	TCP/UDP	SERVERS_CLIENTS net	*	443 (HTTPS)	127.0.0.1	3129	

In the final step we need to block port **80** and **443** for web requests, otherwise the whole traffic will pass by our proxy and no web filtering will be applied. For this purpose, we have to write two rules in our firewall, one for blocking **HTTP** requests on port **80**, and the other one for blocking **HTTPS** requests on port **443**.

Go to the **Firewall => Rules** section in firewall and choose the interface (**SERVERS_CLIENTS** for this lab) which you want to add a new rule. In the content of the rule choose the **Action Block**, **Interface** whatever you want (I choose **SERVERS_CLIENTS**), **Protocol TCP/UDP**, **Source the interface or specific IP** (I choose whole **SERVERS_CLIENTS** interfaces), **“Destination port range” from **HTTP** (**HTTPS** in second rule) to **HTTP** (**HTTPS** in second rule)**, **Category “Block proxy bypass”** and **Description** whatever you want.

Firewall: Rules: SERVERS_CLIENTS

Select category

	Protocol	Source	Port	Destination	Port	Gateway	Schedule		Description	
<small>Automatically generated rules</small>										
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	127.0.0.1	3128	*	*			
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	127.0.0.1	3129	*	*			



Firewall: Rules: SERVERS_CLIENTS

Edit Firewall rule

Action	Block
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	SERVERS_CLIENTS
Direction	in
TCP/IP Version	IPv4
Protocol	TCP/UDP
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	SERVERS_CLIENTS net
Source	Advanced
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	any
Destination port range	from: HTTP to: HTTP

Firewall: Rules: SERVERS_CLIENTS

The changes have been applied successfully.

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description	Actions
<i>Automatically generated rules</i>									
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	127.0.0.1	3128	*	*		
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	127.0.0.1	3129	*	*		
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	*	80 (HTTP)	*	*		

Firewall: Rules: SERVERS_CLIENTS

Edit Firewall rule

Action	Block
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	SERVERS_CLIENTS
Direction	in
TCP/IP Version	IPv4
Protocol	TCP/UDP
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	SERVERS_CLIENTS net
Source	Advanced
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	any
Destination port range	from: HTTPS to: HTTPS



Firewall: Rules: SERVERS_CLIENTS

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description	Actions
<i>Automatically generated rules</i>									
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	127.0.0.1	3128	*	*		
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	127.0.0.1	3129	*	*		
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	*	80 (HTTP)	*	*		
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	*	443 (HTTPS)	*	*		

All rules (2 port forwarding rules and 2 port blocking rules) are displayed in the figure below.

Firewall: Rules: SERVERS_CLIENTS

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description	Actions
<i>Automatically generated rules</i>									
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	127.0.0.1	3128	*	*		
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	127.0.0.1	3129	*	*		
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	*	80 (HTTP)	*	*		
<input type="checkbox"/>	IPv4 TCP/UDP	SERVERS_CLIENTS net	*	*	443 (HTTPS)	*	*		

It is time to test, go to the client machine, open browser and try to go somewhere is blocked. In the figure I tried to go <https://www.instagram.com> and <https://www.tap.az> web sites, but my first request was blocked by firewall because **Instagram** is a social media platform and we had blocked social media platforms in **Services => Squid Web Proxy => Administration => Remote Access Control Lists => categories** section, and the second one is allowed because it is a shopping platform.

The following error was encountered while trying to retrieve the URL:
<https://www.instagram.com/>

Access Denied.

Access control configuration prevents your request from being allowed at this time. Please contact your service provider if you feel this is incorrect.

Your cache administrator is [webmaster](#).

Generated Mon, 25 Mar 2024 16:38:11 GMT by Firewall.cs301.local (squid/6.8)



2.3 ClamAV and C-ICAP plugins

In the previous capture we tested web filtering feature in **Squid Web Proxy** plugin, which blocked our unwanted **HTTP** and **HTTPS** requests, but it doesn't mean that we are protected from harmful actions such as downloading malicious files (ransomwares, trojans, viruses etc.) because with a simple web proxy configuration we can't scan the content of traffic. In this point we need to add **ClamAV** and **C-ICAP** plugins' features to our web proxy.

ClamAV is an open source antivirus software and the **OPNsense** is able to use it as a plugin. We need an additional protocol as a glue between **Squid Web Proxy** and **ClamAV** which is called **ICAP** and **OPNsense** has its plugin, **C-ICAP**, too.

First, download **ClamAV** plugin, go to the **System => Firmware => Plugins** section, in the search bar type “**os-clamav**” and install the result.

Version	Size	Tier	Repository	Comment
1.8	47.7KiB	3	OPNsense	Antivirus engine for detecting malicious threats

Open **Services => ClamAV => Configuration**, enable “**Enable clamd service**”, “**Enable freshclam service**” and “**Enable TCP port**” options and leave rest of them as default. At the right corner of the page, click the “**Download signatures**” button and **Save** changes. Our antivirus plugin will work with the help of these signatures.

General	Signatures	Versions
<input checked="" type="checkbox"/> Enable clamd service <input checked="" type="checkbox"/> Enable freshclam service <input checked="" type="checkbox"/> Enable TCP port <input type="text"/> Maximum number of threads running <input type="text"/> Maximum number of queued items		

Once the **ClamAV** plugin setup finished, install the **C-ICAP** plugin with the “**os-c-icap**” keyword.

Version	Size	Tier	Repository	Comment
1.7_4	50.2KiB	3	OPNsense	c-icap connects the web proxy with a virus scanner

After the installation, **C-ICAP** plugin will be located in **Services** section. Go to the **Services => C-ICAP => Configuration => General** section and configure parameters like displayed in the figure.



Later open the **Services => C-ICAP => Configuration => Antivirus** section and enable the “**Enable ClamAV**” option and leave other options default. The most interesting part of this configuration is “**Scan of filetypes**” option which lets us to control whatever we want to scan in our antivirus plugin.

Now we need to enable **ICAP** option in our web proxy, go to the **Service => Squid Web Proxy => Administration => Forward Proxy => ICAP Settings** and enable the “**Enable ICAP**” option and type “**icap://[:1]:1344/avscan**” to both “**Request Modify URL**” and “**Response Modify URL**” sections.



Services: Squid Web Proxy: Administration

General Proxy Settings ▾ **Forward Proxy** ▾ **Proxy Auto-Config** ▾ **Remote Access Control Lists** **Support**

advanced mode

i Enable ICAP

i Request Modify URL

i Response Modify URL

i Exclusion List

At the end of the whole installations and configurations, we must reboot the system for applying configurations, go to the **Power => Reboot** and wait for rebooting firewall.



After the reboot, our firewall is ready for caching malicious traffic with antivirus plugin. In order to test if it works, go to the machine where we installed our certificate and open the browser. In the search bar type the <https://eicar.eu> and open the web page. You will meet a few anti malware test files what we will try to download.

After the reboot, our firewall is ready for caching malicious traffic with antivirus plugin. In order to test if it works, go to the machine where we installed our certificate and open the browser. In the search bar type the <https://eicar.eu> and open the web page. You will meet a few anti malware test files what we will try to download.

Download area using the standard protocol: HTTP

eicar.com	eicar.com.txt	eicar_com.zip	eicarcom2.zip
68 Bytes	68 Bytes	184 Bytes	308 Bytes (nested ZIP)

Download area using the secure, SSL enabled protocol : HTTPS

eicar.com	eicar.com.txt	eicar_com.zip	eicarcom2.zip
68 Bytes	68 Bytes	184 Bytes	308 Bytes (nested ZIP)

Additional notes:

1. This file used to be named ducklin.htm or ducklin-html.htm or similar based on its original author Paul Ducklin and was made in cooperation with CARO.
2. The definition of the file has been refined 1 May 2003 by Eddy Willems in cooperation with all vendors.
3. The content of this documentation (title-only) was adapted 1 September 2006 to add verification of the activity of anti-malware or anti-spyware products. It was decided not to change the file itself for backward-compatibility reasons.

Provided by [mars solutions GmbH](#)



Whenever you click on the download link in the page it will be blocked by our antivirus and will display the [OPNsense](#) default warning page to inform you about the accident.

After the reboot, our firewall is ready for caching malicious traffic with antivirus plugin. In order to test if it works, go to the machine where we installed our certificate and open the browser. In the search bar type the <https://eicar.eu/eicar.com.txt> and open the web page. You will meet a few anti malware test files what we will try to download.

MALWARE FOUND

You tried to upload/download a file that contains the malware: Win.Test.EICAR_HDB-1
The HTTP location is: <https://eicar.eu/eicar.com.txt>

For more information contact your system administrator

Web Protection by OPNsense
Antivirus engine: clamd-130/27225
This message generated by C-ICAP service: avscan?allow204=on&mode=simple





2.4 Zenarmor plugin

Like other open source firewalls, the **OPNsense** works in **L4**, which means in a traditional mode we can't control the **Application** layer. In this point, the **Zenarmor** plugin helps us to gain some next generation firewall features such as **Application Control**, **TLS Inspection**, **Network Analytics** and more.

Before we download **Zenarmor** packages, we have to install its repository what is called "**os-sunnyvalley**".

System: Firmware					
Status Settings Changelog Updates Plugins Packages					
	Version	Size	Tier	Repository	Comment
os-sunnyvalley	1.4_3	2.39KiB	3	OPNsense	Vendor Repository for Zenarmor (a.k.a Sensei, Next Generation Firewall Extensions)

After downloading the repository, we can install the three packages, "**os-sensei**" (**Zenarmor** plugin) and "**os-sensei-update**" (updater for **Zenarmor**).

System: Firmware					
Status Settings Changelog Updates Plugins Packages					
	Version	Size	Tier	Repository	Comment
os-sensei	1.16.4	228MiB	3	SunnyValley	Next Generation Firewall Extensions for OPNsense (ZENARMOR)
os-sensei-agent	1.16.6	115MiB	3	SunnyValley	ZENARMOR (Sensei) Connectivity Agent for Cloud Central Management
os-sensei-updater	1.16	4.11KiB	3	SunnyValley	OPNsense ZENARMOR Plugin Updater

Once we finish the installation of packages, the **Zenarmor** will be located in settings column with **Zenarmor** name. Apart from other plugins we installed in previous captures, this one has its own environment to work on it, that is why it is not in the Services.

When we click on the **Zenarmor => Dashboard**, the installation page will meet us. The installation process is very simple, just follow the directives step by step.

1-Accept the **Terms of Services**, **Privacy Policies** and go to next.

The screenshot shows the initial configuration screen for the Zenarmor plugin. The left sidebar lists various system components like System, Interfaces, Firewall, VPN, Services, and Zenarmor. The Zenarmor section is expanded, showing sub-options: Dashboard, Reports, Live Sessions, Policies, Settings, and Notifications. The main panel displays a five-step setup wizard. Step 1 is 'Welcome', which includes the Zenarmor logo and a message: "You're ready to do the initial configuration." A checkbox labeled "Check here to indicate that you have read and agree to the Terms of Service and Privacy Policy." is checked. At the bottom are "Uninstall" and "I Agree" buttons.

2-Cooche the "**Install a local Mongodb Database**" and **Install Database**. After the installation is finished click **Next**.



3-Choose “Deployment mode” as “Routed Mode (L3 Mode, Reporting + Blocking) with emulated netmap driver”, define the interfaces which you want to control and monitor via the Zenarmor and set the security zone for each interface (I choose only GUESTS interface because we already have protection on others).

4-Choose the “Get me a 15-day Free Trial of Business Subscription(*)” and type your email for subscribing and click Next button.

It will send an activation email for creating new account, open it and click on the “Activate Account”.

In the new page choose the **Business** or **Home** (it doesn't matter) version and fill the boxes. When you sign up successfully, it will give you an activation key, copy it.



Just one more step.

Select the Edition for your Free Trial

*If you would like to explore all of Zenarmor's features during your trial, please select Business.

Home
Business

First name: Yagub

Last name: Hajiyev

Company name: CS301

[Start Your Free 15-day Business Subscription Trial](#)

[Compare Plans](#)

Welcome to your Zenconsole!

To start your 15-day Business Subscription Trial, simply install the activation key below on your Zenarmor instance

Activation key: **3f9399c7-863a-43ce-bcd9-46bda723e60e** [copy](#)

Paste your activation key and go on the next step.

- [Lobby](#)
- [Reporting](#)
- [System](#)
- [Interfaces](#)
- [Firewall](#)
- [VPN](#)
- [Services](#)
- [Zenarmor](#)
 - [Dashboard](#)
 - [Reports](#)
 - [Live Sessions](#)
 - [Policies](#)
 - [Settings](#)
 - [Notifications](#)
- [Power](#)
- [Help](#)

[Want more power?](#) Zenconsole is the cloud central manager for Zenarmor. [Connect to Zenconsole](#)

1 2 3 4 5

Welcome Database Settings Interface Settings Subscription Finish

Let's make the most of your Zenarmor!
Start a 15-day Free Trial of Business Subscription to experience zenarmor with all of its benefits!
Or, if you have already purchased a subscription, enter the activation key to start protecting your network!

[Compare Plans & Pricing](#)

Get me a 15-day Free Trial of Business Subscription(*)
 I already have my subscription key
 Get me the Free Edition

Advanced Security, Unlimited Policies & Cloud Management, Active Directory Integration and many more...
Enter your activation key

Email Address:

Activation Key:

[Previous](#) [Next](#)

5-Complete the installation and our Zenarmor is ready to use.

- [System](#)
- [Interfaces](#)
- [Firewall](#)
- [VPN](#)
- [Services](#)
- [Zenarmor](#)
 - [Dashboard](#)
 - [Reports](#)
 - [Live Sessions](#)
 - [Policies](#)
 - [Settings](#)
 - [Notifications](#)
- [Power](#)

1 2 3 4 5

Welcome Database Settings Interface Settings Subscription Finish

Almost done.

Your installation of zenarmor is almost complete.
Click 'Complete' to write the final configuration to disk and start zenarmor

[Previous](#) [Complete](#)

Now our plugin is ready to use, go to the **Zenarmor => Policies** section and click on the "**Create New Policy**" button on the right corner of the page and in the popped up window, type the name you want to give to your policy.

- [Lobby](#)
- [Reporting](#)
- [System](#)
- [Interfaces](#)
- [Firewall](#)
- [VPN](#)
- [Services](#)
- [Zenarmor \(Business\)](#)
 - [Dashboard](#)
 - [Devices](#)
 - [Reports](#)
 - [Live Sessions](#)
 - [Activity Explorer](#)
 - [Policies](#)
 - [Settings](#)

Policies

Default

Did you know?
Please note that the 'AND' logical operator is used in the policy conditions. So, for a particular type of traffic to match a specific policy, it must pass through all the conditions defined in the policy. For instance, if you've created a policy and specified two conditions: 'Device MAC address' and 'Port 80', both conditions must be met for the traffic to be matched by the policy.

[Create New Policy](#)

Create policy

Policy name: GUESTS!.policy

[Cancel](#) [Create](#)



After creating the policy click on the policy name for editing it. This will open the editing page, from the **Interfaces** section choose the interface (**em4**) to apply your policy and enable the **Status**.

Policies

- GUESTS_policy (Status: Enabled)
- Default (Status: Enabled)

GUESTS_policy

- Configuration**: Status is Enabled.
- Security**: Name is GUESTS_policy. No Internet access is blocked (overrides all configured policy rules). Block Untrusted Devices is disabled.
- App Controls**: Interface em4 is selected (Inbound and Outbound).
- Web Controls**: VLAN IDs are listed.
- Exclusions**: None.

Later open the **Security** options page and choose the security policies for selected interface input and output traffic and “**Apply Changes**”. In our lab we will test a few features of **Zenarmor** such as malware blocking, web filtering and creating whitelist.

Malware blocking: go to the **Security** section and enable the **Malware/Virus** option in your policy. Later open any machine in **GUESTS** network and search for www.eicar.eu website. In this site try to open the eicar.com.txt file and you will see it.

Policies

- GUESTS_policy (Status: Enabled)
- Default (Status: Enabled)

GUESTS_policy

- Configuration**: Status is Enabled.
- Security**: Category name is DNS over HTTPS (Allowed), Malware/Virus is Blocked, and Phishing is Allowed.
- App Controls**: None.
- Web Controls**: None.
- Exclusions**: None.

Browsers

- Left: Download Anti Malware Testfile - EICAR (HTTP) - Shows download links for various files.
- Right: eicar.eu (HTTPS) - Shows a cloud icon and the message "Hmmm... can't reach this page". Below it says "It looks like eicar.eu closed the connection".

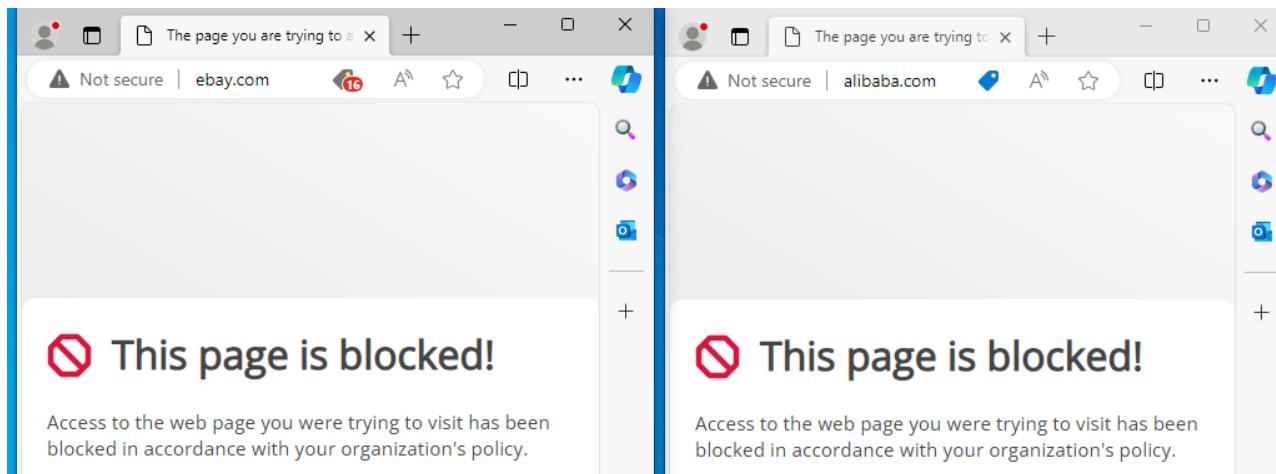
Web filtering: again go to the “**Web Controls**” in the policy and enable the **Shopping** option. In **GUESTS** interface, try to open any shopping site.

Policies

- GUESTS_policy (Status: Enabled)
- Default (Status: Enabled)

GUESTS_policy

- Configuration**: Status is Enabled.
- Security**: None.
- App Controls**: None.
- Web Controls**: Shareware and Freeware is Allowed, Shopping is Blocked, and Social Networks is Allowed.
- Exclusions**: None.



Whitelist: we saw that all shopping sites are blocked by Zenarmor plugin, now we will add [alibaba.com](http://www.alibaba.com) to the whitelist. Open your policy and in the **Exclusions** section, add the domain you want to add into the whitelist (or black list, you can change the mode in **List** option) and click on the “**Add Exclusion**” button and apply changes. When you visit www.alibaba.com address, firewall will let your browser to load page.

Policy	Status	Action
GUESTS_policy	On	<input type="button" value="Edit"/>
Default	On	<input type="button" value="Edit"/>

Did you know?

Please note that the 'AND' logical operator is used to evaluate all policy criteria in Policy Configuration, not the 'OR' logical operator.

So, for a particular type of traffic to match a specific policy, all criteria must be met. (The only exception to this rule are the Devices and MAC addresses, where they can be used interchangeably).

For instance, if you've created a policy and specified VLAN ID, IP, and username criteria, A session must match all of those.

[More Information](#)

GUESTS_policy

- Configuration
- Security
- App Controls
- Web Controls
- Exclusions**

Add New Exclusions

Host/IP/Network Addr: alibaba.com

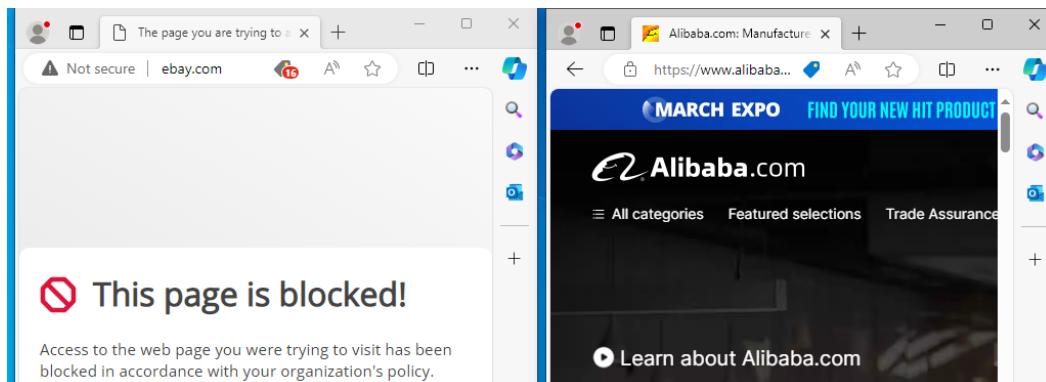
List: White list

Description:

Global:

Feedback: Send this to Zenarmor to improve web categorization.

[+ Add Exclusion](#)





OPNsense additional features

3.1 GeoIP blocking

One of the most essential feature of firewalls' is blocking IPs in whole geographical region such as city, country. In **OPNsense** to enable **GeoIP** blocking function, go to the **Firewall => Aliases => GeoIP settings**. In the **URL** section we need to add the source to download the IP ranges for regions. There numerous sources available on the **Internet** and we will use **Maxmind** platform in this lab.

Go to the <https://www.maxmind.com> address and create an account, next enter to your profile and generate new license key.

The screenshot shows the Maxmind website's 'License Keys' section. On the left, a sidebar menu includes 'Account Summary', 'Account Information', 'Manage License Keys' (which is selected), 'Manage Account Services', 'Manage Users', 'Account Activity', 'Edit My Info', and 'Sign-In Security'. Below the sidebar is a 'Billing' section. The main content area has a heading 'License Keys'. It contains instructions about purchasing web services and generating license keys. A table titled 'Account ID: 991450' lists columns for 'Description', 'License key', 'Key created', and 'Last used'. A button 'Generate new license key' is at the bottom of the table. A success message at the bottom states: 'New license key successfully created'. It includes a note that the key is stored in hashed format and may take up to five minutes to activate. A yellow box says: 'This will be the only time this key is displayed to you in full. Please copy the key to a safe location for your future reference.' Below this, an 'Account ID' section shows '991450' and a 'License key' section with the value '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' and a copy icon. A 'For Usage with GeoIP Update' section notes a config file was generated and provides a 'Download Config' button.

After getting our license key we must create the URL. For the **Maxmind**, we can use a template URL: https://download.maxmind.com/app/geoip_download?edition_id=GeoLite2-Country-CSV&license_key=YOUR_KEY&suffix=zip Replace **YOUR_KEY** with the key you created and paste the new URL into **Firewall => Aliases => GeoIP settings => URL** section and **Apply**.

The screenshot shows the OPNsense 'Firewall: Aliases' configuration page. On the left, a sidebar lists 'Lobby', 'Reporting', 'System', 'Interfaces', 'Firewall' (selected), 'Aliases', 'Automation', 'Categories', 'Groups', and 'NAT'. The main panel has tabs for 'Aliases' (selected) and 'GeoIP settings'. Under 'Aliases', there are sections for 'Last updated' (2024-03-22T18:23:56), 'Total number of ranges' (773307), and 'Url' (with a text input field containing 'https://download.maxmind.com/app/geoip_download?edition_id=GeoLite2-Country-CSV&license_key=YOUR_KEY&suffix=zip'). A red 'Apply' button is at the bottom right. The right side of the screen shows a preview of the 'Aliases' configuration with the same information.



Now we have the **GeolP** source and the next step is to create an aliases to in our rule to block input/output traffic for selected regions. Open **Firewall => Aliases** section and add new one.

Firewall: Aliases

Aliases		GeolP settings		Search		Filter type	Categories	7	grid
Enabled	Name	Type	Description	Content	Loaded#	Last updated	Command		
<input type="checkbox"/>	<input checked="" type="checkbox"/> bogons	External (advanced)	bogon networks (internal)		10				
<input type="checkbox"/>	<input checked="" type="checkbox"/> bogonsv6	External (advanced)	bogon networks IPv6 (inte...		76				
<input type="checkbox"/>	<input checked="" type="checkbox"/> virusprot	External (advanced)	overload table for rate limi...		0				
<input type="checkbox"/>	<input checked="" type="checkbox"/> sshlockout	External (advanced)	abuse lockout table (inter...		0				
<input type="checkbox"/>	<input checked="" type="checkbox"/> __wan_network	Internal (automatic)	WAN net		1				
<input type="checkbox"/>	<input checked="" type="checkbox"/> __lan_network	Internal (automatic)	SERVERS net		1				
<input type="checkbox"/>	<input checked="" type="checkbox"/> __lo0_network	Internal (automatic)	Loopback net		2				

In the alias editing window, give the to your alias in the **Name** section, choose the **GeolP** option in **Type** section and choose countries you want to block.

Edit Alias

Enabled	<input checked="" type="checkbox"/>																						
Name	blocked_countries																						
Type	GeolP																						
Categories																							
Content	<table border="1"> <thead> <tr> <th>Region</th> <th>Countries</th> </tr> </thead> <tbody> <tr> <td>Africa</td> <td>Egypt</td> </tr> <tr> <td colspan="2">1 out of 50 selected</td> </tr> <tr> <td>America</td> <td>Dominican Republic</td> </tr> <tr> <td colspan="2">1 out of 53 selected</td> </tr> <tr> <td>Antarctica</td> <td>Antarctica</td> </tr> <tr> <td colspan="2">1 out of 1 selected</td> </tr> <tr> <td>Arctic</td> <td>Svalbard & Jan Mayen</td> </tr> <tr> <td colspan="2">1 out of 1 selected</td> </tr> <tr> <td>Asia</td> <td>Armenia</td> </tr> <tr> <td colspan="2">1 out of 49 selected</td> </tr> </tbody> </table>	Region	Countries	Africa	Egypt	1 out of 50 selected		America	Dominican Republic	1 out of 53 selected		Antarctica	Antarctica	1 out of 1 selected		Arctic	Svalbard & Jan Mayen	1 out of 1 selected		Asia	Armenia	1 out of 49 selected	
Region	Countries																						
Africa	Egypt																						
1 out of 50 selected																							
America	Dominican Republic																						
1 out of 53 selected																							
Antarctica	Antarctica																						
1 out of 1 selected																							
Arctic	Svalbard & Jan Mayen																						
1 out of 1 selected																							
Asia	Armenia																						
1 out of 49 selected																							

In the last step we will create a new rule for blocking IPs in region. In order to block this traffic, we must append this rule to our **WAN** interface, because only it is able to communicate with outside of our networks (**Internet**). Go to the **Firewall => Rules => WAN**, set the **Action** as **Block** and choose the alias we created above in the **Source** section.



With this rule we said our firewall to block all input traffic from countries we defined in the alias.

Firewall: Rules: WAN

Select category

The changes have been applied successfully.								
	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description
<input type="checkbox"/>	IPv4 *	blocked_countries	*	*	*	*	*	Automatically generated rules

3.2 Captive Portal

As you remember, in previous captures, we mentioned that the **GUESTS** network should be isolated from our environment. We fulfilled it with firewall rules, and in this capture we will setup a specific service to authenticate our guests to give an access to the **Internet** over our **GUESTS** network. This service is called **Captive Portal**, which is a customized login page that guests must address before connecting to the **Internet**. This service has additional features to use for authenticating and limiting network traffic in it, but we will use only one method, **Vouchers**, to control authentication process. In this method we will create a login page and login credentials for guests, what is called voucher. Each voucher will have a validity time limit and we will provide our guests with this vouchers whenever they ask for credentials to connect to the **Internet**.

Before we start configuring **Captive Portal**, we need to create an access server for using it in our service. Go to the **System => Access => Servers** and add new server.

In the configuration window, type a “**Descriptive name**”, choose **Voucher** option in **Type**, put the tic in “**Use**



simple passwords (less secure)" box (it is a little bit user friendly way to creating passwords) and define the credentials' lengths.

System: Access: Servers

Descriptive name	Voucher server
Type	Voucher
Use simple passwords (less secure)	<input checked="" type="checkbox"/>
Username length	7
Password length	7
Save	

System: Access: Servers

Server Name	Type	Host Name	
Voucher server	Voucher	Firewall	
Local Database	Local Database	Firewall	

In the next step, we should generate our vouchers for guests, go to the **Services => Captive Portal => Vouchers** section and press the “Create Vouchers” button.

Lobby
Reporting
System
Interfaces
Firewall
VPN
Services
C-ICAP
Captive Portal
Administration
Sessions
Vouchers
Log File

Services: Captive Portal: Vouchers

Voucher	Valid from	Valid to	Expires at	State
No results found!				

Showing 0 to 0 of 0 entries

In the “Generate vouchers” window, select the options for your need and enter the **Groupname**.

Generate vouchers

Setting	Value
Validity	1 week
Expires in	Custom (hours) 1
Number of vouchers	10
Groupname	guests/vouchers
Generate Close	

Once you press the **Generate** button, vouchers will be in the **Services => Captive Portal => Vouchers** page



and will be downloaded into your machine in **CSV** file format.

<input type="checkbox"/> Voucher	Valid from	Valid to	Expires at	State
<input type="checkbox"/> SUyTCfW	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused
<input type="checkbox"/> WxPGAKi	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused
<input type="checkbox"/> TVhTBK7	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused
<input type="checkbox"/> gzDWTiM	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused
<input type="checkbox"/> cewBEtQ	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused
<input type="checkbox"/> b3F6sHJ	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused
<input type="checkbox"/> Jgz98kG	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused
<input type="checkbox"/> 5cq6Z3X	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused
<input type="checkbox"/> rEhZ9n4	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused
<input type="checkbox"/> w25cefG	Apr 6, 2024 11:48 PM	Apr 13, 2024 11:48 PM	Apr 7, 2024 12:48 AM	unused

Showing 1 to 10 of 10 entries

Expire selected vouchers Drop expired vouchers Create vouchers

You can see the login credentials, username and password, in this file.

	A	B	C	D	E
1	username	password	vouchergr	expirytime	validity
2	SUyTCfW	xZDRhpE	guests vauc	#####	604800
3	WxPGAKi	tnLJaUD	guests vauc	#####	604800
4	TVhTBK7	6USFgGU	guests vauc	#####	604800
5	gzDWTiM	HgY2QUQ	guests vauc	#####	604800
6	cewBEtQ	PcrXFLA	guests vauc	#####	604800
7	b3F6sHJ	YFLYHpy	guests vauc	#####	604800
8	Jgz98kG	ka8pYx3	guests vauc	#####	604800
9	5cq6Z3X	rQEimwH	guests vauc	#####	604800
10	rEhZ9n4	NYJjjVA	guests vauc	#####	604800
11	w25cefG	DTCzC63	guests vauc	#####	604800

After creating vouchers, go to the **Services => Captive Portal => Administration** page and press add button.

Enabled	Description
No results found!	

Showing 0 to 0 of 0 entries

Commands

In the “Edit zone” window, put the tic in **Enabled** box, choose the interface in **Interfaces**, in “Authentication



using" section, select the authentication server name which we created, give a description and **Save**.

Edit zone

<input type="checkbox"/> advanced mode	full help
<input checked="" type="checkbox"/> Enabled	
<input type="checkbox"/> Zone number	0
<input type="checkbox"/> Interfaces	GUESTS
Clear All	
<input type="checkbox"/> Authenticate using	Voucher server
Clear All	
<input type="checkbox"/> Always send accounting requests	<input type="checkbox"/>
<input type="checkbox"/> Enforce local group	None
<input type="checkbox"/> Idle timeout (minutes)	0
<input type="checkbox"/> Hard timeout (minutes)	0
<input type="checkbox"/> Concurrent user logins	<input checked="" type="checkbox"/>
<input type="checkbox"/> SSL certificate	None
<input type="checkbox"/> Hostname	
<input type="checkbox"/> Allowed addresses	Clear All
<input type="checkbox"/> Transparent proxy (HTTP)	<input type="checkbox"/>
<input type="checkbox"/> Transparent proxy (HTTPS)	<input type="checkbox"/>
<input type="checkbox"/> Custom template	None
<input type="checkbox"/> Description	Captive portal for GUESTS
Cancel Save	

That was the last step of **Captive Portal** configuration for our lab, **Apply** changes and go to the next setup.

Services: Captive Portal: Administration

Zones		Templates	
<input type="checkbox"/>	Enabled	<input type="checkbox"/>	Description
<input checked="" type="checkbox"/>	Captive portal for GUESTS		
« » 1 2 3			
Showing 1 to 1 of 1 entries			
After changing settings, please remember to apply them with the button below			
Apply			

Now we must write new rules for **GUESTS** interface to allow network to access DNS and Captive Portal ports. Go to the **Firewall => Rules => GUESTS** and add new rule.

Firewall: Rules: GUESTS

	Protocol	Source	Port	Destination	Port	Gateway	Schedule		Description					
Automatically generated rules														
<input type="checkbox"/>		IPv4 *	*	S_C_J net	*	*	*							
<input type="checkbox"/>		IPv4 *	*	*	*	*	*							

In the rule, select **Pass** for **Action**, select **GUESTS** in **Interfaces**, select **TCP** as **Protocol**, select "**GUESTS net**"



as **Source**, select “**GUESTS address**” as **Destination**, select “**(other)**” option in “**Destination port range**” and type **from: 8000, to:10000** (**Captive Portal** works in this port range in **OPNsense** firewall).

Firewall: Rules: GUESTS

Edit Firewall rule	
Action	Pass
Disabled	<input type="checkbox"/> Disable this rule
Quick	<input checked="" type="checkbox"/> Apply the action immediately on match.
Interface	GUESTS
Direction	in
TCP/IP Version	IPv4
Protocol	TCP
Source / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Source	GUESTS net
Source	Advanced
Destination / Invert	<input type="checkbox"/> Use this option to invert the sense of the match.
Destination	GUESTS address
Destination port range	from: <input type="text" value="(other)"/> <input type="button" value="▲"/> to: <input type="text" value="(other)"/> <input type="button" value="▲"/> <input type="text" value="8000"/> <input type="text" value="10000"/>

Firewall: Rules: GUESTS

The firewall rule configuration has been changed. You must apply the changes in order for them to take effect.								<input type="button" value="Apply changes"/>	
	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description	<input type="button" value="+"/>
<input type="checkbox"/>	IPv4 *	GUESTS net	*	S_C_J net	*	*	*	Automatically generated rules	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Copy"/> <input type="button" value="Paste"/>
<input type="checkbox"/>	IPv4 *	*	*	*	*	*	*		<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Copy"/> <input type="button" value="Paste"/>
<input type="checkbox"/>	IPv4 TCP	GUESTS net	*	GUESTS address	8000 - 10000	*	*		<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Copy"/> <input type="button" value="Paste"/>

After creatin new rule, arrange all rules as shown in the figure below.

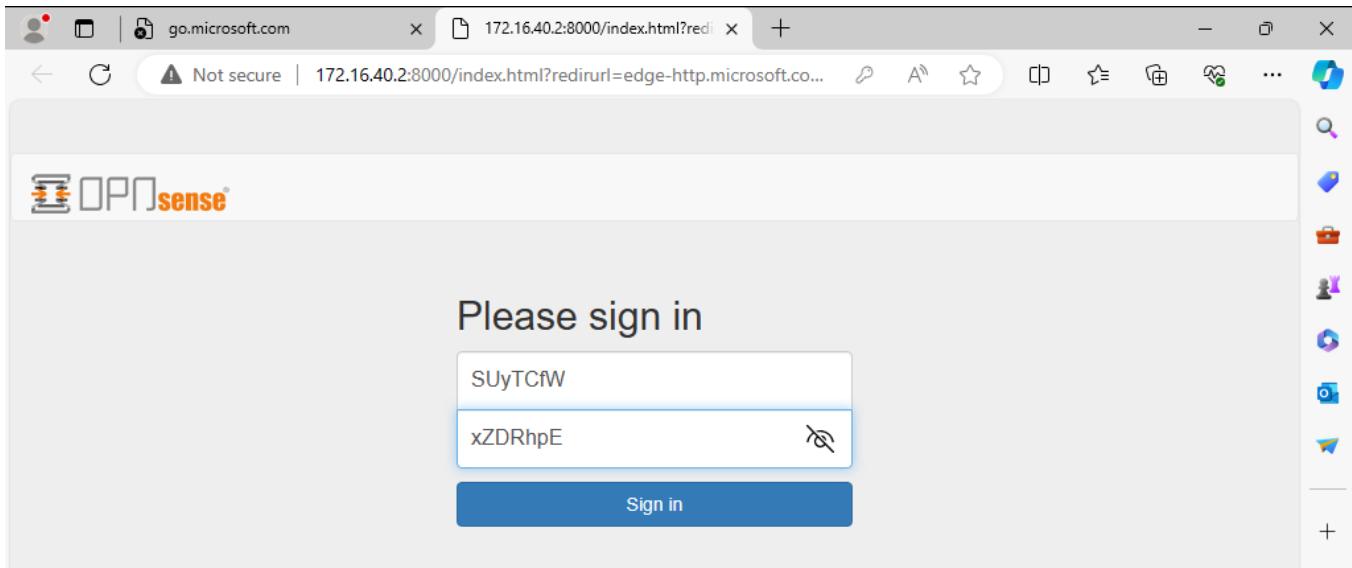
Firewall: Rules: GUESTS

The changes have been applied successfully.									
	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description	<input type="button" value="+"/>
<input type="checkbox"/>	IPv4 TCP	GUESTS net	*	GUESTS address	8000 - 10000	*	*	Automatically generated rules	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Copy"/> <input type="button" value="Paste"/>
<input type="checkbox"/>	IPv4 *	GUESTS net	*	S_C_J net	*	*	*		<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Copy"/> <input type="button" value="Paste"/>
<input type="checkbox"/>	IPv4 *	*	*	*	*	*	*		<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Copy"/> <input type="button" value="Paste"/>

Now it is time to test our setup, go to the guest machine which you connect to the **GUESTS** interface (for our lab, **VMNET4** adapter), then try to go any address, and the default login page will meet you. Enter one of the vouchers in it and “**Sign up**”. Our guest machine could access to the **Internet** until the voucher



expires.



We can observe and finish the sessions in the **Services => Captive Portal => Sessions** section.

Username	MAC address	IP address	Connected since
SUyTCfW	00:0c:29:61:df:0f	172.16.40.10	Invalid date

3.3 IDS/IPS configuration

If you analyze the firewall rules and proxies' configurations carefully, you will notice that we didn't fully protect **SERVERS** and **CLIENTS** networks. If bad actors try to scan these networks and machines inside them, they will be successful because our infrastructure is vulnerable to recon tools, such as **Nmap**, **Metasploit** etc. In this capture, we will try to build an extra security layer by configure IDS/IPS feature of **OPNsense** firewall.

IDS (Intrusion Detection System) listen the network traffic and search for suspicious patters to match with signatures (rules) in its database and gives an alert if they match. **IDS** doesn't intervene any action, but **IPS** (Intrusion Prevention System) can drop the requests if it is suspicious.

In the earliest step, we will configure the **IDS**, go to the **Services => Intrusion Detection => Administration** section and toggle the “**advanced mode**”, then put ticks in **Enabled**, “**Promiscuous mode**”, “**Enable syslog alerts**” boxes, select **Hyperscan** in “**Pattern matcher**” section, select **SERVERS** and **CLIENTS** in **Interfaces**, type these interfaces subnet IPs in “**Home networks**” section and **Apply**. (you can get detailed information



about these options in **OPNsense** documentation in its web page)

Services: Intrusion Detection: Administration

Settings	Download	Rules	User defined	Alerts	Schedule
<input checked="" type="checkbox"/> advanced mode					
Enabled	<input checked="" type="checkbox"/>				
IPS mode	<input type="checkbox"/>				
Promiscuous mode	<input checked="" type="checkbox"/>				
Enable syslog alerts	<input checked="" type="checkbox"/>				
Enable eve syslog output	<input type="checkbox"/>				
Syslog verbosity	DEFAULT (0)				
Pattern matcher	Hyperscan				
Detect Profile	Default				
Interfaces	CLIENTS, SERVERS				
Home networks					
<input type="text"/> 172.16.10.0/24 <input type="text"/> 172.16.20.0/24					
<input type="button"/> Clear All <input type="button"/> Copy <input type="button"/> Text					
default packet size					
Rotate log	Weekly				
Save logs	4				
Log package payload	<input type="checkbox"/>				
Apply					

Later go to the **Download** page and put the tic in “**abuse.ch/URLhaus**” option and press “**Enable selected**” button and press “**Download & Update Rules**” button. Each of these option is a **IDS/IPS** database and they contain hundreds of signatures in them, but we will use only one of them for better understanding.

Services: Intrusion Detection: Administration

Settings	Download	Rules	User defined	Alerts	Schedule																																																
Rulesets	<input checked="" type="button"/> Enable selected <input type="button"/> Disable selected																																																				
	<input type="text"/> Search																																																				
	<table border="1"> <thead> <tr> <th>Description</th> <th>Last updated</th> <th>Enabled</th> <th>Edit</th> </tr> </thead> <tbody> <tr> <td>abuse.ch/Feodo Tracker</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>abuse.ch/SSL Fingerprint Blacklist</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>abuse.ch/SSL IP Blacklist</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>abuse.ch/ThreatFox</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>abuse.ch/URLhaus</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>ET open/3coresec</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>ET open/botcc</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>ET open/botcc.portgrouped</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>ET open/clarmy</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>ET open/compromised</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>ET open/drop</td> <td>not installed</td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> </tbody> </table>					Description	Last updated	Enabled	Edit	abuse.ch/Feodo Tracker	not installed	<input type="button"/>	<input type="button"/>	abuse.ch/SSL Fingerprint Blacklist	not installed	<input type="button"/>	<input type="button"/>	abuse.ch/SSL IP Blacklist	not installed	<input type="button"/>	<input type="button"/>	abuse.ch/ThreatFox	not installed	<input type="button"/>	<input type="button"/>	abuse.ch/URLhaus	not installed	<input type="button"/>	<input type="button"/>	ET open/3coresec	not installed	<input type="button"/>	<input type="button"/>	ET open/botcc	not installed	<input type="button"/>	<input type="button"/>	ET open/botcc.portgrouped	not installed	<input type="button"/>	<input type="button"/>	ET open/clarmy	not installed	<input type="button"/>	<input type="button"/>	ET open/compromised	not installed	<input type="button"/>	<input type="button"/>	ET open/drop	not installed	<input type="button"/>	<input type="button"/>
Description	Last updated	Enabled	Edit																																																		
abuse.ch/Feodo Tracker	not installed	<input type="button"/>	<input type="button"/>																																																		
abuse.ch/SSL Fingerprint Blacklist	not installed	<input type="button"/>	<input type="button"/>																																																		
abuse.ch/SSL IP Blacklist	not installed	<input type="button"/>	<input type="button"/>																																																		
abuse.ch/ThreatFox	not installed	<input type="button"/>	<input type="button"/>																																																		
abuse.ch/URLhaus	not installed	<input type="button"/>	<input type="button"/>																																																		
ET open/3coresec	not installed	<input type="button"/>	<input type="button"/>																																																		
ET open/botcc	not installed	<input type="button"/>	<input type="button"/>																																																		
ET open/botcc.portgrouped	not installed	<input type="button"/>	<input type="button"/>																																																		
ET open/clarmy	not installed	<input type="button"/>	<input type="button"/>																																																		
ET open/compromised	not installed	<input type="button"/>	<input type="button"/>																																																		
ET open/drop	not installed	<input type="button"/>	<input type="button"/>																																																		
	Download & Update Rules																																																				



After we successfully download the checked database's rules, they will appear in **Rules** section. In order to test our **IDS**, choose a rule and press **Edit** icon.

Services: Intrusion Detection: Administration

Settings Download Rules User defined Alerts Schedule

Filters Search 10

sid	Action	Source	ClassType	Message	Edit	Enabled
83670412	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>
83670411	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>
83670410	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>
83670409	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>
83670408	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>
83670406	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>
83670404	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>
83670405	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>
83670403	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>
83670402	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected ...	<input type="button"/>	<input checked="" type="checkbox"/>

Alert Drop Showing 1 to 10 of 29409 entries

Apply

In the popped window, copy the **URL** address in **reference_html** section.

Rule details

full help

Signature Id	83670412
Revision	1
Message	URLhaus Known malware download URL detected (2807312)
classtype	trojan-activity
created_at	2024_04_10
reference_html	urlhaus.abuse.ch/url/2807312/
source	abuse.ch.urlhaus.rules
status	enabled
Action	Alert

Cancel Save

Open your browser in a virtual machine which belongs to one of **SERVERS** or **CLIENTS** networks, then in the search bar, paste and go to that address. In the opened web page, again copy the address in **URL:** section and go to that address. It will download a file whose signature is exist in our **IDS**'s database.



ID:	2807312
URL:	http://115.63.49.20:33619/bin.sh
URL Status:	Online (spreading malware for 12 minutes)
Host:	115.63.49.20
Date added:	2024-04-10 10:44:06 UTC
Threat:	Malware download
Reporter:	geenensp
Abuse complaint sent (?:)	Yes (2024-04-10 10:45:07 UTC to zhaoyz3[at]chinaunicom[dot]cn)
Tags:	32-bit, elf, mips, MozI

Now back to the **Services => Intrusion Detection => Administration => Alert** section in **OPNsense GUI**, here is our alert. When we downloaded this malicious test file from **URLhaus** web page, **IDS** matched this file's patters with its existing rule and gave an alert to us.

Timestamp	SID	Action	Interface	Source	Port	Destination	Port	Alert	Info
2024-04-10T14:57:44.019610+0400	83670412	allowed	SERVERS	172.16.10.10	52995	115.63.49.20	33619	URLhaus Known malware download URL detected (2807312)	

We will try it again but this time IPS will be active. Go to the **Services => Intrusion Detection => Administration => Settings** section and put tic in “**IPS mode**” option and **Apply**.



Services: Intrusion Detection: Administration

Settings Download Rules User defined Alerts Schedule

advanced mode

Enabled

IPS mode

Promiscuous mode

Enable syslog alerts

Enable eve syslog output

Syslog verbosity: DEFAULT (0)

Pattern matcher: Hyperscan

Detect Profile: Default

Interfaces: CLIENTS, SERVERS

Home networks: 172.16.10.0/24, 172.16.20.0/24

default packet size

Rotate log: Weekly

Save logs: 4

Log package payload

Apply

After activating **IPS mode**, go to the **Rules** section and choose the same rule we checked above and press the **Drop** button (or choose **Action** as **Drop** in “**Rules Details**” window) and **Apply**.

Services: Intrusion Detection: Administration

Settings Download **Rules** User defined Alerts Schedule

Filters

sid	Action	Source	ClassType	Message	Info / Enabled
<input checked="" type="checkbox"/> 83670412	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 83670411	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 83670410	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 83670409	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 83670408	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 83670406	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 83670404	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 83670405	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 83670403	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 83670402	alert	abuse.ch.urlhaus.rules	trojan-activity	URLhaus Known malware download URL detected (...)	<input type="checkbox"/> <input checked="" type="checkbox"/>

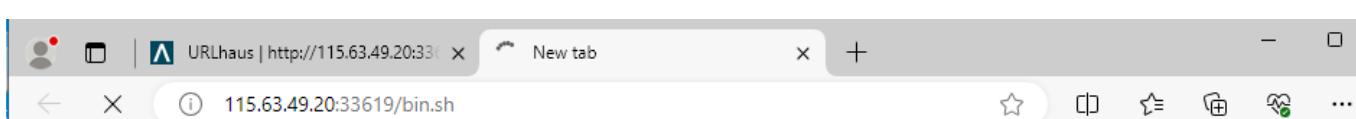
Alert Drop

1 2 3 4 5 > *

Showing 1 to 10 of 29409 entries

Apply

Back to the browser and retry to go the address to download the same file again.





You'll see that the action will be blocked and when you check the **Alert** section there will be new alert with action **blocked**. This is the main difference between **IDS** and **IPS**. **IDS** just detect the suspicious traffic and give alert but **IPS** detect and block it.

Timestamp	SID	Action	Interface	Source	Port	Destination	Port	Alert	Info
2024-04-10T15:01:22.875534+0400	83670412	blocked	SERVERS	172.16.10.10	53096	115.63.49.20	33619	URLhaus Known malware download URL detected (2807312)	
2024-04-10T15:01:22.875534+0400	83670412	blocked	SERVERS	172.16.10.10	53096	115.63.49.20	33619	URLhaus Known malware download URL detected (2807312)	
2024-04-10T14:57:44.019610+0400	83670412	allowed	SERVERS	172.16.10.10	52995	115.63.49.20	33619	URLhaus Known malware download URL detected (2807312)	

Although **OPNsense IDS** comes with thousands of default rules, you may want to add your own custom rules in it. Until here we did almost every configuration in **GUI** interface but this time we obliged to use the **CLI**. But before we start to configuration in firewall, we need to prepare a database for our custom rules. As you know we have a **WEB** server in our infrastructure and we deployed the web application (**DVWA**) in it. We will use this **WEB** server as our database in this task. Open your **Oracle Linux** machine and go to the **/var/www/html** path and create new file named **nmap.rules** (or whatever you want for name) in this path.

```
[root@web ~]# cd /var/www/html
[root@web html]# nano nmap.rules
```

Now we have to add our custom rules in this file. Instead of typing all rules one by one, I used the ready **Github** repository for detecting **Nmap** scanning activities. Open the link below, copy and paste all rules in it to your **nmap.rules** file.

Nmap rules: <https://github.com/aleksibovellan/opnsense-suricata-nmaps/blob/main/local.rules>

```
# opnsense-suricata-nmaps
# OPNsense's Suricata IDS/IPS Detection Rules Against Nmap Scans
# v. 1.4.3 / March 24th 2024 by Aleksi Bovellan
# https://github.com/aleksibovellan/opnsense-suricata-nmaps

# For Nmap Detections between scan speeds of -T5-T0.

# Nmap -sS scans at -T0, this rule has a limited port range to avoid constant alerts from generic ports
alert tcp any any -> any [21,22,23,25,80,88,110,135,137,138,139,161,389,443,445,465,514,587,63$]
alert tcp any any -> any [21,22,23,25,80,88,110,135,137,138,139,161,389,443,445,465,514,587,63$]

# Nmap -sU scans at -T0, this rule has a limited port range to avoid constant alerts from generic ports
alert udp any any -> any [53,67,68,69,123,161,162,389,520,1026,1027,1028,1029,1434,1900,11211,$

# Nmap -f scans at -T0:
alert ip any any -> any any (msg:"POSSBL SCAN NMAP KNOWN FRAGM (type -f)"; fragbits:M+D; thresh$)
alert ip any any -> any any (msg:"POSSBL SCAN NMAP FRAGM (type -f)"; fragbits:M; threshold:typ$)

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text   ^J Justify   ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace    ^U Uncut Text ^T To Spell  ^L Go To Line
```



```
[root@web ~]# cd /var/www/html  
[root@web html]# pwd  
/var/www/html  
[root@web html]# nano nmap.rules  
[root@web html]# ll nmap.rules  
-rw-r--r-- 1 root root 5095 Apr 10 22:45 nmap.rules  
[root@web html]#
```

After creating our rules database, we are ready to configure **Suricata IDS** (OPNsense use **Suricata** as **IDS/IPS** service as default) to pull rules from this database.

As you remember we enabled **SSH** service for connecting our firewall over the **SERVERS** network and defined port as **3022** for security reason. For easy management, I recommend to connect firewall with **SSH**, how I did in this task (you can use its console too).

Go to the `/usr/local/opnsense/scripts/suricata/metadata/rules` path and create new file named `custom.xml`.

Inside this file we have to define our database's location **URL** (which is our **WEB** server's **IP address**) and its name (**nmap.rules**). You can get the syntax of configuration from this source below.

<https://forum.opnsense.org/index.php?topic=7209.0>



```
Windows PowerShell          x + v
GNU nano 7.2                                     custom.xml
<?xml version="1.0"?>
<ruleset documentation_url="http://docs.opnsense.org/">
    <location url="http://172.16.10.11" prefix="Custom"/>
    <files>
        <file description="custom scan rules">nmap.rules</file>
        <file description="Custom" url="inline::rules/nmap.rules">nmap.rules</file>
    </files>
</ruleset>
```

```
yagub@Firewall:/usr/local/opnsense/scripts/suricata/metadata/rules % ll
total 20
-rw-r--r-- 1 root  wheel  314 Apr 10 22:42 custom.xml
-rw-r--r-- 1 root  wheel  6267 Apr  5 23:38 et-open.xml
-rw-r--r-- 1 root  wheel   722 Apr  5 23:38 opnsense.xml
-rw-r--r-- 1 root  wheel  1289 Apr  5 23:38 sslbl.xml
yagub@Firewall:/usr/local/opnsense/scripts/suricata/metadata/rules % |
```

After configuring **Suricata**, back to the **GUI** and restart the **IDS** service. Go to the **Download** section, find your custom rules, enable and download.

Services: Intrusion Detection: Administration

Settings Download Rules User defined Alerts Schedule

Rulesets

Enable selected | Disable selected

Search

	Description	Last updated	Enabled	Edit
<input type="checkbox"/>	abuse.ch/Feodo Tracker	not installed		
<input type="checkbox"/>	abuse.ch/SSL Fingerprint Blacklist	not installed		
<input type="checkbox"/>	abuse.ch/SSL IP Blacklist	not installed		
<input type="checkbox"/>	abuse.ch/ThreatFox	not installed		
<input type="checkbox"/>	abuse.ch/URLhaus	not installed		
<input checked="" type="checkbox"/>	Custom/custom rules	not installed		
<input type="checkbox"/>	ET open/3coresec	not installed		
<input type="checkbox"/>	ET open/botcc	not installed		
<input type="checkbox"/>	ET open/botcc.portgrouped	not installed		
<input type="checkbox"/>	ET open/clarmy	not installed		
<input type="checkbox"/>	ET open/compromised	not installed		

Download & Update Rules

In the **Rules** section you will find all your custom rules which are available.



Services: Intrusion Detection: Administration

<input type="checkbox"/> sid	Action	Source	ClassType	Message	Info / Enabled
<input type="checkbox"/> 1000001	alert	nmap.rules	attempted-recon	SUSP PORT PROBE KNOWN TCP (type -sS)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 1000002	alert	nmap.rules	attempted-recon	POSSBL SCAN NMAP KNOWN TCP (type -sS)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 1000003	alert	nmap.rules	attempted-recon	POSSBL SCAN NMAP KNOWN UDP (type -sU)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 1000004	alert	nmap.rules	attempted-recon	POSSBL SCAN NMAP KNOWN FRAGM (type -f)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 1000005	alert	nmap.rules	attempted-recon	POSSBL SCAN NMAP FRAGM (type -f)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 1000006	alert	nmap.rules	attempted-recon	POSSBL SCAN NMAP KNOWN TCP (type -sS)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 1000007	alert	nmap.rules	attempted-recon	POSSBL SCAN NMAP TCP (type -sS)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 1000008	alert	nmap.rules	attempted-recon	POSSBL SCAN NMAP TCP (type -sT)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 1000009	alert	nmap.rules	attempted-recon	POSSBL SCAN NMAP KNOWN UDP (type -sU)	<input type="checkbox"/> <input checked="" type="checkbox"/>
<input type="checkbox"/> 1000010	alert	nmap.rules	attempted-recon	POSSBL SCAN NMAP UDP (type -sU)	<input type="checkbox"/> <input checked="" type="checkbox"/>

Alert Drop

< < 1 2 > >>

Showing 1 to 10 of 16 entries

Apply

In order to test your rules, go to your **Kali** machine and run a scan on your **DC** (or any other machine in **SERVERS** or **CLIENTS** network).

```
yagub@kali: ~
[yagub㉿kali)-[~]
$ sudo nmap 172.16.10.10 -sS
[sudo] password for yagub:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-10 23:56 +04
Nmap scan report for 172.16.10.10
Host is up (0.055s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
3268/tcp  open  globalcatLDAP
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 62.23 seconds
```

After completing **Nmap** scan, back to the **Alerts** section and you will see the tens of alerts generated by **IDS** and all of them are allowed to pass.



Services: Intrusion Detection: Administration

Timestamp	SID	Action	Interface	Source	Port	Destination	Port	Alert	Info
2024-04-10T17:00:37.745914+0400	1000008	allowed	SERVERS	172.16.10.1	53418	172.16.10.10	1201	POSSBL SCAN NMAP TCP (type -sT)	
2024-04-10T17:00:37.745914+0400	1000008	allowed	SERVERS	172.16.10.1	53432	172.16.10.10	55555	POSSBL SCAN NMAP TCP (type -sT)	
2024-04-10T17:00:37.745914+0400	1000008	allowed	SERVERS	172.16.10.1	53477	172.16.10.10	912	POSSBL SCAN NMAP TCP (type -sT)	
2024-04-10T17:00:37.745914+0400	1000008	allowed	SERVERS	172.16.10.1	53486	172.16.10.10	25735	POSSBL SCAN NMAP TCP (type -sT)	
2024-04-10T17:00:37.745914+0400	1000008	allowed	SERVERS	172.16.10.1	53429	172.16.10.10	44501	POSSBL SCAN NMAP TCP (type -sT)	
2024-04-10T17:00:37.745914+0400	1000008	allowed	SERVERS	172.16.10.1	53276	172.16.10.10	427	POSSBL SCAN NMAP TCP (type -sT)	
2024-04-10T17:00:37.745914+0400	1000008	allowed	SERVERS	172.16.10.1	53499	172.16.10.10	3914	POSSBL SCAN NMAP TCP (type -sT)	

If you remember we enabled **IPS** mode for this service and tested it on **URLhaus**, this time we will test it on **Nmap** scans. All we need to do is to change the rules actions from alert to drop. Instead of doing it one by one manually, **OPNsense** lets us to modify whole database at the same time. In order to do it, go to the **Services => Intrusion Detection => Policy** section and add new policy.

Services: Intrusion Detection: Policy

Policies	Rule adjustments		
<input type="checkbox"/> Enabled	Priority	Description	Commands
No results found!			
Showing 0 to 0 of 0 entries			
Apply			

In the popped window, put tic in **Enabled** box, assign a **Priority**, select the custom rule in **Rulesets**, select the rule types in **Rules**, select **Drop** as “**New Action**” and **Save**. This will change all rules’ conditions from **allowed** to **blocked**.

Rule details

Enabled	<input checked="" type="checkbox"/>	full help
Priority	1	
Rulesets	nmap.rules	
Action	Alert	
Rules	classtype	attempted-recon, trojan-activity
New action	Drop	
Description		
		Cancel Save



After creating new policy, press **Apply** button.

Services: Intrusion Detection: Policy

Policies		Rule adjustments
<input type="checkbox"/> Enabled	Priority	Description
<input checked="" type="checkbox"/>	1	
		Commands
		+
Showing 1 to 1 of 1 entries		
Apply		

Now back to your **Kali** machine and try to scan again. You will see that new alerts appear in the **Alerts** section, but this time their condition become **blocked**.

Services: Intrusion Detection: Administration

Alerts										Schedule				
										Search		2024/04/11 0:00	7	
Timestamp	SID	Action	Interface	Source	Port	Destination	Port	Alert	Info					
2024-04-11T00:00:56.259069+0400	1000008	blocked	SERVERS	172.16.10.1	59647	172.16.10.10	90	POSSBL SCAN NMAP TCP (type -sT)						
2024-04-11T00:00:56.259069+0400	1000008	blocked	SERVERS	172.16.10.1	59647	172.16.10.10	90	POSSBL SCAN NMAP TCP (type -sT)						
2024-04-11T00:00:56.226928+0400	1000008	blocked	SERVERS	172.16.10.1	59643	172.16.10.10	1122	POSSBL SCAN NMAP TCP (type -sT)						
2024-04-11T00:00:56.226928+0400	1000008	blocked	SERVERS	172.16.10.1	59643	172.16.10.10	1122	POSSBL SCAN NMAP TCP (type -sT)						
2024-04-11T00:00:56.182879+0400	1000008	blocked	SERVERS	172.16.10.1	59633	172.16.10.10	22939	POSSBL SCAN NMAP TCP (type -sT)						
2024-04-11T00:00:56.182879+0400	1000008	blocked	SERVERS	172.16.10.1	59633	172.16.10.10	22939	POSSBL SCAN NMAP TCP (type -sT)						
2024-04-11T00:00:56.181017+0400	1000008	blocked	SERVERS	172.16.10.1	59635	172.16.10.10	7625	POSSBL SCAN NMAP TCP (type -sT)						
										Showing 1 to 7				

NOTE: sometimes **IPS** mode doesn't block actions, this is because we are using virtual environment and open source tool, but the important thing is to understand the logic of **IDS/IPS** and their usage with next generation firewalls.



Conclusion

This lab was designed to understand fundamental security concepts. During the lab we used different methods to protect our infrastructure. Although we spent a lot of effort to make as possible as secure our infrastructure, still there are some weaknesses. Bad virtualization and open source solutions sometimes don't work properly and cause unpredictable errors and bugs. But despite all of these problems, we gained an ability to work with firewalls and design a basic infrastructure.

Now lets analyze our project and give a general summarize what we did:

- We prepared a foundation for our lab in pre-requirements capture
- Installed **OPNsense** firewall, added 5 different networks adapters and configured them
- For the security reason, we created a new user with root privileges and disabled **root** user
- Changed defaults access ports for **HTTPS** and **SSH** access, and defined only **SERVERS** network for firewall access and blocked for all other networks
- Isolated all networks from each other with firewall rules and applied Jump server logic to access our servers (although we didn't mention about this method and complete fully installation and testing it, you can find numerous sources to learn its logic)
- Installed a reverse proxy (**Nginx**) to hide our **WEB** server behind it and applied **WAF** function of **Nginx** plugin for protecting our web application (**DVWA**) against cyberattacks by using **Naxsi** policies
- Installed transparent **Squid Web Proxy** in our firewall and applied web filtering on **SERVERS** and **CLIENTS** networks to protect them from malicious files and web sites
- Installed **Zenarmor** plugin for web filtering **GUESTS** network's web traffic
- Configured **GeoIP** feature of firewall to block internet traffic with some selected countries
- Configured **Captive Portal** to authenticate and control the guests' **Internet** access in **GUESTS** network
- Configured **IDS/IPS** and added our custom rules to protect our **SERVERS** and **CLIENTS** network against suspicious network traffic

I hope this documentation became useful for you. Your feedbacks are precious for me, feel free to contact with me if you have a question or comment about pros and cons of this lab.