University of Piraeus

Department of Digital Systems

MSc in “Security of Digital Systems”

Master Thesis: “*Cyber Wargame Environment*”

Georgios Sextos
MTE 1331

Supervisor: Prof. Christos Xenakis December 2015

Table of Contents

Table of Figures

1 INTRODUCTION

During the semesters on the department of Security of Digital Systems in the University of Piraeus a postgraduate student has the ability to learn and clearly understand the fundamentals of Information Security in regards to Application security, network security, mobile security, while also trying to learn how to comply with related security regulations using specific security frameworks.

Apart from having such knowledge in a theoretic level, a postgraduate student has the ability to test his/hers skills in laboratory sessions trying to perform exploitations and other security related challenges, mostly on UNIX oriented operating systems.

As a result the specific thesis comes in front to help the students develop their knowledge and test their skills in the "Real World" field, as the developed infrastructure simulates a real enterprise environment consisting of front-end firewalls with different zones of security (DMZ zone, LAN, etc.), intrusion detection system for monitoring security incidents that may happen within the infrastructure, as well as a Windows infrastructure like in almost all Enterprises across the Globe. During the engagement of the students with an infrastructure like this, they will gain certain hands-on experience of a small scale enterprise infrastructure, on UNIX and Windows based systems and the strengths and weaknesses that may be hiding behind any of them.

Using an infrastructure like this, different scenarios can be performed to educate the students like:

- Intruders that are trying to bypass the security mechanisms in order to jump from one system to the other in order to gain access to the internal systems (e.g. Domain Controller).

- Internal "employees" that are trying to find out how a specific security breach has happened and proceed with the respective changes on the systems to block such breaches from happening again.

- Real cyberwarfare scenario, where an attacking team tries to penetrate the systems and a defending team that tries to analyze the attacking methods, find security holes in the infrastructure and try to avoid sensitive internal data leakage.

This Master Thesis is a part of a biggest project in order to create a Cyber Warfare Emulation Environment. The other part of this project consists of the postgraduate student Master Thesis Stamatis Mandilas under the name "*Cyber Warfare Emulation Environment*" this part contains the rest elements which complete the Emulation Environment (PFsense, Snort IDS, Domain Controller, Sql Box, Database).

2 CYBER COMPETITIONS - CHALLENGES

There are many categories of cyber warfare competitions – challenges such as: Wargames, Challenge based competition, Capture the flag, etc.

2.1 Wargames

These games take place on given server, where you start with an ssh login and try to exploit setuid-binaries to gain higher permissions. These games are usually available 24/7 and you can join whenever you want.

- Over The Wire
- Smash The Stack
- Intruded

2.2 Challenge based competitions

These games will present numerous tasks that you can solve separately. The challenges mostly vary from exploitation, CrackMes, crypto, forensic, web security and more. These games are usually limited to a few days and the team with the most tasks solved is announced the winner. Some of the listed below:

- Defcon Quals
- Codegate Quals
- CSAW CTF (usually during summer)
- Hack.lu CTF 2011 (end of September) and Hack.lu CTF 2010
- PlaidCTF

2.3 Capture the flag (CTF)

These require you to capture and protect "flags". The best known is probably iCTF. This game is also limited to a certain time frame. Contestants are typically equipped with a Virtual Machine that they are to connect to a VPN. Your task is to analyze the presented machine, find security bugs, patch them and exploit the bugs on other machines in your VPN. The "flags" are stored and retrieved by a central game-server that checks a team's availability and whether previously stored flags have not been stolen.

- iCTF (typically in December)
- CIPHER CTF (will be renewed by new organizers this year)
- RuCTF and RuCTFe (a Russian CTF and its international version)

2.4 Other

There are also some downloadable virtual machines available to play offline, which is some kind of mix between 3 and 2 categories.

- Vulnerable Web Application
- Damn Vulnerable Linux
- Google Jarlsberg

The specific thesis and implementation is based and inspired by Cyber Protector CDX.

Cyber Protector is a live technical Blue/Red Team CDX. Our choice about implementation has to do with flexibility and modularity of the topology. Modular topology is the first priority in order to provide flexibility and scalability in the exercise. Topology can be set based in the experience of the participants and the preferable difficulty and complexity. Also a live cyber exercise is given a more realistic sense.

- Blue Team have to defend a pre-built network consisting of up to 15 virtual machines against
- Blue Team will login remotely to the virtual infrastructure from their own premises.

- Red Team attacks. The infrastructure is initially unknown to them and contains vulnerabilities.
- The Red Team objective is to conduct attacks in Blue Team network.

Advantages

✓ Modular topology (set based on educational needs).
✓ Many levels of difficulty based on Blue / Red team experience.
✓ Security controls, tools, process and procedures deployed for Cyber Defense on networks.
✓ Identification of security gaps.
✓ Evolve and make progress year by year.
✓ Get acquainted with New Technologies and New Cyber Security Solutions.

3 IMPLEMENTATION SCENARIO& COMPONENTS

During the implementation various operating systems, security products and mechanisms have been selected in order to create a topology that could cover a wide specter of security products and technologies.

The related components can be found on the following chapters, where they are being presented in detail.

3.1 Network Architecture Topology

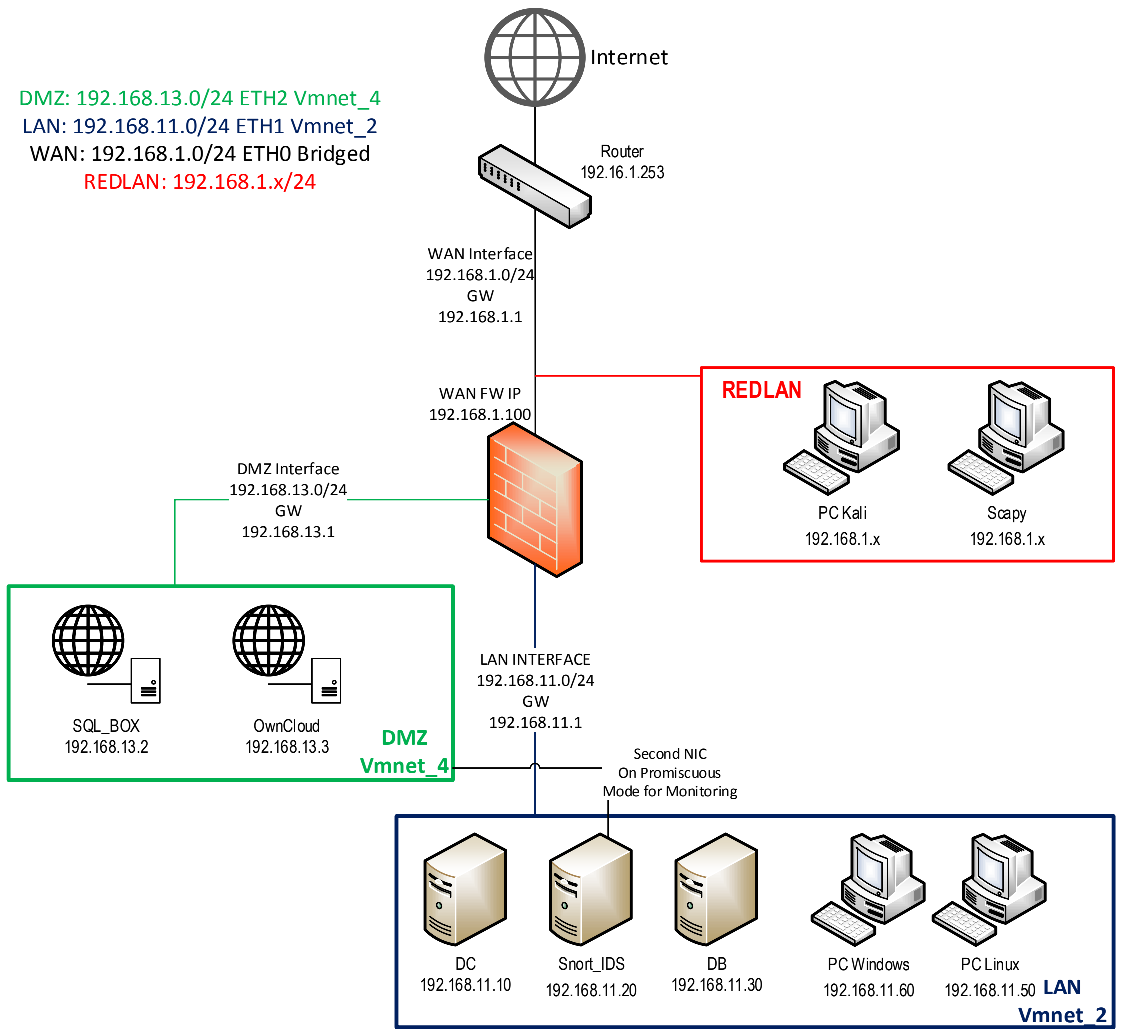


Figure 1: Network Architecture Topology

- LAN Network: 192.168.11.0/24(Workstations, DC, DB, and Snort_IDS)
- DMZ Network: 192.168.13.0/24 (Webservers:Unix [Apache], Windows [IIS])

- WAN Network: 192.168.1.0/24 (WAN network, Network that RED team resides as well)

As the implementation has taken place on a virtual laboratory environment the WAN IP of the firewall is considered to be the Public IP of the infrastructure and thus will be the IP that all the services will be mapped in order for the RED team to be able to attack.

3.2 Firewall

Firewall can be defined as *a "collection of systems*" installed at the connection point of the protected area-network to other networks, which requires a predefined security policy. Installing a firewall in an organization aims in optimization of the existing level of protection of data and computing resources of the organization from attackers.

The main role of the firewall is to prevent unauthorized access to a safe area and prevent unauthorized information output from an area. The preferred practice is to set the firewall to reject all connections except those permitted by the Administrator i.e. (Default-Deny). The main purpose of placing a firewall is to prevent attacks on the local network.

3.2.1 Firewall Categories

Packet Filtering

Most of the network layer firewall (packet filtering) are simple routers (routers). Routers are generally referred to as filtering routers (filtering or screening routers) or network level firewall (network level or packet filter firewalls).

As is known, an IP packet consists of a header and information (data). In header packet there is the information that a router takes in order to make routing. Also with this Firewall decide whether to PERMIT a packet or to DROP this packet. This information is the Source IP, Destination IP, Source Port and Destination Port. The advantages and disadvantages of packet filter firewalls are listed below.

Advantages

- ✓ Installed and configured very easily.
- ✓ It is transparent to users: Because safety of this class do not deal at all with the data portion of the packet, it is not necessary for users to learn any special commands to handle.
- ✓ They have great execution speed because they only deal with part of the IP packet header. Moreover, the whole processing performed in the operating system kernel which is able to process the information faster.
- ✓ Low cost option.

Disadvantages

- ✗ Logging of incidents is simplistic.
- ✗ Do not offer alarm mechanisms and monitoring mechanisms (auditing) in sufficient level.
- ✗ They have not authentication mechanisms at user level. It is less secure than application layer security

Application firewall

The application layer gateways or application gateways are programmed to understand the traffic on the application level of the TCP / IP. They provide access controls on user level and in application-level protocols.

Application gateways adopted in order to eliminate some of the shortcomings that emerged in the implementation of filters in routers. Use of software applications promote and filter connections for services such as HTTP, TELNET and FTP.

Advantages

- ✓ Provide more security and better access control
- ✓ Provide better logging

Disadvantages

- ✗ Difficult implementation.
- ✗ There is no user transparency.
- ✗ The speed along with the performance is unsatisfactory.

Hybrid firewalls (hybrid)

Usually it is a combination of network-layer and application-layer firewall, in order to be able to solve combined problems.

IP TABLES

Iptables is an application program that allows a system administrator to configure the tables provided by the firewall kernel Linux (Kernel) and chains (ip chains). Different modules and the kernel used for various protocols. Ip tables requires elevated privileges to operate and must be performed by the root, otherwise it cannot work properly. On most Linux systems, the ip tables installed on (/ usr / sbin / iptables). It can also be found in (/ sbin / iptables), but since the iptables is more like a service, remains the preferred location (/ usr / sbin).

Functions- operation

Ip tables has 3 built-in tables which are shown below. From those tables the Filter Table is the one used most, as it contains all the firewalling rules we want for our system. Nat Table mainly used for Port Forwarding, where this is appropriate, but also for the SNAT outgoing connections. More rarely the Mangle Table is used, when there is need for QoS.

NAT TABLE

This table should be used for Network address translation (with other words translates the destination address)

FILTER TABLE

Packet Filtering, determines whether packet will be accepted or not depending on where it comes from, what is the destination, how many packages will accept in a given port.

MANGLE TABLE

This particular table can change data of a packet and a number of values found in header. It consists of five sequences shown below

CHAINS

INPUT

Packets arriving inbound in firewall. Located in Filter and Mangle tables and process the incoming connections.

OUTPUT

Packets that are created locally by the firewall located in all tables and process outgoing connections.

FORWARD

Packets reach the firewall but with destination another machine behind firewall. Located in Filter and Mangle tables.

PREROUTING

Packets destined for services Nat / port forwarding. Located in the tables NAT, MANGLE.

POSTROUTING

Packets destined for services Nat / port forwarding. It is about outgoing packets that have already passed through the chains INPUT and FORWARD. Located in Filter and Mangle tables
It should also be emphasized that there are so-called ‘targets” which are used to indicate what action will be taken when a rule matches the packet and also specifies the policies of the chains.

TARGET

ACCEPT
Allows packet to continue in the next chain

DROP
"drop" the packet

LOG
Log the packet to syslog.

REJECT
"drop" the packet and simultaneously sends the appropriate reply message.

RETURN
Continues processing the packet with the so-called chain

MASQUERADE
Translates the hostname (NAT)

The criteria of matching one packet in order to make the action of this rule may be one or more of the following.

Match
DESCRIPTION
--Source (-s)
Matching a source ip address or network.
--destination (-d)
Matching a destination ip address or network.
--protocol (-p)
It corresponds to the packet's protocol to elaborate
--in-interface (-i)
Input interface
--out- interface (-o)
Output interface
--state (-m)
Matching a connection status
--string (-m)
Corresponds to the Application layer Sequence

Packet Travelling

When a packet reaches the Linux Kernel, the first Chain that meets is the PREROUTING CHAIN (Point 1). In this point are the tables Mangle and Nat, where it is highlighted the package for some kind QoS or become DNAT. In point 2, will be the routing of the packet, depending on the destination. If the IP Destination refers to the same Firewall, then will be transferred to INPUT chain (point 3). If the IP Destination refers to a machine that is protected by Firewall, then will be transferred to FORWARD chain (point 4).

The packet in the INPUT chain (point 3), will pass by the rules in tables Mangle and Filter. If in Table Filter exist rule that allows the entrance, then the packet will be forwarded to the Local System. Otherwise it will be DROP or REJECT.

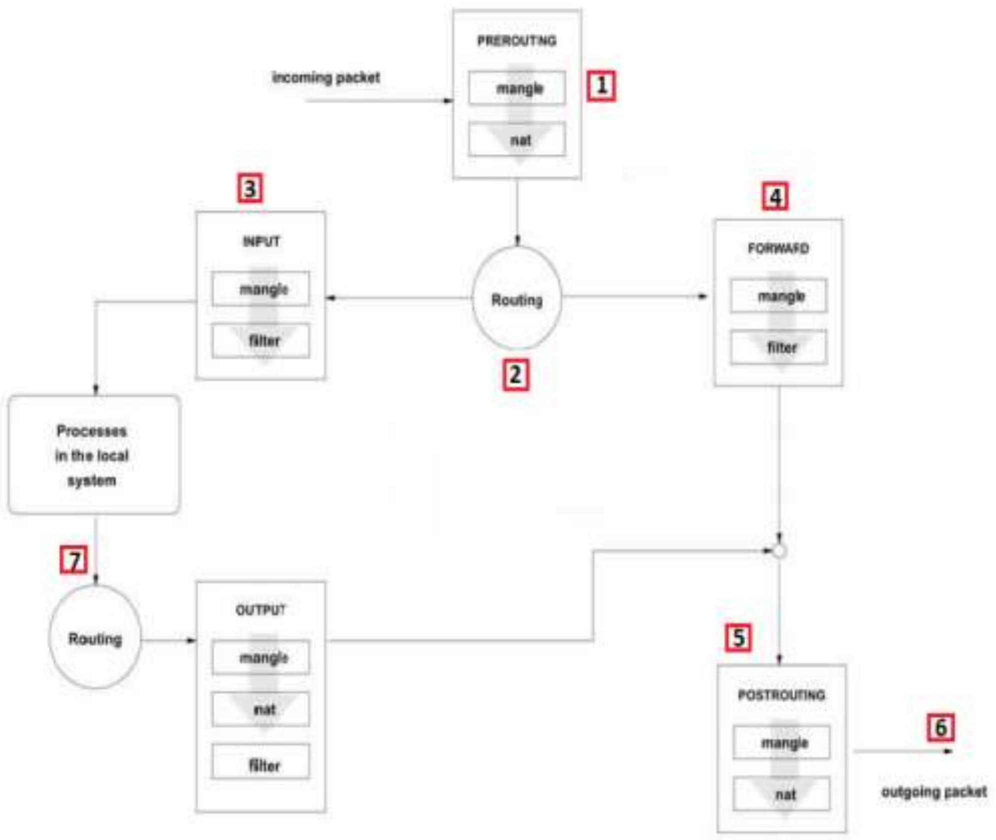


Figure 2: Firewall Chain Routing

The packet in Forward chain (point 4) will go through the rules we have in tables Mangle and Filter. If the Table Filter has no rule that allows entry, the packet will be forwarded to chain Postrouting (point 5). Otherwise it will be DROP or REJECT. In Postrouting chain we have tables Mangle and Nat. In Nat, there might be a rule for SNAT in packet. Finally packet reach in (point 6) and forwarded to its final destination.

If a packet is created from the actual Firewall, then is located at (point 7). Output goes to the chain, where we can have rules in tables, Mangle, Nat, Filter. If the Table Filter has rule that allows outgoing, then the packet will be promoted in Postrouting chain (point 5). Otherwise it will be DROP or REJECT. After Postrouting the packet will start to final destination.

3.2.2 Iptables Firewall Topology and Configuration

The Firewall implementation except than firewalling operationhas the routing operation between the three Interfaces. In order to activate routing operation:

- We enter file "*etc / sysctl.conf*" and change the value of "*net.ipv4.ip_forward*" to **1**.
- Then we give command: "*sysctl -p /etc/sysctl.conf*" in order this change become permanent.
- Finally restart of the Service with command"*service network-manager restart*"

Problems

A major problem to solve is that by default the ip-tables rules are not saved. To solve this problem the first solution was to get in:

- *"/ Etc / network / interfaces*"
- "*sudo nano / etc / network / interfaces*"

and add just below the interfaces the command:

- "*pre-up iptables-restore </etc/iptables.rules*"

This solution didn't work, but alsocreate a problem as the Linux machine could no longer Boot, even without network interfaces.

As a second solution we tried to download the version *"Persistent"sudo apt-get install iptables-persistent*

- We created the file rules.v4 the file "*/ etc / iptables*"

Every time we wanted to save some changes in the rules of ip-tables we gave the command:

- *"root @ ubuntu: / home / admin # iptables-save> /etc/iptables/rules.v4"*

This method worked well.

3.2.2.1 NAT (Network Address Translation)

In order to be able for the external users to access the "companies" webservers some NAT rules have to be created, and match the internal IP's of the servers to external ones that can be accessible from the public WAN network. In this case as we have only one "public" IP (in this case 192.168.1.100) Port forwarding has to be considered as an option for multiple services listening on the same ports.

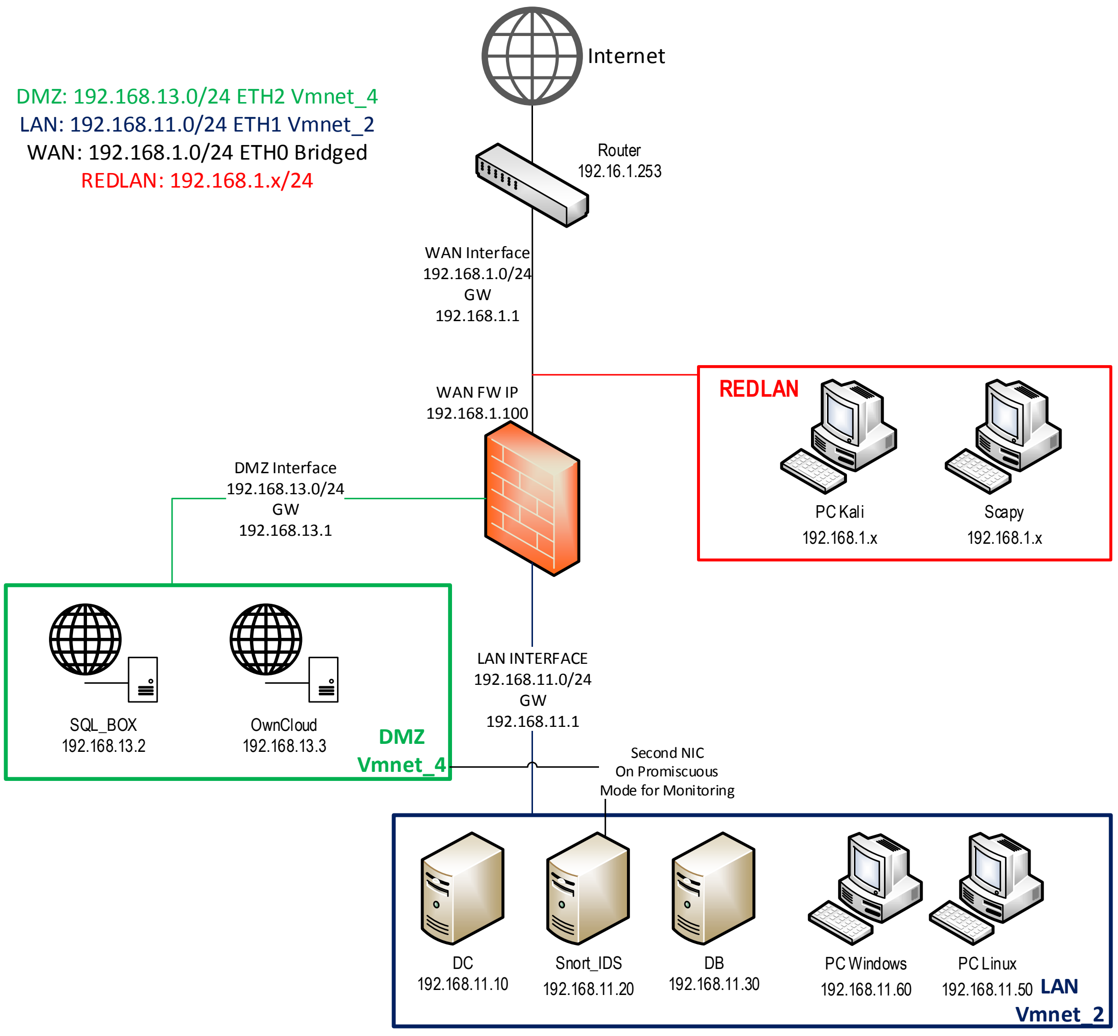


Figure 3: Network Address Translation

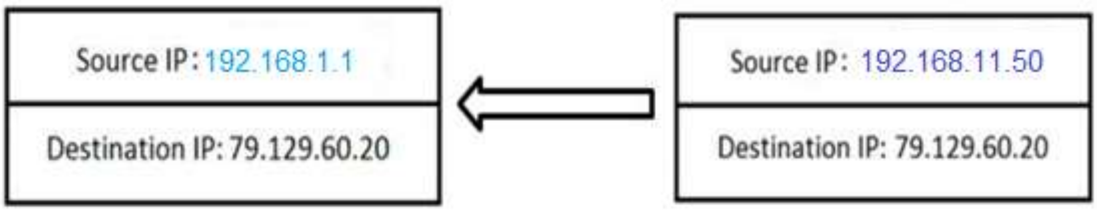


Figure 4: NAT Example

All PC's and Server's located in LAN and DMZ will have different private IPs, and the Firewall will be responsible for Network Address Translation (NAT). When a package starts from an internal network, and comes to the Internet will have the SNAT, which will translate the packet IP Source and gets the IP address of the WAN interface of the firewall (192.168.1.1).

All rules regarding the NAT come into table Nat. There are two ways to achieve the SNAT. The first is masquerade: “*iptables -t nat -A POSTROUTING -o eth0 -j masquerade”*

With this command, any packet exits the WAN interface, change the Source Address, and take Wan Source Address.

The second way is with the command SNAT. In this case, we declare the outset what will replace the IP source address. We chose the second way, as we had static IP WAN and would have faster the process of SNAT.

“iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 192.168.1.1”

For DNAT operation (also known as Port Forward) we should create appropriate rules for all the services we want to allow the WAN to the DMZ. These rules also come into table NAT. Only DNAT is certainly not enough. Then you will need the corresponding firewall rule.

- HTTP OWNCLOUD

iptables -t nat -A PREROUTING -i eth0 -o eth1 -p tcp --dport 8181 -j DNAT --to-destination 192.168.13.3:80

- SQL Port to SQLBOX

iptables -t nat -A PREROUTING -i eth0 -o eth1 -p tcp --dport 8282 -j DNAT --to-destination 192.168.13.2:80

- Remote Desktop to Windows PC

iptables -t nat -A PREROUTING -i eth0 -o eth2 -p tcp --dport 3389 -j DNAT --to-destination 192.168.11.60

- SSH to Ubuntu

iptables -t nat -A PREROUTING -i eth0 -o eth2 -p tcp --dport 22 -j DNAT --to-destination 192.168.11.50

- MySQL access on the DB

iptables -t nat -A PREROUTING -i eth0 -o eth2 -p tcp --dport 3306 -j DNAT --to-destination 192.168.11.30

3.2.2.2 Firewalling

When we implement a Firewall policy to iptables, we should know that is necessary not only to allow the original (New Connection) package to pass through the Firewall, but also to create the appropriate rule to allow RELATED and ESTABLISHED packets or associated with this connection.

We have two choices. In the first, we can state in all Chain that ESTABLISHED or RELATED packets are allowed. e.g. for the three default chains:

```
iptables -A OUTPUT -m state --state RELATED, ESTABLISHED -j ACCEPT
iptables -A INPUT -m state --state RELATED, ESTABLISHED -j ACCEPT
iptables -A FORWARD -m state --state RELATED, ESTABLISHED -j ACCEPT
```

In the second option, we need in every rule that we make, to create the "inverse" rule ESTABLISHED packets.

```
iptables -A OUTPUT -o eth0 -p tcp --dport 80 -m state --state NEW, ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

3.2.3 Level 1 Firewall

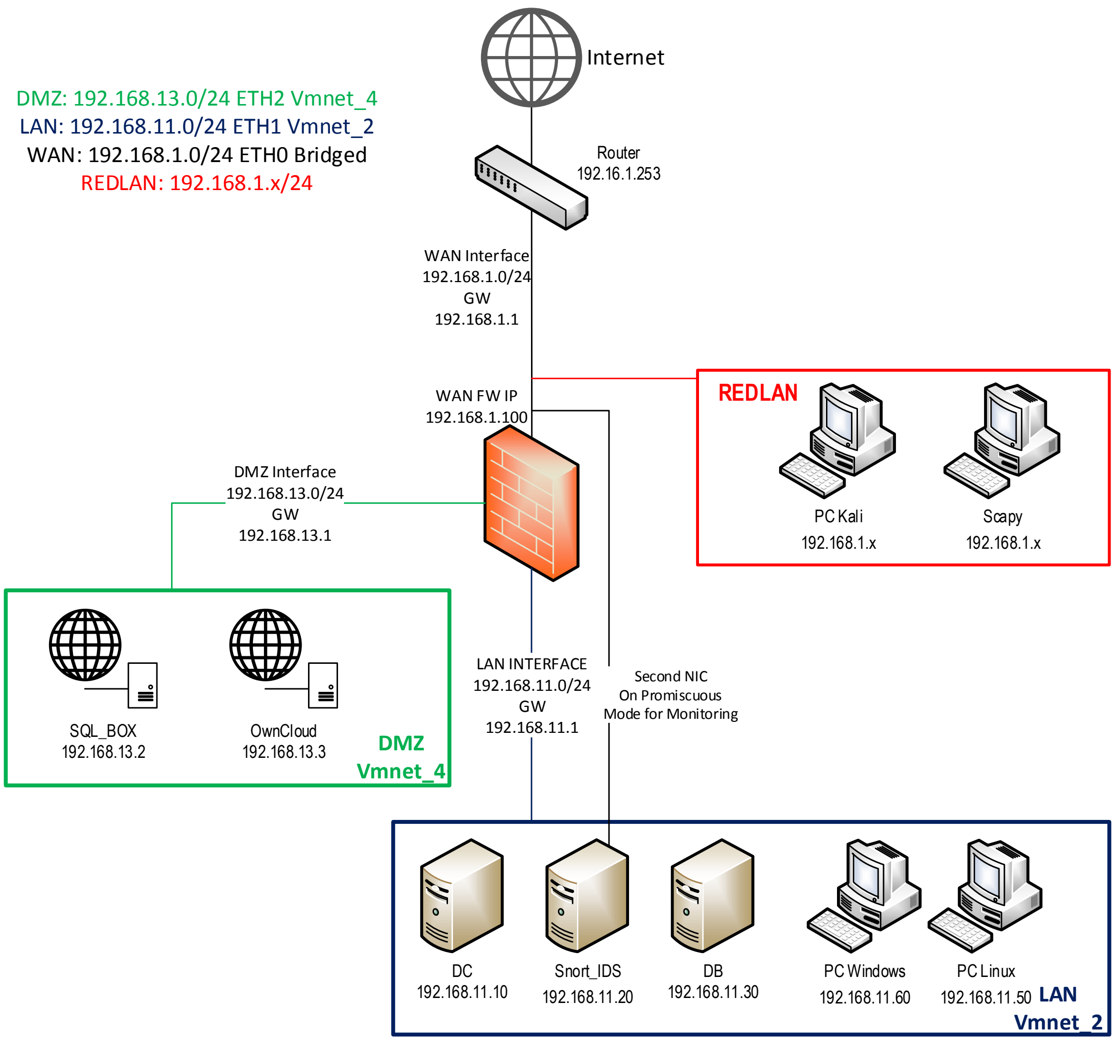


Figure 5: Level 1 Firewall

3.2.3.1 NAT

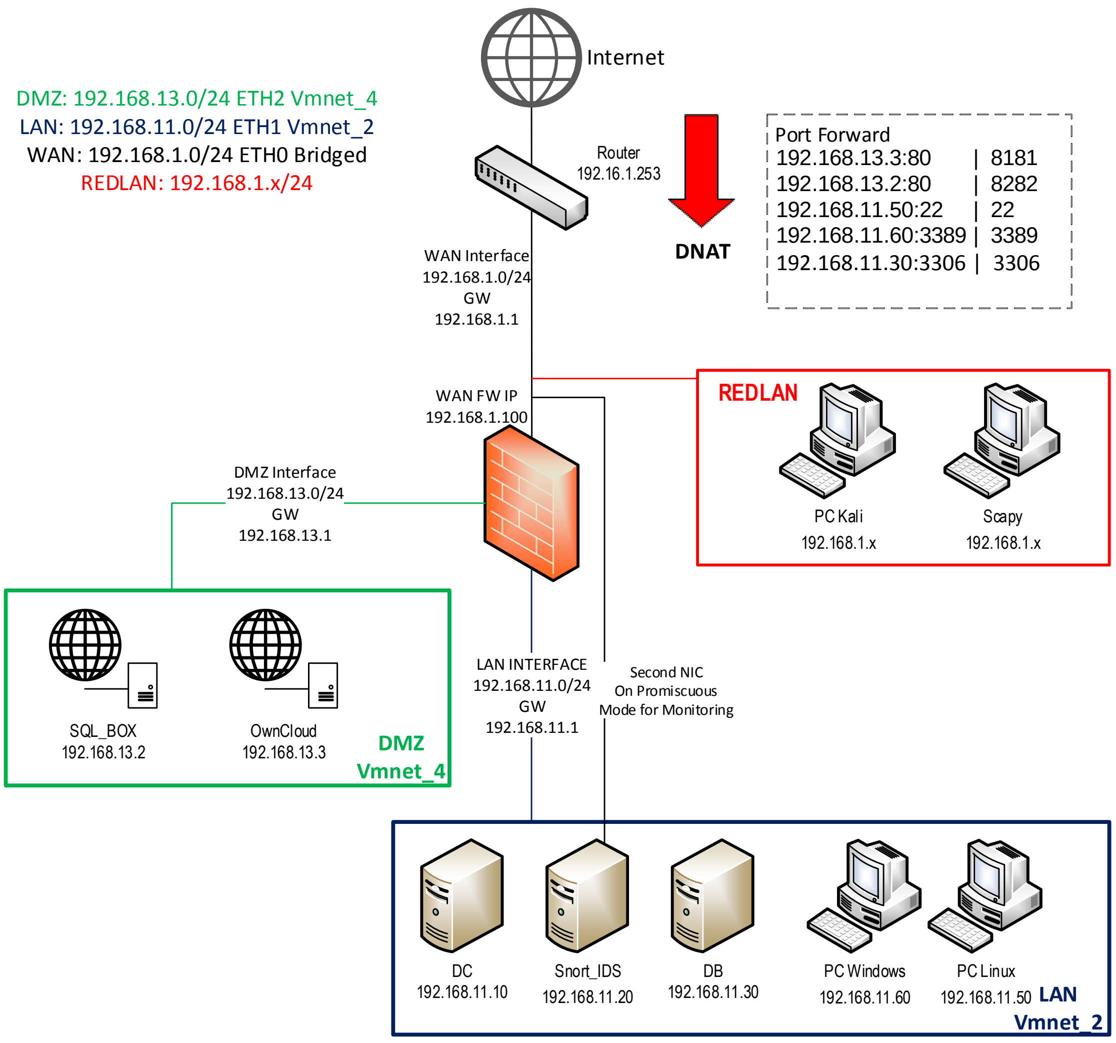


Figure 6: Level 1 Firewall NAT

iptables -t nat -A POSTROUTING -o eth0 -j masquerade

iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 192.168.1.100

iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 8181 -j DNAT --to-destination 192.168.13.3:80

iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 8282 -j DNAT --to-destination 192.168.13.2:80

iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 22 -j DNAT --to-destination *192.168.11.50:22*

iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 3389 -j DNAT --to-destination 192.168.11.60:3389

iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 3306 -j DNAT --to-destination 192.168.11.30:3306

3.2.4 Level 2 Firewall

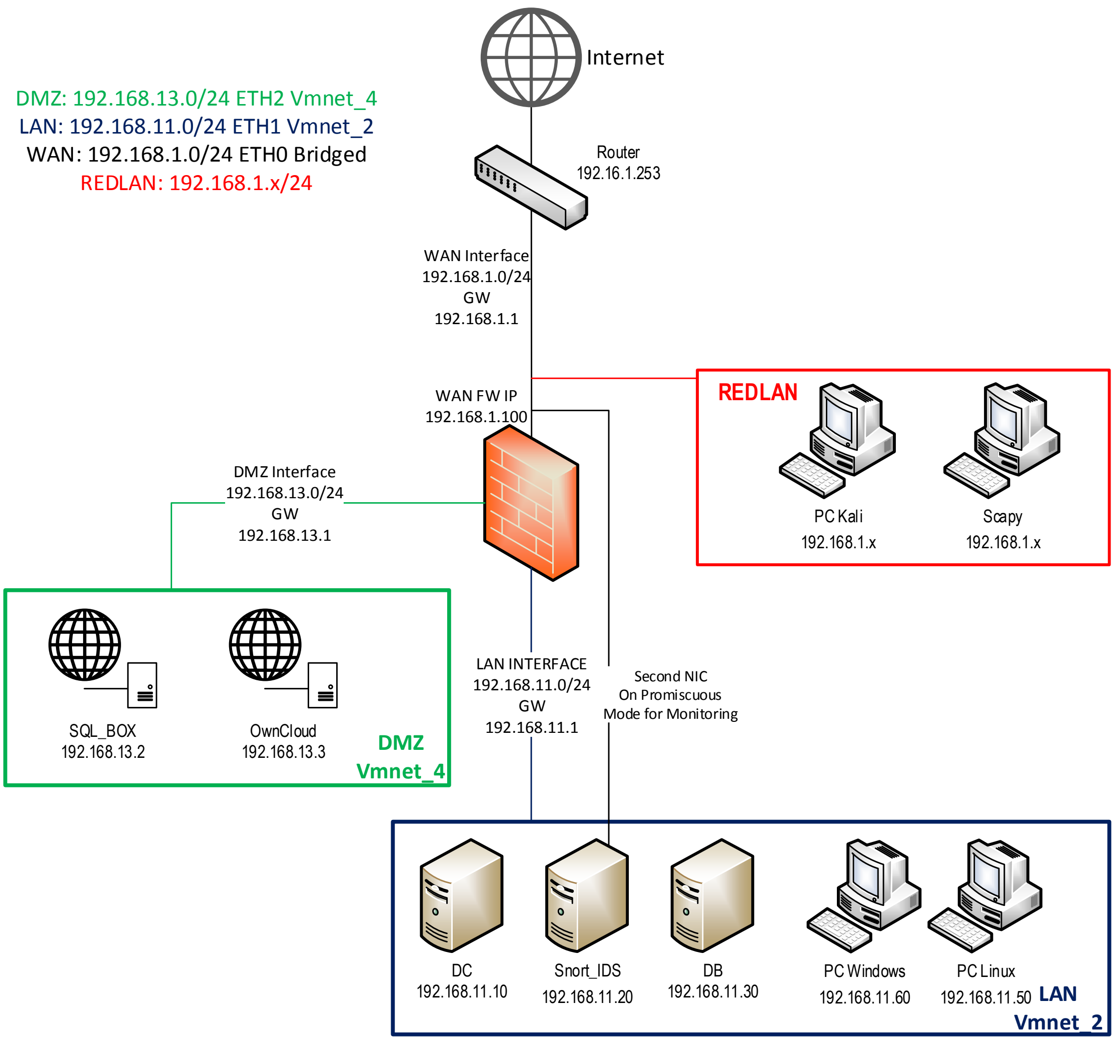


Figure 7: Level 2 Firewall

3.2.4.1 NAT

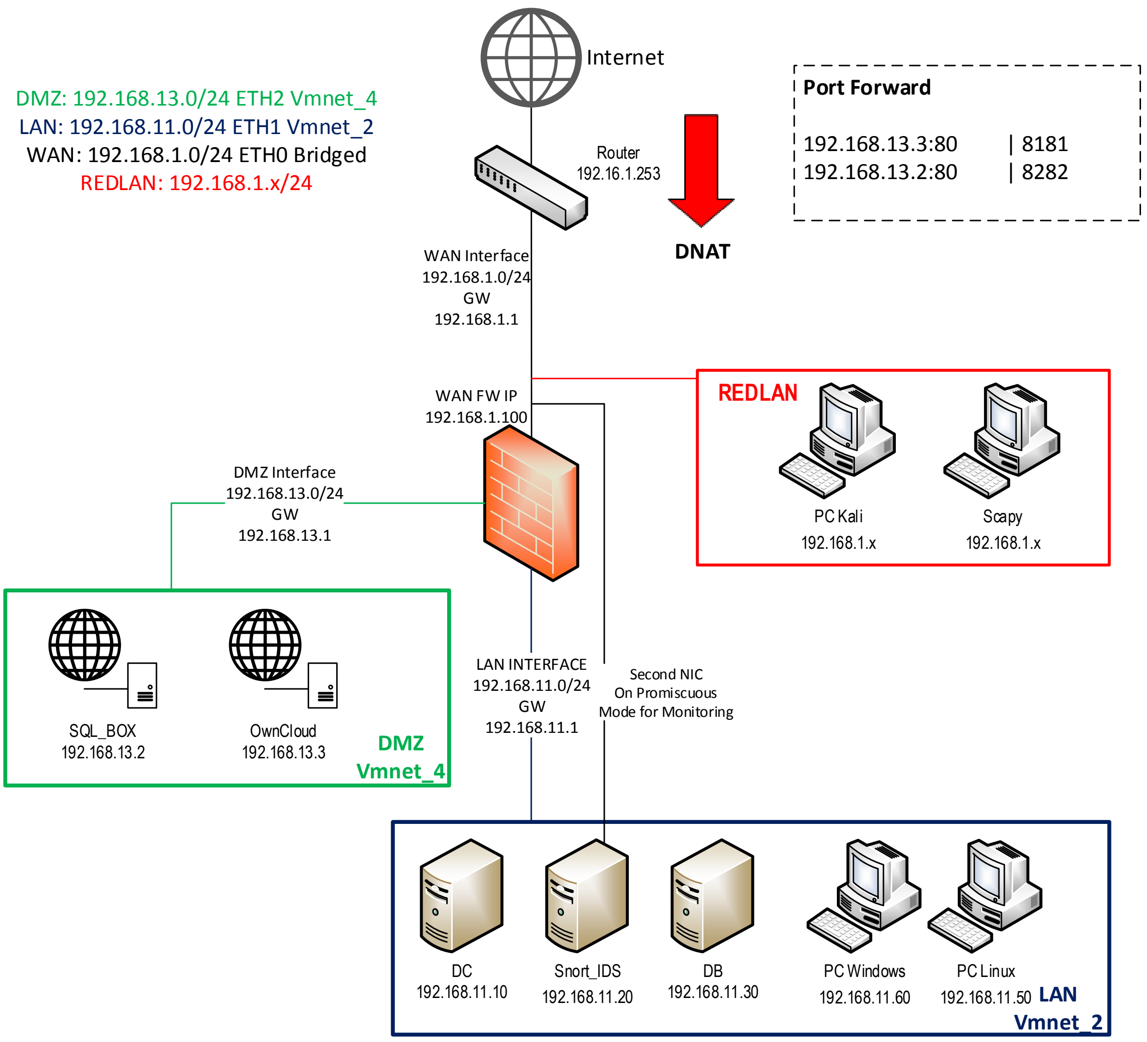


Figure 8: Level 2 Firewall NAT

iptables -t nat -A POSTROUTING -o eth0 -j masquerade

iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 192.168.1.100

iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 8181 -j DNAT --to-destination 192.168.13.3:80

iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 8282 -j DNAT --to-destination 192.168.13.2:80

3.2.4.2 WAN to DMZ Traffic

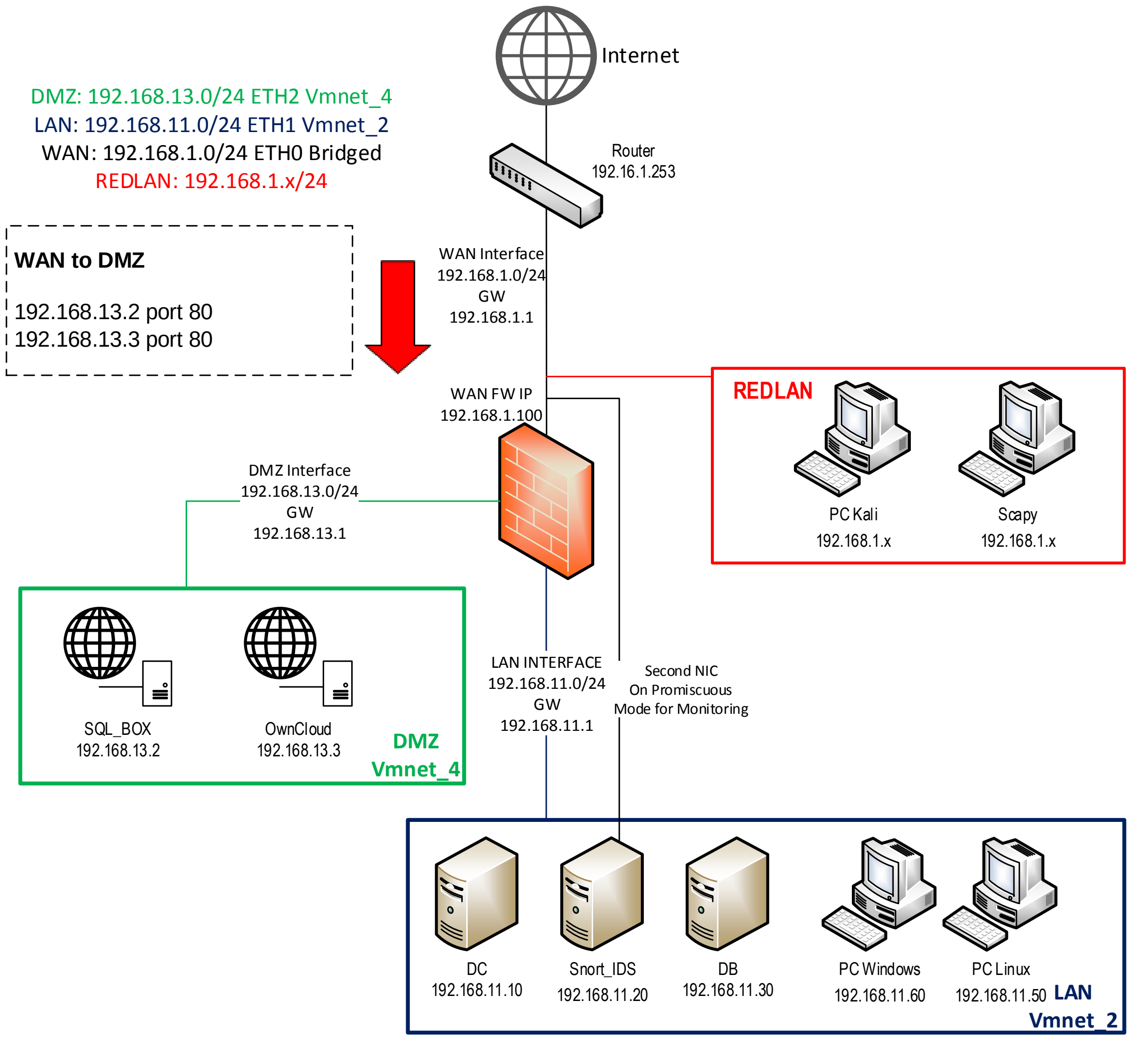


Figure 9: Level 2 Firewall WAN to DMZ

```
iptables -A FORWARD -d 192.168.13.2/32 -p tcp -m tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -s 192.168.13.2/32 -p tcp -m tcp --sport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -d 192.168.13.3/32 -p tcp -m tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -s 192.168.13.3/32 -p tcp -m tcp --sport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
```

3.2.4.3 DMZ to LAN Traffic

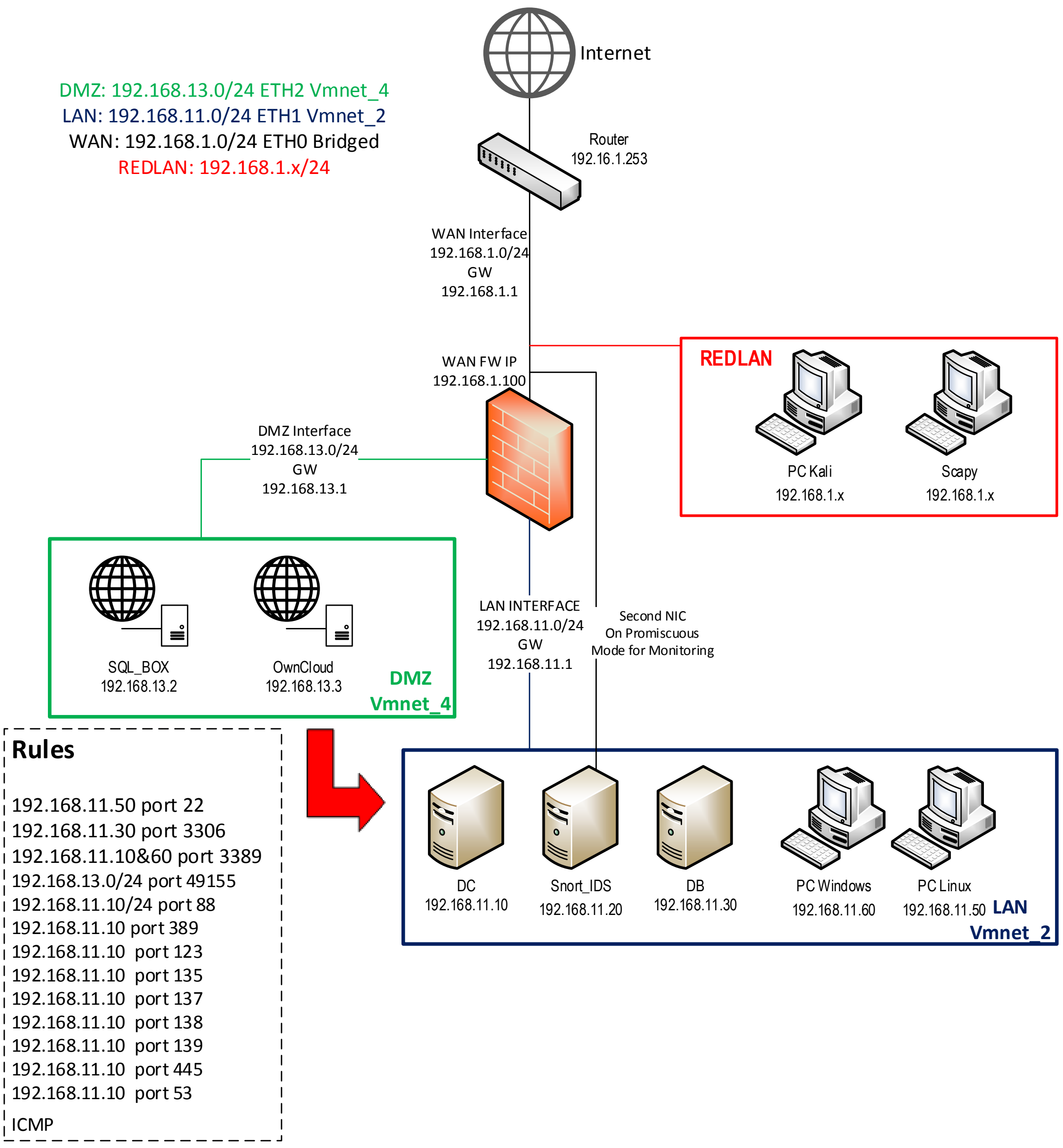


Figure 10: Level 2 Firewall DMZ to LAN

Domain Services

iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p tcp --dport 49155 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p tcp --dport 49155 -m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p tcp --dport 88 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p tcp --dport 88 -m state --state ESTABLISHED -j ACCEPT

SSH

iptables -A FORWARD -d 192.168.11.50/32 -s 192.168.13.2 -p tcp -m tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.11.50/32 -d 192.168.13.2 -p tcp -m tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.11.50/32 -s 192.168.13.3 -p tcp -m tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.11.50/32 -d 192.168.13.3 -p tcp -m tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

RDP to Win7

iptables -A FORWARD -d 192.168.11.60/32 -s 192.168.13.2 -p tcp -m tcp --dport 3389 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.11.60/32 -d 192.168.13.2 -p tcp -m tcp --sport 3389 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.11.10/32 -s 192.168.13.2 -p tcp -m tcp --dport 3389 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.11.10/32 -d 192.168.13.2 -p tcp -m tcp --sport 3389 -m state --state NEW,ESTABLISHED -j ACCEPT

MySQL

iptables -A FORWARD -s 192.168.13.2/32 -d 192.168.11.30 -p tcp -m tcp --dport 3306 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.2/32 -s 192.168.11.30 -p tcp -m tcp --dport 3306 -m state --state ESTABLISHED -j ACCEPT

LDAP

iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p udp --dport 389 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p udp --dport 389 -m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p tcp --dport 389 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p tcp --dport 389 -m state --state ESTABLISHED -j ACCEPT

Client-Server Communication

```
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p udp --dport 135 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p udp --dport 135 -m state --
state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p tcp --dport 135 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p tcp --dport 135 -m state --
state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p tcp --dport 137 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p tcp --dport 137 -m state --
state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p udp --dport 137 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p udp --dport 137 -m state --
state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p tcp --dport 138 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p tcp --dport 138 -m state --
state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p udp --dport 138 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p udp --dport 138 -m state --
state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p tcp --dport 139 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p tcp --dport 139 -m state --
state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p udp --dport 139 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p udp --dport 139 -m state --
state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p udp --dport 445 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p udp --dport 445 -m state --
state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p tcp --dport 445 -m state --
state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p tcp --dport 445 -m state --
state ESTABLISHED -j ACCEPT
```

DNS

```
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p udp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p udp --dport 53 -m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10 -p tcp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10 -p tcp --dport 53 -m state --state ESTABLISHED -j ACCEPT
```

NTP

```
iptables -A FORWARD -s 192.168.13.0/24 -d 192.168.11.10/32 -p udp --dport 123 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 192.168.11.10/32 -p udp --dport 123 -m state --state ESTABLISHED -j ACCEPT
```

ICMP

```
iptables -A FORWARD -i eth2 -o eth1 -p icmp -m icmp --icmp-type 0 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth2 -o eth1 -p icmp -m icmp --icmp-type 8 -m state --state NEW,ESTABLISHED -j ACCEPT
```

3.2.4.4 Internet Access for LAN

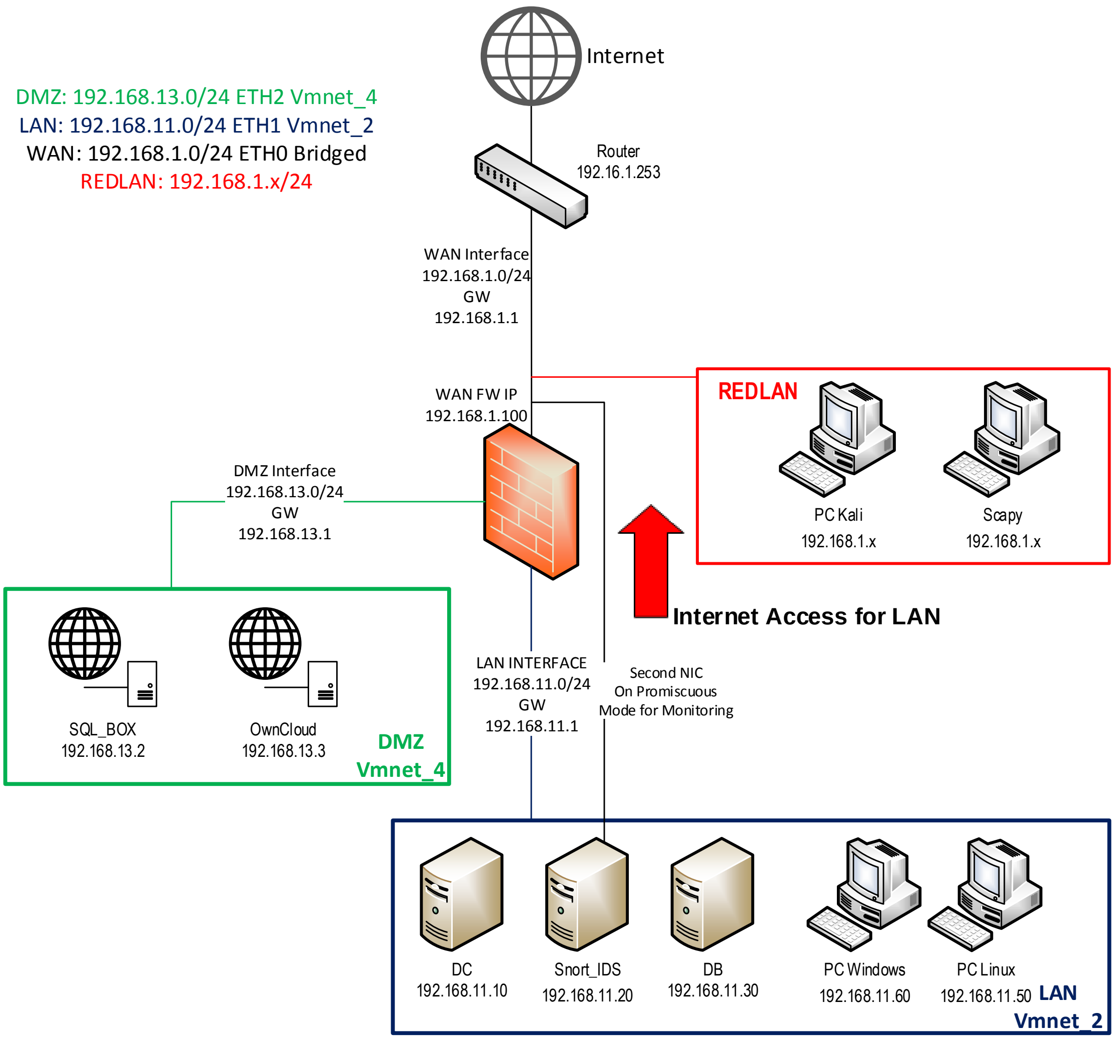


Figure 11: Level 2 Firewall Internet for LAN

```
iptables -A FORWARD -i eth1 -p tcp -j ACCEPT

iptables -A FORWARD -i eth0 -p tcp -m state --state ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth1 -p udp -j ACCEPT

iptables -A FORWARD -i eth0 -p udp -m state --state ESTABLISHED -j ACCEPT
```

3.2.4.5 ICMP ANY to WAN

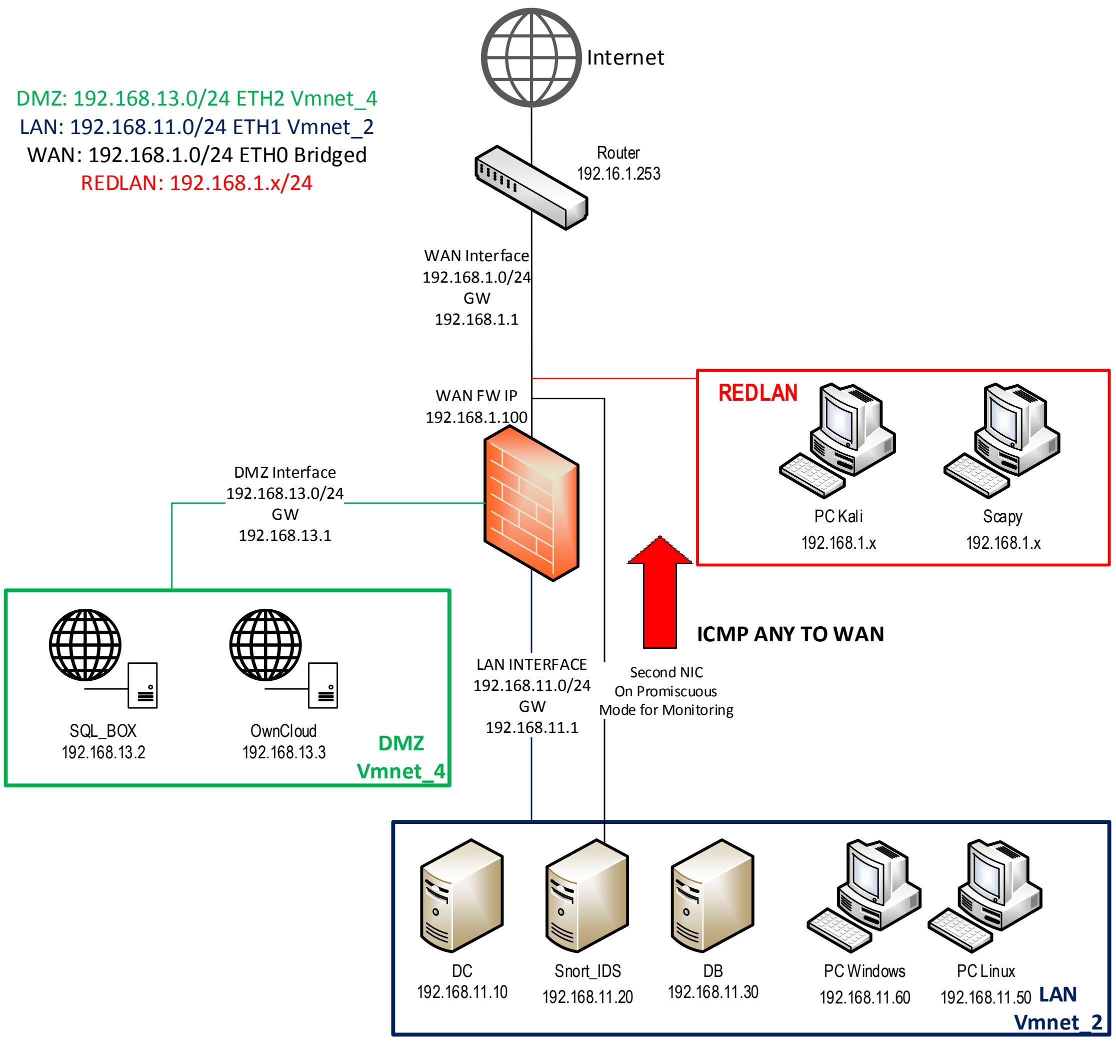


Figure 12: Level 2 Firewall ICMP ANY WAN

iptables -A FORWARD -p icmp -m icmp --icmp-type 0 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -p icmp -m icmp --icmp-type 8 -m state --state NEW,ESTABLISHED -j ACCEPT

3.2.4.6 Internet Access for DMZ

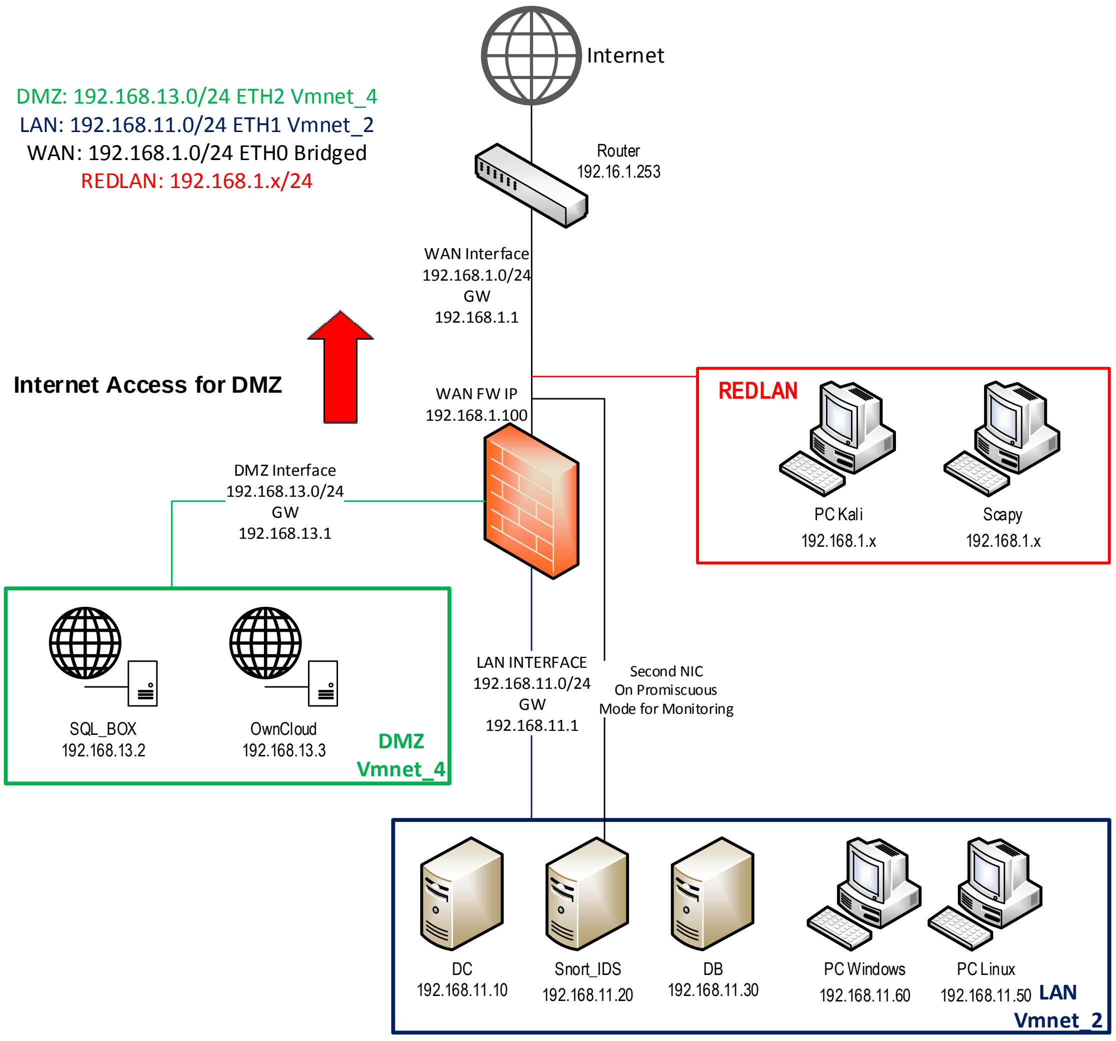


Figure 13: Level 2 Internet Access for DMZ

iptables -A FORWARD -i eth2 -o eth0 -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth0 -o eth2 -p tcp -m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth2 -o eth0 -p udp -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth0 -o eth2 -p udp -m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 8.8.8.8 -p udp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 8.8.8.8 -p udp --dport 53 -m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 192.168.13.0/24 -d 8.8.8.8 -p tcp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.13.0/24 -s 8.8.8.8 -p tcp --dport 53 -m state --state ESTABLISHED -j ACCEPT

3.2.5 Logging

In iptables is possible to log the traffic that passes through the firewall. To achieve this we need to access the file.

- **nano /etc/rsyslog.d/50-default.conf**

and add at the end of the file the following command. This command is normally done by Log "level 4" that warning, will be recorded in the file iptables.log.

- **kern.warning /var/log/iptables.log**

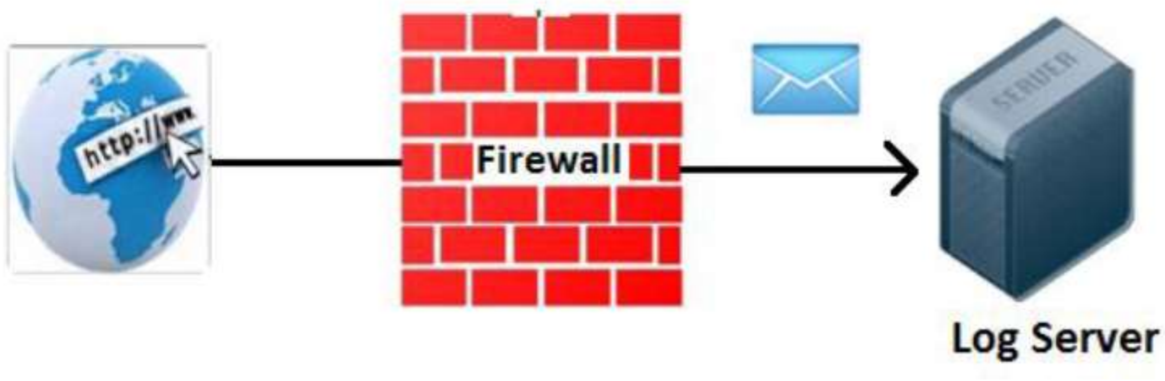


Figure 14: Logging

In order to record all traffic that is in DROP INPUT chain, we use the following commands. They created a new Chain of Logging and transfer all the traffic that should be DROP. This chain makes log movement and then DROPS the packet. If there are several consecutive same Log, they will be recorded only five same per minute.

```
iptables -N LOGGING
iptables -A INPUT -j LOGGING
iptables -A LOGGING -m limit --limit 5/min -j LOG --log-prefix "Input-Dropped: " --log-level 4
iptables-ALOGGING -jDROP
```

If we want to have two levels Logging then we could make some movement with Log "log-level 7" (debug) and send it to the second archeio.p.ch, with the command:

```
iptables -I FORWARD 1 -m limit --limit 15/minute -j LOG --log-level 7 --log-prefix
"FORWARD-Traffic: "
```

3.3 Integrating Snort with Barnyard and Snorby

SNORT

3.3.1 Deploying SNORT with Pulledpork, Barnyard2 and Snorby

With the documented installation steps bellow the following components will be deployed:

- Snort – This is the sensor component it's responsible for monitoring the raw traffic and comparing the traffic to rules.
- PullledPork – This is our rule management application.
- Barnyard2 – This processes the alerts generated by snort and processes them in to a database format.
- Snorby – This is the visual front end to the event data that is written in to the database.

For the deployment we use as user the user "sensor"

Installation Steps:

3.3.1.1 Barnyard2 Install

First we need to configure the SQL database to store the alerts.

Secure the SQL install with

1. mysql_secure_installation

Grab the SQL Schema

1. cd /home/sensor/tmp_build/
2. wget https://raw.githubusercontent.com/firnsy/barnyard2/master/schemas/create_mysql

Create users and tables to access the DB

1. mysql -u root -p
2. create database snort;
3. grant all on snort.* to snort@'localhost' IDENTIFIED BY 'set a strong password here';
4. flush privileges;
5. use snort;
6. source create_mysql;
7. show tables;

With SQL working we need to install barnyard.

1. wget https://github.com/firnsy/barnyard2/archive/v2-1.13.tar.gz
2. tar zxf v2-1.13.tar.gz

3. cd barnyard2-2-1.13/
4. ./autogen.sh
5. ./configure --with-mysql --with-mysql-libraries=/usr/lib/x86_64-linux-gnu
6. make && make install

Barnyard2 – Configure

1. mv /usr/local/etc/barnyard2.conf /etc/snort/
2. nano /etc/snort/barnyard2.conf

we need to tell barnyard how to connect to our new sql database. At the end of the file add the following line:

- output database: log,mysql, user=snort password=yourpasshere dbname=snort host=localhost

To uniquely identify this sensor in the database modify theses two lines as appropriate

- config hostname snort-ids
- config interface eth1

Set some file paths.

1. mkdir /var/log/barnyard2
2. chown snort.snort /var/log/barnyard2
3. touch /var/log/snort/barnyard2.waldo
4. chown snort.snort /var/log/snort/barnyard2.waldo

Once all the changes have been made test barnyard runs correctly with

1. barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.log -w /var/log/snort/barnyard2.waldo -g snort -u snort

Once you see this line " –== Initialization Complete ==–" appear stop barnyard with Ctrl+C and check the sql database to make sure events are being stored.

1. mysql -u snort -p snort

Barnyard2 – Boot

1. nano /etc/init/barnyard2.conf

Add the following lines to the file

- description "Barnyard2 Alerts"
- start on runlevel [2345]
- stop on runlevel [!2345]
- script
- exec barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.log -w /var/log/snort/barnyard2.waldo -g snort -u snort -D
- end script

Set and start the service with

1. chmod +x /etc/init/barnyard2.conf
2. service barnyard2 start

Once started check its still running

1. ps -A | grep barn

Reboot your host and make sure that after the reboot the services for SNORT and BARNYARD2 have been started

3.3.1.2 SNORBY Install

Connect as Root

1. sudo su

Snorby Setup & DB Creation

1. mysql -u root -p
   > create database snorby;
   >grant all on snorby.* to snort@localhost identified by 'snortsqlpassword';
   > flush privileges;
   > quit

Grab snorby and save it in to the web dir.

1. cd /home/sensor/tmp_build
2. wget -O snorby.zip --no-check-certificate https://github.com/Snorby/snorby/archive/master.zip
3. unzip snorby.zip
4. mkdir /var/www/html/snorby
5. mv snorby-master/* /var/www/html/snorby

Now we need to edit a couple of files

1. cd /var/www/html/snorby
2. nano Gemfile

Make the following changes

- gem 'rake', '0.9.2' to gem 'rake', '> 0.9.2'
- after gem 'json', '~> 1.7' add gem 'thin'

under the group(:development) section comment out the gem thin line as show below:

- group(:development) do
- gem "letter_opener"
- # gem 'thin'

Save and Exit

1. nano Gemfile.lock

1. change the line rake (0.9.2) to rake (0.9.2.2)

Install the bundles and set the config files

1. gem install rails bundler passenger
2. bundle install
3. cp config/snorby_config.yml.example config/snorby_config.yml
4. cp config/database.yml.example config/database.yml

Edit the config file

1. nano config/snorby_config.yml

Under production modify the rules section to match. This will allow Snorby to display the full snort rule that triggered the event.

2. rules:
    - "/etc/snort/rules"

Edit the database config file

1. nano config/database.yml

Set the username and password to the values we created when we configured the snorby database

Run With Apache2

1. passenger-install-apache2-module

Follow the onscreen instructions. Once the install has finished scroll through the output and copy the lines similar to those below we will need them:

```
LoadModule passenger_module /var/lib/gems/1.9.1/gems/passenger-4.0.58/buildou$


 PassengerRoot /var/lib/gems/1.9.1/gems/passenger-4.0.58
 PassengerDefaultRuby /usr/bin/ruby1.9.1
```

1. nano /etc/apache2/sites-availiable/snorby.confs

Paste the code we copied above and then add the code below making sure to change the Servername to match your own IP address or hostname

```
Servername <hostname or IP>
DocumentRoot /var/www/html/snorby/public

AllowOverride all
Order allow,deny
Allow from all
Options -MultiViews
```

Enable the site with

1. ln -s /etc/apache2/sites-available/snorby.conf /etc/apache2/sites-enabled/snorby.conf
2. rm /etc/apache2/sites-enabled/000-default.conf

Set file permissions on the Snorby dir

1. chown -R www-data:www-data /var/www/html/snorby

The final thing we need to change is to modify our barnyard install to write events in to Snorby's database. As mentioned earlier you can modify the existing line or you can use both, I Modify the existing line.

1. nano /etc/snort/barnyard2.conf

Edit the output database line to match Snorby's database

3. output database: log,mysql, user=snort password=yourpasswordhere dbname=snorby host=localhost

Stop barnyard if it is already running.

1. service barnyard stop

Run barnyard from the command line to check the new database settings applied without error. This can take several minutes as it did previously depending on how many rules are enabled.

1. barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.log -w /var/log/snort/barnyard2.waldo -g snort -u snort

As before once you see barnyard has successfully initialized stop with Ctrl+C.

Then start Snorby.

1. RAILS_ENV=production bundle exec rake snorby:setup

After that, snorby should be accessible through your Web Browser on port 80

The default credential to access the site are: **snorby@snorby.org:snorby**

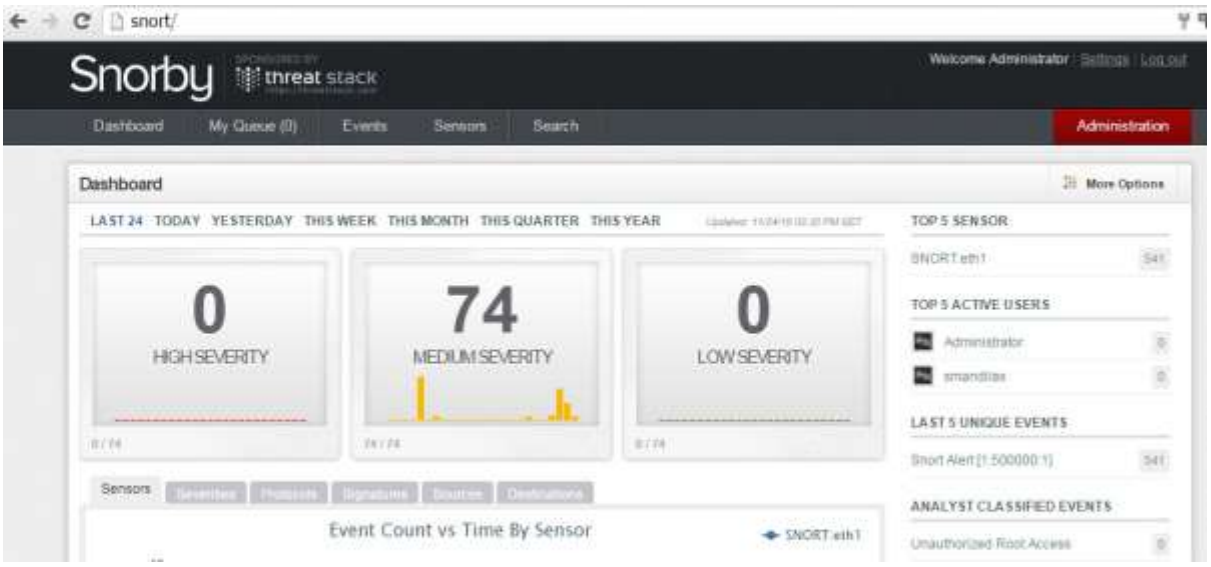


Figure 15: Snorby Administration

3.4 Collaboration Site

Cloud is a crucial component in modern corporate environment. In our master thesis we use cloud in order to have a share point for topology workstations.

We choose for web interface and management (Active directory) Owncloud solution. Owncloud is a self-controlled free and open source cloud. We implement Owncloud in Ubuntu Server 14.04.1 operating system in order to have a fully free software implementation with the advantage of hindsight. Scalability is a characteristic that we want in our topology, for this reason we implement different versions of Owncloud with a variety of vulnerabilities. For example we can see below vulnerabilities and versions that we used in our topology.

Version 6.0.3 June 23rd 2014

SECURITY: Multiple XSS (oC-SA-2014-010)
SECURITY: Improper authorization checks in contacts (oC-SA-2014-011)
SECURITY: Improper authorization checks in files_external (oC-SA-2014-012)
SECURITY: Improper authorization checks in documents (oC-SA-2014-013)
SECURITY: CSRF in documents (oC-SA-2014-014)
SECURITY: Enumeration of shared files in documents (oC-SA-2014-015)
SECURITY: Improper authorization checks in core (oC-SA-2014-016)
SECURITY: Deserialization of Untrusted Data in core (oC-SA-2014-017)

Version 5.0.8 July 10th 2013

SECURITY: XSS vulnerability in "Share Interface" (oC-SA-2013-029)

SECURITY: Authentication bypass in "user_webdavauth" (oC-SA-2013-030)

Version 5.0.6 May 14th 2013
SECURITY: SQL Injection (oC-SA-2013-019)
SECURITY: Multiple directory traversals (oC-SA-2013-020)
SECURITY: Multiple XSS vulnerabilities (oC-SA-2013-021)
SECURITY: Open redirector (oC-SA-2013-022)
SECURITY: Password auto completion (oC-SA-2013-023)
SECURITY: Privilege escalation in the calendar application (oC-SA-2013-024)
SECURITY: Privilege escalation and CSRF in the API (oC-SA-2013-025)
SECURITY: Incomplete blacklist vulnerability (oC-SA-2013-026)
SECURITY: Information disclosure: CSRF token + username (oC-SA-2013-027)

Virtualization of the system takes place with VMware Workstation. Below follows the installation of each component of cloud system.

3.4.1 Ubuntu server Installation via VMware Workstation

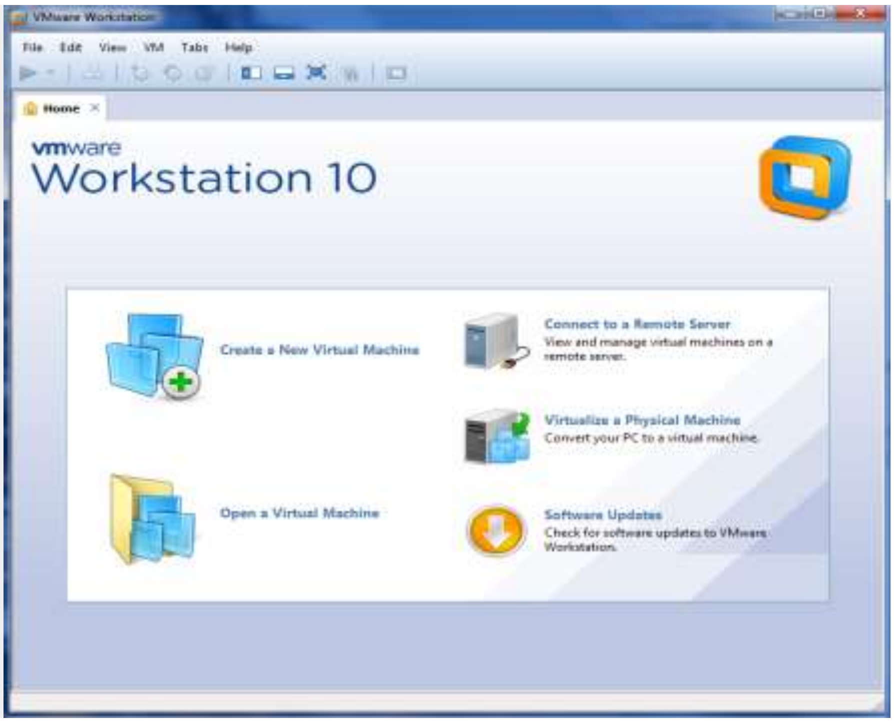
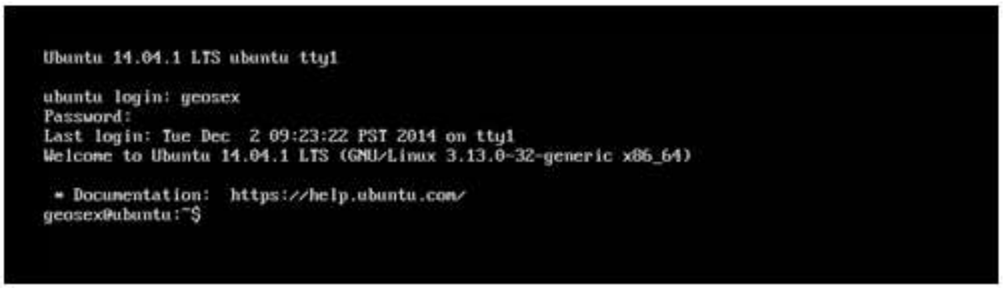

VMware Workstation
File Edit View VM Tabs Help
Home
vmware Workstation 10
Create a New Virtual Machine
Connect to a Remote Server
View and manage virtual machines on a remote server.
Virtualize a Physical Machine
Convert your PC to a virtual machine.
Open a Virtual Machine
Software Updates
Check for software updates to VMware Workstation.

Figure 16: VMware Workstation

3.4.2 Owncloud Installation in Ubuntu Server

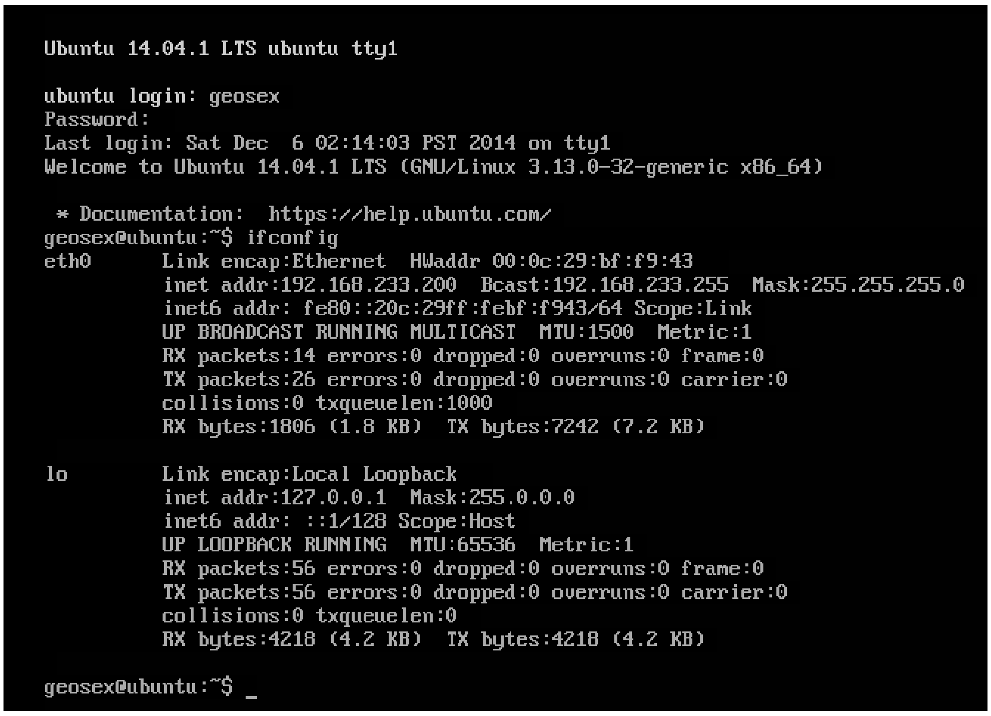


```
Ubuntu 14.04.1 LTS ubuntu tty1

ubuntu login: geosex
Password:
Last login: Tue Dec  2 09:23:22 PST 2014 on tty1
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
geosex@ubuntu:~$
```

Figure 17: OwnCloud Ubuntu Server 14.4.1

- Check your ip's

➢ ***Ifconfig***



```
Ubuntu 14.04.1 LTS ubuntu tty1

ubuntu login: geosex
Password:
Last login: Sat Dec  6 02:14:03 PST 2014 on tty1
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
geosex@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:bf:f9:43
          inet addr:192.168.233.200  Bcast:192.168.233.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:febf:f943/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1806 (1.8 KB)  TX bytes:7242 (7.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:56 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4218 (4.2 KB)  TX bytes:4218 (4.2 KB)

geosex@ubuntu:~$ _
```

Figure 18: Ifconfig in Ubuntu Server

- Make all updates-upgrades

  - ***sudo apt-get update***
  - ***sudo apt-get upgrade***

- Install Apache web server on your Ubuntu 14.04 VPS if it is not already installed

- ***sudo apt-get install apache2***

- Install PHP on your server

- ***sudo apt-get install php5 php5-mysql***

- And the following PHP modules required by Owncloud

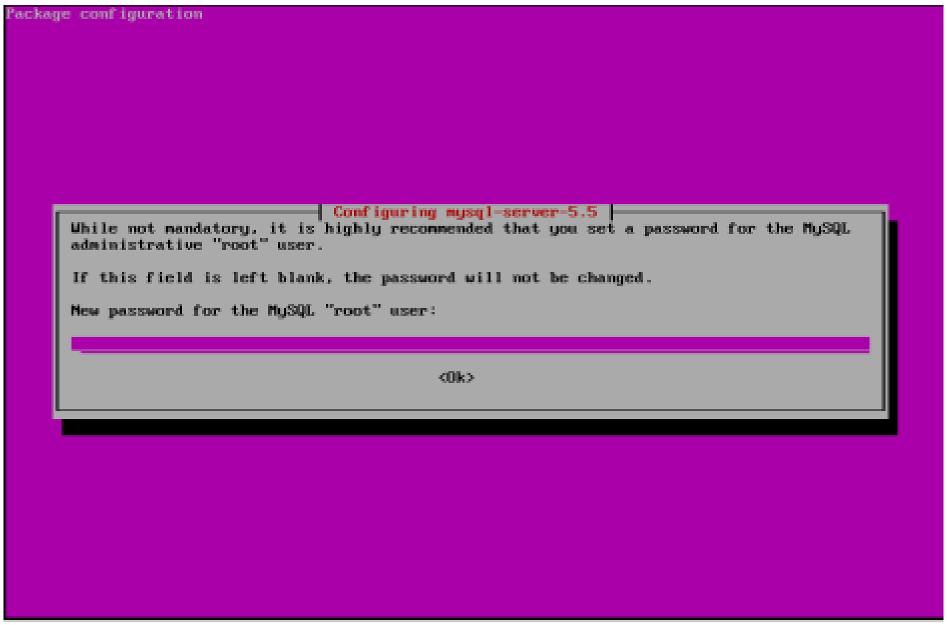
- ***sudo apt-get install php5-gd php5-json php5-curl php5-intl php5-mcrypt php5-imagick***

- Install MySQL database server

- ***sudo apt-get install mysql-server***

- setting of SQL password

  “YOURPASSWORD”



```
Package configuration

                 ┤ Configuring mysql-server-5.5 ├
 While not mandatory, it is highly recommended that you set a password for the MySQL
 administrative "root" user.

 If this field is left blank, the password will not be changed.

 New password for the MySQL "root" user:

                                   <Ok>
```

Figure 19: SQL password

- After MySQL server is installed, run the 'mysql_secure_installation' and use the following settings

- ***Set** root password? [Y/n] **y***
- ***Remove** anonymous users? [Y/n] **y***
- ***Disallow** root login remotely? [Y/n] **y***
- ***Remove** test database and access to it? [Y/n] **y***
- ***Reload** privilege tables now? [Y/n] **y***

- Download the release of your choice of Owncloud on your server.

- wget https://download.owncloud.org/community/owncloud-7.0.0.tar.bz2

- Unpack the downloaded archive to the Apache's document root directory

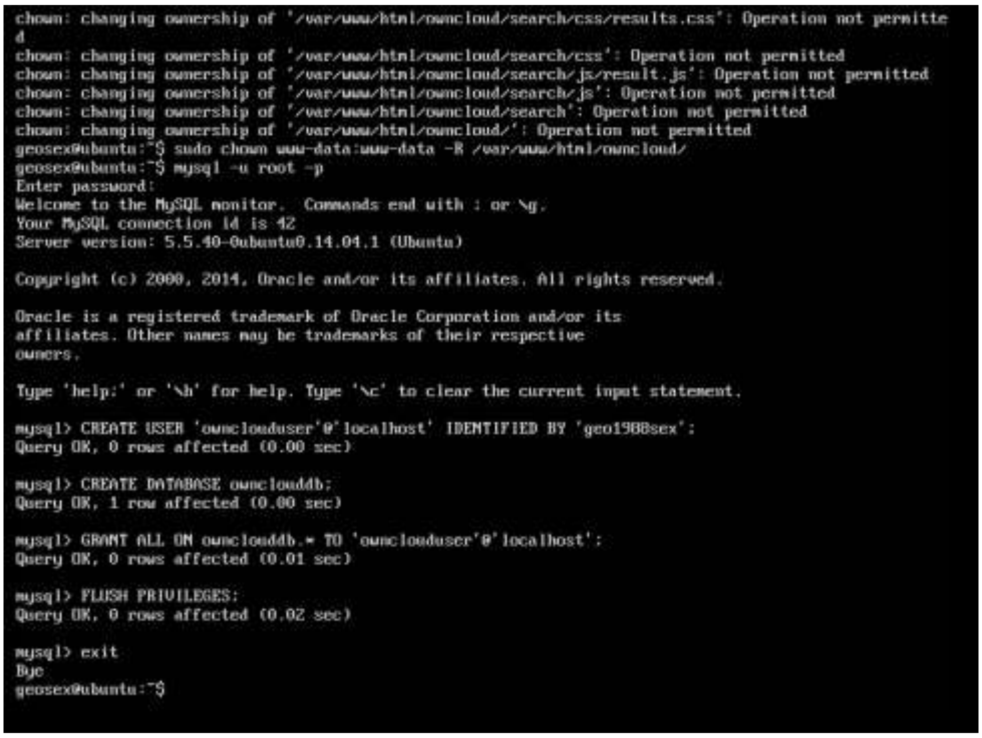
- sudo tar -xvf owncloud-7.0.0.tar.bz2 -C /var/www/html

- The user running the web server have to be the owner of the Owncloud files, so we will change the ownership

- sudo chown www-data:www-data -R /var/www/html/owncloud/

- Log in to your MySQL server and create a user and database for OwnCloud.

- mysql -u root -p
- Enter password:
- mysql> CREATE USER 'ownclouduser'@'localhost' IDENTIFIED BY 'YOURPASSWORD';
- mysql> CREATE DATABASE ownclouddb;
- mysql> GRANT ALL ON ownclouddb.* TO 'ownclouduser'@'localhost';
- mysql> FLUSH PRIVILEGES;
- mysql> exit



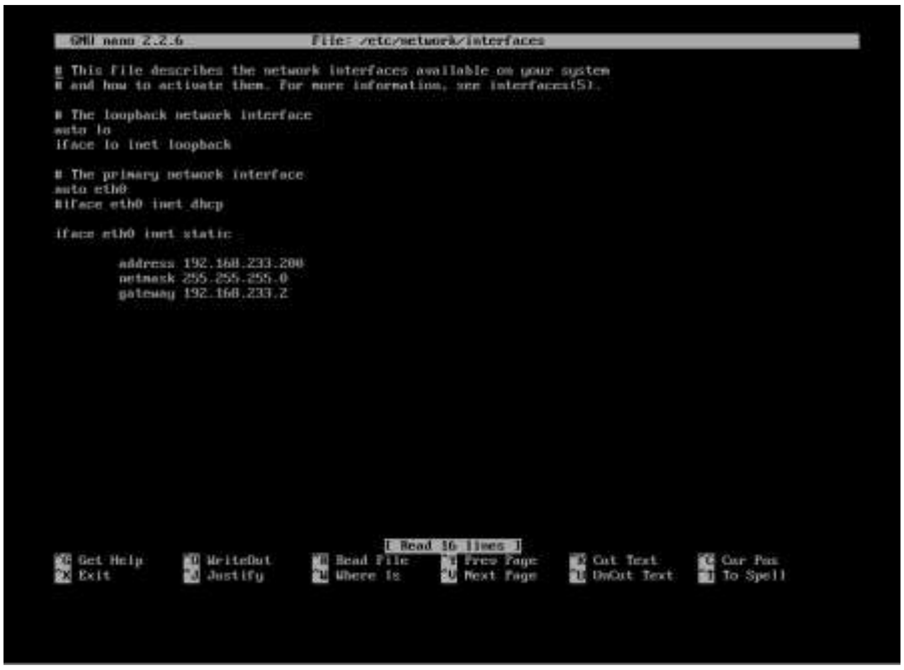
```
chown: changing ownership of '/var/www/html/owncloud/search/css/results.css': Operation not permitte
d
chown: changing ownership of '/var/www/html/owncloud/search/css': Operation not permitted
chown: changing ownership of '/var/www/html/owncloud/search/js/result.js': Operation not permitted
chown: changing ownership of '/var/www/html/owncloud/search/js': Operation not permitted
chown: changing ownership of '/var/www/html/owncloud/search': Operation not permitted
chown: changing ownership of '/var/www/html/owncloud/': Operation not permitted
geosex@ubuntu:~$ sudo chown www-data:www-data -R /var/www/html/owncloud/
geosex@ubuntu:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'ownclouduser'@'localhost' IDENTIFIED BY 'geo1988sex';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE DATABASE ownclouddb;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL ON ownclouddb.* TO 'ownclouduser'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> exit
Bye
geosex@ubuntu:~$
```

Figure 20: SQL create user

Don’t forget to change ‘yourpassword‘with an actual strong password.

- Make the necessary changes in /etc/network/interfaces and /etc/resolv.conf

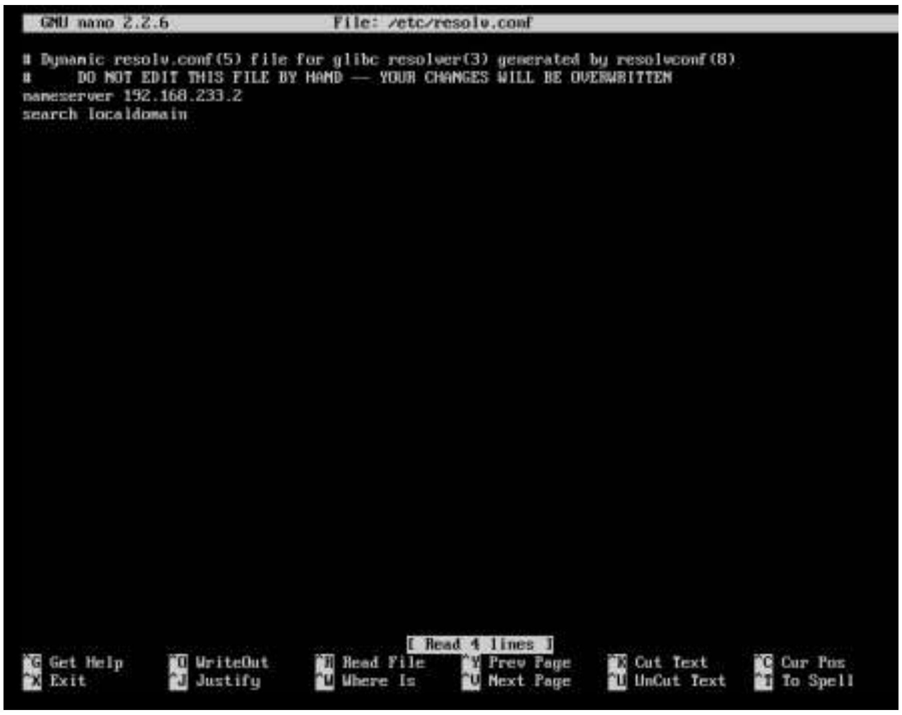
- ***sudo nano /etc/network/interfaces***
- ***iface eth0 inet static***
  - address 192.168.13.3
  - netmask 255.255.255.0
  - gateway 192.168.1.1



```
  GNU nano 2.2.6                File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
#iface eth0 inet dhcp

iface eth0 inet static

        address 192.168.233.200
        netmask 255.255.255.0
        gateway 192.168.233.2

                              [ Read 16 lines ]
^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

Figure 21: network interfaces

- ***sudo nano /etc/resolv.conf***
- ***nameserver 192.168.233.2***



```
  GNU nano 2.2.6                File: /etc/resolv.conf

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.233.2
search localdomain

                              [ Read 4 lines ]
^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

Figure 22: localhost settings

- Restarting the interfaces
  - ***/etc/init.d/networking restart***

- Reboot Ubuntu Server in order changes load in network interfaces.
  - ***sudo reboot***

- Navigate your browser to http://YOURDOMAIN.TLD/owncloud.

There you will create a new administrator user and enter the information of the MySQL database we created earlier in this tutorial.

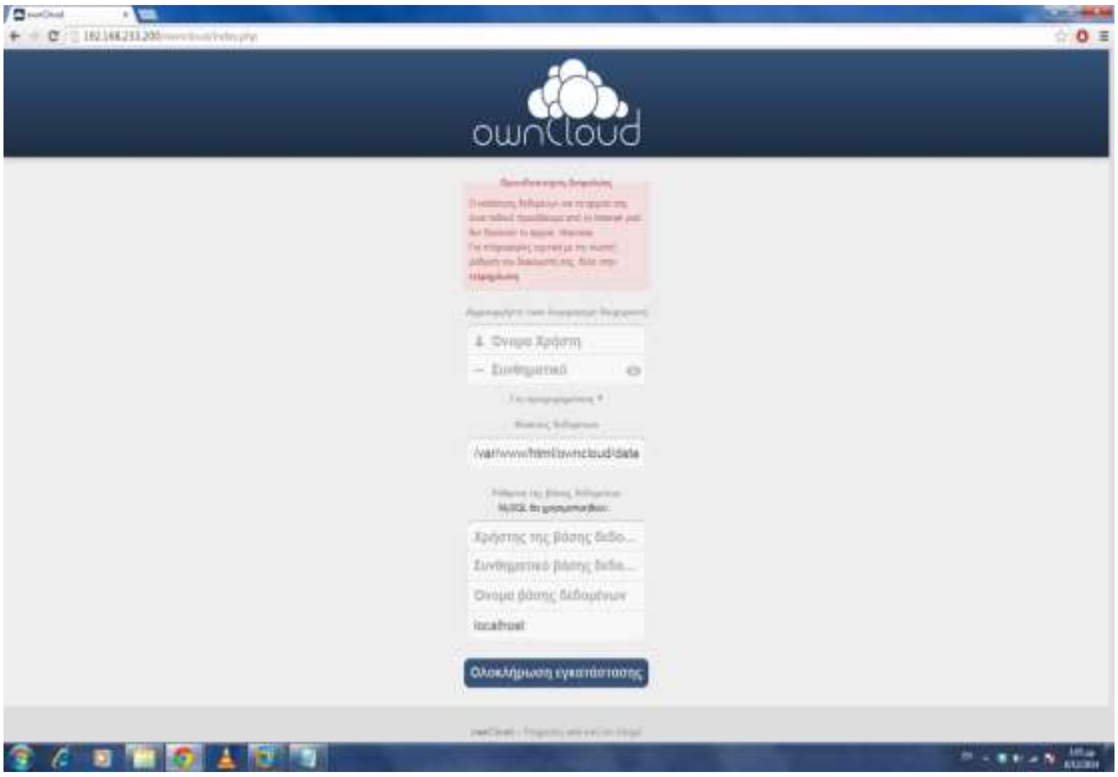


Figure 23: Owncloud Web Interface image 1

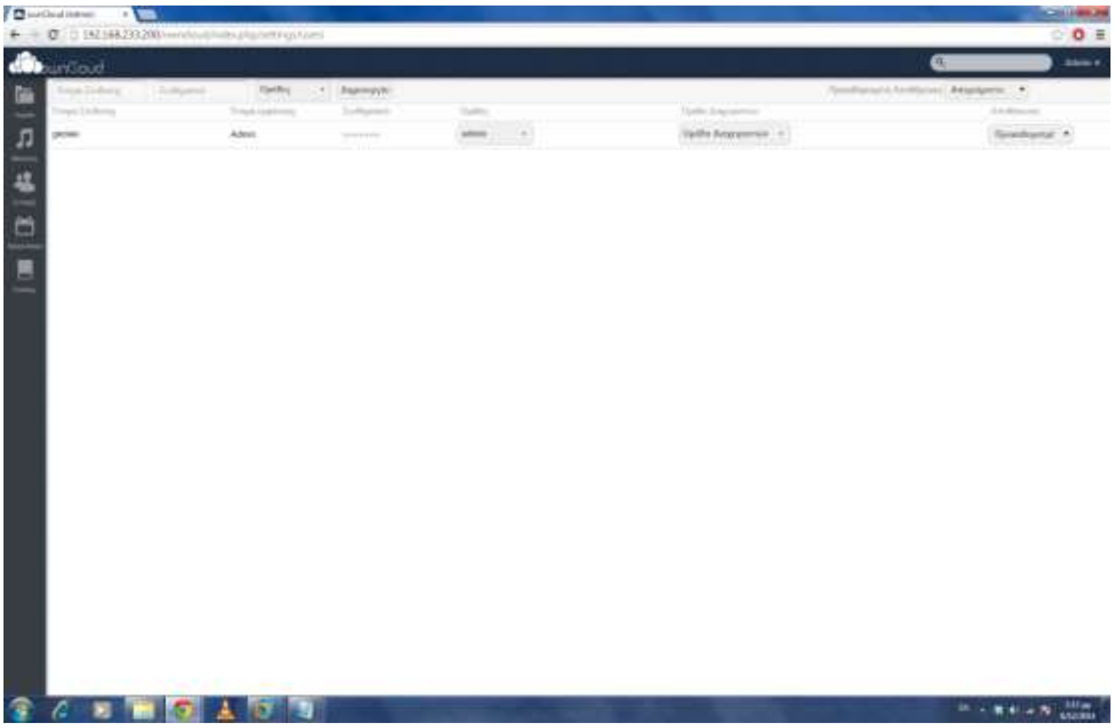


Figure 24: Owncloud Web Interface image 2

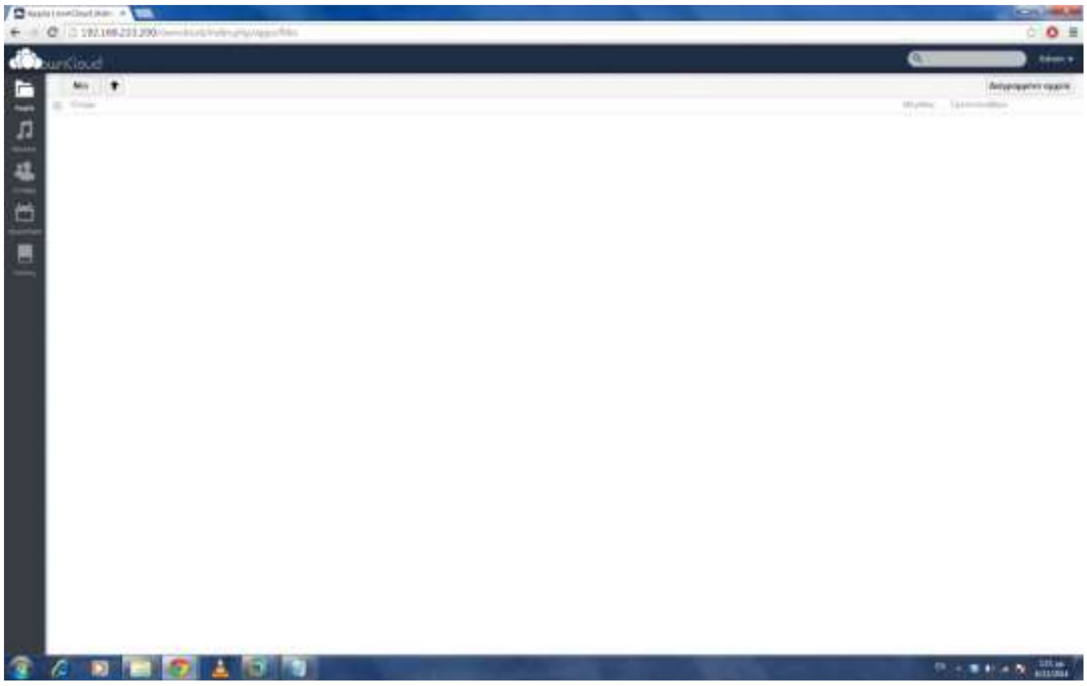


Figure 25: Owncloud Web Interface image 3

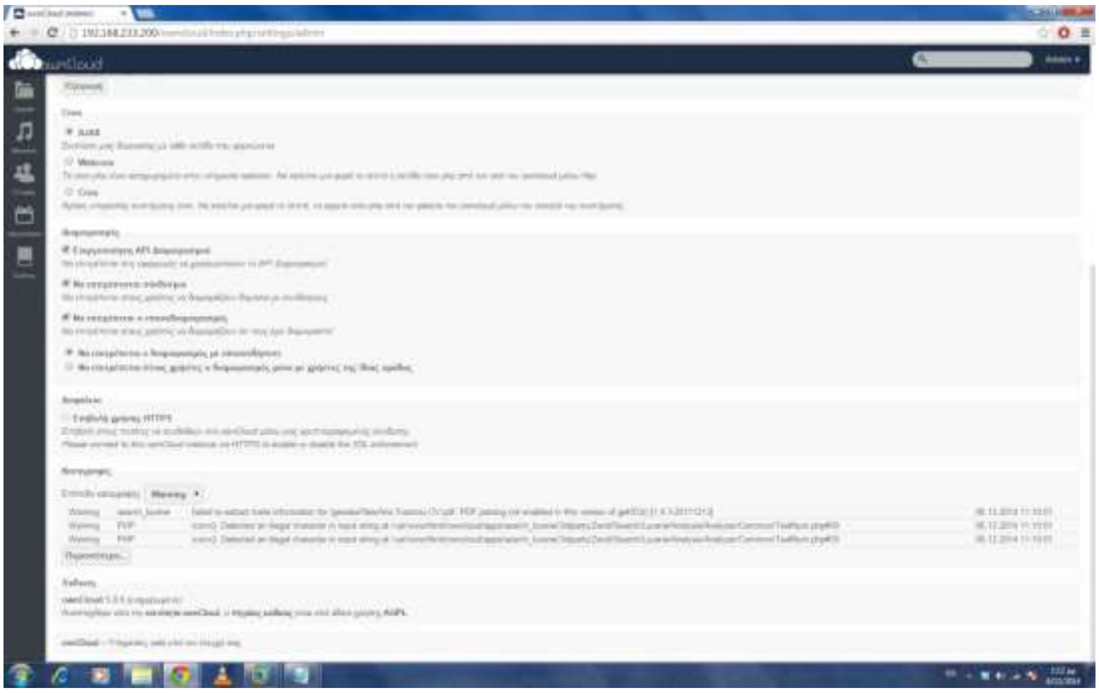


Figure 26: Owncloud Web Interface image 4

3.5 Scapy

Scapy is a powerful interactive packet manipulation program. It is able to forge or decode packets of a wide number of protocols, send them on the wire, capture them, match requests and replies. It can easily handle most classical tasks like scanning, trace routing, probing, unit tests, attacks or network discovery.

We use Scapy in our topology in order to simulate traffic. It's crucial to simulate traffic in order to have more realistic scenarios. Below we can find guideline for Scapy installation and scripting and also the scripts that we use to simulate traffic in topology.

3.5.1 Installing Scapy

```
sudo apt-get install python
sudo apt-get install python-scapy

$ cd /tmp
$ wget scapy.net
$unzip scapy-latest.zip
$ cd scapy-2.*
$ sudo python setup.py install

sudo apt-get update
sudo apt-get install python-scapy python-pyx python-gnuplot
```

To run Scapy interactively

```
sudo scapy
```

The scapy shell will be displayed :

```
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>>
```

To quit Scapy

```
>>>quit()
```

3.5.2 Scripting in Scapy

“3 way handshake”

```
 #! /usr/bin/env python
from scapy.all import *
import random
sp=random.randint(1024,65535)

ip=IP(src="10.0.0.234/29",dst="www.google.com")
 SYN=TCP(sport=sp, dport=80,flags="S",seq=10)
 SYNACK=sr1(ip/SYN)

 my_ack=SYNACK.seq+1
 ACK=TCP(sport=sp,dport=80,flags="A",seq=11,ack=my_ack)
send(ip/ACK)
payload="SEND TCP"

 PUSH=TCP(sport=sp,dport=80,flags="PA",seq=11,ack=my_ack)
send(ip/PUSH/payload)
```

sending icmp

```
#! /usr/bin/env python
from scapy.all import send, IP, ICMP
```

send(IP(src="10.0.99.100",dst="10.1.99.100")/ICMP()/"good traffic")

sending icmp in random destination

#! /usr/bin/env python

from scapy.all import send, IP, ICMP

import random

***sp**=random.randint(1024,65535)*

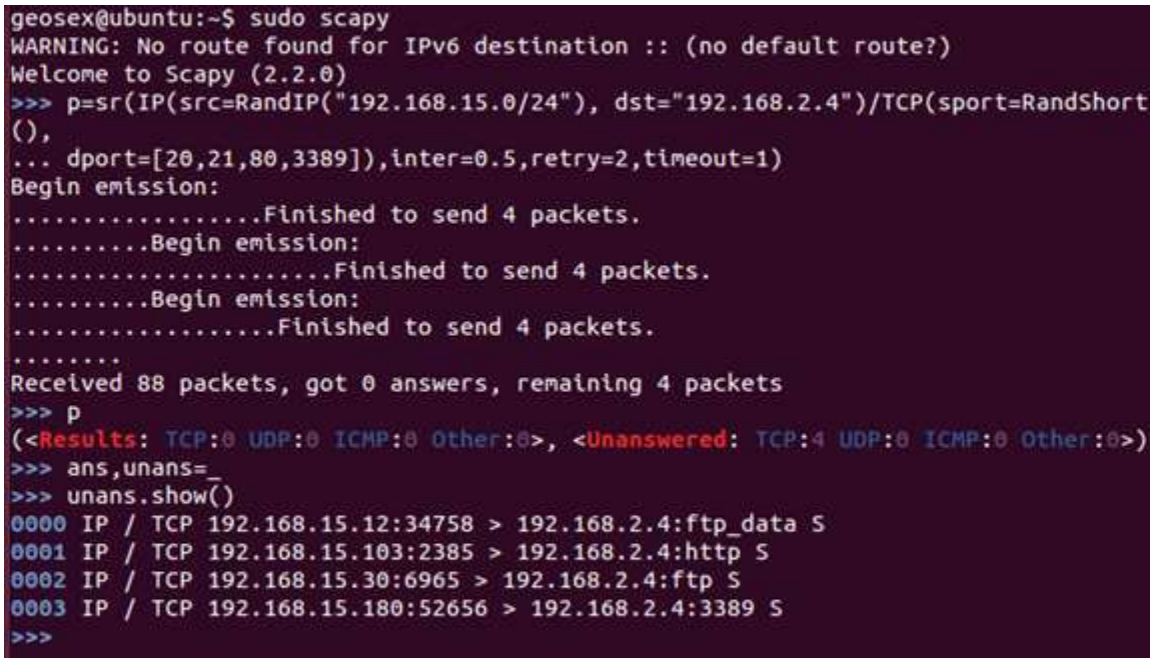
send(IP(src="10.0.99.100",dst=sp)/ICMP()/"good traffic")

h=sr1(IP(dst="192.168.2.1")/**ICMP**())

p=sr(IP(dst="192.168.2.1")/**TCP**(dport=23))
p – This time I've called my packet p (no real reason, just fancied a change)
sr– We changed to the sr() function so we can see the unanswered packets as well
/TCP – Yes that's right we moved onto TCP packets instead of ICMP
dport=23 - A TCP packet needs a destination, so you can use dport to specify one and I've chosenport 23 (Telnet) .

p=sr(IP(dst="192.168.2.1")/TCP(dport=[**23,80,53**]))

sr1(IP(dst="192.168.2.1")/UDP()/DNS(rd=1,qd=DNSQR(qname="www…..com")))
sr1 - This is the send & receives function that only returns the first answered packet
(IP(dst="10.1.99.2") - Again we are using an IP packet and the destination IP
/UDP() - DNS is a UDP packet (port 53)
/DNS - This is telling Scapy that we want to create a DNS packet.
rd=1 - Is telling Scapy that recursion is desired
qd=DNSQR(qname="www…..com")



```
geosex@ubuntu:~$ sudo scapy
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> p=sr(IP(src=RandIP("192.168.15.0/24"), dst="192.168.2.4")/TCP(sport=RandShort
(),
... dport=[20,21,80,3389]),inter=0.5,retry=2,timeout=1)
Begin emission:
..................Finished to send 4 packets.
..........Begin emission:
.......................Finished to send 4 packets.
..........Begin emission:
...................Finished to send 4 packets.
........
Received 88 packets, got 0 answers, remaining 4 packets
>>> p
(<Results: TCP:0 UDP:0 ICMP:0 Other:0>, <Unanswered: TCP:4 UDP:0 ICMP:0 Other:0>)
>>> ans,unans=_
>>> unans.show()
0000 IP / TCP 192.168.15.12:34758 > 192.168.2.4:ftp_data S
0001 IP / TCP 192.168.15.103:2385 > 192.168.2.4:http S
0002 IP / TCP 192.168.15.30:6965 > 192.168.2.4:ftp S
0003 IP / TCP 192.168.15.180:52656 > 192.168.2.4:3389 S
>>>
```

Figure 27: Scapy sending random packets

4 ATTACKING SCENARIOS

The attacking scenarios may vary from implementation to implementation and level to level. This means that different strategies of exploitation have to be used when attacking the solution when it's configured on "Level 1" or "Level 2", as well as when Front-End Web servers are Unix or Windows (IIS or Apache)

The following sections describe some examples of possible exploitation scenarios that can take place using our configured infrastructure.

4.1 Attacking Scenarios on Level 2

On the 2nd level of the firewall configuration things are more complex for the attackers as there are specific rules on the firewall configuration, which allow specific communications between the computers of the network.

In this case the attackers have the ability to exploit the following:

Direct Attacking scenarios:

1. SQL BOX (windows server 2008)

- LFI (Local File Inclusion)
  - Using the implemented vulnerabilities on the actual vulnerable application, the attackers have the ability, to navigate within the system and collect valuable information for further use, as the LFI vulnerability allows the attacker to run commands like the following:
    http://name_of_server:port/../../../../../../../etc/sudoers
    and read the content of the related file.
  - Apart from navigating through the operating system, attackers have the ability to modify specific fields within the http queries and thus upload arbitrary codes, in order to further assist them to perform exploits. An example can be something like the following:
    Using tamper data on firefox browser, the attacker has the ability to change fields and make the web server to download malicious files internally. In this scenario, when such files are downloaded, they can be executed and open a reverse shell for the attackers. This can include "php_reverse_shell" as this was described in the documentation of MSc's laboratory sessions (exploiting Unix SQL BOX. As far as it concerns Windows, the approach can be a bit different, as the Windows environment doesn't support the same commands and libraries as Unix systems and thus, attacks like Windows_reverse_tcp

using meterpreter of metasploit can take place in order to create the appropriate packages for the reverse shell.

- SQL Injections
  During our implementation, we have decided to take advantage of the vulnerable server that Mr. Anastasios Stassinopoulos has created and take it to another level, migrating it into a Windows environment as well. For the Windows implementation the MySql database that has been configured, has been moved from the local machine and has been moved to the LAN network, on another SQL dedicated server. Thus, when exploiting the Windows SQL box, through SQL Injection, a shell will be opened directly in the internal network of the infrastructure.
  - SLQ Injection on SQLBOX on Unix system
    As described in the documentation of MSc's laboratory sessions various SQL injection attacks can be performed on this vulnerable application, giving the attackers the ability to grant shell access on the Local Machine.
  - SLQ Injection on SQLBOX on Windows system
    Respectively to the above, the same procedure can be followed also for an SQL injection on the Windows system, but in this way, the malicious sql commands and payload will be uploaded in a MySQL server that will be located in the LAN network.

2. Collaboration Site (OwnCloud)
   As mentioned in the previous chapters three different versions of the OwnCloud solution have been implemented in order for the attackers to have a plethora of different vulnerabilities. By exploiting some of them, various information can be disclosed such as vital information about user and admin credentials of the infrastructure in general. The vulnerabilities that can be exploited in the different versions can be found bellow:

Version 6.0.3 June 23rd 2014
SECURITY: Multiple XSS (oC-SA-2014-010)
SECURITY: Improper authorization checks in contacts (oC-SA-2014-011)
SECURITY: Improper authorization checks in files_external (oC-SA-2014-012)
SECURITY: Improper authorization checks in documents (oC-SA-2014-013)
SECURITY: CSRF in documents (oC-SA-2014-014)
SECURITY: Enumeration of shared files in documents (oC-SA-2014-015)
SECURITY: Improper authorization checks in core (oC-SA-2014-016)
SECURITY: Deserialization of Untrusted Data in core (oC-SA-2014-017)

Version 5.0.8 July 10th 2013
SECURITY: XSS vulnerability in "Share Interface" (oC-SA-2013-029)
SECURITY: Authentication bypass in "user_webdavauth" (oC-SA-2013-030)

Version 5.0.6 May 14th 2013

SECURITY: SQL Injection (oC-SA-2013-019)
SECURITY: Multiple directory traversals (oC-SA-2013-020)
SECURITY: Multiple XSS vulnerabilities (oC-SA-2013-021)
SECURITY: Open redirector (oC-SA-2013-022)
SECURITY: Password auto completion (oC-SA-2013-023)
SECURITY: Privilege escalation in the calendar application (oC-SA-2013-024)
SECURITY: Privilege escalation and CSRF in the API (oC-SA-2013-025)
SECURITY: Incomplete blacklist vulnerability (oC-SA-2013-026)
SECURITY: Information disclosure: CSRF token + username (oC-SA-2013-027)

<u>Note:</u> On the "User" personal account (local account Password: "Simple_Password") a txt file has been created in order to reveal some user credentials to the attackers

<u>Indirect Attacking scenarios:</u>

When access has been granted on one of the above, other vulnerabilities can be discovered and exploited.
Specific scenarios can be found bellow:

1. Winexe
   Winexe is an application that can help a user to run remote commands in many systems, such linux and Windows.
   This application has been installed in the Ubuntu workstation and thus can be used from attackers to jump to any of the Windows Machines.

<u>Note:</u> In the home directory of the Ubuntu Workstation a backdoor file has been created ("login.conf") that includes credentials for the attackers in order to login to the windows domain computers

2. Psexec (for windows systems)
   Like above, psexec is a lightweight telnet replacement that lets any user execute processes on other systems, complete with full interactivity for console application without having to manually install client software.
   This application is installed in the Windows 7 workstation as well as to the Windows SQL Box machine. This application can be used in order to jump between windows domain machines and execute malicious commands.

5 REFERENCES

[1] Scarfone K., Mell P., "Guide to Intrusion Detection and Prevention Systems (IDPS)", Computer Security Resource Center (National Institute of Standards and Technology) (800–94), February 2007.

[2] Mattord V., “Principles of Information Security”, Course Technology. pp. 290–301, ISBN 978-1-4239-0177-8, 2008.

[3] Anderson J., “Computer Security Technology Planning Study Volume 2”, October 1972, available at: http://seclab.cs.ucdavis.edu/projects/history/papers/ande72.pdf

[4] Anderson J., “Computer Security Threat Monitoring and Surveillance”, April 1980, available at: http://seclab.cs.ucdavis.edu/projects/history/papers/ande80.pdf

[5] Denning D., "An Intrusion Detection Model," Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119–131

[6] Kohlenberg T. (Ed.), Alder R., Carter, Dr. Everett F. (Skip), Esler J., Foster J., Jonkman M., Marty R., and Poor M., "Snort IDS and IPS toolkit: featuring Jay Beale and Members of the Snort Team", Syngress, 2007, ISBN 978-1-59749-099-3

[7]Anderson R., “Security Engineering: A Guide to Building Dependable Distributed Systems”, New York: John Wiley & Sons. pp. 387–388, ISBN 978-0-471-38922-4, 2001

[8] Snapp S., Brentano J., Dias, Gihan D., Goan T., Heberlein T., Ho C., Levitt K., Mukherjee B., Smaha S., Granc T., Teal D., Mansur D., "DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and An Early Prototype," The 14th National Computer Security Conference, October, 1991, pages 167–176.

[9] Yongguang Z., Weenkey L., Yi-An H., “Intrusion Detection Techniques for Mobile Wireless Networks”, ACM WINET 2003, available at: http://www.cc.gatech.edu/~wenke/papers/winet03.pdf

[10] Endorf C., Schultz E., Mellander J., “Intrusion Detection & Prevention”, McGraw Hill Professional, ISBN 978-0-072-22954-7, 2004

[11] Ganapathy S., Yogesh P., Kannan A., “Intelligent Agent-Based Intrusion Detection System Using Enhanced Multiclass SVM,” Computational Intelligence and Neuroscience, vol. 2012, Article ID 850259, 10 pages, 2012. doi:10.1155/2012/850259

[12] Dorosz P., Kazienko P., “Systems wykrywania intruzów” VI Krajowa Konferencja Zastosowan Kryptografii ENIGMA 2002, Warsaw May 2002 , p. TIV 47-78, (In Polish only)

[13] Bragg, R., “CISSP: Certified Information Systems Security Professional Training Guide”, Indianapolis IN: QUE Publishing, 2003

[14] Harris, S. (2002). Mike Meyers’ CISSP Certification Passport. Berkley, CA: McGraw-Hill/Osbourne

[15] Kadel L., “Designing and Implementing an Effective Information Security Program: Protecting the Data Assets of Individuals”, Small and Large Businesses, March 2004

[16] Kazienko P., Dorosz P., “Intrusion Detection Systems (IDS) Part I”, April 2003

[17] Kazienko P., Dorosz P., “Intrusion Detection Systems (IDS) Part 2”, June 2004

[18] Shimonski R., “What You Need to Know About Intrusion Detection Systems”, November 2002

[19] Roesch M., Green C., Caswell B., “Snort user’s manual 2.9.6”, available at: http://manual.snort.org/

[20] SNORT, “Snort Official Documentation”, available at: http://www.snort.org/docs/

[21] SNORT, “Snort documentation”, available at: http://www.snort.org/start/documentation

[22] TECHTARGET, “Definition Snort”, available at: http://searchmidmarketsecurity.techtarget.com/definition/Snort

[23] PEARSONHIGHERED, “Dissecting Snort”, available at: http://www.pearsonhighered.com/assets/hip/us/hip_us_pearsonhighered/samplechapter/157870281X.pdf

[24] Weir J., “Building a Debian\Snort based IDS”, August 2012, available at: http://www.snort.org/assets/167/deb_snort_howto.pdf

[25] OINKMASTER, “About Oinkmaster”, available at: http://oinkmaster.sourceforge.net/about.shtml

[26] SECDEV, “Scapy’s documentation”, April 2010, available at: http://www.secdev.org/projects/scapy/doc/index.html

[27] WORKROBOT, “SCAPY packet-crafting reference”, available at: http://www.workrobot.com/sansfire2009/SCAPY-packet-crafting-reference.html

[28] Combs, R., “VRT: Snort 2.9 Essentials: The DAQ”, August 2010, available at: http://vrtblog.snort.org/2010/08/snort-29-essentials-daq.html

[29] Metcalf, W., & Julien, V. (n.d.)n “ snort_inline”, available at: http://snortinline.sourceforge.net/oldhome.html

[30] Sourcefire, Inc. (n.d.), “External DAQ Modules”, available at: http://www.snort.org/snort-downloads/external-daq/

[31] METAFLOWS, “PF_RING Snort multiprocessing (Inline/Passive)”, available at: http://www.metaflows.com/solutions2/pf-ring/

[32] SNORT, “Snort Required Software”, available at: http://www.snort.org/start/requirements OReilly Linux iptables, Pocket Reference (2004)

[33] LINUX FIREWALLS Attack Detection and Response with iptables, psad, and fwsnortby Michael Rash

[34] Firewalls for Dummies, by Brian Komar, Ronald Beekelaar, and Joern Wettern,PhD 2003

[35] A Comprehensive Firewall Testing Methodology Murray Brand , Edith Cowan University 2007

[36] http://rbgeek.wordpress.com/2012/05/14/ubuntu-as-a-firewallgateway-router/

[37] http://www.geekingatschool.com/2011/02/setup-ubuntu-server-as-a-simple-router/

[38] https://help.ubuntu.com/community/IptablesHowTo

[39] http://wiki.centos.org/HowTos/Network/IPTables

[40] http://forums.fedoraforum.org/showthread.php?t=259706

[41] http://www.revsys.com/writings/quicktips/nat.html

[42]http://www.thegeekstuff.com/2012/08/iptables-log-packets/

[43]https://doc.owncloud.org/

[44]http://www.ubuntu.com/download/server/install-ubuntu-server

[45]http://www.amanhardikar.com/mindmaps/Practice.html

[46]https://www.rivy.org/2013/03/building-barnyard2-from-source/

[47]http://opentodo.net/2012/10/snort-from-scratch-part-i/

[48]https://nvd.nist.gov/

[49]https://www.corelan.be/index.php/2011/02/27/cheat-sheet-installing-snorby-2-2-with-apache2-and-suricata-with-barnyard2-on-ubuntu-10-x/

[49] https://nathanhoad.net/how-to-ruby-on-rails-ubuntu-apache-with-passenger

[50]https://www.rivy.org/2013/03/installing-and-configuring-barnyard2/

APPENDIX

Passwords Table

HOST	System		Service	
	Username	Password	Username	Password
Firewall (IPtables)	user	Simple_Password		
	root	toor		
Firewall (pfSense)	admin	pfsense		
	user	Simple_Password		
Domain	Administrator	qwe123!@#QWE		
	ssluser1	Simple_Password		
	ssluser2	Simple_Password		
	ssluser3	Simple_Password		
Domain DB MySQL			root	yuJuBJPaWDv4Spuv
			dbowner	t3ms3cDB0wn3r
Snort	sensor	sensor	***Snorby WebConsole Credentials***	
			snorby@snorby.org	snorby
OwnCloud	user	Simple_Password	***OwnCloud Console Credentials***	
	root	toor	admin	admin
			user	Simple_Password
			<all domain users>	<domain passwords>
Ubuntu WS	user	Simple_Password		
	root	toor		
SQLBox (UNIX)	user	Simple_Password	root	yuJuBJPaWDv4Spuv
	root	toor	dbowner	t3ms3cDB0wn3r

Table 1: Table of Passwords