University of Piraeus

Department of Digital Systems

MSc in “Security of Digital Systems”

Master Thesis: “*Cyber Warfare Emulation Environment*”

Stamatis Mandilas
MTE 1326

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 George Sextos under the name "*Cyber Wargames Environment*" this part contains the rest elements which complete the Emulation Environment IPTables, Colaboration Site (OwnCloud), Scapy, Barnyard, Snorby).

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

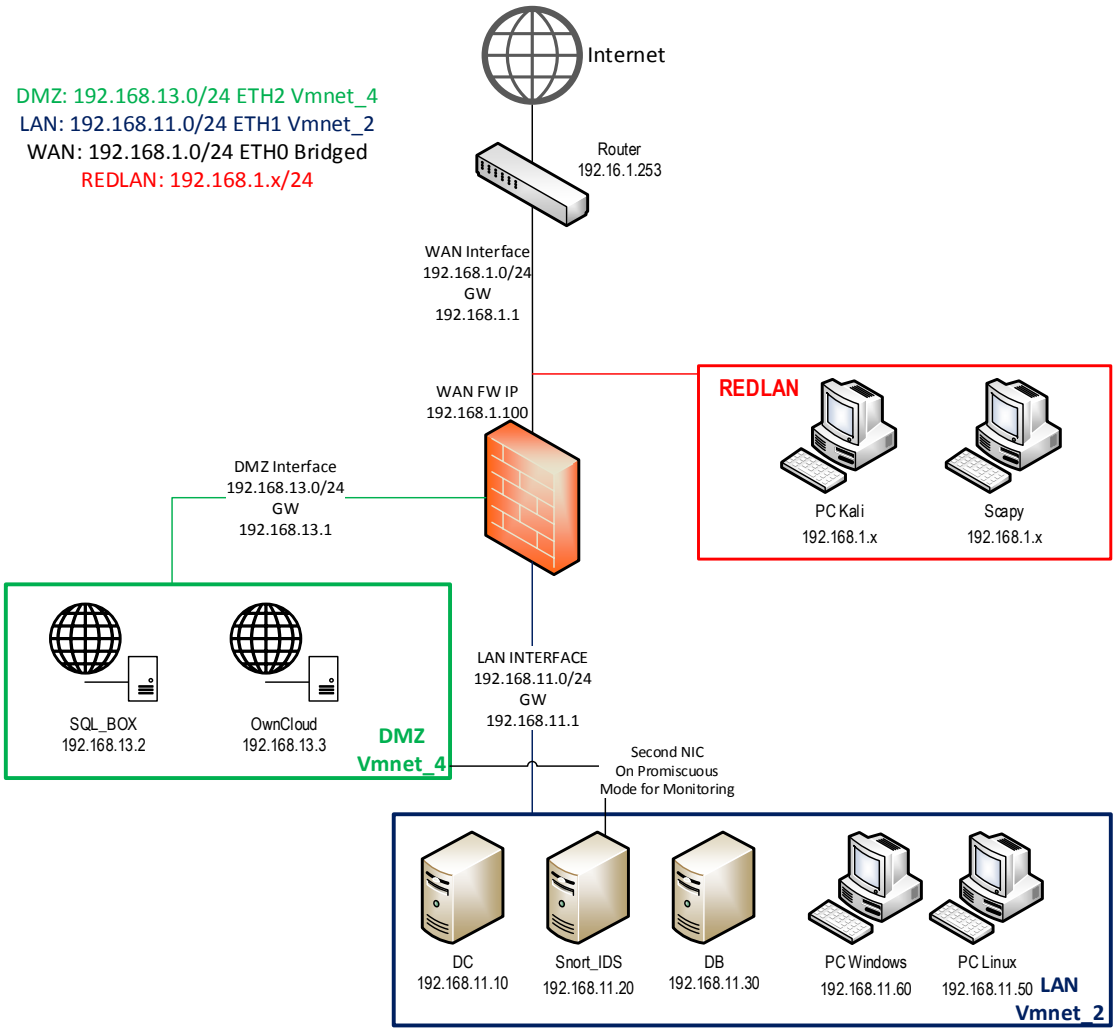

Figure 1: Network Architecture Topology

- 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.

Functions- operation

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.

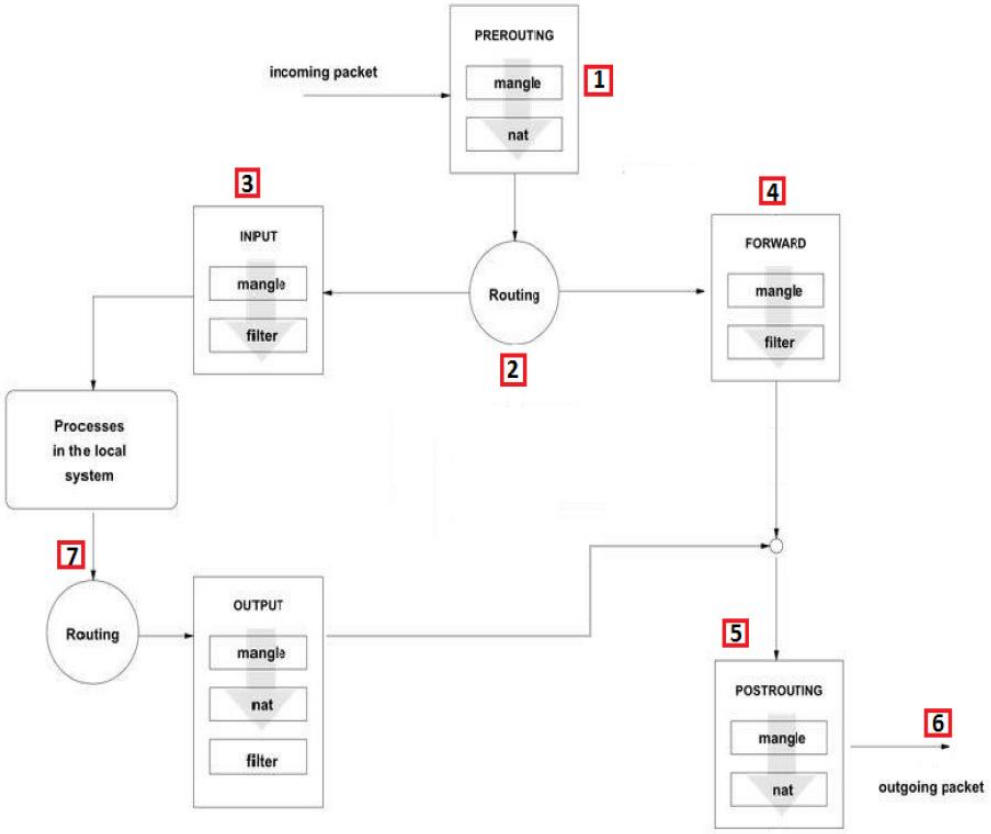

Figure 2: Firewall Chain Routing

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.

3.2.1.1 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.

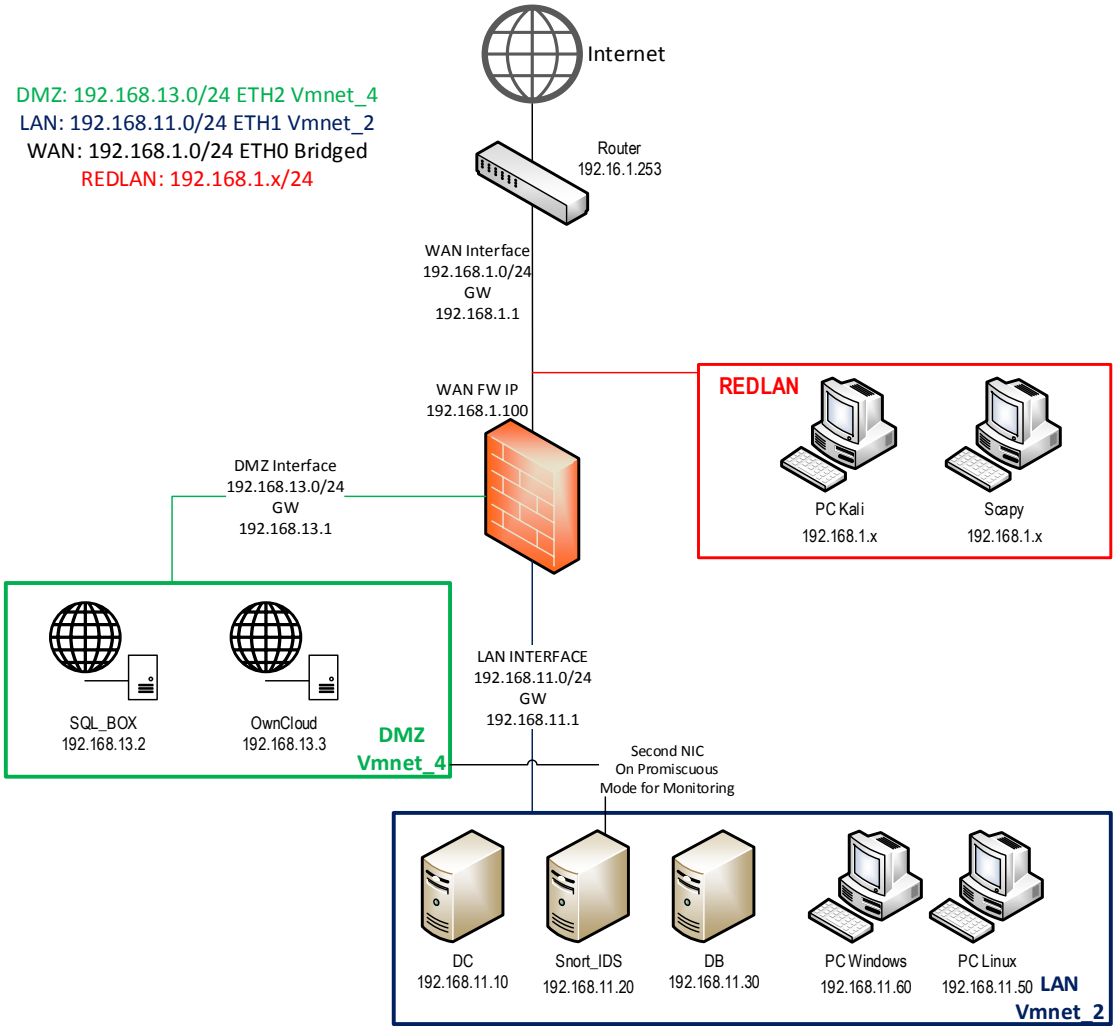

Figure 3: Network Address Translation

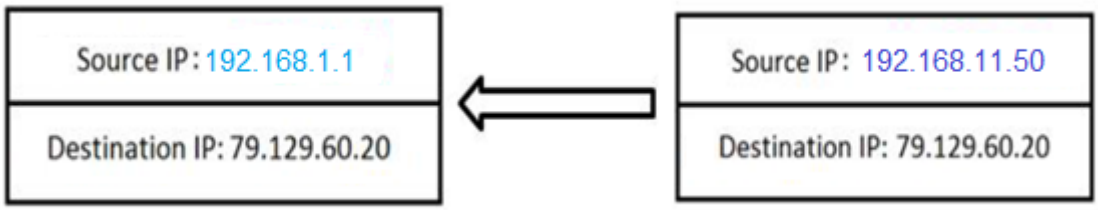

Figure 4: NAT Example

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).

3.2.2 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.

3.2.3 Firewall PfSense

- What is PfSense?

PfSense is an open source firewall/router computer software distribution based on FreeBSD. It is installed on a physical computer or a virtual machine (Virtual machine in our case) to make a dedicated firewall/router for a network and is noted for its reliability and offering features often only found in expensive commercial firewalls.

- Why PfSense?

We have selected to add pfSense as a firewall component in our infrastructure because it utilizes a really nice graphical user interface, which is pretty much the same as an idea with any of the firewalls that many enterprises are using. With an interface like that any of the students can get familiar with the concepts of network address translation, firewalling, etc. Apart from the nice graphical interface, by using pfsense anyone can also use many out of the box components of the system, like VPN (IPsec, L2TP, OpenVPN, PPTP) and many other features that can also be downloaded. Due to its great performance and features for an open source firewall many companies use it as their main firewall as well.

3.2.3.1 PfSense install

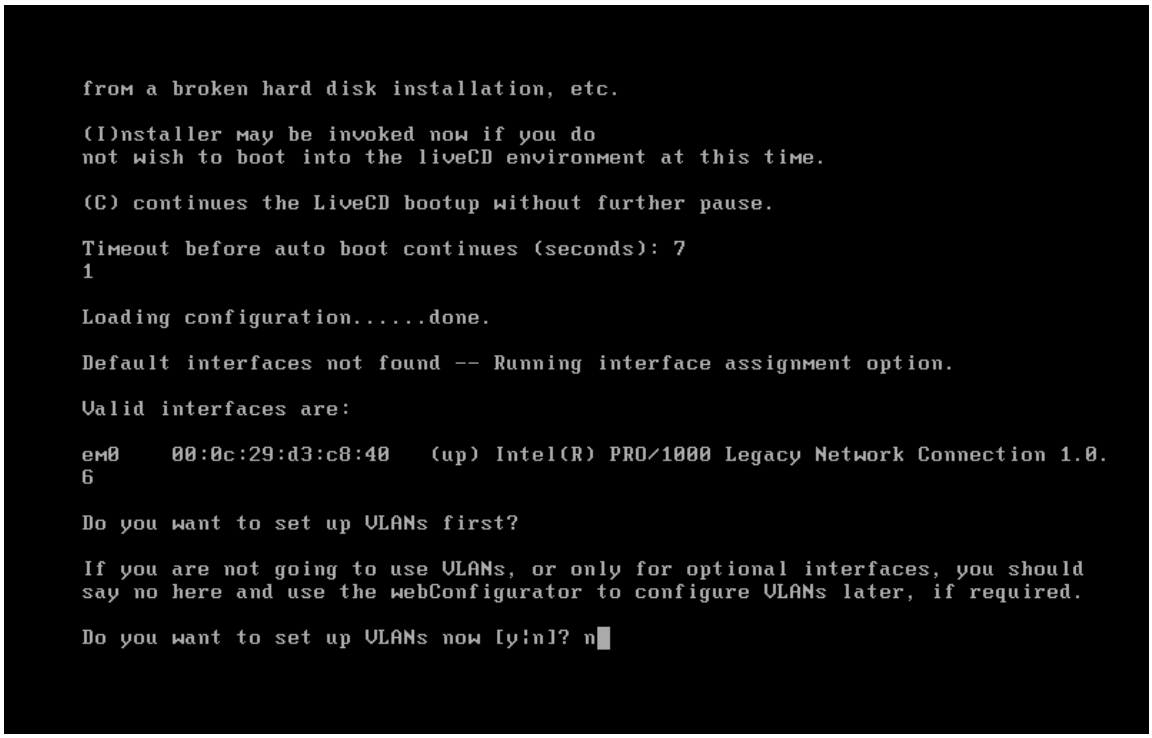

Figure 5: pfSense install 1

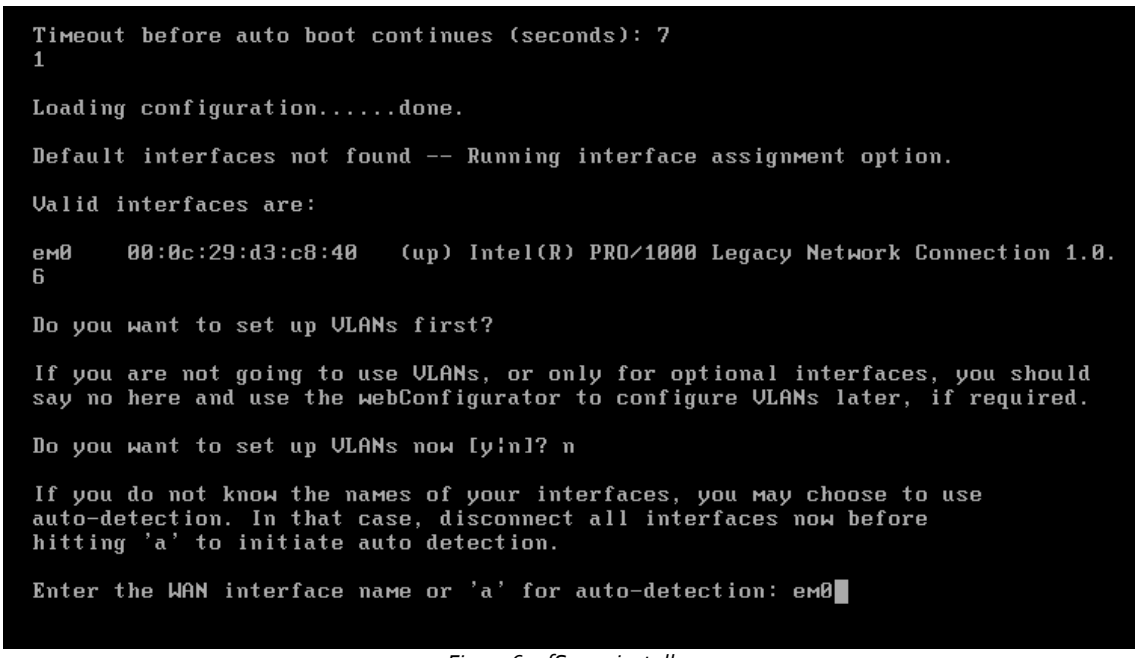

Figure 6: pfSense install 2

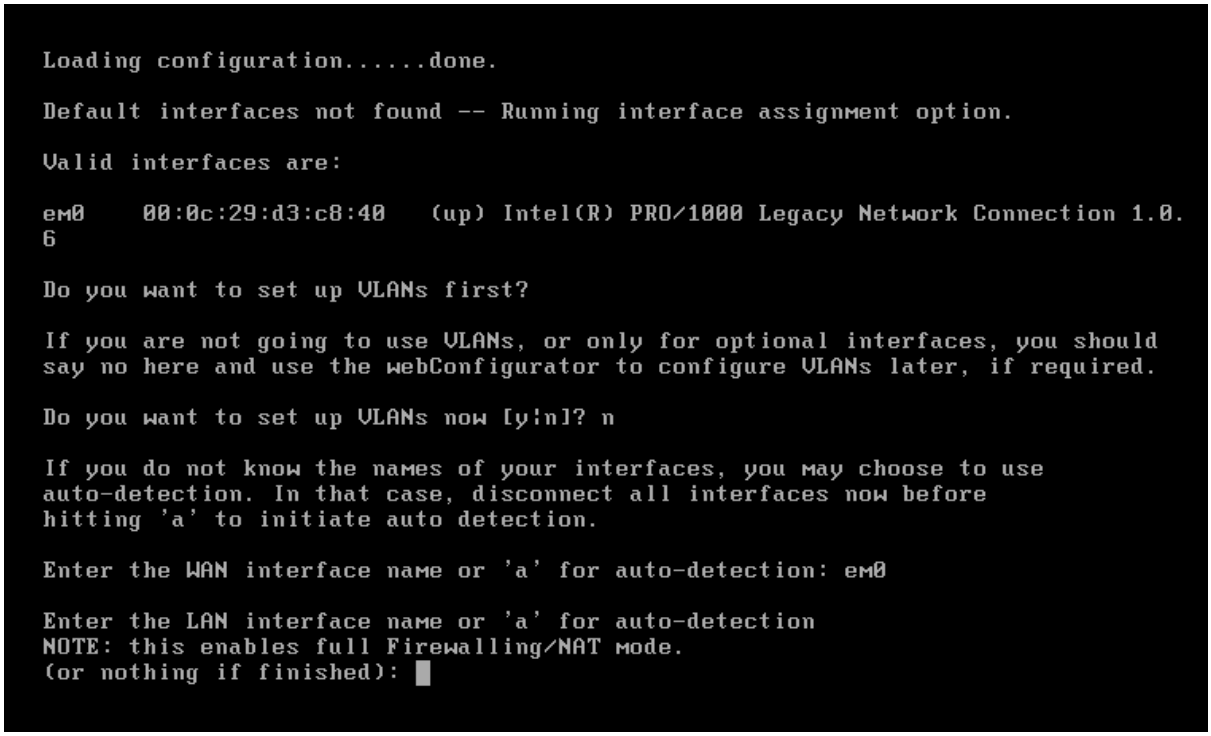

Figure 7: pfSense install 3

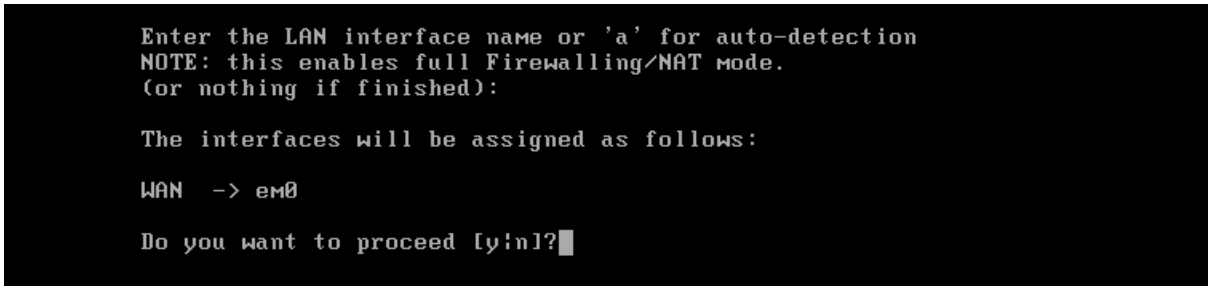

Figure 8: pfSense install 4

During the following step press “99” in order to install pfSense on the hard drive and be able to save any changes after restarts.

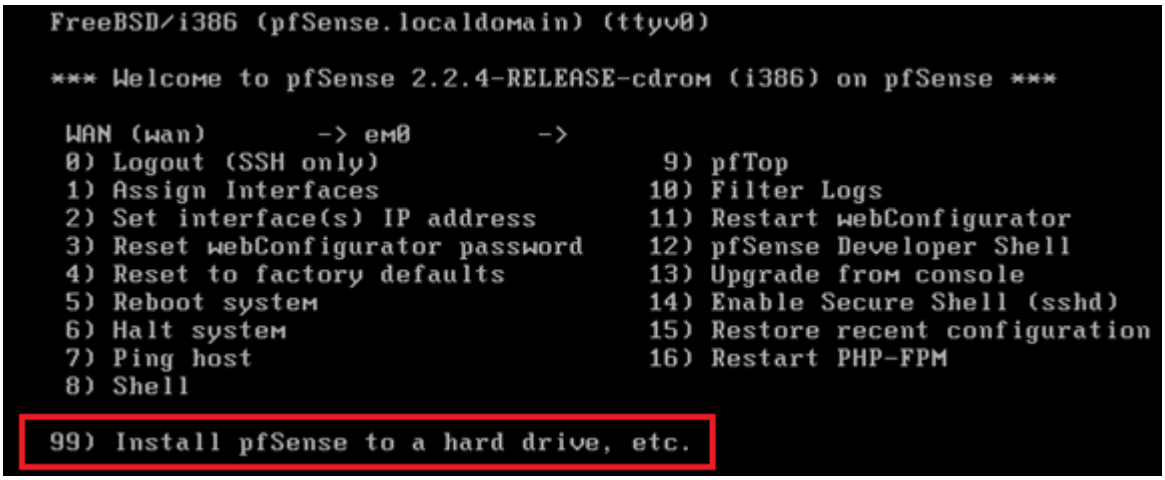

Figure 9: pfSense install 5

After the installation it is time to configure the interfaces

```
FreeBSD/i386 (pfSense.localdomain) (ttyv0)

*** Welcome to pfSense 2.2.4-RELEASE-cdrom (i386) on pfSense ***

 WAN (wan)       -> em0        ->
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) pfSense Developer Shell
 4) Reset to factory defaults         13) Upgrade from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

99) Install pfSense to a hard drive, etc.

Enter an option: 2
```

Figure 10: pfSense install 6

```
Configure IPv4 address WAN interface via DHCP? (y/n) n

Enter the new WAN IPv4 address.  Press <ENTER> for none:
> 192.168.10.100

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
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 31):
> 24

For a WAN, enter the new WAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
> 192
not an IPv4 IP address!

For a WAN, enter the new WAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
> 192.168.10.1

Configure IPv6 address WAN interface via DHCP6? (y/n)
```

Figure 11: pfSense install 7

```
Configure IPv6 address WAN interface via DHCP6? (y/n) n

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

Do you want to enable the DHCP server on WAN? (y/n) n

Please wait while the changes are saved to WAN...
 Reloading filter...
 Reloading routing configuration...
 DHCPD...

The IPv4 WAN address has been set to 192.168.10.100/24
You can now access the webConfigurator by opening the following URL in your web
browser:
                http://192.168.10.100/

Press <ENTER> to continue.
```

Figure 12: pfSense install 8

```
                http://192.168.10.100/

Press <ENTER> to continue.
*** Welcome to pfSense 2.2.4-RELEASE-cdrom (i386) on pfSense ***

 WAN (wan)       -> em0        -> v4: 192.168.10.100/24
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) pfSense Developer Shell
 4) Reset to factory defaults         13) Upgrade from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

99) Install pfSense to a hard drive, etc.

Enter an option:
```

Figure 13: pfSense install 9

Now that the installation has been finished, more configurations on the interfaces and the firewalling can be done through the web console.

The web console is accessible through http or https, regarding what the user has chosen during the initial configuration of the solution.

The default credentials to access the console can be found bellow:

- Username: admin
- Password: pfSense

Figure 14: pfSense install 10

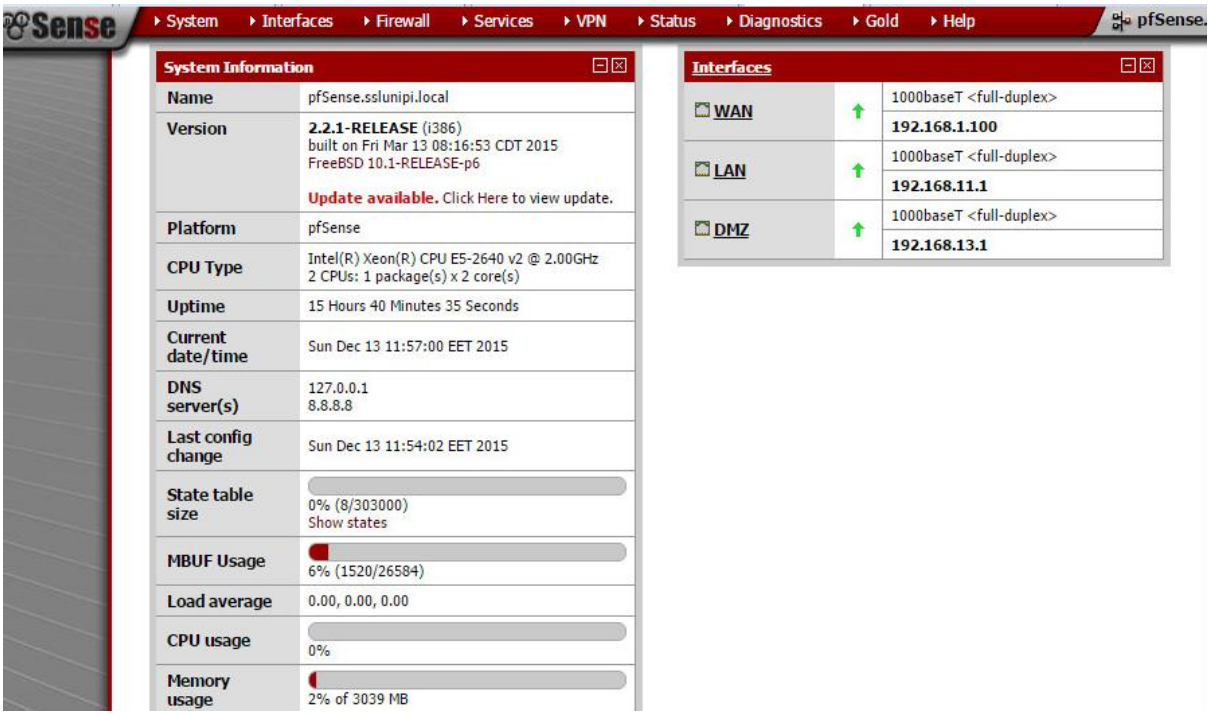

Figure 15: pfSense install 11

The rules that have been set on the pfSense solution are identical to those that have been describe as well in the previous chapters, so the infrastructure can be modular, while also utilizing the same functionalities.

3.2.3.2 Level 1 Rules

NAT

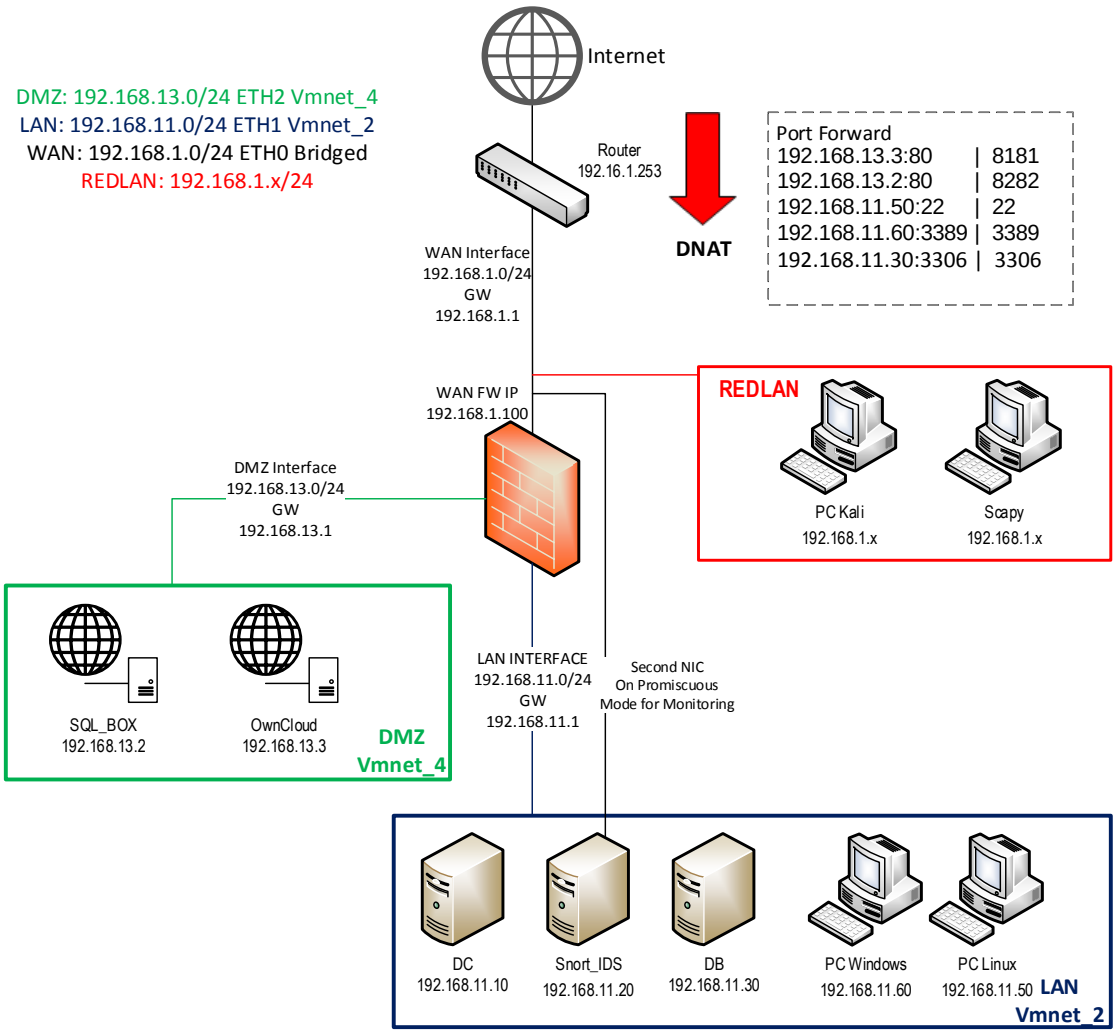

Figure 16: Level 1 NAT

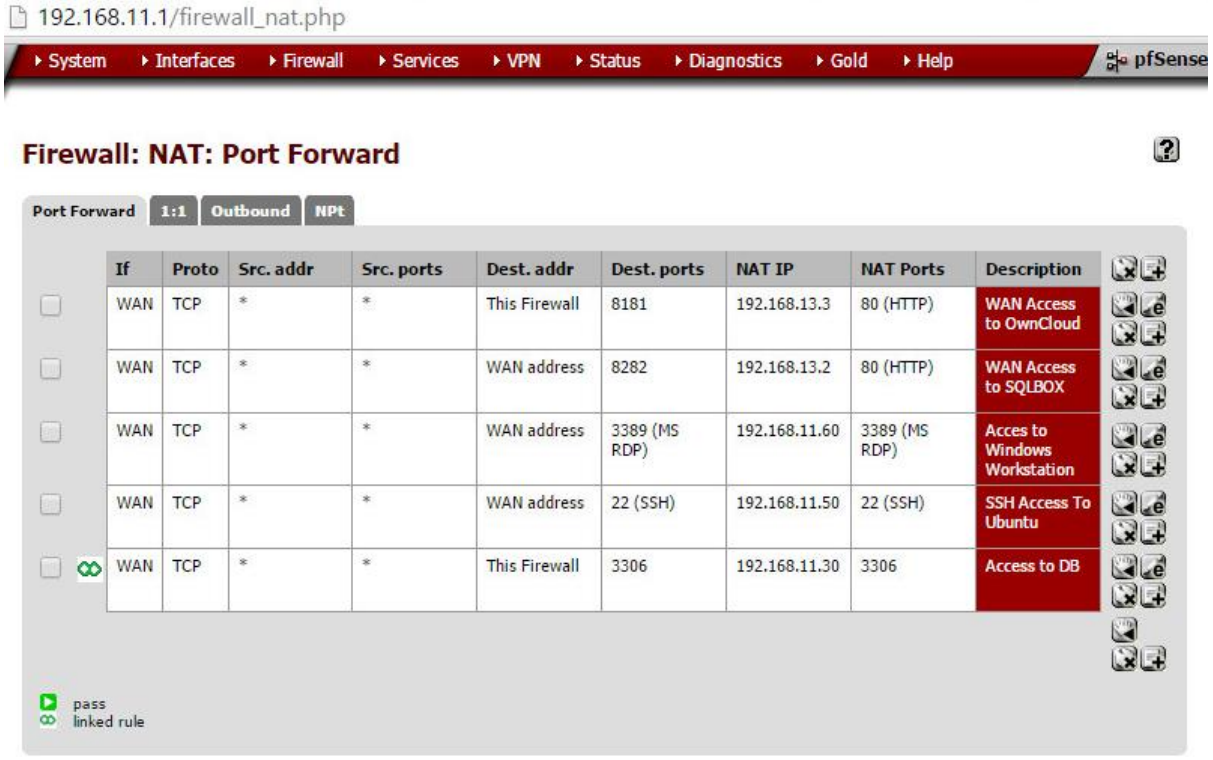

Figure 17: pfSense Level 1 NAT

WAN Rules

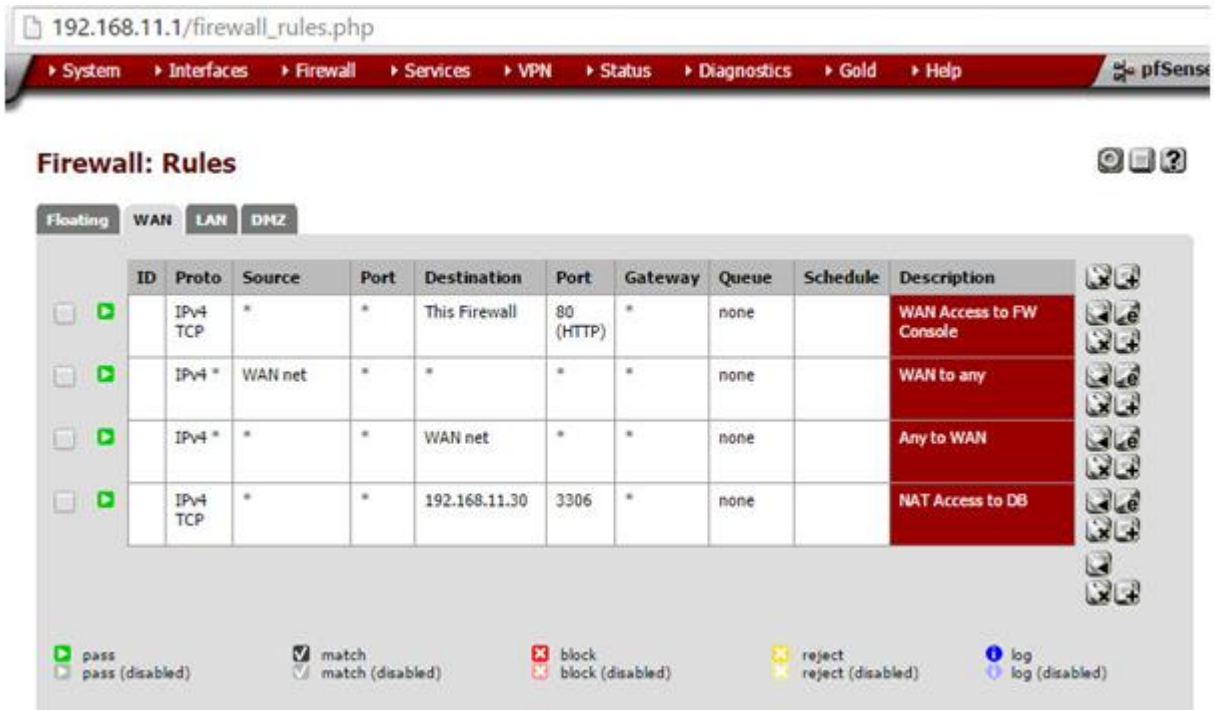

Figure 18: pfSense Level 1 WAN Rules

LAN Rules

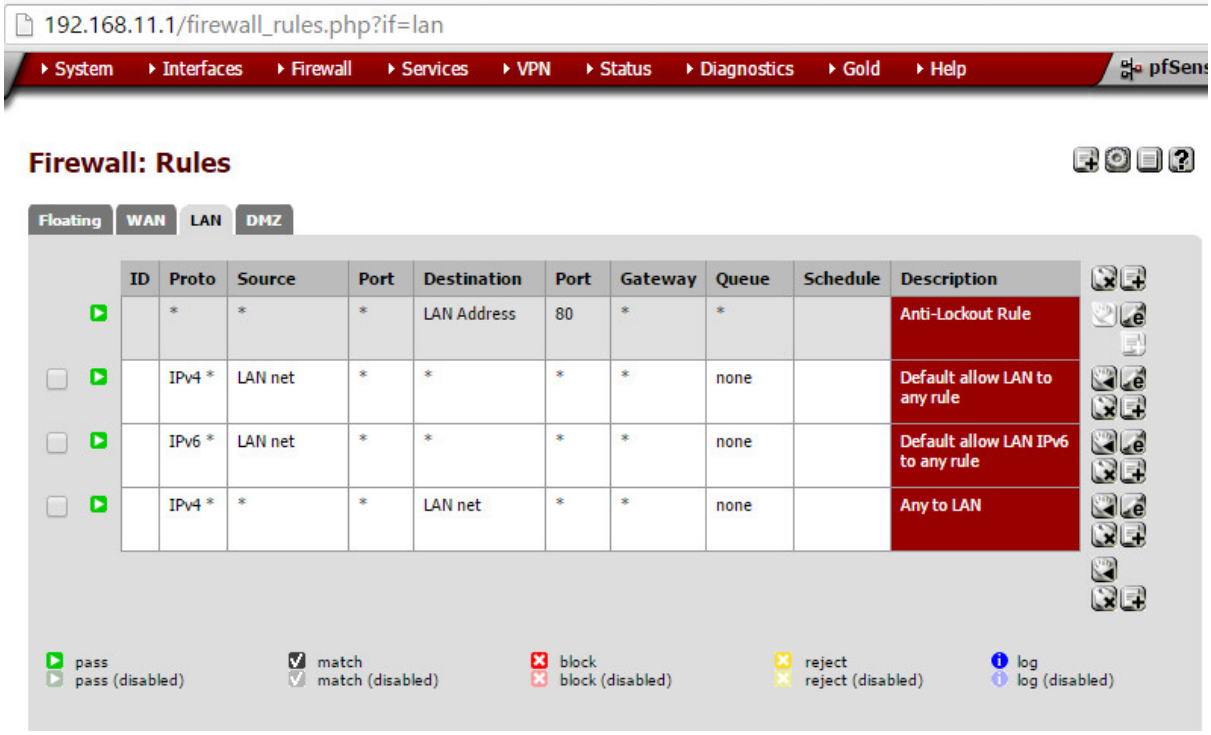

Figure 19: pfSense Level 1 LAN Rules

DMZ Rules

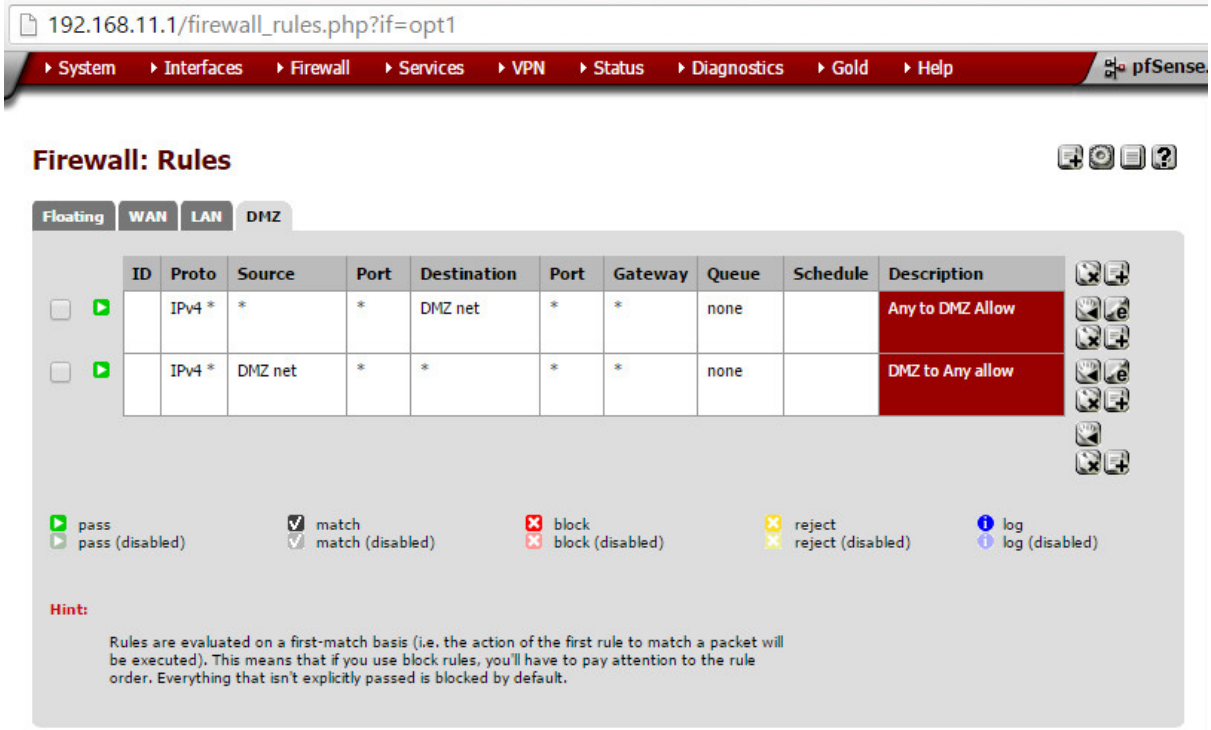

Figure 20: pfSense Level 1 DMZ Rules

3.2.3.3 Level 2 Rules

NAT

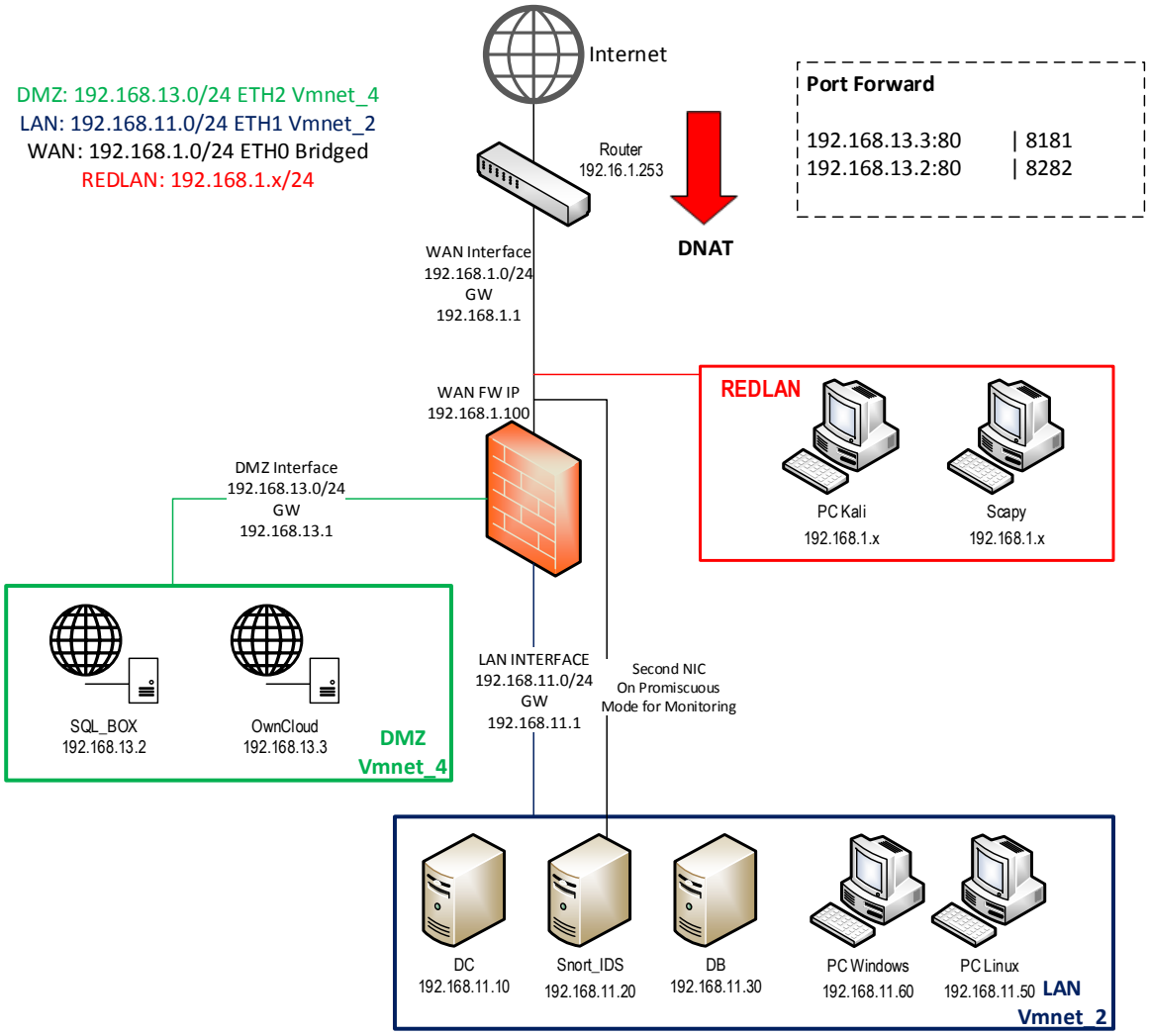

Figure 21: Level 2 NAT

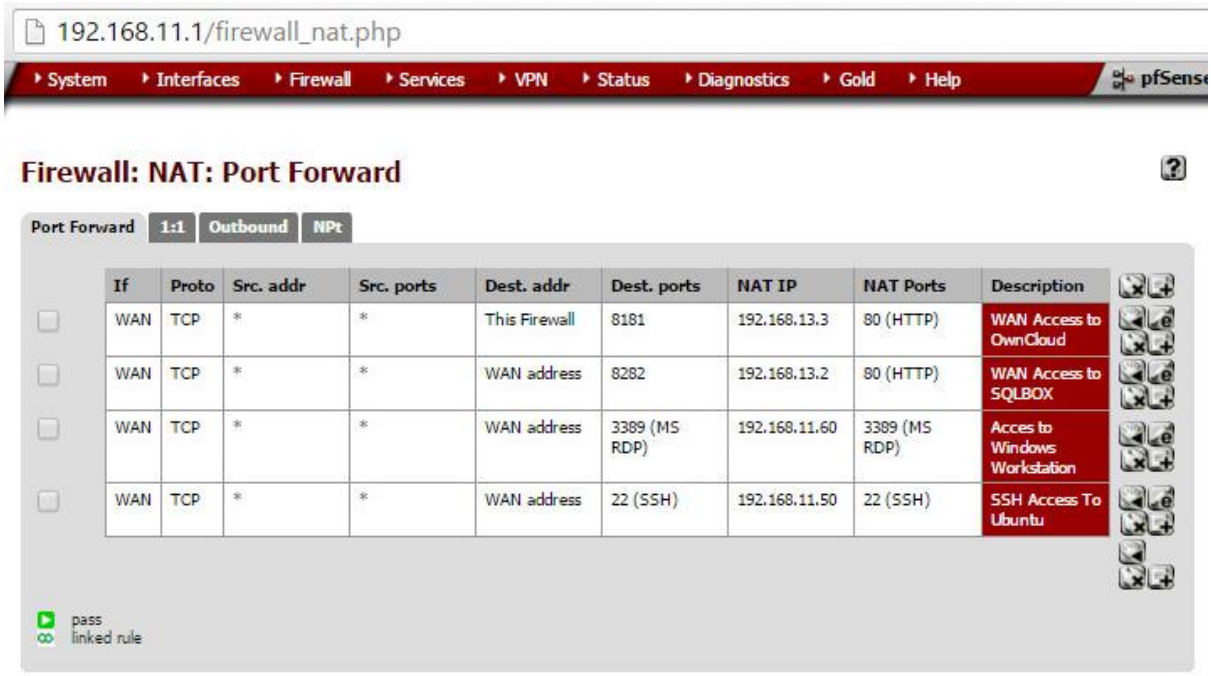

Figure 22: pfSense Level 2 NAT

WAN Rules

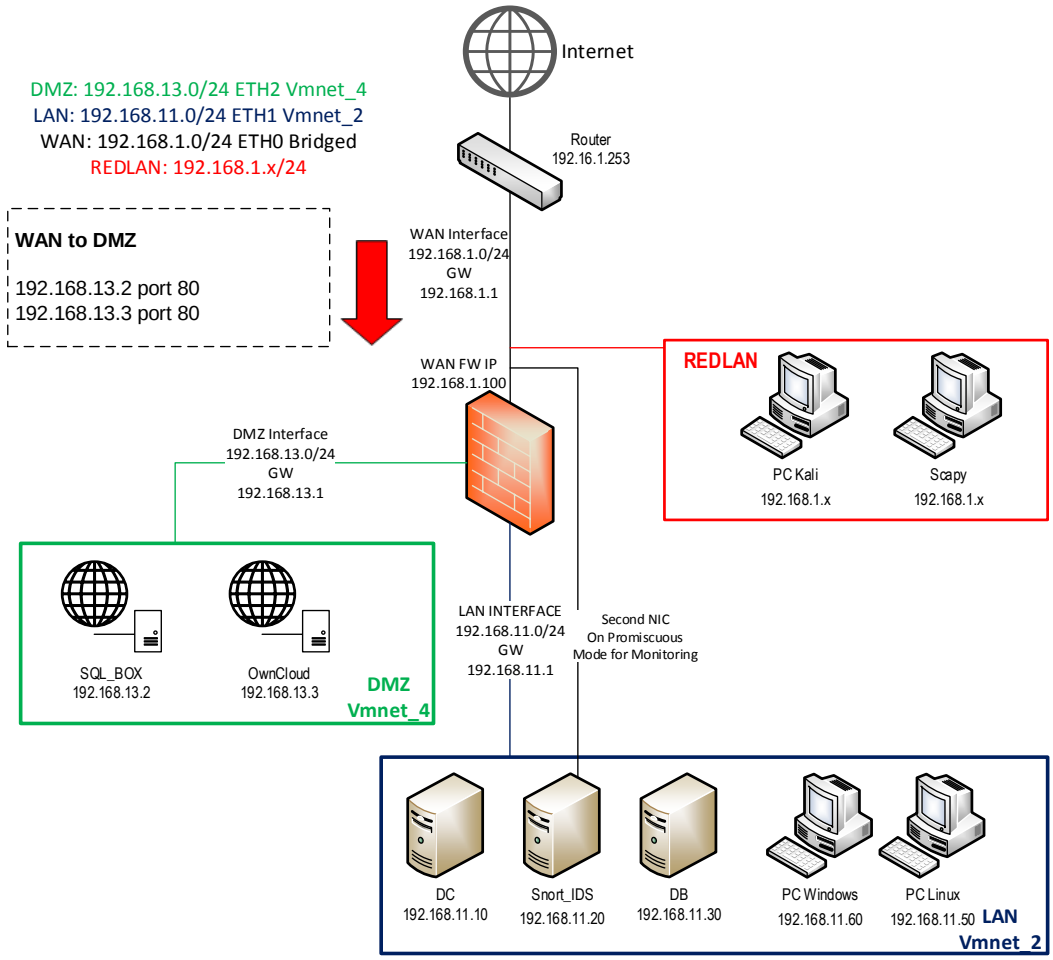

Figure 23: Level 2 WAN Rules

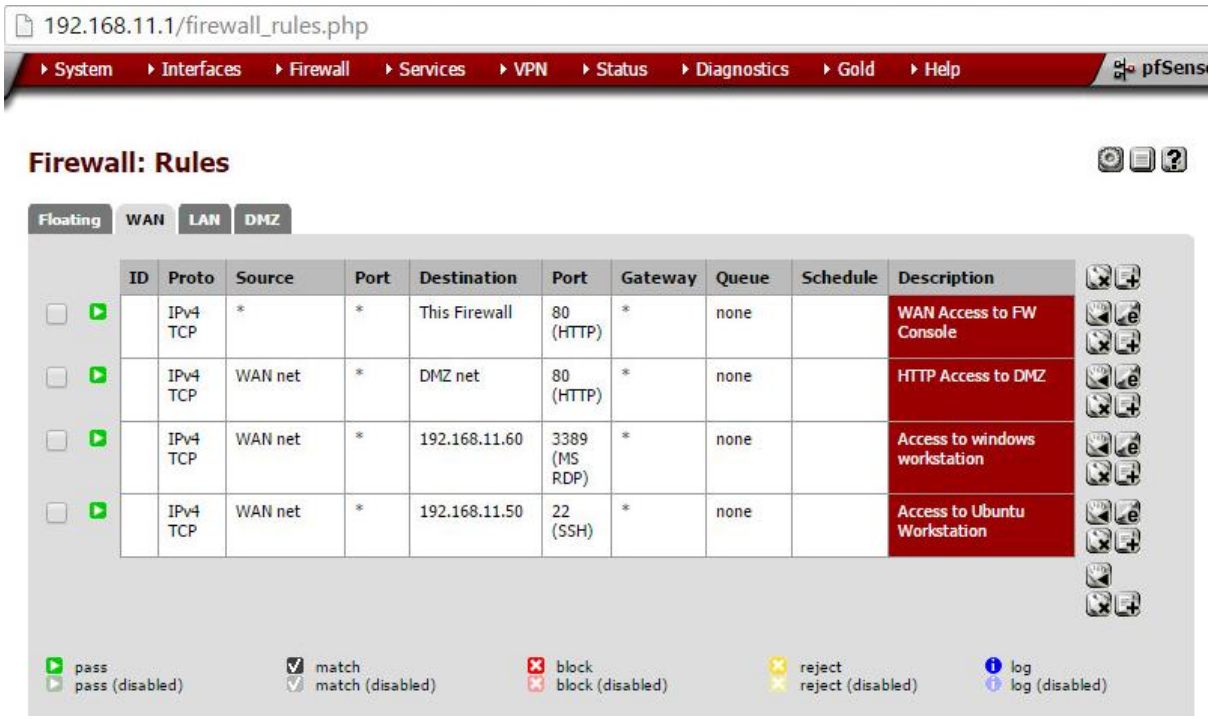

Figure 24: pfSense Level 2 WAN Rules

LAN Rules

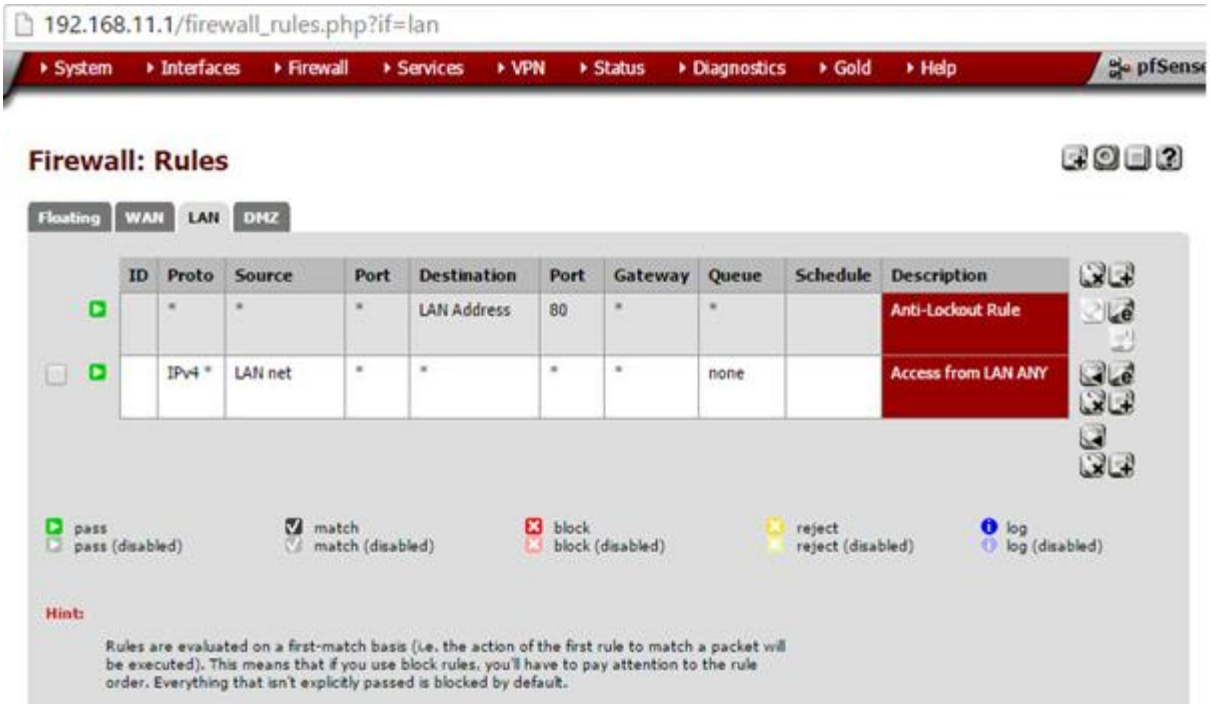

Figure 25: pfSense Level 2 LAN Rules

DMZ Rules

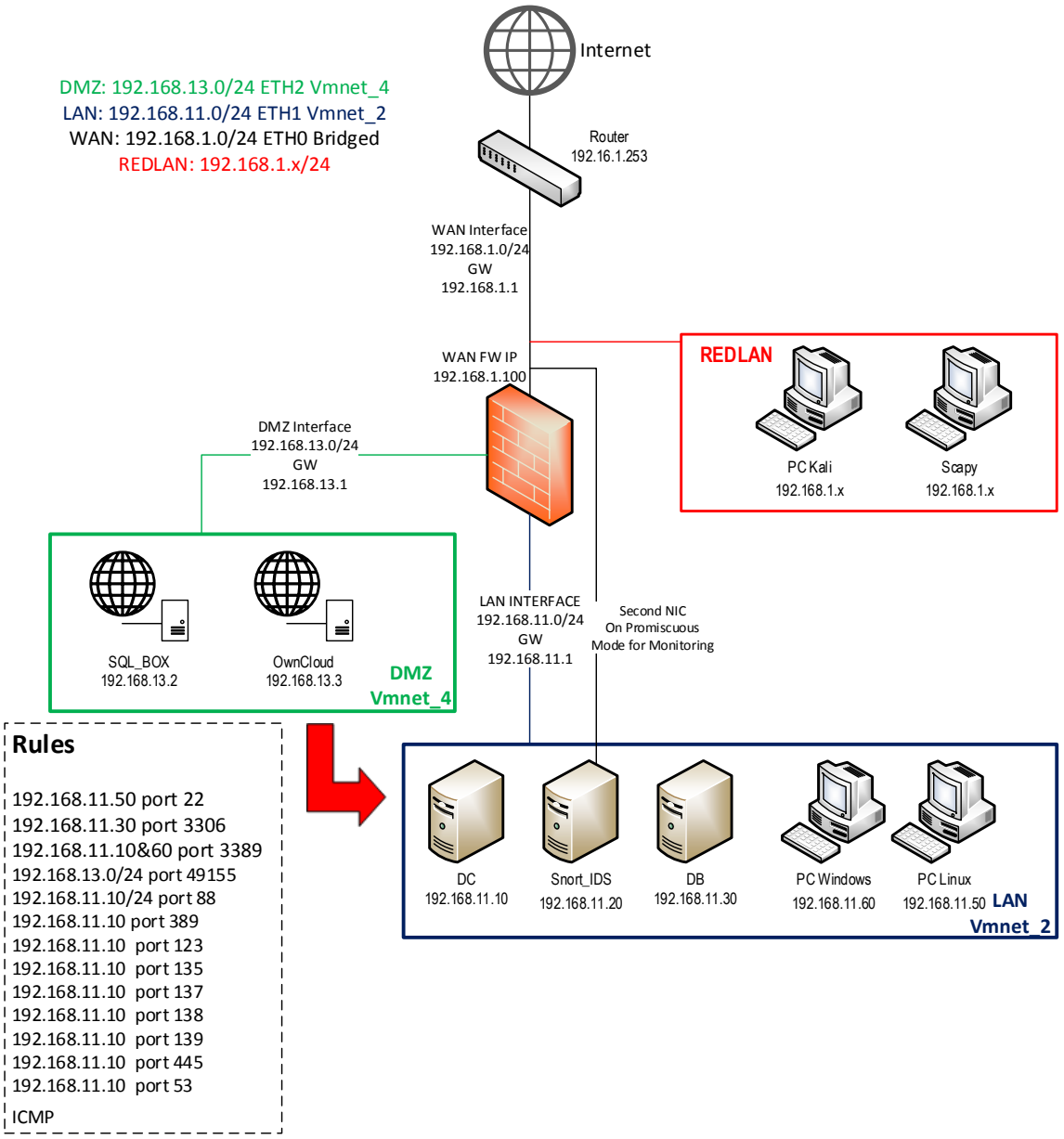

Figure 26: Level 2 DMZ Rules

	ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
		IPv4 *	DMZ net	*	WAN net	*	*	none		Access from DMZ to WAN ANY
		IPv4 ICMP	DMZ net	*	*	*	*	none		ICMP to ANY
		IPv4 TCP/UDP	DMZ net	*	192.168.11.10	53 (DNS)	*	none		DNS Access to LAN
		IPv4 TCP	192.168.13.2	*	192.168.11.30	3306	*	none		SQLBOX to MySQL Access
		IPv4 TCP/UDP	DMZ net	*	192.168.11.10	389 (LDAP)	*	none		LDAP Access
		IPv4 UDP	DMZ net	*	192.168.11.10	123 (NTP)	*	none		NTP Access
		IPv4 TCP	DMZ net	*	192.168.11.50	22 (SSH)	*	none		SSH Access
		IPv4 TCP	DMZ net	*	192.168.11.60	3389 (MS RDP)	*	none		RDP to Workstation
		IPv4 TCP	DMZ net	*	192.168.11.10	3389 (MS RDP)	*	none		RDP to DC
		IPv4 TCP	DMZ net	*	*	80 (HTTP)	*	none		HTTP Access
		IPv4 TCP	DMZ net	*	*	443 (HTTPS)	*	none		HTTPS Access
		IPv4 TCP/UDP	DMZ net	*	LAN net	135 - 139	*	none		NETBIOS & RPC
		IPv4 TCP/UDP	DMZ net	*	192.168.11.10	445 (MS DS)	*	none		MS DS
		IPv4 TCP	192.168.13.2	*	LAN net	88	*	none		Kerberos - System Authentication

Figure 27: pfSense Level 2 DMZ Rules

3.3 Windows Infrastructure

As described in the beginning, experience on windows systems for the postgraduate students is really limited, as they do not get familiar with concepts like the Domain, which could be really useful for them during their involvement later on as employees on any organization.

What is a Windows Domain?

Windows Domains are generally made up of computers on the same local network. The main difference of the domain with any traditional workgroup, is that all the computers in the domain, are being registered to a domain controller. The Domain controller is main server that hosts the domain, which controls almost everything within the domain. Things that can be managed from the domain controller is the set of users that can access computers and servers within the domain, privileges that these users have in any of the computers, files, shares, etc. Also it can control and apply specific policies on all the domain computers, like the enforcement of strong passwords, the expiration and history of them, etc, in order for any domain to comply with any security regulations.

3.3.1 Domain Controller

In the domain that we have created for the purposes of this thesis, we have created a domain controller using a Windows Server 2012 R2 operating system and assigned the name "**sslunipi.com"** as the domain name. The procedure to create a domain on a domain controller as well as installing the role of a DNS Server, can be found on the screenshots bellow:

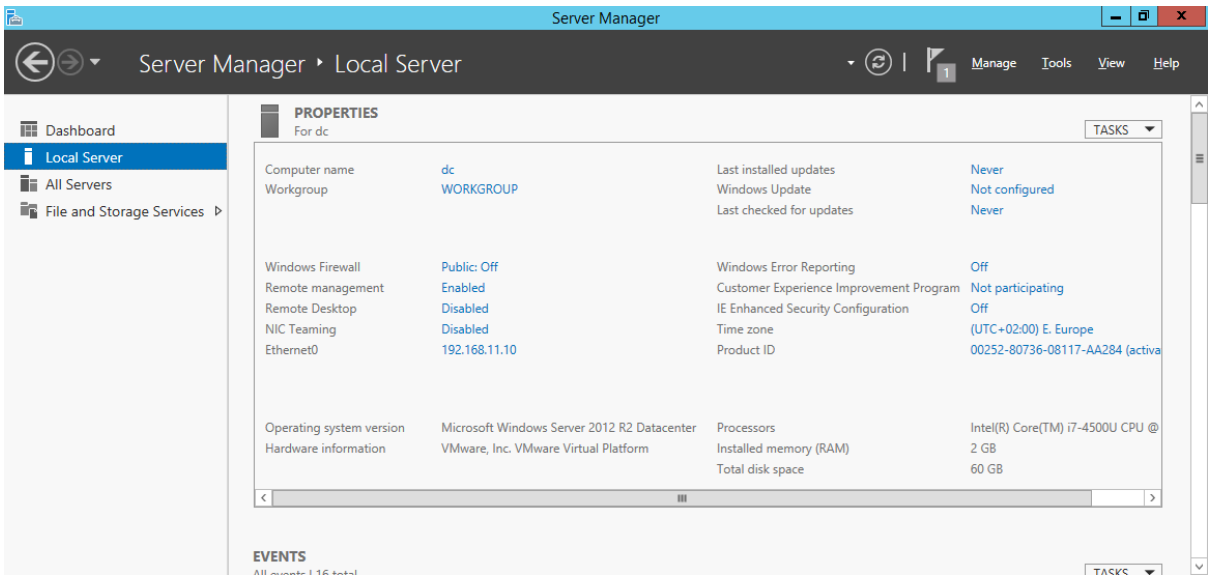

Figure 28: Domain Install 1

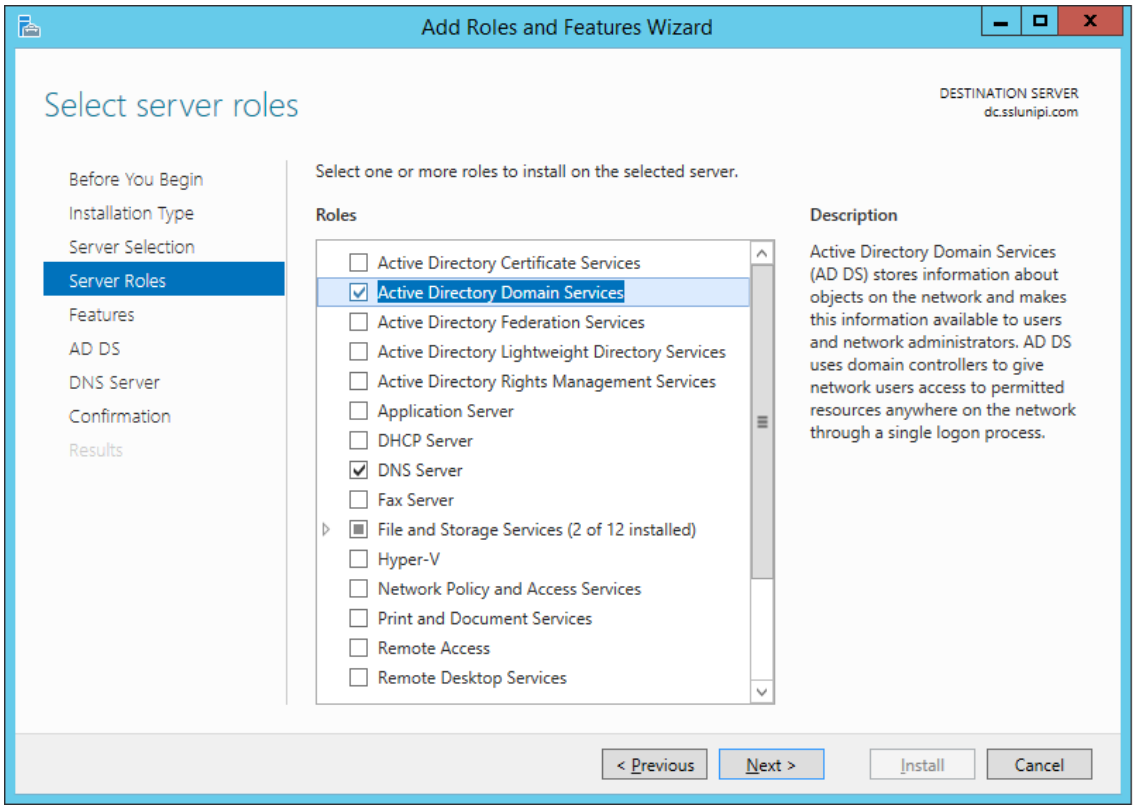

Figure 29: Domain Install 2

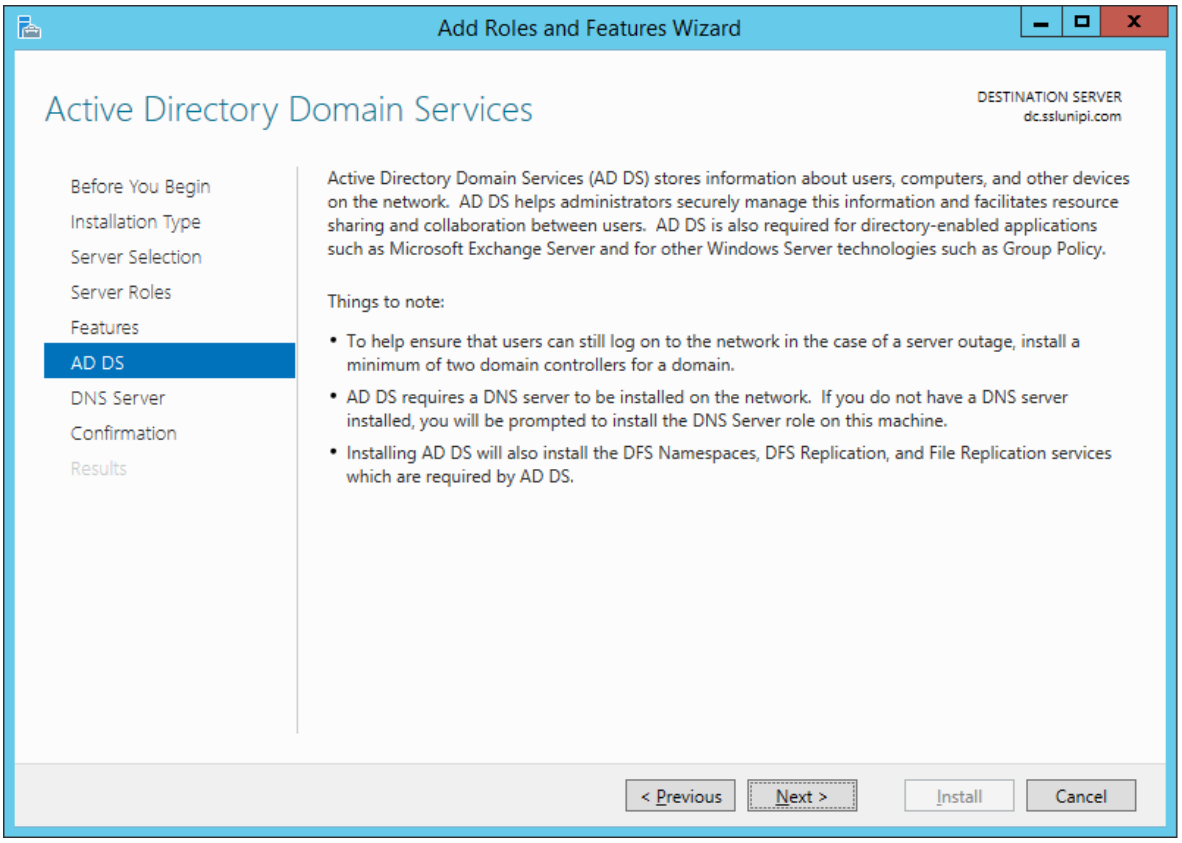

Figure 30: Domain Install 3

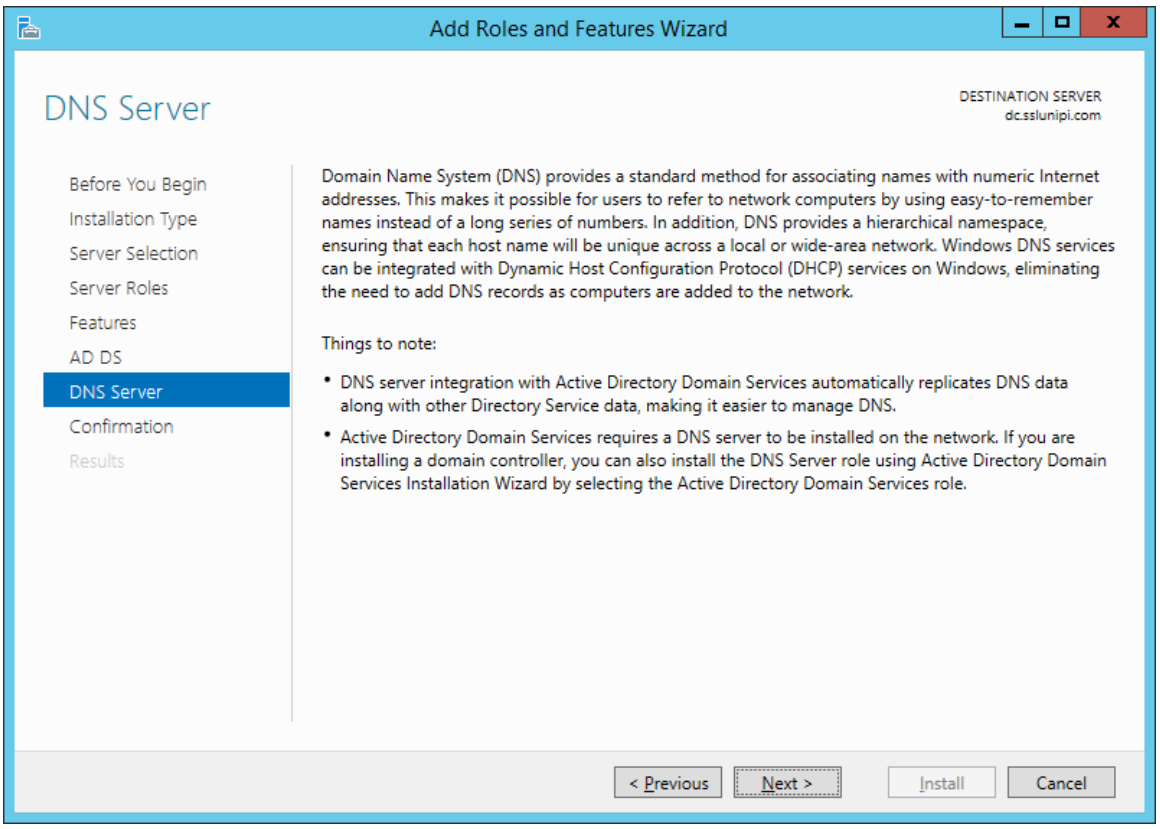

Figure 31: Domain Install 4

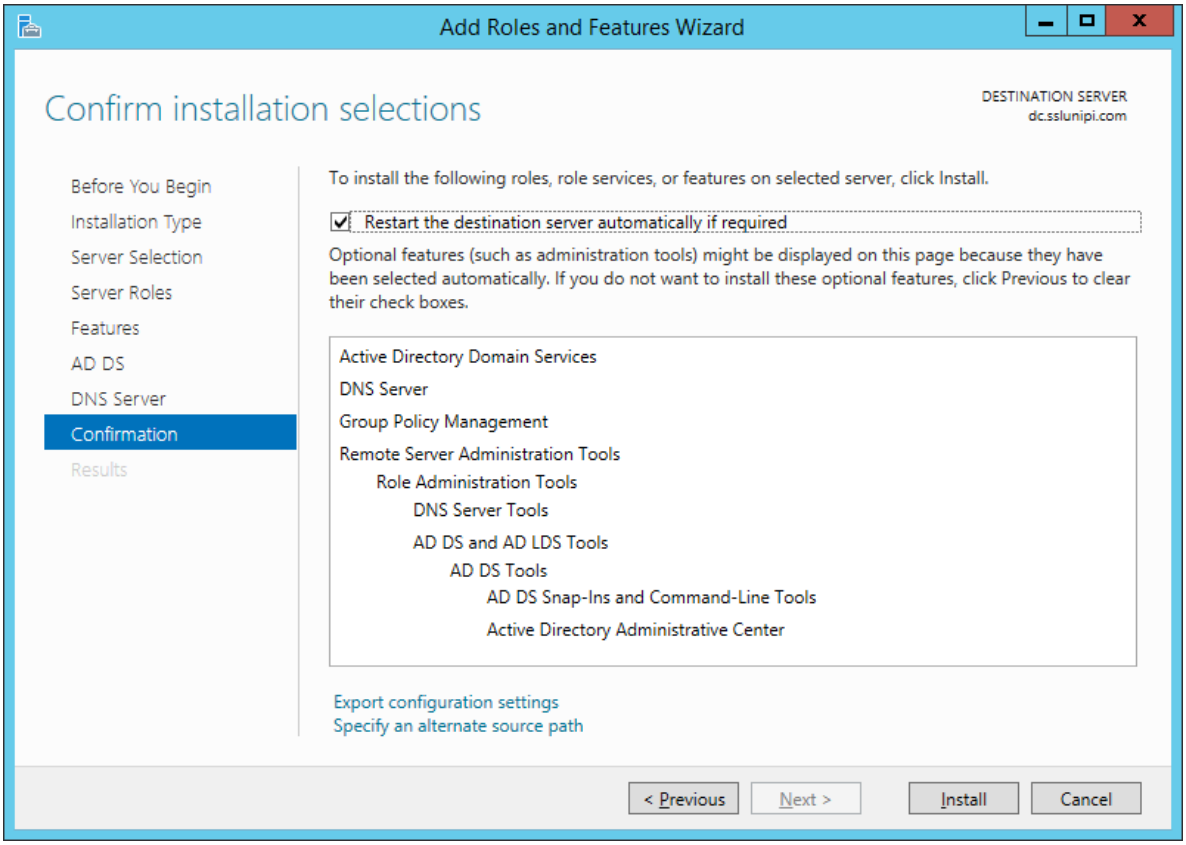

Figure 32: Domain Install 5

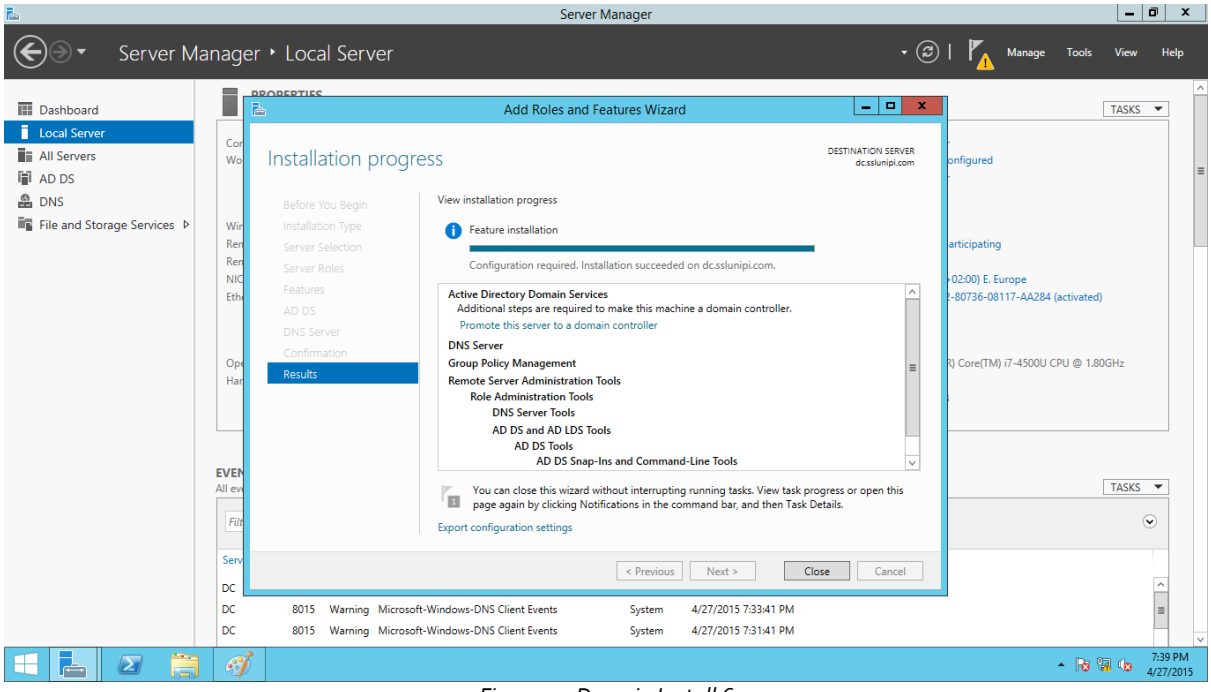

Figure 33: Domain Install 6

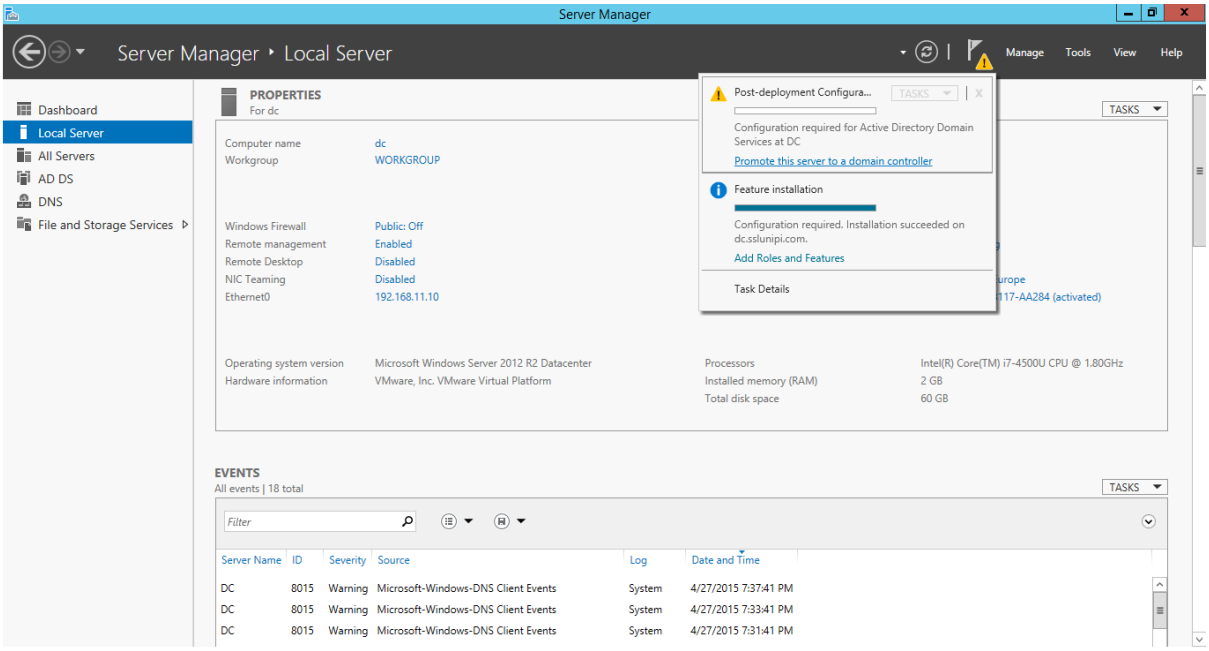

Figure 34: Domain Install 7

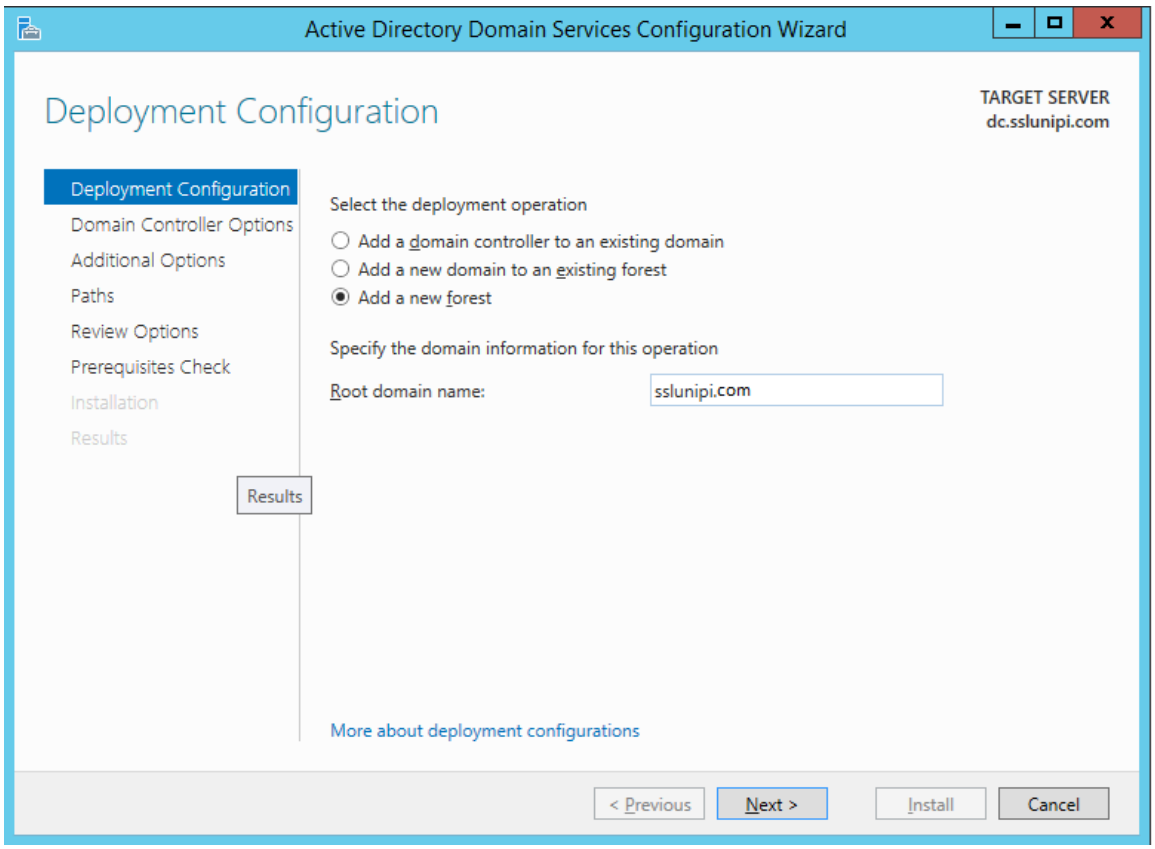

Figure 35: Domain Install 8

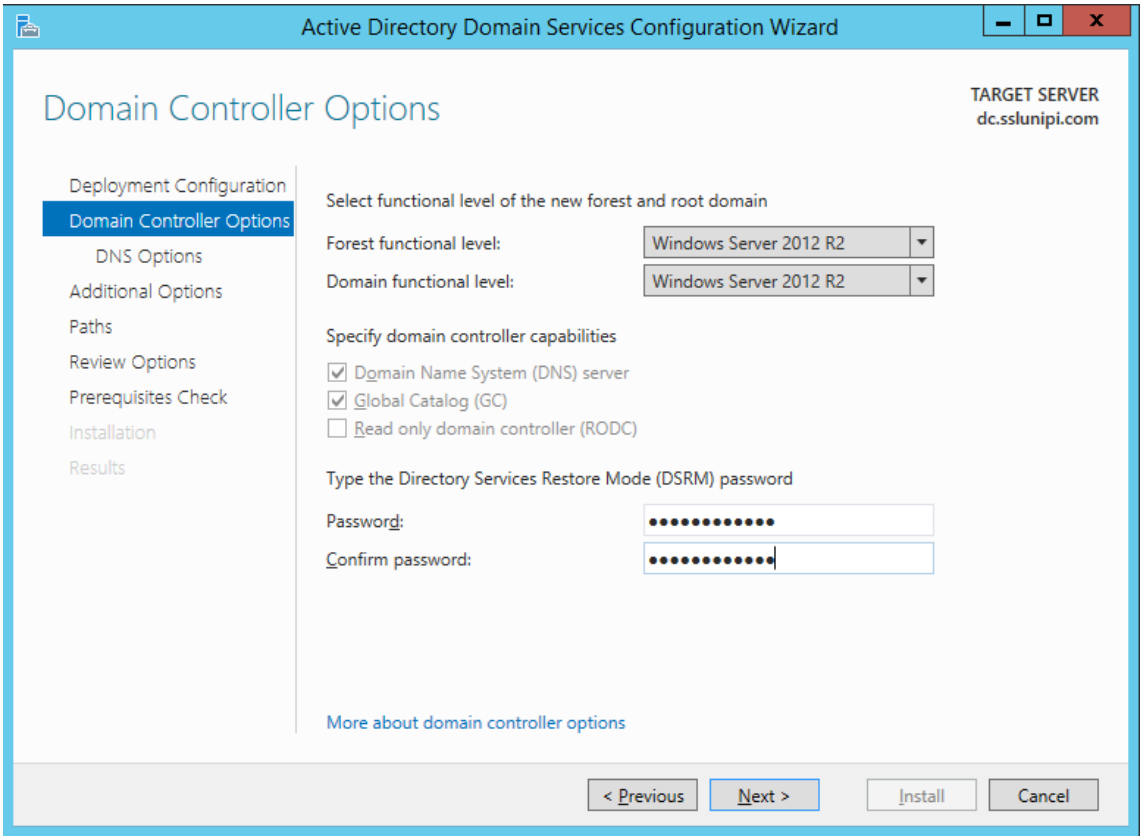

Figure 36: Domain Install 9

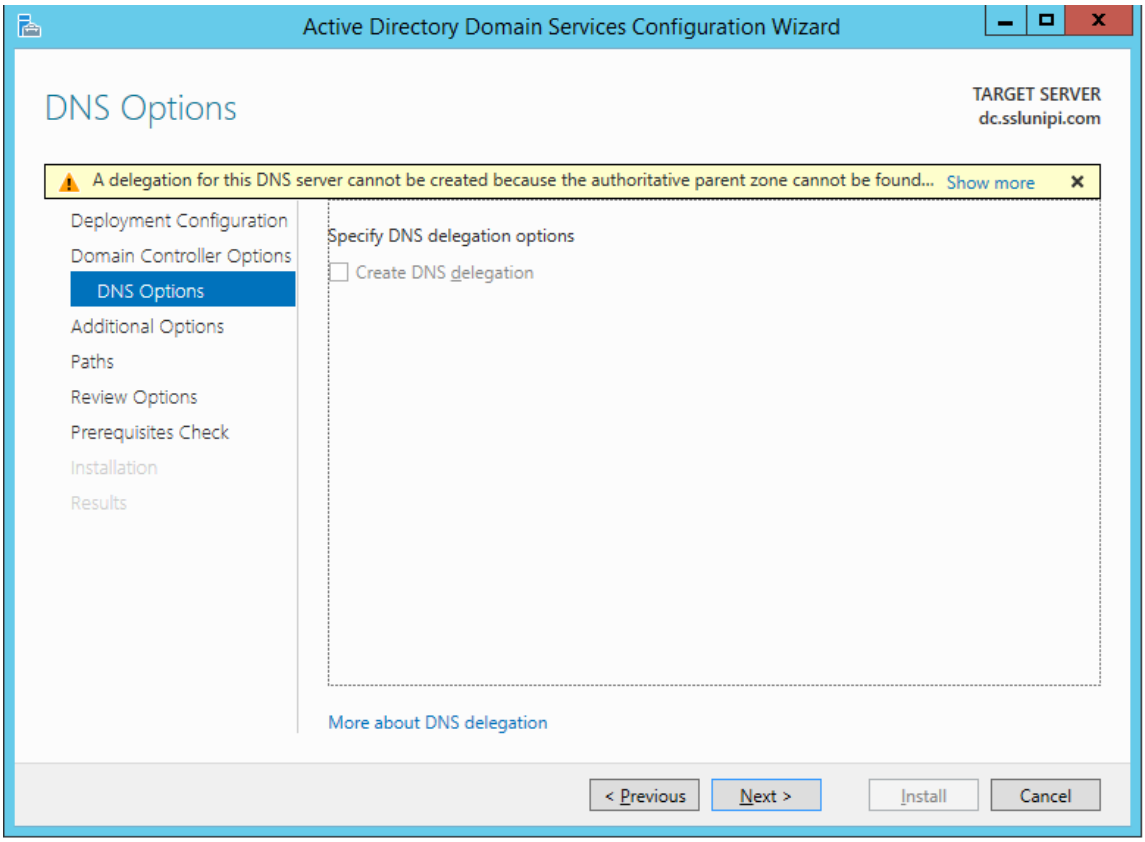

Figure 37: Domain Install 10

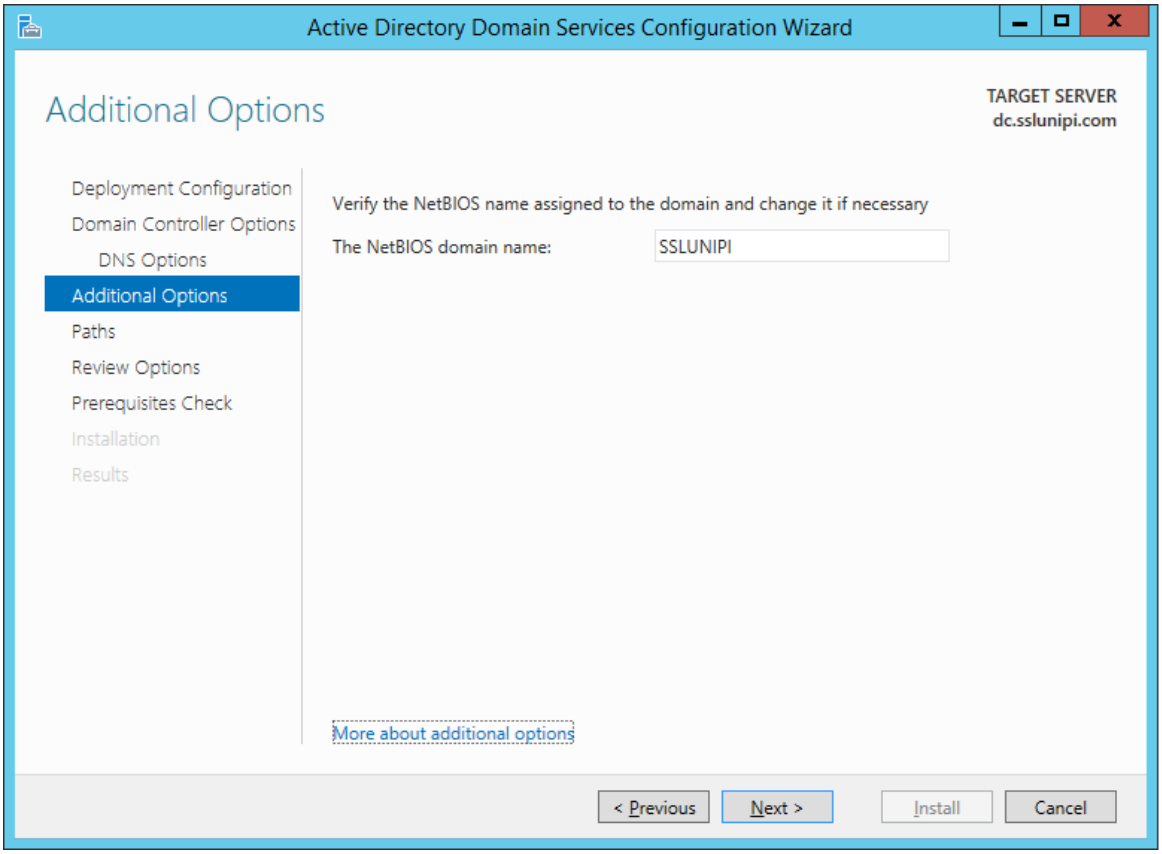

Figure 38: Domain Install 11

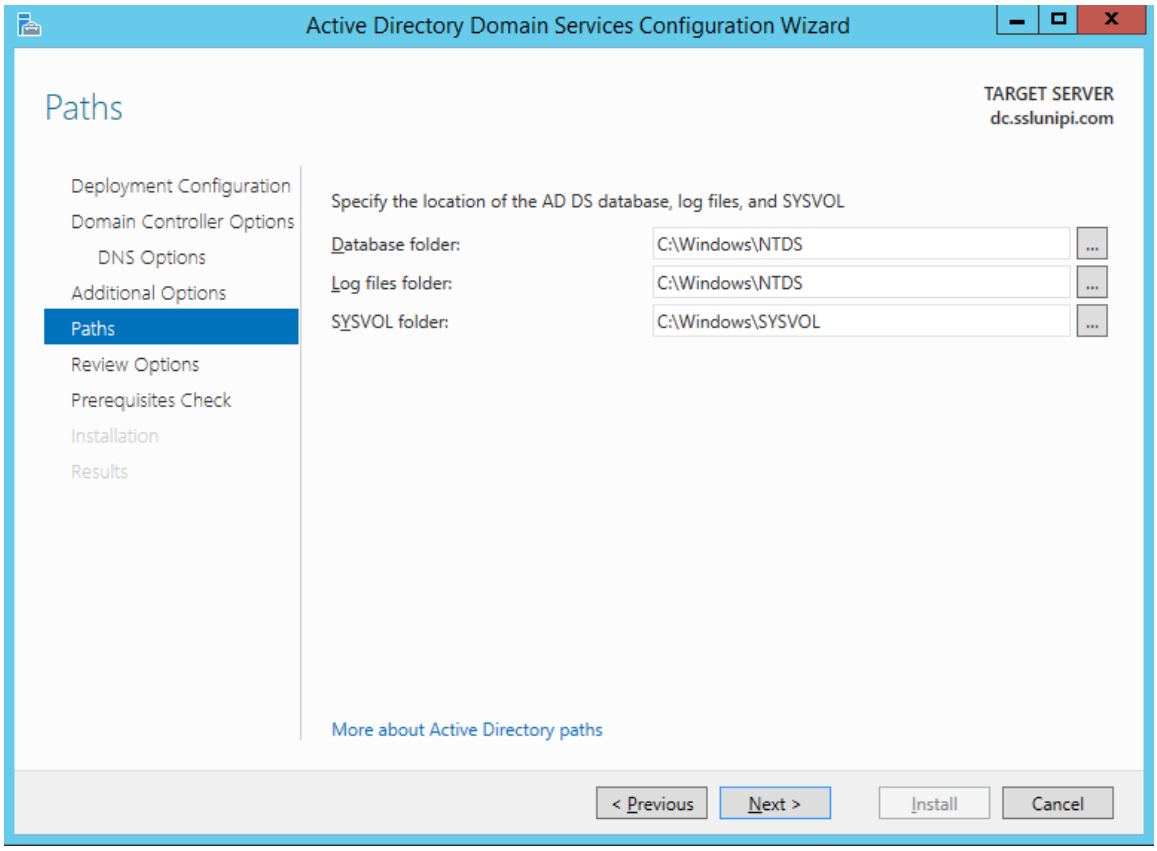

Figure 39: Domain Install 12

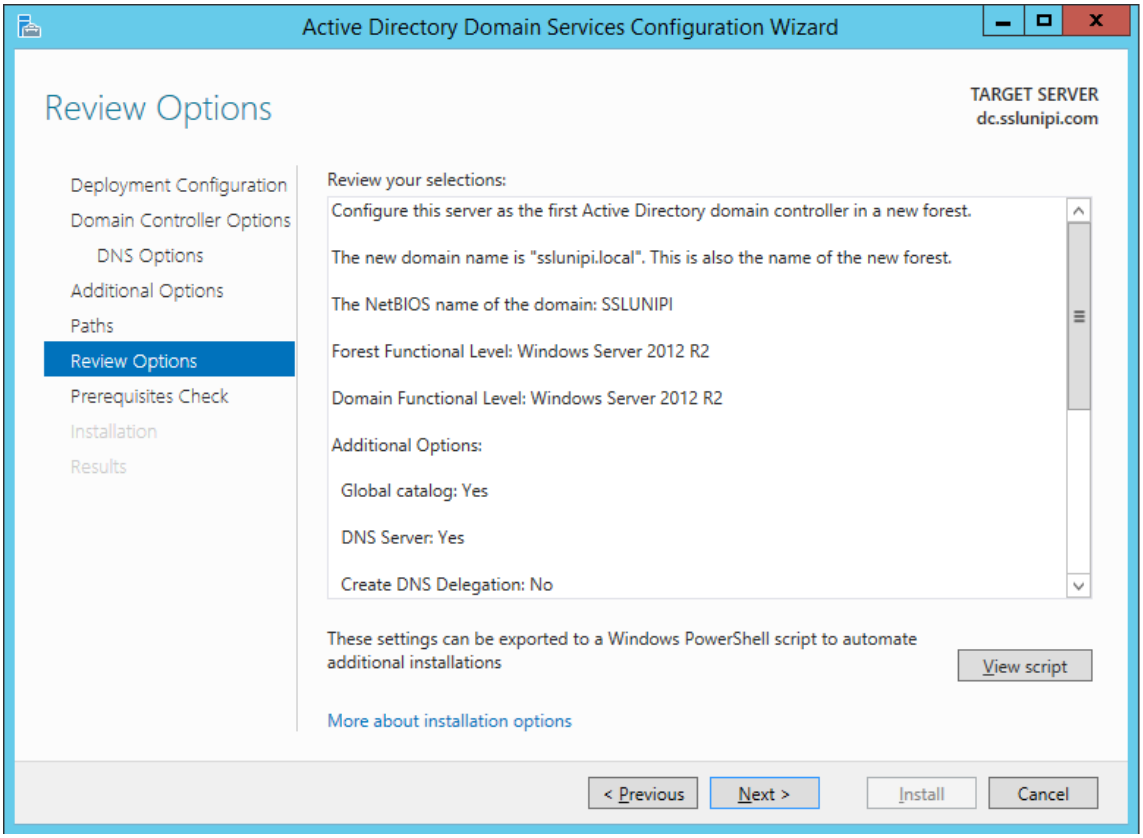

Figure 40: Domain Install 13

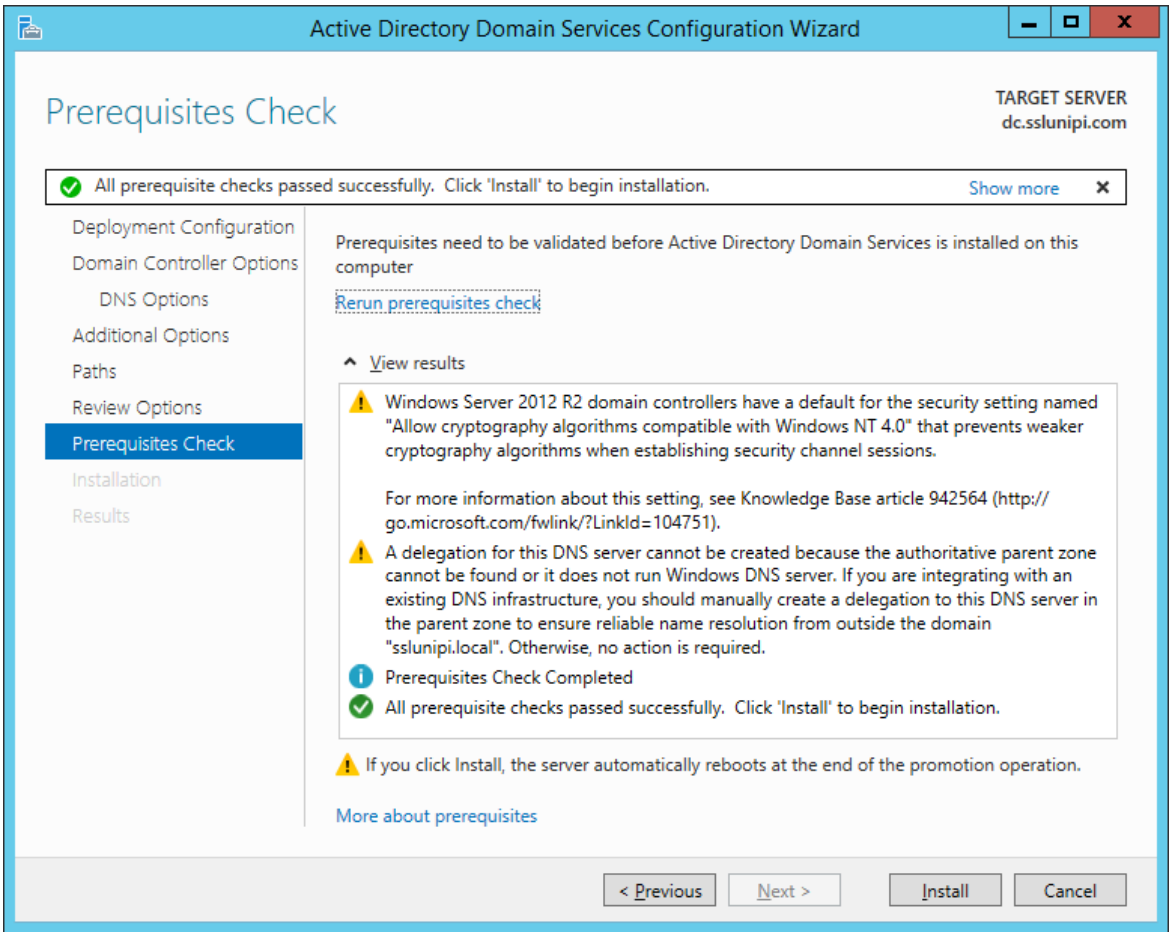

Figure 41: Domain Install 14

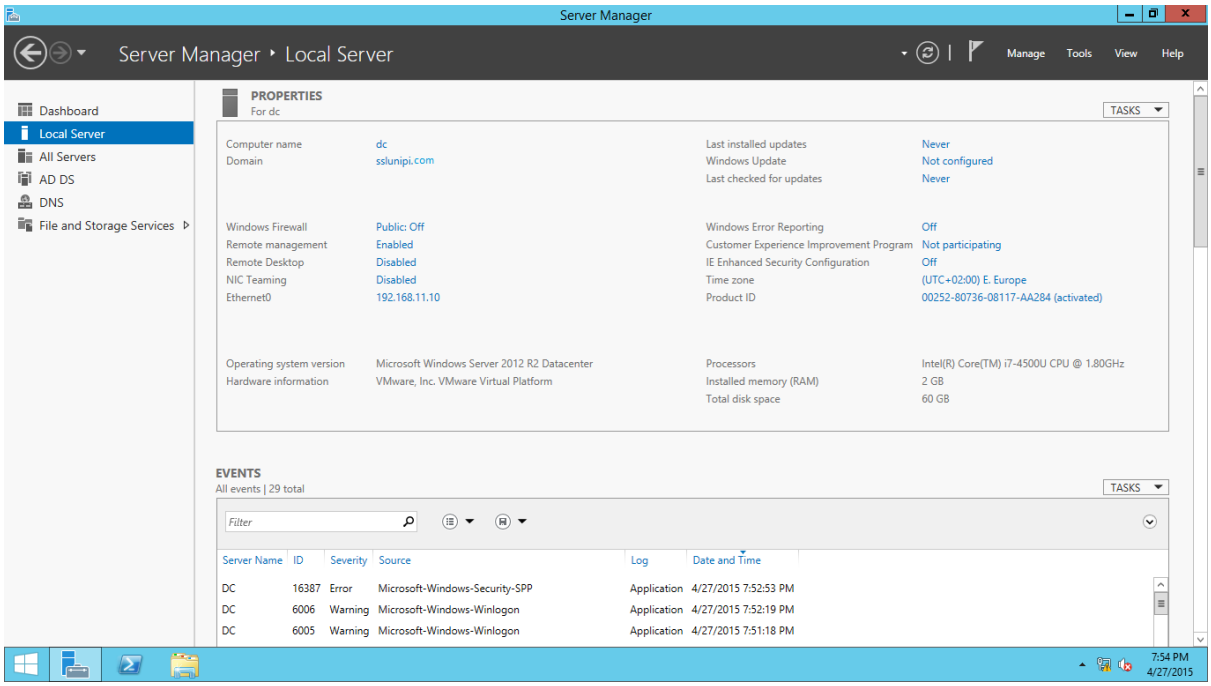

Figure 42: Domain Install 15

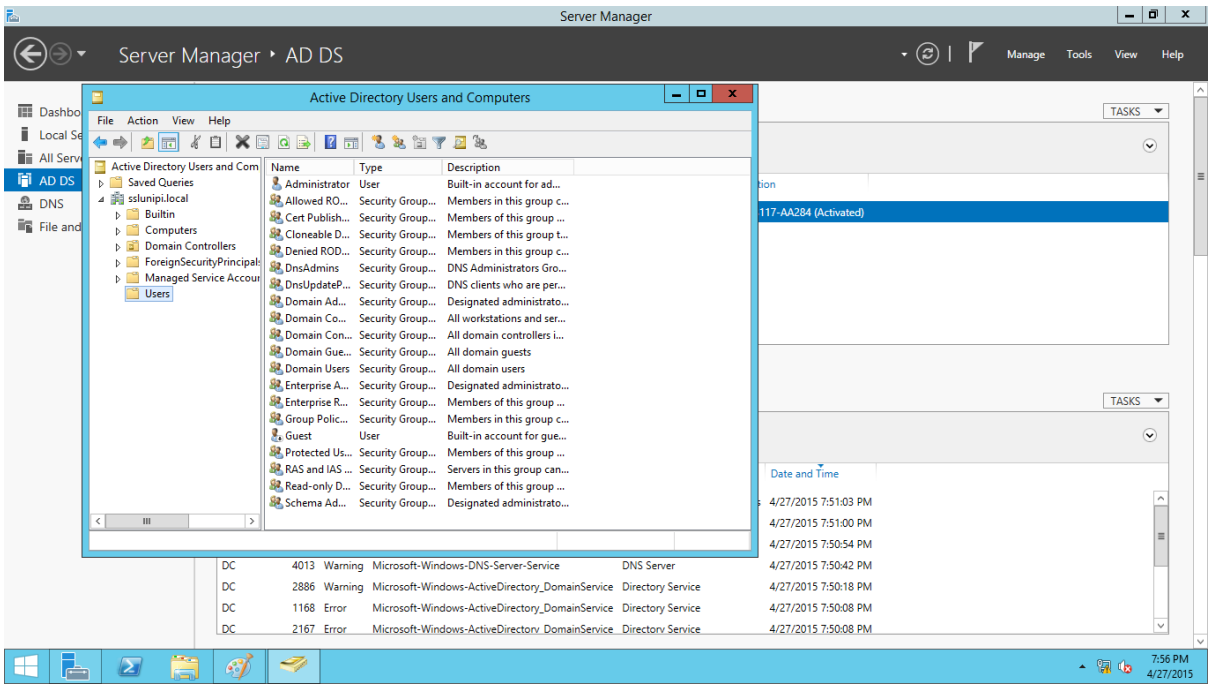

Figure 43: Domain Install 16

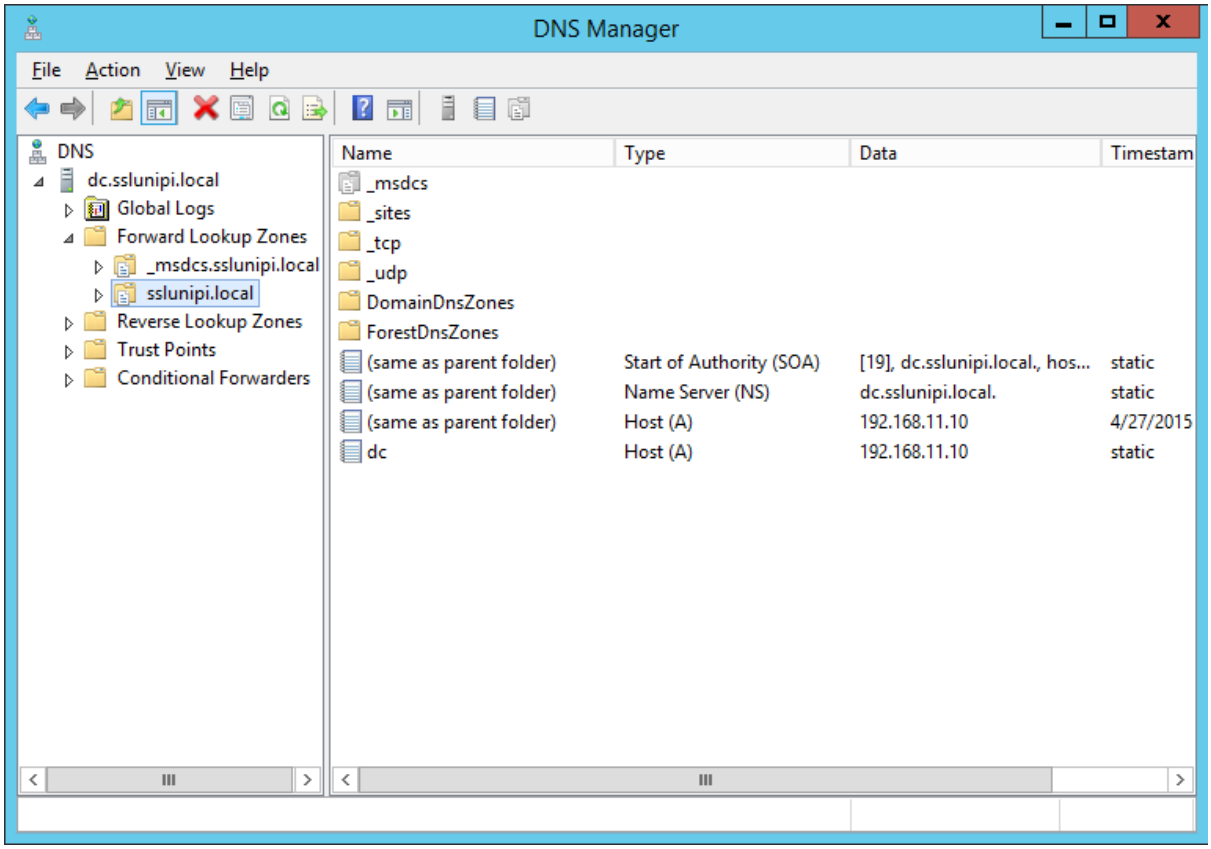

Figure 44: Domain Install 17

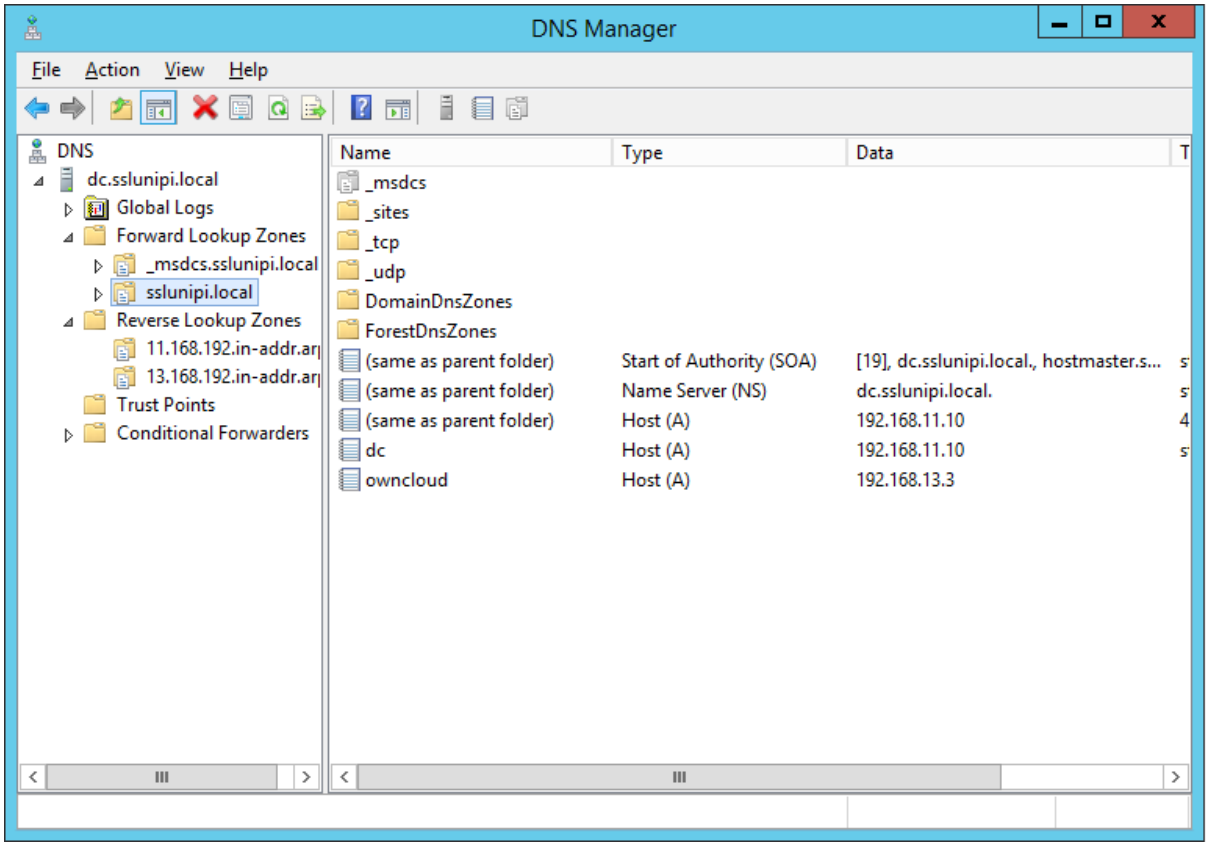

Figure 45: Domain Install 18

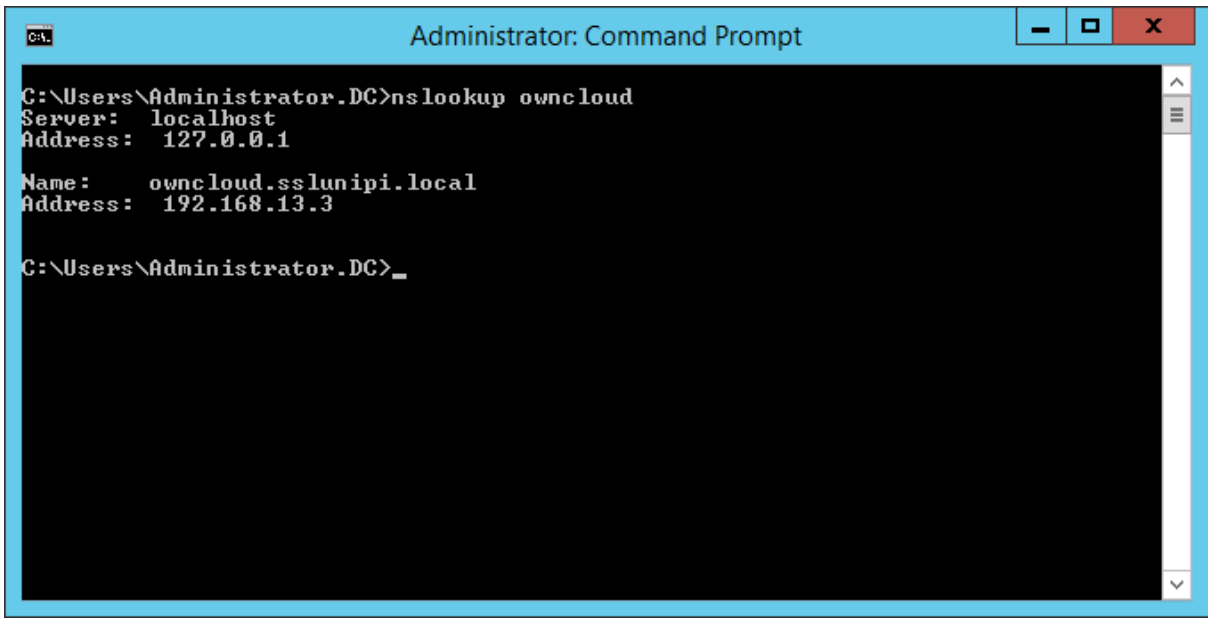

Figure 46: Domain Install 19

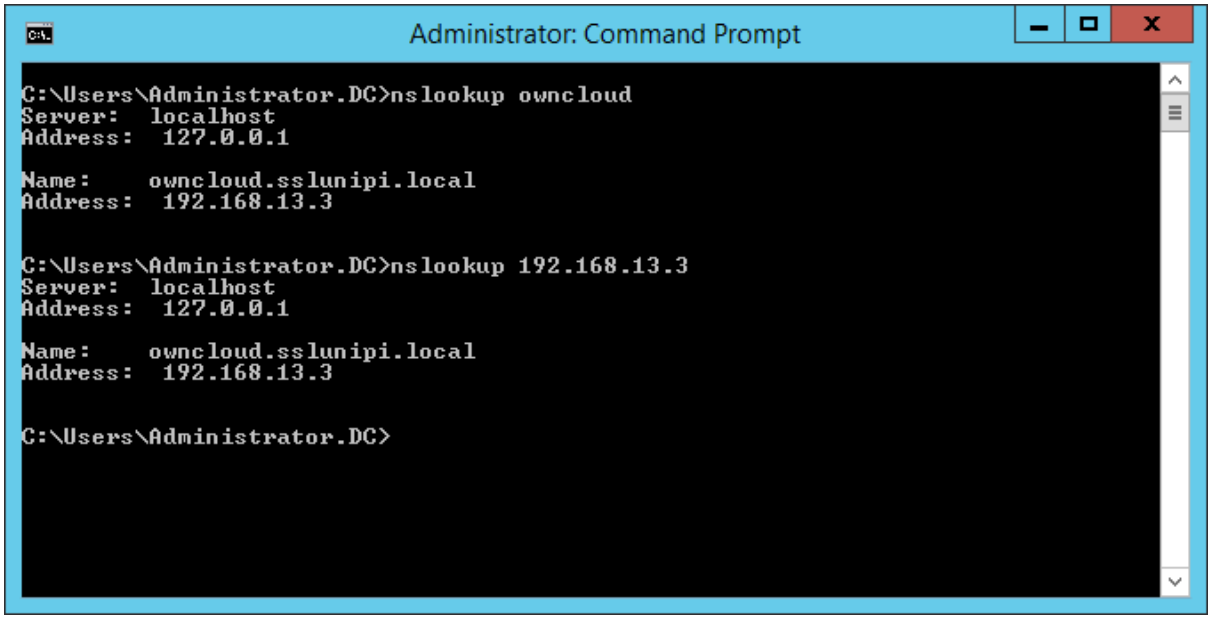

Figure 47: Domain Install 20

Credentials for the Domain Users:

Username	Password	Role
Administrator	qwe123!@#QWE	Domain Administrator
Smandilas	qwe123!@#	Domain Administrator
ssluser1	Simple_Password	Domain User
ssluser2	Simple_Password	Domain User
ssluser3	Simple_Password	Domain User

Computers registered to the domain

Computer Name	IP	Services on Server	OS
dc.sslunipi.com	192.168.11.10	Active Directory DNS	Windows Server 2012 R2
db.sslunipi.com	192.168.11.30	MySQL	Windows Server 2008 R2
sqlbox.ssluni.com (SQLBox on windows)	192.168.13.2	IIS	Windows Server 2008 R2
win7.sslunipi.com	192.168.11.60		Windows 7

3.3.2 Domain SQLBOX

SQLBox is a vulnerable set of sites created for Web Penetration testing by Anastasios Stasinopoulos © and being used by the team of postgraduate students in order to test their Web Pen Testing skills. The initial SQLBox has been based on a CentOS operating system with an embedded database.

In order to create some more scenarios for this thesis, we have moved the vulnerable sites to a windows system and to the IIS web server instead of apache, so that the users

could exploit the vulnerable sites and gain shell on a windows system that is registered on a domain. The SQLBox vulnerable platform has different levels of exploitation for test SQL Injection and LFI skills of the students. Apart from changing the operating system, the Database has been moved from the server itself and was moved in to the LAN on another windows server, so the attackers could possible jump from the one system to the other by using remote MySQL commands.

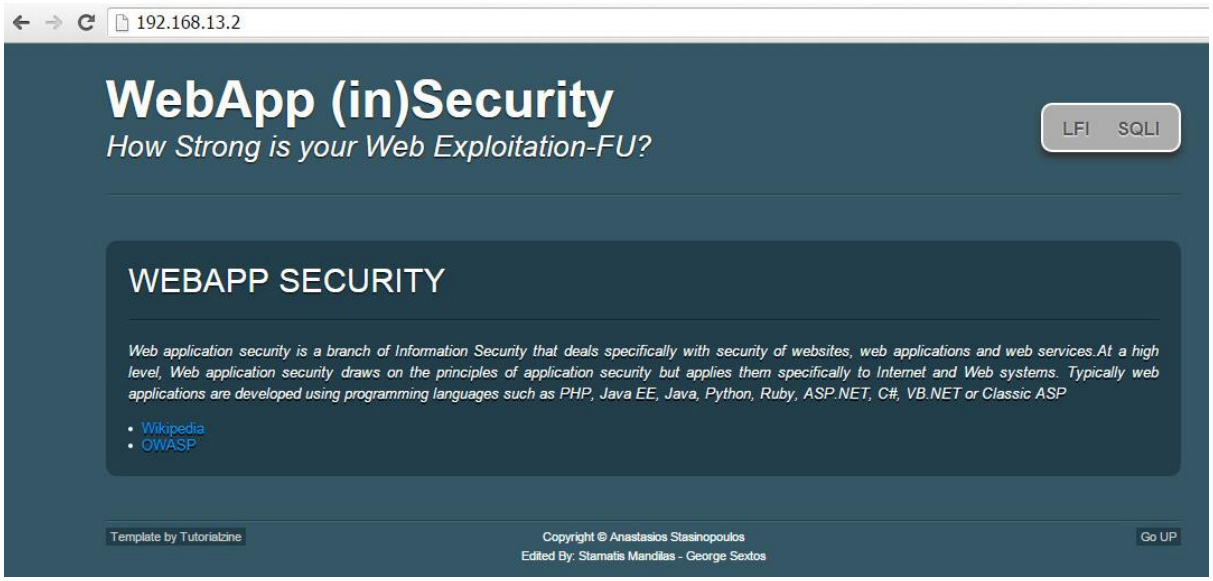

Figure 48: SQLBox Windows 1

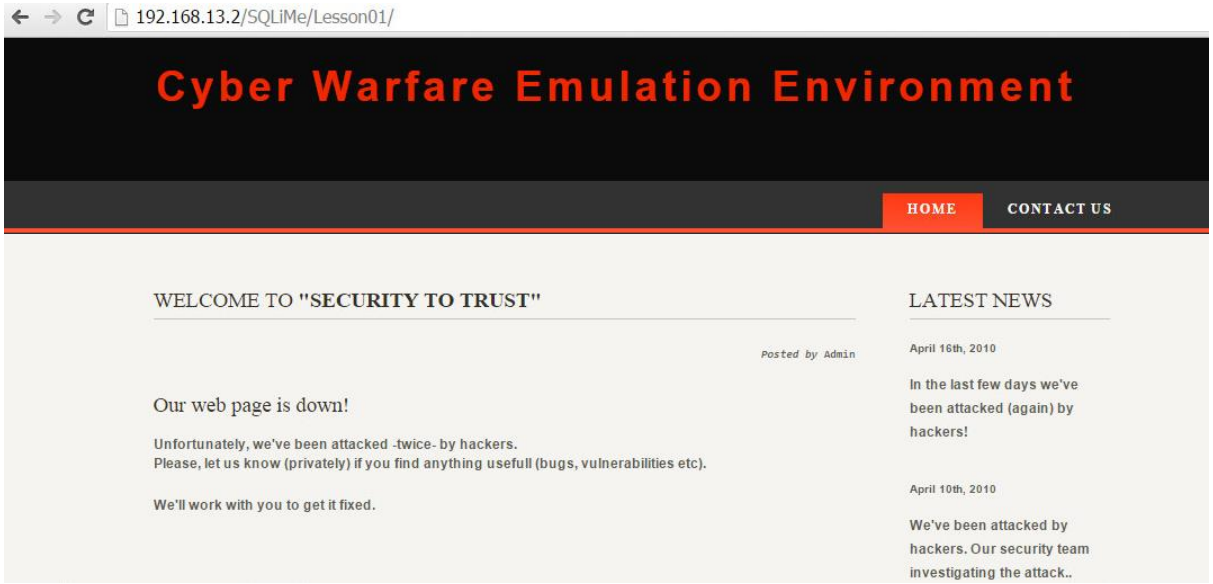

Figure 49: SQLBox Windows 2

3.3.3 Domain DB

As described previously, the specific database has been created in order to support the SQLBox application, which requires the existence of a MySQL database. The decision to place the specific database on the LAN network has been take in order to be used as a jumping point from the DMZ zone to the LAN network.

The MySQL database has been created using an old version of Xamp in order to add as much vulnerability as possible to actual database environment. The versions of all installed components can be found bellow:

Apache Friends XAMPP (Basis Package) version 1.7.3

1. + Apache 2.2.14 (IPV6 enabled)
2. + MySQL 5.1.41 (Community Server) with PBXT engine 1.0.09-rc
3. + PHP 5.3.1 (PEAR, Mail_Mime, MDB2, Zend)
4. + Perl 5.10.1 (Bundle::Apache2, Apache2::Request, Bundle::Apache::ASP, Bundle::Email, Bundle::DBD::mysql, DBD::SQlite, Randy Kobes PPM)
5. + XAMPP Control Version 2.5.8 (ApacheFriends Edition)
6. + XAMPP CLI Bundle 1.6
7. + XAMPP Port Check 1.5
8. + XAMPP Security 1.1
9. + SQLite 2.8.17
10. + SQLite 3.6.20
11. + OpenSSL 0.9.8l
12. + phpMyAdmin 3.2.4
13. + ADOdb v5.10
14. + FPDF v1.6
15. + Zend Framework 1.9.6 Minimal Package (via PEAR)
16. + Mercury Mail Transport System v4.72
17. + msmtp 1.4.19 (a sendmail compatible SMTP client)
18. + FileZilla FTP Server 0.9.33
19. + Webalizer 2.21-02 (with GeoIP lite)
20. + apc 3.1.3p1 for PHP
21. + eAccelerator 0.9.6-rc1 for PHP
22. + Ming 0.4.3 for PHP
23. + PDF with pdflib lite v7.0.4p4 for PHP
24. + rar 2.0.0-dev for PHP
25. + Xdebug 2.0.6-dev for PHP
26. + libapreq2 v2.12 (mod_apreq2) for Apache

MySQL Credentials:

Username	Password
dbowner	t3ms3cDB0wn3r
root	yuJuBJPaWDv4Spuv

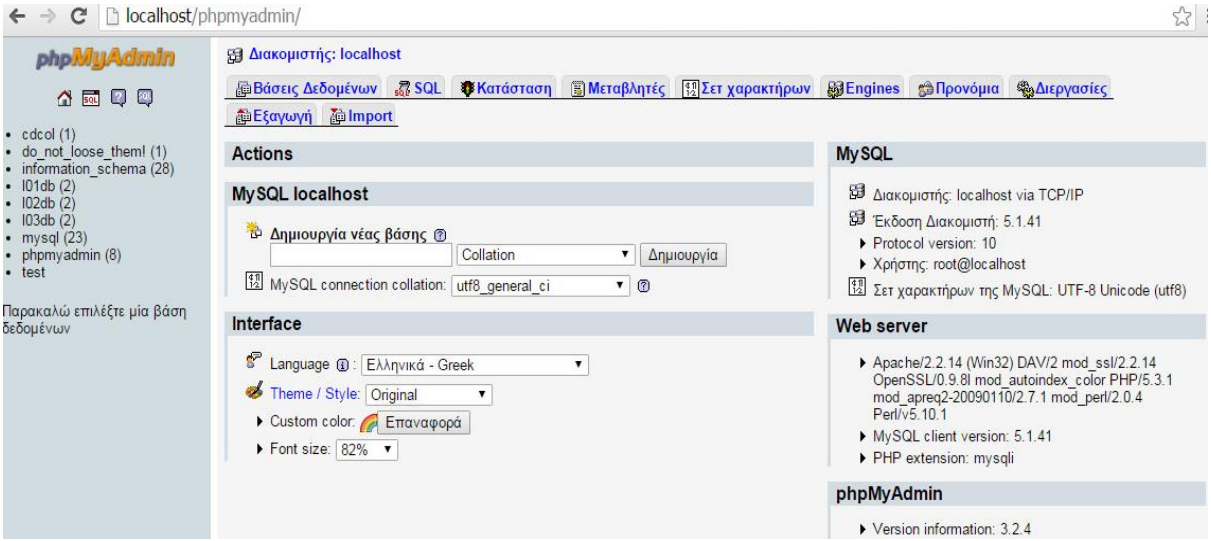

Figure 50: DB phpmyadmin

3.4 SnortIDS

Primarily, an IDS is concerned with the detection of hostile actions. There are two main techniques that are being used by an IDS. The first one, **anomaly detection**, explores issues in intrusion detection associated with deviations from normal system or user behavior.

The second employs **signature detection**, this is the technique that we used, to discriminate between anomaly or attack patterns (signatures) and known intrusion detection signatures. Both methods have their distinct advantages and disadvantages as well as suitable application areas of intrusion detection.

Basic Characteristics of Signature-based IDS:

A signature-based IDS examines ongoing traffic, activity, transactions, or behavior for matches with known patterns of events specific to known attacks. A signature-based IDS requires access to a current database of attack signatures to actively compare and match current behavior against a large collection of signatures. Except when entirely new, uncataloged attacks occur, this technique works extremely well. Other advantages of signature-based IDS: very low false alarm rate, simple algorithms, easy creation of attack signature databases, easy implementation and typically minimal system resource usage.

When considering the application area being the source of data used for intrusion detection, it's possible to distinguish IDSes on the basis of the kinds of activities, traffic, transactions, or systems they monitor. In this case, IDSes may be divided into **network-based** (**NIDS**), **host-based** (**HIDS**), and **application-based** IDS types. IDSes that exploit information obtained from a whole segment of a local network and look for attack signatures are called network-based IDSes, whereas those that operate on hosts defend and monitor the operating and file systems for signs of intrusion and are called host-based IDSes. Some IDSes monitor only specific applications and are called application-based IDSes. Snort is a HIDS product and so we will focus only on this type of classification.

Basic Characteristics of Host-based IDS:

Host-based IDS can analyze activities on the host it monitors at a high level of detail; it can often determine which processes and/or users are involved in malicious activities. Though they may each focus on a single host, many host-based IDS systems use an agent-console model where agents run on (and monitor) individual hosts but report to a single centralized console (so that a single console can configure, manage, and consolidate data from numerous hosts). Host-based IDSes can detect attacks undetectable to the network-based IDS and can gauge attack effects quite accurately. Host-based IDSes can use host-based encryption services to examine encrypted traffic, data, storage, and activity. Host-based IDSes have no difficulties operating on switch-based networks, either.

Some primary types of HIDS can be distinguished:

- Systems that monitor incoming connection attempts (RealSecure Agent, PortSentry). These examine host-based incoming and outgoing network connections. These are particularly related to the unauthorized connection attempts to TCP or UDP ports and can also detect incoming port scans.

- Systems that examine network traffic (packets) that attempts to access the host. These systems protect the host by intercepting suspicious packets and looking for aberrant payloads (packet inspection).

- Systems that monitor login activity onto the networking layer of their protected host (HostSentry). Their role is to monitor log-in and log-out attempts, looking for unusual activity on a system occurring at unexpected times, particular network locations or detecting multiple login attempts (particularly failed ones).

- Systems that monitor actions of a super-user (root) who has the highest privileges (LogCheck). IDS scans for unusual activity, increased super-user activity or actions performed at particular times, etc.

- Systems that monitor file system integrity (Tripwire, AIDE). Tools that have this ability (*integrity checker*) allow the detection of any changes to the files that are critical for the operating system.

- Systems that monitor the system register state (Windows platform only). They are designed to detect any illegal changes in the system register and alert the system administrator to this fact.

Kernel based intrusion detection systems [Els00]. These are especially prevalent within Linux (LIDS, OpenWall). These systems examine the state of key operating system files and streams, preventing buffer overflow, blocking unusual interprocess communications, preventing an intruder from attacking the system. In addition, they can block a part of the actions undertaken by the super-user (restricting privileges).

The HIDS reside on a particular computer and provide protection for a specific computer system. They are not only equipped with system monitoring facilities but also include other modules of a typical IDS, for example the response module. Snort performs this type of monitoring.

Beginning with version 2.9, Snort uses a new mechanism for capturing packets. The Data Acquisition (DAQ) libraries were first announced in August 2010 on the VRT Blog. Previously, Snort integrated packet capture and other network functions. With DAQ, these functions have been separated out into modules that can be selected when invoking Snort. DAQ was created to integrate inline functionality and to provide flexibility for new modules.

The separation of DAQ also allows developers to create their own modules. There are already three externally developed DAQ modules – PF_RING, Napatech and PCAPRR. DAQ has two prerequisites: libpcap for packet capture and libdnet for other network functions. Snort no longer uses libnet -- which hasn't been maintained or updated for many years (Roberts) -- for packet construction.

Four of the six DAQ modules allow Snort to operate inline and drop packets. Previously, to use this functionality you had to compile Snort with a patch. The “inline” term generally refers to the position of the sensor placed in between two devices or networks.

These are the DAQ modules included with Snort:

Pcap: the default mode, used for sniffer and IDS modes
Afpacket: inline on Linux using two bridged interfaces
Ipq: inline on Linux using netfilter, replaces the snort_inline project patch
Nfq: inline on Linux using netfilter
Ipfw: inline on OpenBSD and FreeBSD using divert sockets with the pf and ipfw firewalls
Dump: allows testing of inline and normalization mechanisms

SOFTWARE ASSETS

Some basic information system assets required for running Snort.

Software Assets:
OS: Linux based OS
Snort: As intrusion detection system.
BASE: Basis analysis and security engine (Graphical detection Engine).
MySQL: Database to log of alerts and intrusions.
PHP: To setup up base on browser.
Pear packages: To set Graphical environment on BASE.
Libpcap: To set up network adapter on packet capture mode on network.(Win cap in case of windows environment).
Ado Db: To setup connectivity between BASE and mysql.
Apache2: To run the system as a server on network (having static IP address), running on ports 80 and 443 for the web.

Threats

A threat requires vulnerability. This means that in order for a threat to become realistic, the associated vulnerability must exist, and be exploitable. Most threats can never be eliminated but the vulnerabilities that they exploit can be controlled, thereby reducing the probability of the threat becoming an incident. In order to control these threats, they must first be identified.

Since intrusion detection systems deal with potential threats on the system, let us further discuss of their taxonomy. Defending against potential attackers is much easier when the attack tactics are known. Listed below are just a few of the many types of hacking tactics that a systems administrator may have to face:

Those related to unauthorized access to the resources (often as introductory steps toward more sophisticated actions):

- Password cracking and access violation,
- Trojan horses,
- Interceptions; most frequently associated with TCP/IP stealing and interceptions that often employ additional mechanisms to compromise operation of attacked systems (for example by flooding); man in the middle attacks),
    - Spoofing (deliberately misleading by impersonating or masquerading the host identity by placing forged data in the cache of the named server i.e. DNS spoofing)
    - Scanning ports and services, including ICMP scanning (Ping), UDP, TCP Stealth Scanning TCP that takes advantage of a partial TCP connection establishment protocol.) etc.
    - Remote OS Fingerprinting, for example by testing typical responses on specific packets, addresses of open ports, standard application responses (banner checks), IP stack parameters etc.,
    - Network packet listening (a passive attack that is difficult to detect but sometimes possible),
    - Stealing information, for example disclosure of proprietary information,
    - Authority abuse; a kind of internal attack, for example, suspicious access of authorized users having odd attributes (at unexpected times, coming from unexpected addresses),
    - Unauthorized network connections,
    - Usage of IT resources for private purposes, for example to access pornography sites,
    - Taking advantage of system weaknesses to gain access to resources or privileges,
- Unauthorized alteration of resources (after gaining unauthorized access):
    - Falsification of identity, for example to get system administrator rights,
    - Information altering and deletion,
    - Unauthorized transmission and creation of data (sets), for example arranging a database of stolen credit card numbers on a government computer (e.g. the spectacular theft of several thousand numbers of credit cards in 1999),
    - Unauthorized configuration changes to systems and network services (servers).
- Denial of Service (DoS):
    - Flooding – compromising a system by sending huge amounts of useless information to lock out legitimate traffic and deny services:
    - Ping flood (Smurf) – a large number of ICMP packets sent to a broadcast address,
    - Send mail flood - flooding with hundreds of thousands of messages in a short period of time; also POP and SMTP relaying,
    - SYN flood – initiating huge amounts of TCP requests and not completing handshakes as required by the protocol,
    - Distributed Denial of Service (DDoS); coming from a multiple source,
    - Compromising the systems by taking advantage of their vulnerabilities:
    - Buffer Overflow, for example Ping of Death — sending a very large ICMP (exceeding 64 KB),
    - Remote System Shutdown,
- Web Application attacks; attacks that take advantage of application bugs may cause the same problems as described above.

It is important to remember, that most attacks are not a single action, rather a series of individual events developed in a coordinated manner.

SNORT

3.4.1 What is Snort

Snort is a free and open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS) initially created by Martin Roesch in 1998 but now developed by sourcefire, of which Roesch is the founder and CTO. In 2009, snort entered InfoWorld`s Open Source Hall of Fame as one of the “greatest *pieces of+ open source software of all time”. Snort has the ability to perform real-time traffic analysis and packet login on Internet Protocol (IP) networks. Moreover has the ability to perform protocol analysis, content searching, and content matching. IT implements many services which can be used to detect probes or attacks and has plenty of mods that snort can be configured to run. Finally, snort can be installed to a variety of Operating Systems (OS). All this functionalities will be analyzed in the next section.

3.4.2 Requirements of snort.

Snort requirements in order to achieve basic functionality are Libpcap, PCRE, Libdnet, Barnyard2 and DAQ, For windows users only WinPcap and barnyard are required, but what are all those libraries and programs? Let`s analyze them one by oneLibpcap (or WinPcap for windows) .

In the field of computer network administration, pcap (packet capture) consists of an application programming interface (API) for capturing network traffic. Unix-like systems implement pcap in the libpcap library; Windows uses a port of libpcap known as WinPcap.
Monitoring software may use libpcap and/or WinPcap to capture packets traveling over a network. libpcap and WinPcap also support saving captured packets to a file and reading files containing saved packets. Snort uses these files to read network traffic and analyze it.
PCRE (Perl Compatible Regular Expressions)

Is a regular expression C library inspired by Perl’s external interface, written by Philip Hazel. The PCRE library is incorporated into a number of prominent open-source programs such as the Apache HTTP Server, the PHP and R scripting languages, and Snort.
A version of PCRE > 8.31 is recommended

- **Libdnet**

Libdnet is a generic networking API that provides access to several protocols.

- **Barnyard2**

Barnyard is an output system for Snort. Snort creates a special binary output format called *unified.* Barnyard2 reads this file, and then resends the data to a database back-end. Unlike the database output plug-in, Barnyard2 manages the sending of events to the database and stores them when the database temporarily cannot accept connections.

- **DAQ**

DAQ is the Data-Acquisition API that is necessary to use Snort version 2.9.0 and above. Some network cards have features named "Large Receive Offload" (LRO) and "Generic Receive Offload" (GRO). With these features enabled, the network card performs packet

reassembly before they're processed by the kernel. By default, Snort will truncate packets larger than the default snaplen of 1518 bytes. In addition, LRO and GRO may cause issues with Stream5 target-based reassembly. We recommend that you turn off LRO and GRO.
Network interface

Finally a capturing interface is required like Ethernet or Wireless interface, so snort can monitor traffic on that interface.

- Requirements can be found at http://www.snort.org/start/requirements

3.4.3 Snort modes

1. ***Sniffer mode***, which simply reads the packets off of the network and displays them for you in a continuous stream on the console (screen). Sniffer mode can be configured to just print out the TCP/IP packet headers to the screen, or to see the application data in transit and the headers, or showing the data link layer headers too.

2. ***Packet Logger*** *mode,* which logs the packets to disk in either ASCII format, in a directory hierarchy based upon the IP address of one of the hosts in the datagram or relative to the home network, or in tcpdump format to a single binary file in the logging directory specified. If you're on a high speed network or you want to log the packets into a more compact form for later analysis, you should consider logging in binary mode. Snort can also read the packets back by using the -r switch, which puts it into playback mode. Packets from any tcpdump formatted file can be processed through Snort in any of its run modes.

3. ***Network Intrusion Detection System (NIDS) mode***, which performs detection and analysis on network traffic. This is the most complex and configurable mode. This mode trigger rules written in plain text at snort.conf and logs only the packets matched with some rule specified.

There are many ways to configure NIDS mode, default is full mode that logs in decoded ASCII format and uses full alerts. The full alert mechanism prints out the alert message in addition to the full packet headers. Moreover NIDS can be configured to log in both ASCII and binary format.

3.4.4 Deploying SNORT with Pulledpork, Barnyard2 and Snorby

With the documented installation steps bellowthe 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:

3.4.4.1 Host Preparation

1. sudo su

Update the host with

2. apt-get update && apt-get upgrade

Configure our interfaces

3. nano /etc/network/interfaces

add or edit the following lines to match your network (The address on eth0 will be used for the management and the eth1 will monitor the respective network)

```
# The primary network interface
auto etho
iface etho inet static
address<address you like>
netmask<netmask>
gateway<gateway address>
dns-nameservers<DNS Servers>
# The monitor Interface
auto eth1
iface eth1 inet manual
up ifconfig eth1 up promisc
down ifconfig eth1 down -promisc
```

Reset the interfaces to their new config with the following commands.

4. ifdown eth0
5. ifup eth0
6. ifup eth1

Check both interfaces came up and eth1 lists PROMISC. Make sure that traffic is being captured on eth1.

3.4.4.2 Install Dependencies

1. apt-get -y install build-essential libtool automake gcc flex bison libnet1 libnet1-dev libpcre3 libpcre3-dev autoconf libcrypt-ssleay-perl libwww-perl git zlib1g zlib1g-dev libssl-dev mysql-server libmysqlclient-dev apache2 imagemagick wkhtmltopdf

ruby1.9.3 libyaml-dev libxml2-dev libxslt1-dev openssl libreadline6-dev unzip libcurl4-openssl-dev apache2-threaded-dev libapr1-dev libaprutil1-dev

Install manually the following components

2. mkdir tmp_build
3. cd tmp_build
4. wget https://libdnet.googlecode.com/files/libdnet-1.12.tgz
5. tar zxf libdnet-1.12.tgz
6. cd libdnet-1.12
7. ./configure
8. make && make install
9. cd ..
10. wget http://www.tcpdump.org/release/libpcap-1.6.2.tar.gz
11. tar zxf libpcap-1.6.2.tar.gz
12. cd libpcap-1.6.2
13. ./configure
14. make && make install
15. ldconfig

3.4.4.3 SNORT Installation

1. cd ..
2. wget https://snort.org/downloads/snort/snort-2.9.7.0.tar.gz
3. wget https://snort.org/downloads/snort/daq-2.0.4.tar.gz

Install DAQ (Data Acquisition Library)

1. tar zxf daq-2.0.4.tar.gz
2. cd daq-2.0.4
3. ./configure
4. make && make install
5. ldconfig

Install Snort Engine

1. cd ..
2. tar zxf snort-2.9.7.0.tar.gz
3. cd snort-2.9.7.0
4. ./configure --enable-sourcefire
5. make && make install
6. ln -s /usr/local/lib/libdnet.1.0.1 /usr/lib/libdnet.1

Some directories and users creation for better management

1. groupadd snort
2. useradd snort -d /var/log/snort -s /sbin/nologin -c SNORT_IDS -g snort
3. mkdir /var/log/snort
4. chown snort:snort /var/log/snort
5. mkdir /etc/snort
6. mv etc/* /etc/snort/

As snort needs rules to function, you can register on www.snort.org, register yourselves and download the latest snort rulesets. For this deployment we will use pulledpork, which will automatically download snort rules for us:

1. cd ..
2. wget https://pulledpork.googlecode.com/files/pulledpork-0.7.0.tar.gz
3. tar zxf pulledpork-0.7.0.tar.gz
4. cd pulledpork-0.7.0
5. cp pulledpork.pl /usr/sbin/
6. chmod 755 /usr/sbin/pulledpork.pl
7. cp etc/* /etc/snort/
8. cpan install LWP::Protocol::https
9. cpan install Crypt::SSLeay
10. cpan Mozilla::CA IO::Socket::SSL

At the time of writing pulled pork didn't like grabbing items over SSL, to fix this we set pulled pork to not validate ssl certificates. If pulledpork generates an error 500 you might want to try this as well.

1. nano /usr/sbin/pulledpork.pl

After Vars here and before # we are gonna need these add the following line.

$ENV{'PERL_LWP_SSL_VERIFY_HOSTNAME'}=0;$ENV{'PERL_LWP_SSL_VERIFY_HOSTNAME'} = 0;

Edit the pulled pork conf file

1. nano /etc/snort/pulledpork.conf

Do the following:

- anywhere you see <oinkcode> replace it with your code
- rule_path=/etc/snort/rules/snort.rules
- local_rules=/etc/snort/rules/local.rules

- sid_msg=/etc/snort/sid-msg.map
- snort_path=/usr/local/bin/snort
- config_path=/etc/snort/snort.conf
- distro= Ubuntu-10-4
- black_list=/etc/snort/rules/black_list.rules
- snort_control=/usr/bin/snort_control

Comment out

- #IPRVersion=/usr/local/etc/snort/rules/iplists

Uncomment

- rule_url=https://rules.emergingthreatspro.com/|emerging.rules.tar.gz|open

Before we can pull down the rules we need to configure snort.

1. nano /etc/snort/snort.conf

Set the network you need to monitor

- ipvar HOME_NET <e.g. 192.168.1.0/24>
- ipvar EXTERNAL_NET !$HOME_NET

set rule paths to be absolute

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

Remove all the “include $RULE_PATH/…. “ adding the following two lines.

- include $RULE_PATH/local.rules
- include $RULE_PATH/snort.rules

Set the output format as unified2, which will be used later-on from Barnyard2

- output unified2: filename snort.log, limit 128

Set perimissions on the following paths

1. cd /etc/snort
2. chown -R snort:snort *
3. mkdir -p /usr/local/lib/snort_dynamicrules
4. mkdir /etc/snort/rules
5. touch /etc/snort/rules/so_rules.rules
6. touch /etc/snort/rules/local.rules
7. touch /etc/snort/rules/white_list.rules

With all the config files set lets pull down some rules.

1. pulledpork.pl -c /etc/snort/pulledpork.conf

If pulled pork runs without errors we can set a cron job so it will run every day. Snort needs to be restarted in order to apply new rules. If you prefer to do this manually to avoid possible errors don't add the second cron job.

1. nano /etc/crontab

- 0 0 * * * root /usr/sbin/pulledpork.pl -c /etc/snort/pulledpork.conf
- 0 15 * * * root service snort restart

We can sniff traffic, we have rules and we are configured and run a quick test.

1. snort -T -i ens192 -u snort -g snort -c /etc/snort/snort.conf

Snort – Boot

1. nano /etc/init/snort.conf

add the following in to the file

- description "Snort Service"
- start on runlevel [2345]
- stop on runlevel [!2345]
- script
- exec snort -q -i eth1 -u snort -g snort -c /etc/snort/snort.conf -D
- end script

Set and start the service with

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

Once started check its still running

1. ps -A | grep snort

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 1

On the 1st level of the firewall configuration things are pretty simple for the attackers when the have already compromised a server in the DMZ, as the rules on the firewall configuration, allow pretty much any traffic to all the zones of the internal network. This means, that any logging will not show any Drop of traffic and thus will be more difficult for the ones in the defense to identify any possible try of attack in the internal network. Moreover some Network Address Translation (NAT) [Not only the necessary ones] have been configured in order to expose more vulnerabilities.

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

Direct Attacking scenarios:

1. Windows 7 Workstation

- Remote Desktop Vulnerability
    - In the first level of configuration a NAT has been configured in order to expose the RDP protocol (port 3389) to the internet and thus to the attacker network. The specific version of Windows 7 and RDP have a couple of exploits that can be also found in Metasploit as "MS14-066" and "MS12-020". These Vulnerabilities in Remote Desktop could allow remote code execution.

2. Ubuntu 12.04 Workstation

- OpenSSH
    - As above, in the first level of configuration a NAT has been configured in order to expose the OpenSSH protocol (port 22) to the internet and thus to the attacker's network.
      More information about vulnerabilities of the protocol can be found at:
      http://www.openssh.com/security.html

3. 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.

4. 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)

Note: 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

Indirect Attacking scenarios:

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.

Note: 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