INSTITUTE FOR ADVANCED COMPUTING

AND

SOFTWARE DEVELOPMENT

AKURDI, PUNE

DOCUMENTATION ON

# "DEMILITARIZED ZONE ARMOUR – SECURING INTERNAL NETWORKS WITH HONEYPOT, PFSENSE AND IPTABLES "

SUBMITTED BY:

**GROUP NO. 14**

**NIKHIL B. KOTHARE (233424)**

**BHAVESH R. PATIL (233429)**

| | |
|---|---|
| **MR. KARTIK AWARI** | **MR. ROHIT PURANIK** |
| **PROJECT GUIDE** | **CENTRE CO-ORDINATOR** |

# ABSTRACT

The Demilitarized Zone Armor project focuses on strengthening network security through the implementation of advanced tools. The primary objective is to establish a secure Demilitarized Zone (DMZ) that hosts a web server, ensuring the protection of the internal LAN against external threats. By segregating and analyzing incoming external traffic, the project mitigates the potential for malicious infiltrations into the core network.

One crucial component of the project involves the strategic deployment of a honeypot. This deceptive system imitates vulnerable targets, enticing potential attackers. As these attackers interact with the honeypot, their methods and intentions are surreptitiously observed, generating insightful data to enhance defensive strategies.

To reinforce the security infrastructure, the project leverages the power of iptables—a packet-level tool. By crafting meticulous filtering rules, the system gains precise control over packet transmission, thwarting unauthorized access attempts and effectively countering diverse forms of attacks. The integration of these tools culminates in a multi-layered defense mechanism, significantly elevating the network's overall security posture.

# TABLE OF CONTENTS

**Topics** **Page No.**

**ABSTRACT** **ii**

**LIST OF ABBRIEVIATONS** **iv**

**LIST OF FIGURES** **iv**

**LIST OF TABLES** **iv**

1. **INTRODUCTION** **1**

1.1 Problem Statement 2

2. **LITERATURE SURVEY** **3**

3. **METHODOLOGY**

3.1 System Architecture 5

4. **REQUIREMENT SPECIFICATION**

4.1 Software Requirement 7

5. **WORKING** **8**

6. **IMPLEMENTATION** **11**

7. **HONEYPOT** **22**

8. **IPTABLES** **26**

9. **APPLICATIONS** **27**

10. **ADVANTAGES & DISADVANTAGES** **28**

11. **CONCLUSION** **30**

12. **REFERENCES** **31**

# LIST OF ABBREVIATIONS

| Sr. No. | Abbreviation | Full-Form |
|---|---|---|
| 1. | DMZ | Demilitarized zone |
| 2. | IDS | Intrusion Detection System |
| 3. | IPS | Intrusion Prevention System |
| 4. | WAF | Web Application Firewall |
| 5. | VPN | Virtual Private Network |
| 6. | ACL | Access Control List |
| 7. | NAT | Network Address Translation |
| 8. | LAN | Local Area Network |
| 9. | WAN | Wide Area Network |
| 10. | WLAN | Wireless Local Area Network |

# LIST OF FIGURES

| Figure No. | Figure Name | Page No. |
|---|---|---|
| Figure 1. | System architecture of DMZ network | 5 |
| Figure 2. | Working diagram of DMZ | 8 |

# 1. INTRODUCTION

In an increasingly interconnected digital landscape, the paramount concern for organizations is the security of their networks. The Demilitarized Zone Armor project emerges as a response to this pressing need, aiming to fortify network defenses using a strategic amalgamation of advanced tools. This project centers on the creation and fortification of a Demilitarized Zone (DMZ) that houses a web server. The ultimate goal is to establish a safeguarded environment for hosting online services while warding off potential threats to the internal Local Area Network (LAN).

The fundamental principle underlying this initiative is the concept of isolation and controlled access. By segregating the DMZ from the internal LAN, the project intends to erect an impregnable barrier against malicious actors seeking unauthorized access. The DMZ acts as an intermediary zone between the public internet and the private LAN, allowing external users to access designated services without infiltrating the internal network.

To complement this isolation strategy, the project employs a pivotal tool known as a honeypot. The honeypot serves as a virtual decoy, deliberately designed to appear vulnerable and enticing to potential attackers. Its purpose is to lure malicious entities away from critical network components and applications, effectively reducing the risk of successful intrusions. As attackers engage with the honeypot, the project gains invaluable insights into their tactics and methods, thereby enhancing its ability to counteract sophisticated threats.

A cornerstone of the Demilitarized Zone Armor project is the utilization of iptables, a versatile firewall management tool. Iptables operates at the granular level of individual data packets, allowing for precise control over their flow. Through the formulation of rule-based filters, iptables serves as a formidable gatekeeper, regulating traffic and thwarting unauthorized access attempts. This tool empowers the project to erect a multi-layered defense mechanism, bolstering the security posture of the network.

In summary, the Demilitarized Zone Armor project stands as a testament to the commitment to network security in the face of evolving cyber threats. By ingeniously combining the concepts of isolation, deception through honeypots, and the precision of iptables, the project endeavors to establish a robust shield that safeguards the integrity and confidentiality of the network. This report delves into the various tools employed within the project, their functionalities, and their collective role in achieving a fortified network infrastructure.

# 1.1 Problem Statement

`In the contemporary digital landscape, the proliferation of cyber threats poses a formidable challenge to the security of organizational networks. As businesses increasingly rely on online services and data exchange, the vulnerability of network infrastructures to external attacks becomes a pressing concern. The advent of sophisticated attack methodologies and the potential for data breaches underscore the need for proactive measures to safeguard critical resources.

Traditional security mechanisms, while effective to some extent, often fall short in addressing the evolving nature of cyber threats. The direct exposure of internal networks to external environments creates vulnerabilities that can be exploited by malicious actors. The absence of a robust demarcation between the public internet and private internal networks allows attackers to capitalize on security gaps, potentially leading to unauthorized access, data theft, and service disruption.

Furthermore, the lack of insightful data on contemporary attack techniques limits the ability to design effective defense strategies. Conventional security measures fail to capture the nuances of modern attack vectors, leaving organizations ill-equipped to anticipate and counteract evolving threats. In this context, the need arises for an innovative approach that not only segregates the internal network from external influences but also strategically gathers intelligence on emerging attack methodologies. This approach should provide a fortified platform for hosting web services while concurrently studying attacker behavior to enhance defense mechanisms. The project seeks to address these pressing issues by designing a Demilitarized Zone (DMZ) Armor that integrates a DMZ for secure web hosting and deploys advanced tools such as honeypots and iptables to bolster network security.

This project report delves into the multifaceted challenges posed by contemporary cyber threats, emphasizing the necessity for an integrated solution that fortifies network perimeters, gathers intelligence, and establishes a proactive defense against a dynamic landscape of cyber risks.

# 2. LITERATURE SURVEY

Network security in today's digital landscape is a critical concern due to the escalating sophistication of cyber threats. Researchers and practitioners have extensively explored various strategies to fortify network defenses, with a specific focus on establishing secure demarcations between internal networks and external environments. The following literature survey highlights key findings and insights from existing research in this field.

**1. Demilitarized Zones (DMZs) and Network Segmentation:**

The concept of creating DMZs, borrowed from military terminology, has been widely adopted in network security. Authors like Tanenbaum and Wetherall (2011) emphasize the importance of isolating publicly accessible servers, such as web servers, in a DMZ to prevent unauthorized access to sensitive internal resources. This segregation reduces the attack surface and limits lateral movement for potential attackers. Such network segmentation is a foundational practice in modern network security architectures.

**2. Honeypots for Deception:**

The utilization of honeypots as deception mechanisms to divert and study attackers has gained significant attention. Spitzner (2003) introduced the concept of honeypots as traps designed to detect, deflect, or study unauthorized use of information systems. Researchers like Provos and Holz (2009) have explored the efficacy of honeypots in capturing real-time data on attack techniques and patterns. Their findings highlight the value of honeypots in understanding attacker behavior and enhancing threat intelligence.

**3. Firewall Management with iptables:**

Iptables, a popular firewall management tool, has been extensively studied for its capabilities in packet filtering and access control. Authors like Turner (2005) have provided comprehensive insights into iptables' rule-based approach, detailing its effectiveness in filtering and shaping network traffic. Researchers also emphasize iptables' ability to create intricate rulesets that align with specific security policies, allowing for dynamic and precise control over data packets.

**4. Multi-Layered Defense Strategies:**

The amalgamation of various security tools to create multi-layered defense mechanisms has gained traction. Authors like Anderson (2008) advocate for the integration of intrusion detection systems, firewalls, and deception techniques to establish comprehensive security postures. Such strategies create redundancy in defense and enhance the ability to detect, prevent, and respond to a wide range of attacks.

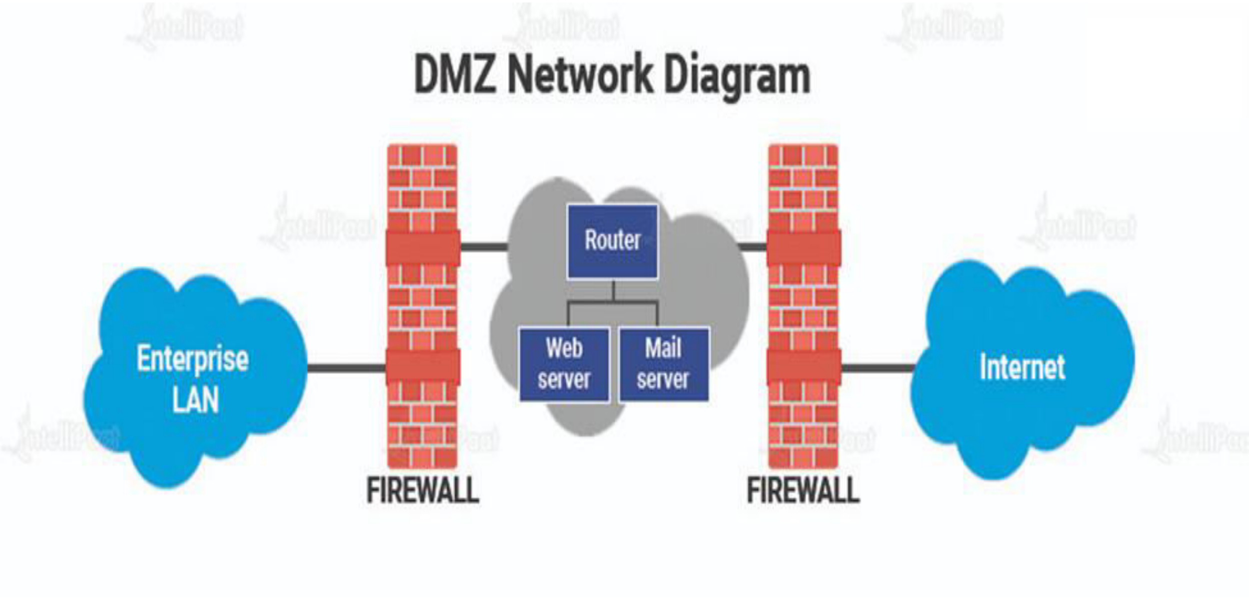
**5. Real-world Implementations:**

Real-world case studies demonstrate the practical application of the aforementioned concepts. Researchers like Natarajan and Ponnavaikko (2018) showcase the deployment of DMZs and honeypots in corporate networks. They highlight the reduction of attack surfaces and the collection of attack data for analysis and prevention. Additionally, studies by Porras et al. (2008) emphasize the role of network segmentation and honeypots in mitigating attacks against critical infrastructure.

In conclusion, the existing literature underscores the significance of robust network security strategies that incorporate DMZs, honeypots, and advanced firewall management tools. The surveyed literature forms a foundational basis for the Demilitarized Zone Armor project, emphasizing the need for an integrated approach that addresses the contemporary challenges of network security while gathering actionable intelligence to enhance defensive capabilities.

# 3. METHODOLOGY

## 3.1 SYSTEM ARCHITECTURE



**System Architecture: Demilitarized Zone (DMZ)**

In this system architecture, a Demilitarized Zone (DMZ) is implemented to safeguard the internal LAN network from external threats. The DMZ consists of two distinct firewalls, a web server, and a dedicated machine hosting a honeypot.

- **External Network:** The external network represents the untrusted public internet, where potential threats and attacks originate.

- **Firewall 1 (pfSense):**

The first line of defense is provided by a pfSense firewall facing the external network. This firewall is designed to filter incoming traffic and prevent unauthorized access to the internal LAN and the web server. It employs stateful packet inspection and access control policies to block malicious traffic and only allow essential services to pass through to the DMZ.

- **Demilitarized Zone (DMZ):**

  The DMZ is a segregated network segment that sits between the two firewalls. It is a semi-trusted environment where public-facing services like the web server are hosted. The DMZ acts as a buffer, reducing the exposure of the internal LAN to potential threats.

- **Web Server:**

  Within the DMZ, a web server hosts online services that need to be accessible to external users. The web server is configured to respond to external requests and provide the required services while maintaining isolation from the internal LAN. It is configured to have limited access to the internal network resources.

- **Firewall 2 (iptables):**

  The second firewall, implemented using iptables, stands between the DMZ and the internal LAN. This firewall enforces additional security measures, filtering and controlling traffic flowing between the DMZ and the internal LAN. It ensures that only legitimate and authorized communication is permitted to cross the boundary, preventing any potential threats from spreading to the internal network.

- **Internal LAN:**

  The internal LAN houses sensitive resources, databases, and confidential information. It is well-protected by the dual-layer firewall setup and remains isolated from external threats by the DMZ.

- **Honeypot**:

  A dedicated machine within the DMZ hosts a honeypot. The honeypot is designed to mimic vulnerable systems and attract potential attackers. It serves as a distraction, diverting attackers away from critical resources. As attackers engage with the honeypot, their tactics and methods are monitored and analyzed to enhance overall threat intelligence.

This system architecture combines the strength of dual firewalls, DMZ segmentation, a protected web server, and a honeypot to create a multi-layered security approach. It ensures that external threats are intercepted at different stages and prevents unauthorized access to the internal LAN while gathering valuable insights into evolving attack strategies.

# 4.REQUIREMENT SPECIFICATION

## 4.2.SOFTWARE REQUIREMENTS:

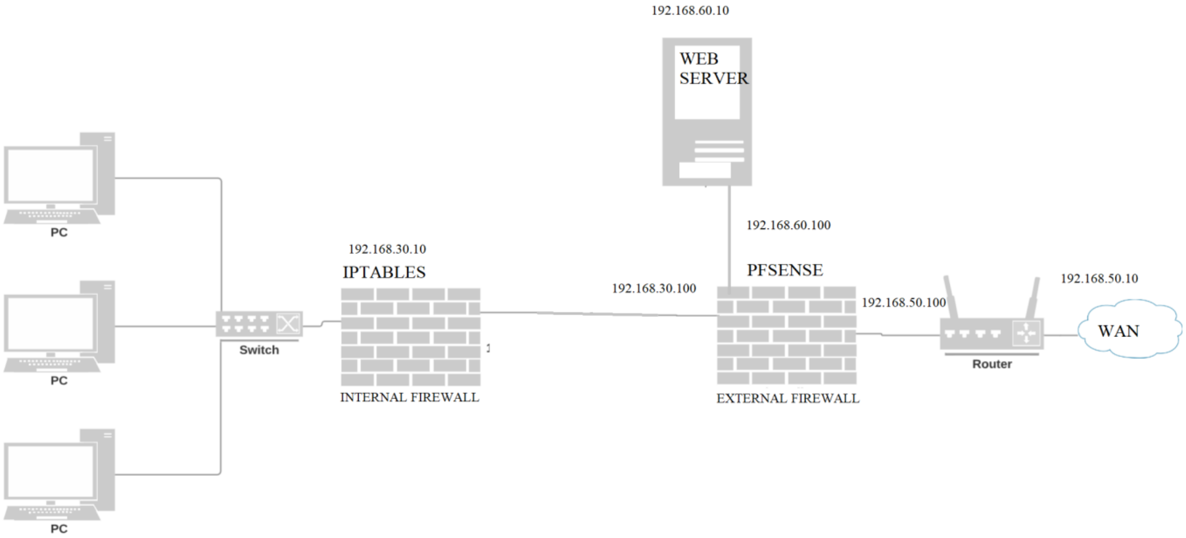
HARDWARE:

- RAM-8gb
- Stoarge-500gb
- CPU-2.5GHz
- Network-100mbps(host-only)

**SOFTWARE:**

- Operating system – LINUX-debian-10
- VMWARE® workstation 17 pro 17.0.0
- Firewall- pfsense ,iptables
- Web server- apache

# 5.WORKING



**Step 1: External Firewall (pfSense):**

1. Incoming Traffic: Incoming traffic from the external network (internet) first reaches the external firewall, pfSense.

2. Traffic Filtering: pfSense filters and inspects incoming traffic based on pre-configured firewall rules. Suspicious or unauthorized traffic is blocked.

3. Allowed Traffic: Legitimate traffic is allowed to pass through pfSense's external interface and enters the DMZ.

**Step 2: DMZ Segment:**

1. Traffic Entry: The allowed traffic from the external network enters the DMZ, the isolated segment between the external and internal firewalls.

**Step 3: Internal Firewall (iptables):**

1. Traffic to Internal Firewall: Traffic from the DMZ segment is directed towards the internal firewall, which is a separate Debian 10 machine running iptables.

2. Traffic Filtering: iptables enforces additional security measures by filtering and inspecting the traffic from the DMZ. Unauthorized or malicious traffic is blocked.

3. Allowed Traffic: Legitimate traffic, which is necessary for communication between the DMZ and the internal network, is allowed to proceed.

**Step 4: Internal LAN:**

1. Traffic to Internal LAN: Approved traffic from the DMZ that passes through the internal firewall reaches the internal LAN, which contains sensitive resources and data.

**Step 5: Honeypot (Cowrie):**

1. Honeypot Setup: A separate Debian 10 machine in the DMZ hosts the Cowrie honeypot software.

2. Attracting Attackers: Cowrie emulates vulnerable services, attracting potential attackers who attempt to exploit its simulated vulnerabilities.

3. Attacker Interaction: Attackers interact with the honeypot, unknowingly providing insights into their tactics and intentions.

4. Data Collection: Cowrie records attacker interactions, commands, and activities for analysis and threat intelligence.

**Step 6: Apache Web Server:**

1. Web Server Setup: Another virtual machine in the DMZ hosts an Apache web server.

2. Hosting Services: The Apache web server hosts public-facing services that need to be accessible from the internet.

3. Traffic Handling: The web server manages incoming requests, serving web pages or applications to users.

**Benefits of the Setup:**

- External Firewall (pfSense) provides the first line of defense by blocking malicious traffic from entering the network.

- DMZ offers a buffer zone that isolates external-facing services from the internal network, limiting the potential attack surface.

- Internal Firewall (iptables) ensures that only authorized communication between the DMZ and internal LAN is allowed.

- Honeypot (Cowrie) helps gather insights into attacker tactics, improving threat detection and defense strategies.

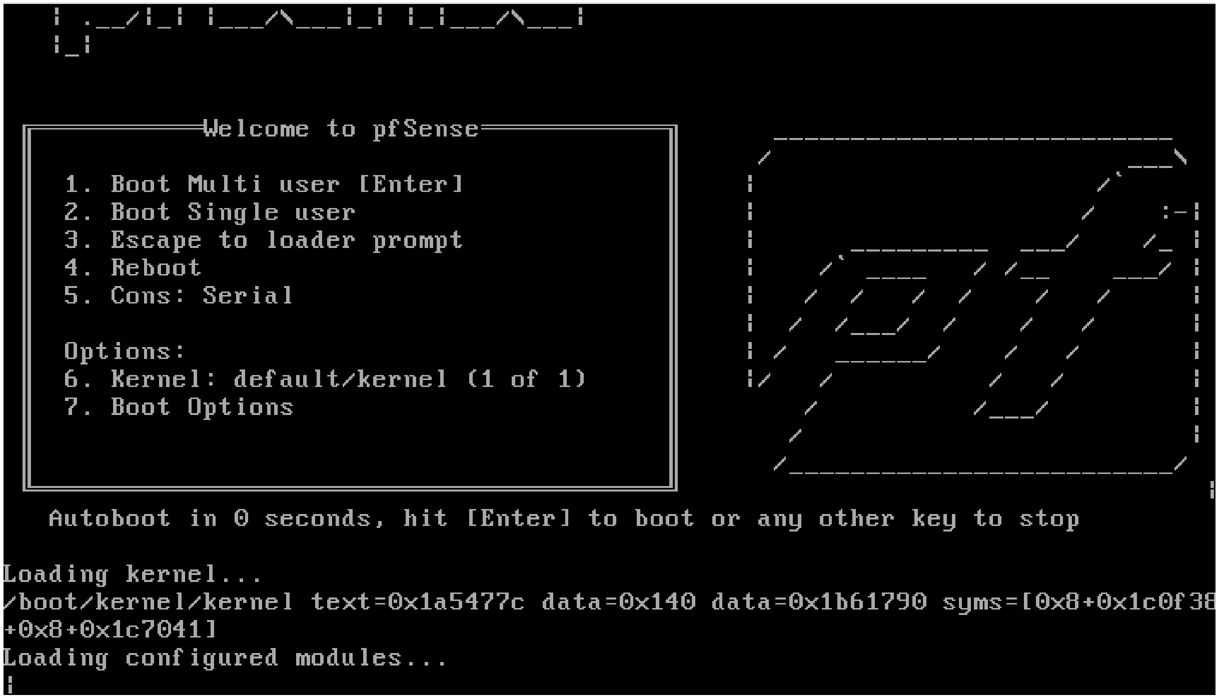
- Apache Web Server offers public services in a controlled environment, minimizing risks to the internal network.

This setup showcases a multi-layered security approach, allowing you to control and monitor traffic between different segments, detect potential threats, and safeguard critical resources. It provides a practical demonstration of DMZ architecture and security mechanisms using the specified components

# 6. IMPLEMENTATION

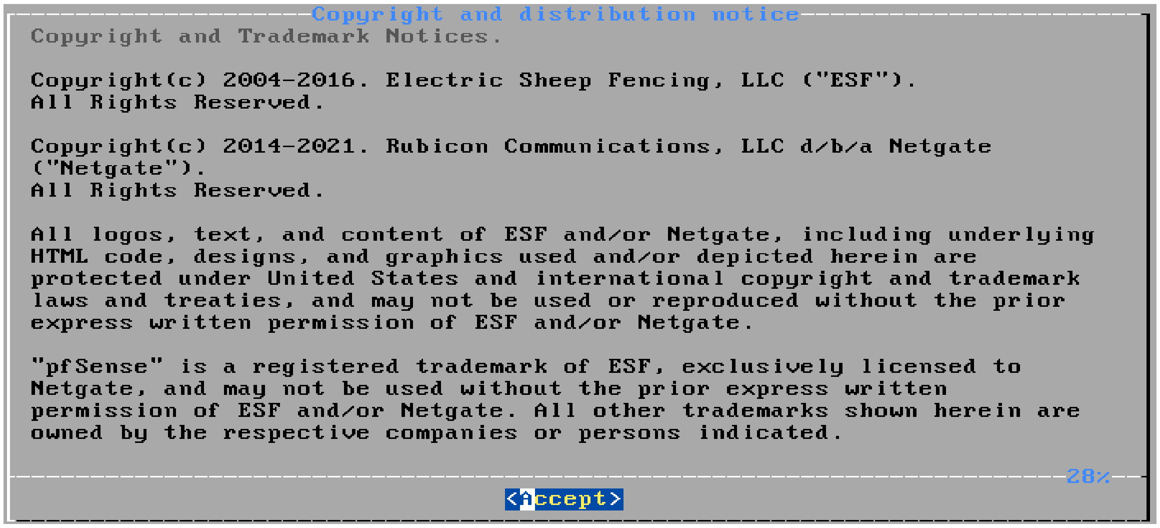
Installing Pfsense:

To begin with Pfsense installation, insert the disk or USB drive containing the bootable Pfsense ISO image to boot. In the screenshot below you can see the first installation screen you’ll see. You don’t need to select an option, the installation process will start automatically



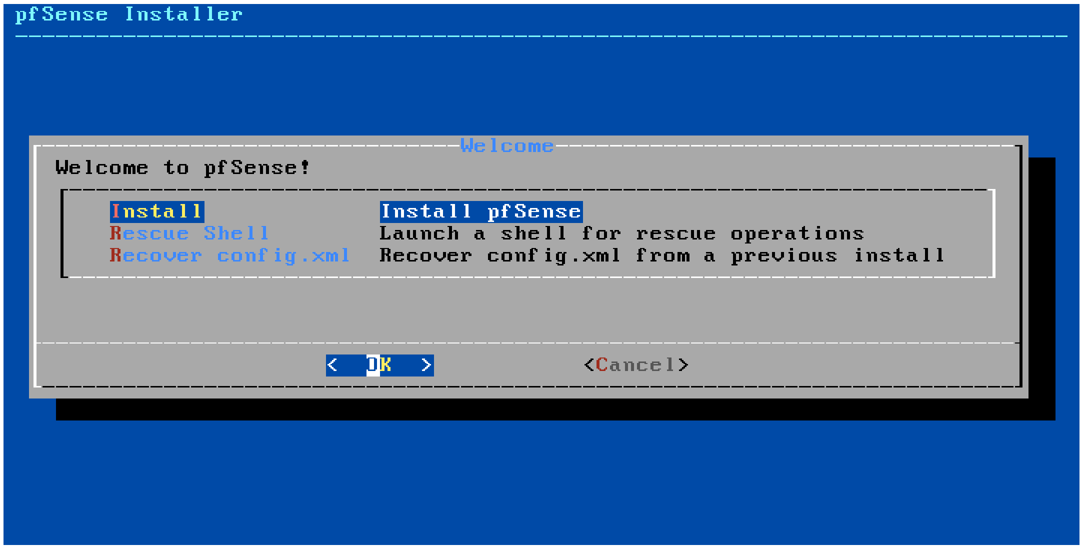
```
   Welcome to pfSense

   1. Boot Multi user [Enter]
   2. Boot Single user
   3. Escape to loader prompt
   4. Reboot
   5. Cons: Serial

   Options:
   6. Kernel: default/kernel (1 of 1)
   7. Boot Options

   Autoboot in 0 seconds, hit [Enter] to boot or any other key to stop

Loading kernel...
/boot/kernel/kernel text=0x1a5477c data=0x140 data=0x1b61790 syms=[0x8+0x1c0f38
+0x8+0x1c7041]
Loading configured modules...
```

The second screen contains a Copyright and distribution notice you need to accept by pressing the ACCEPT button, as shown below.



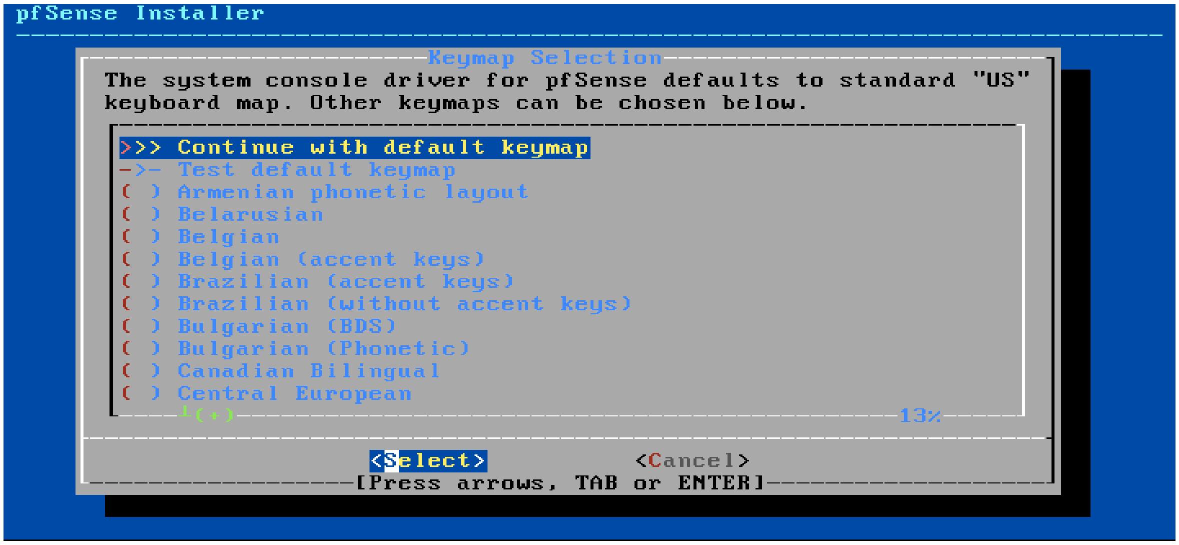
```
Copyright and distribution notice

Copyright and Trademark Notices.

Copyright(c) 2004-2016. Electric Sheep Fencing, LLC ("ESF").
All Rights Reserved.

Copyright(c) 2014-2021. Rubicon Communications, LLC d/b/a Netgate
("Netgate").
All Rights Reserved.

All logos, text, and content of ESF and/or Netgate, including underlying
HTML code, designs, and graphics used and/or depicted herein are
protected under United States and international copyright and trademark
laws and treaties, and may not be used or reproduced without the prior
express written permission of ESF and/or Netgate.

"pfSense" is a registered trademark of ESF, exclusively licensed to
Netgate, and may not be used without the prior express written
permission of ESF and/or Netgate. All other trademarks shown herein are
owned by the respective companies or persons indicated.
                                                                    28%
                              <Accept>
```

You can start a fresh Pfsense installation, launch a recovery console or restore a configuration file. To begin a new Pfsense installation, press the ENTER key on the Install option.



```
pfSense Installer

                              Welcome
Welcome to pfSense!

    Install             Install pfSense
    Rescue Shell        Launch a shell for rescue operations
    Recover config.xml  Recover config.xml from a previous install

                  <  OK  >          <Cancel>
```

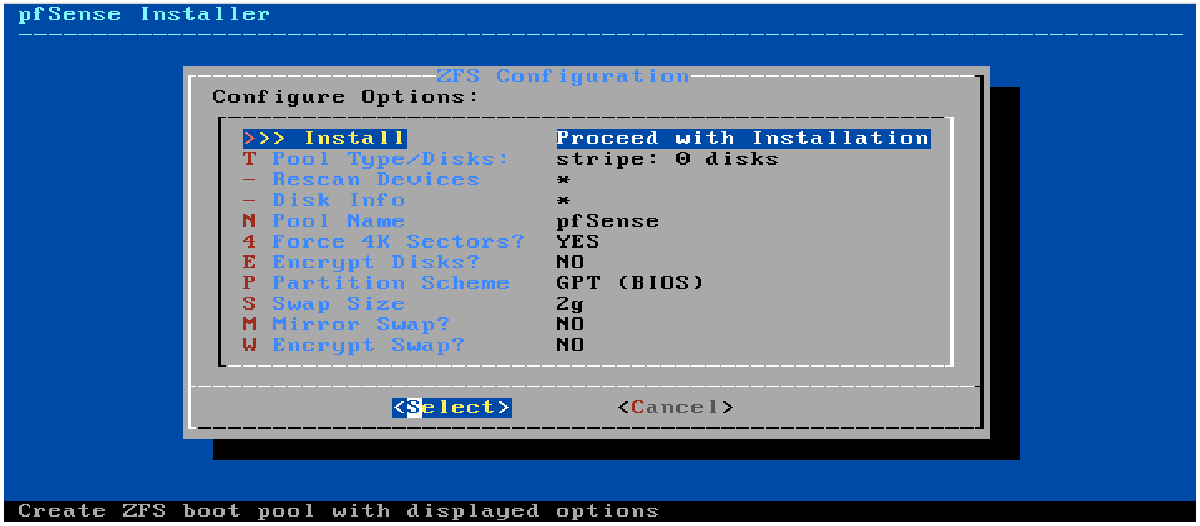
Now, you can select your keymap. For US English, press ENTER to choose the default option.



```
pfSense Installer

                      Keymap Selection
 The system console driver for pfSense defaults to standard "US"
 keyboard map. Other keymaps can be chosen below.

 >>> Continue with default keymap
 ->- Test default keymap
 ( ) Armenian phonetic layout
 ( ) Belarusian
 ( ) Belgian
 ( ) Belgian (accent keys)
 ( ) Brazilian (accent keys)
 ( ) Brazilian (without accent keys)
 ( ) Bulgarian (BDS)
 ( ) Bulgarian (Phonetic)
 ( ) Canadian Bilingual
 ( ) Central European
     (+)                                              13%

                 <Select>          <Cancel>
               [Press arrows, TAB or ENTER]
```

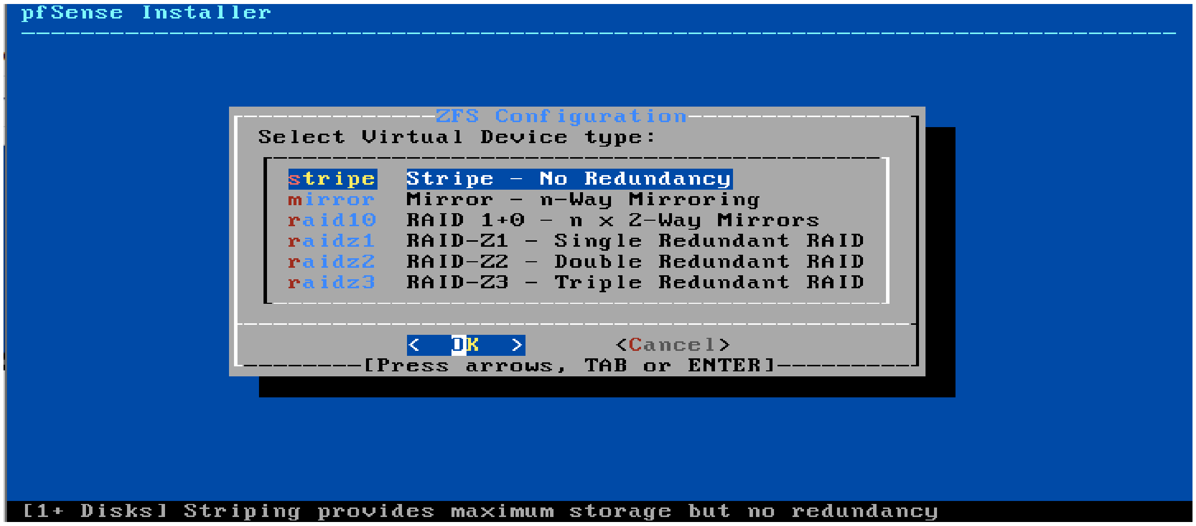
Recent Pfsense versions allow you to select ZFS as a filesystem. ZFS has many features including Inline Data Compression, Inline Data deduplication, ZFS Send/Receive, RAID Z and more. In this tutorial, we will select the ZFS option, but Auto UFS will work correctly if selected. Select the option you want and press ENTER to continue.

After selecting your filesystem, the installation process will allow you to edit some configuration and select additional options such as disk encryption, Swap size, etc. as shown in the image below. You can check the options, or you can proceed with the installation by pressing Install as shown in the following screenshot.



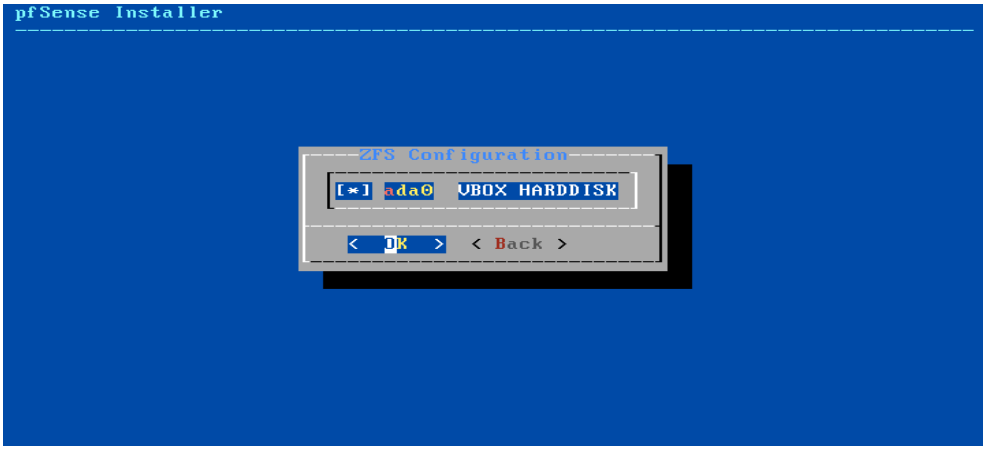
```
pfSense Installer

ZFS Configuration
Configure Options:

>>> Install               Proceed with Installation
T Pool Type/Disks:        stripe: 0 disks
- Rescan Devices          *
- Disk Info               *
N Pool Name               pfSense
4 Force 4K Sectors?       YES
E Encrypt Disks?          NO
P Partition Scheme        GPT (BIOS)
S Swap Size               2g
M Mirror Swap?            NO
W Encrypt Swap?           NO

        <Select>        <Cancel>

Create ZFS boot pool with displayed options
```

Now, you need to select the disk configuration.



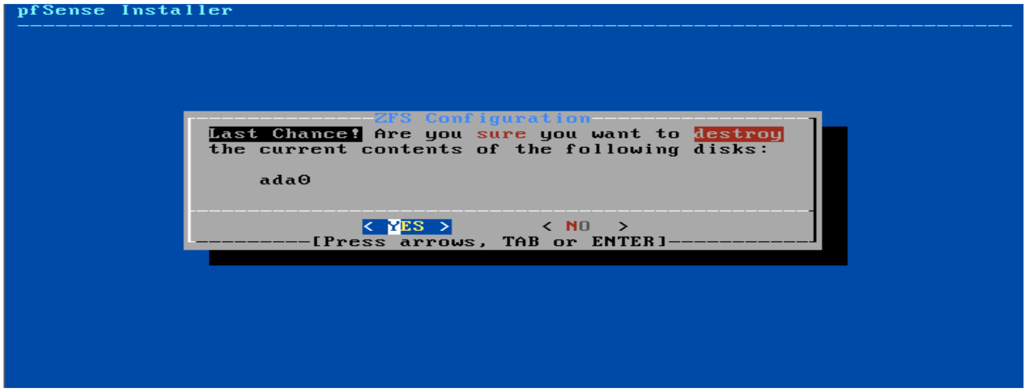
```
pfSense Installer

ZFS Configuration
Select Virtual Device type:

stripe   Stripe - No Redundancy
mirror   Mirror - n-Way Mirroring
raid10   RAID 1+0 - n x 2-Way Mirrors
raidz1   RAID-Z1 - Single Redundant RAID
raidz2   RAID-Z2 - Double Redundant RAID
raidz3   RAID-Z3 - Triple Redundant RAID

        <  OK  >        <Cancel>
--------[Press arrows, TAB or ENTER]--------

[1+ Disks] Striping provides maximum storage but no redundancy
```

You need to select the disk on which Pfsense will be installed. In my case, I'm using VMWARE for this tutorial. Select the disk unit you want and press ENTER.



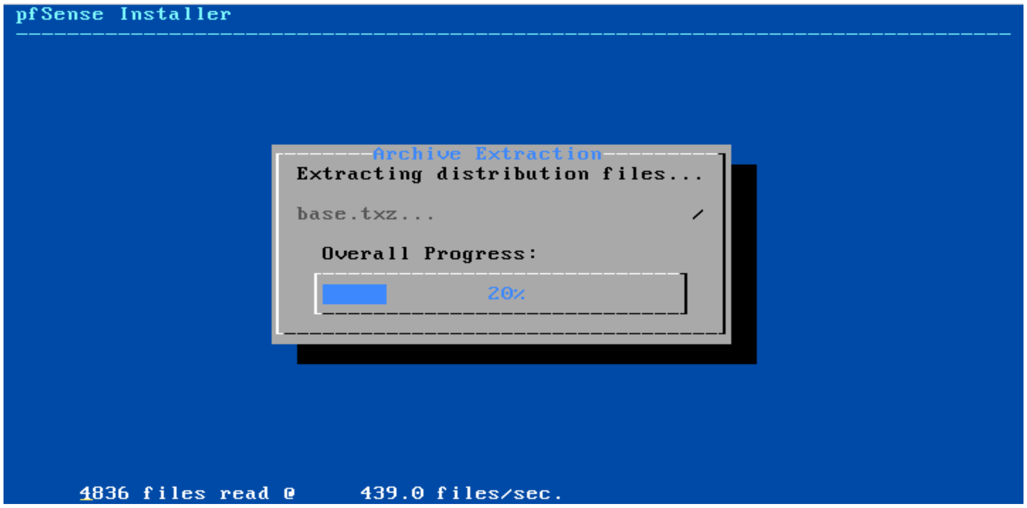
```
pfSense Installer

ZFS Configuration
[*] ada0   VBOX HARDDISK
<  OK  >   < Back >
```

Before starting the installation process, the installer will give you a last chance to stop or edit the installation. If you have nothing to change, press ENTER to start Pfsense installation.



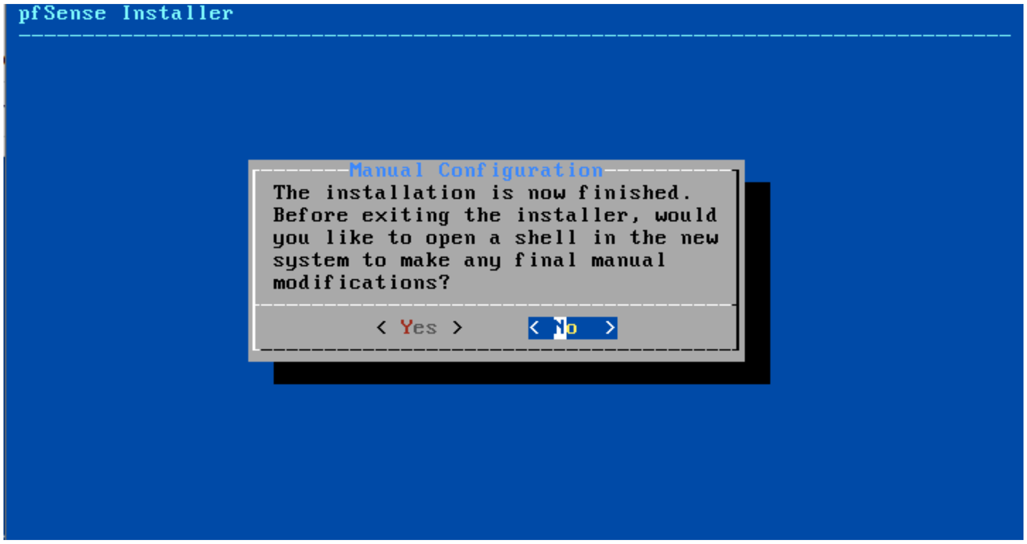
```
pfSense Installer

ZFS Configuration
Last Chance! Are you sure you want to destroy
the current contents of the following disks:

    ada0

< YES >        < NO  >
[Press arrows, TAB or ENTER]
```

As you can see in the screenshot below, the installation process will start. This may take a few minutes to finish.



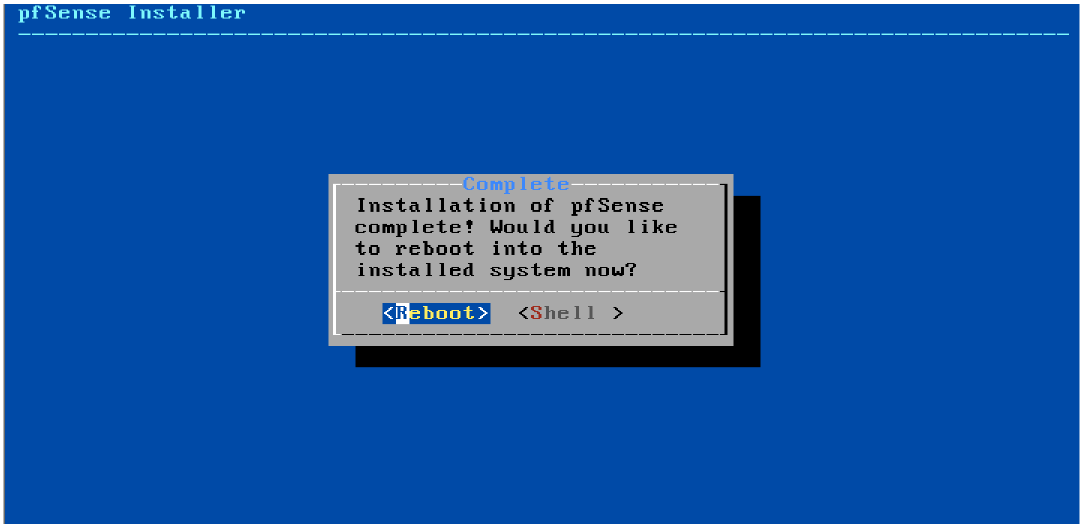
```
pfSense Installer

Archive Extraction
Extracting distribution files...

base.txz...                              /

  Overall Progress:
  [          20%          ]

4836 files read @      439.0 files/sec.
```

Once the installation process ends, you will be offered to make changes. If you have no changes to do, press No to continue.



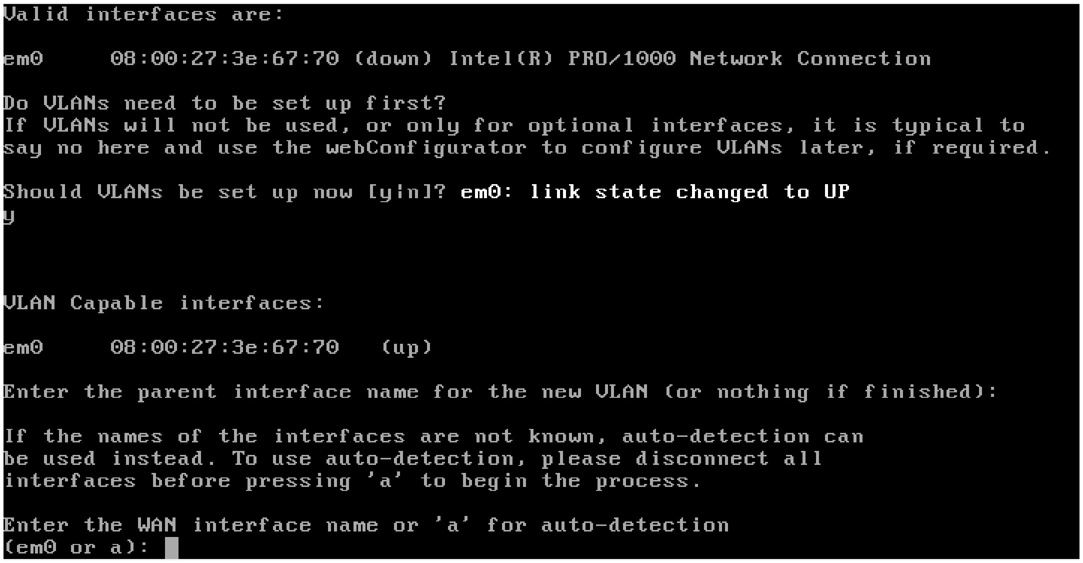
```
pfSense Installer

Manual Configuration
The installation is now finished.
Before exiting the installer, would
you like to open a shell in the new
system to make any final manual
modifications?

        < Yes >      < No  >
```

Finally, you will be asked to reboot into Pfsense. Select and press Reboot to continue to start Pfsense.



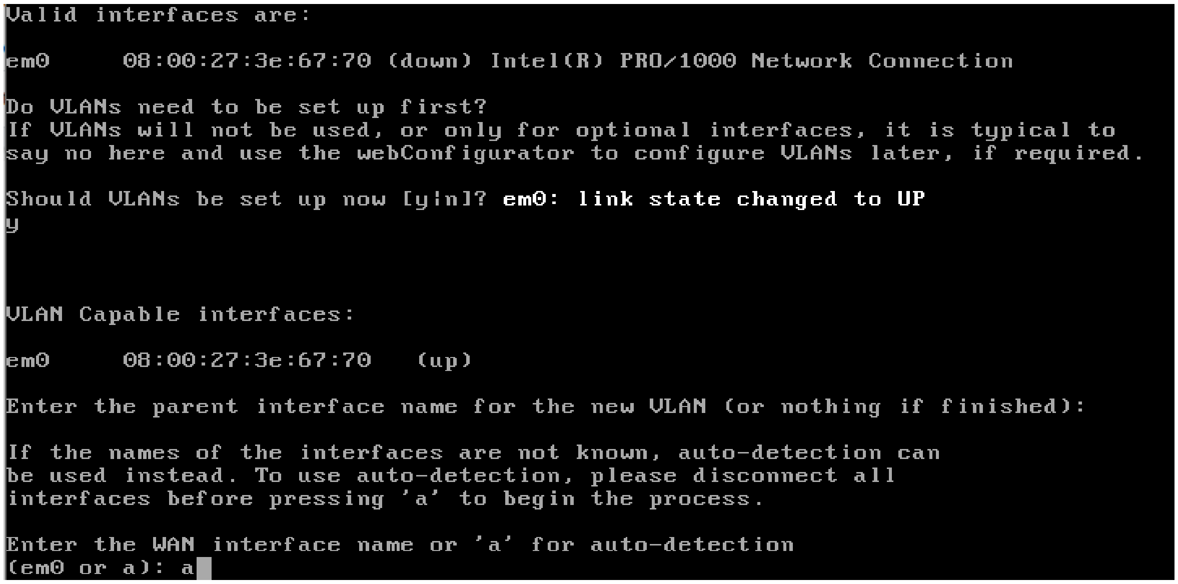
```
pfSense Installer

Complete
Installation of pfSense
complete! Would you like
to reboot into the
installed system now?

<Reboot>  <Shell >
```

On the first reboot, Pfsense will offer you to set up the network/s interface/s. The first interface is virtual. The virtual network interface is **em0**. To configure the network interface up, press Y. You can select N and configure it later through the Web configurator as shown in the screenshot below.



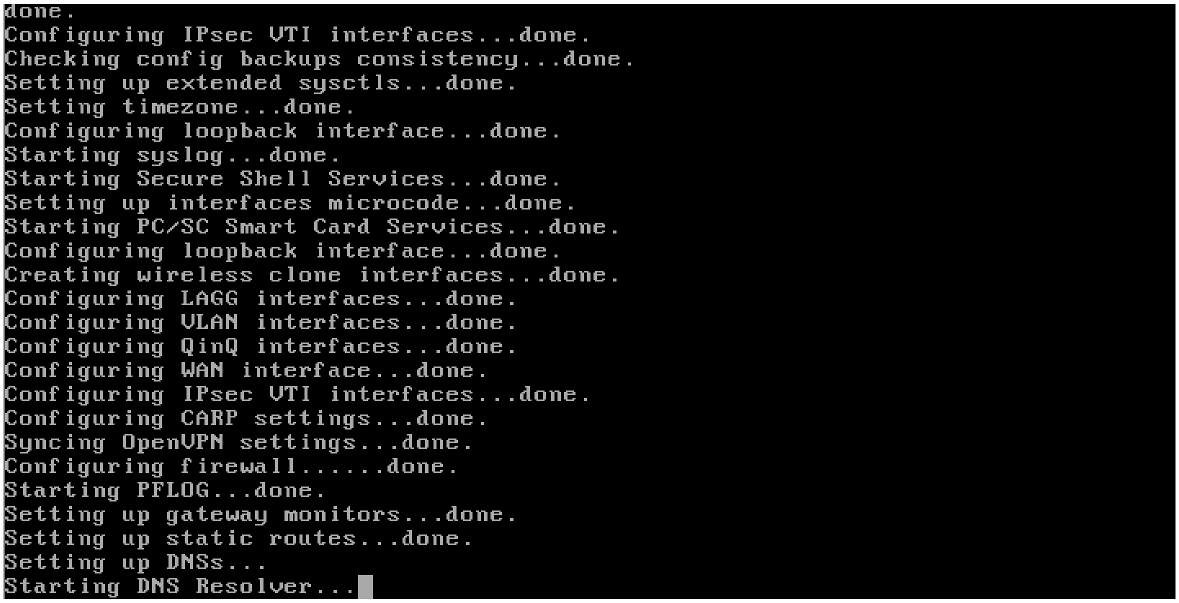
```
Valid interfaces are:

em0     08:00:27:3e:67:70 (down) Intel(R) PRO/1000 Network Connection

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y|n]? em0: link state changed to UP
y


VLAN Capable interfaces:

em0     08:00:27:3e:67:70   (up)

Enter the parent interface name for the new VLAN (or nothing if finished):

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

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

Now, you need to select the WAN interface, you can type it or select *'a'* for autodetection.



```
Valid interfaces are:

em0     08:00:27:3e:67:70 (down) Intel(R) PRO/1000 Network Connection

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y|n]? em0: link state changed to UP
y


VLAN Capable interfaces:

em0     08:00:27:3e:67:70   (up)

Enter the parent interface name for the new VLAN (or nothing if finished):

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

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

If the autodetection was correct, press '*y*' to set up your LAN interface. Then Pfsense will boot as shown in the following screenshot.



```
done.
Configuring IPsec VTI interfaces...done.
Checking config backups consistency...done.
Setting up extended sysctls...done.
Setting timezone...done.
Configuring loopback interface...done.
Starting syslog...done.
Starting Secure Shell Services...done.
Setting up interfaces microcode...done.
Starting PC/SC Smart Card Services...done.
Configuring loopback interface...done.
Creating wireless clone interfaces...done.
Configuring LAGG interfaces...done.
Configuring VLAN interfaces...done.
Configuring QinQ interfaces...done.
Configuring WAN interface...done.
Configuring IPsec VTI interfaces...done.
Configuring CARP settings...done.
Syncing OpenVPN settings...done.
Configuring firewall......done.
Starting PFLOG...done.
Setting up gateway monitors...done.
Setting up static routes...done.
Setting up DNSs...
Starting DNS Resolver...
```

# 7.HONEYTPOT

- **Threat intelligence:**
- **Types of Honeypots-**
- Low Interaction:

  Simulates only some parts (services)

  For ex:- the network stack
- High Interaction

- Provides a real system the attacker can interact with Can compromised completely
- To change the ssh port number we can edit it form this file
- → /etc/ssh/sshd_config
    - Install python 3.8-
- - https://linuxize.com/post/how-to-install-python-3-8-on-debian-10/

**1. Create non-root user-**

- sudo adduser --disabled-password cowrie

**2. Configure authbind-**

- sudo touch /etc/authbind/byport/22

- sudo chown cowrie:cowrie /etc/authbind/byport/22

- sudo chmod 777 /etc/authbind/byport/22

- sudo su - **cowrie** # switch to cowrie user

- pwd

- /home/cowrie**- git clone** **http://github.com/cowrie/cowrie**- cd **cowrie**

**3.Create virtual environment of python-**

# verify that current directory is /home/cowrie/cowrie

**-** virtualenv --python=python3.8 cowrie-env (we can give any name)

**4.We have to activate the env.**

- source cowrie-env/bin/activate
    - #due to these prompt will jumps to virtual env
- deactivate
    - get back from virtual env

# we goes into virtualenv mode

#know we have to install the dependencies for python

#in linux we use apt but in **python we use pip**

- pip install **--**upgrade pip (upgrade pip to latest version)

#All the dependencies are stored in the file requirements.txt so no need to install 1 by 1 packages

- pip install --upgrade -r requirements.txt

Will prompt error cause cowrie needs python version > 3.8

**5.Copy the configuration file-**

# In virtual env

- cp etc/cowrie.cfg.dist etc/cowrie.cfg

**6.Edit the config file to change the port no from 2222 > 22**

- etc/cowrie.cfg
- listen_endpoints = **tcp:22:**interface=0.0.0.0 (line no.585)

**7.Configure the user name for allow and not allowed user**

- **Create a new file**
  - nano etc/userdb.txt
    - root:x:!123456 (root user name allowed but passwd

      123456 (not allowed)
    - root:x:* (except above one everything is allowed)

- **Start cowrie**
  - bin/cowrie start
  - bin/cowrie status
- **Check ports**
  - ss -ant
    - both the ports are open
    - 2222 is for ssh of base machine

- 22 is for honeypot ssh

- **New machine (attacker)**

  - ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no

  root@ip_cowrie

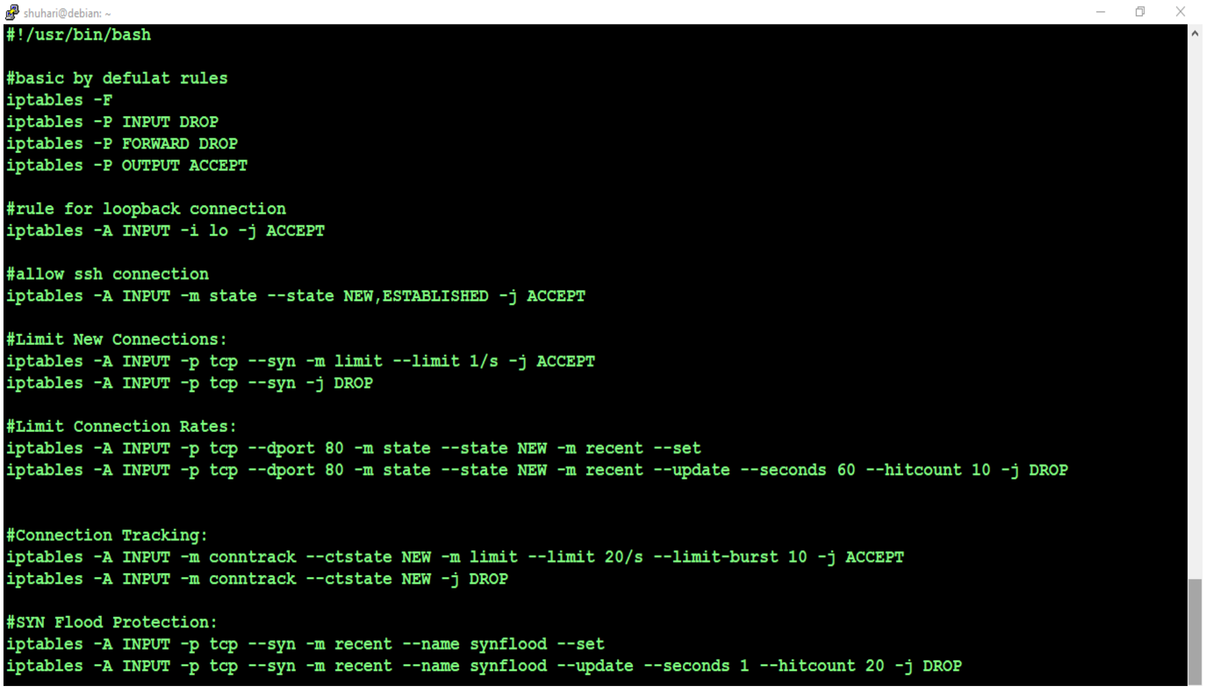
- **Duplicate session of cowrie**

  - tail -f /home/cowrie/cowrie/var/log/cowrie/cowrie.log

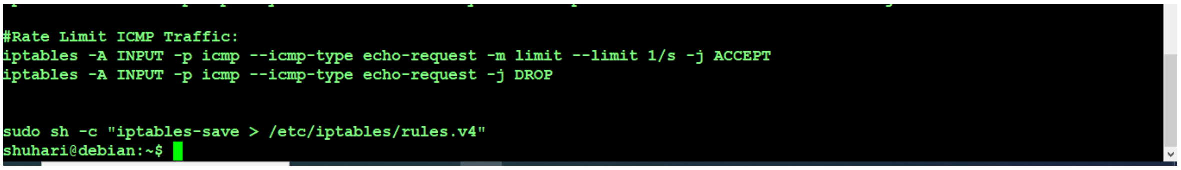
sudo apt install -y git python-virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv

# 8. IPTABLES

- **Iptables Rules:**



```
#!/usr/bin/bash

#basic by defulat rules
iptables -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

#rule for loopback connection
iptables -A INPUT -i lo -j ACCEPT

#allow ssh connection
iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT

#Limit New Connections:
iptables -A INPUT -p tcp --syn -m limit --limit 1/s -j ACCEPT
iptables -A INPUT -p tcp --syn -j DROP

#Limit Connection Rates:
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m recent --set
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 60 --hitcount 10 -j DROP


#Connection Tracking:
iptables -A INPUT -m conntrack --ctstate NEW -m limit --limit 20/s --limit-burst 10 -j ACCEPT
iptables -A INPUT -m conntrack --ctstate NEW -j DROP

#SYN Flood Protection:
iptables -A INPUT -p tcp --syn -m recent --name synflood --set
iptables -A INPUT -p tcp --syn -m recent --name synflood --update --seconds 1 --hitcount 20 -j DROP
```



```
#Rate Limit ICMP Traffic:
iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP


sudo sh -c "iptables-save > /etc/iptables/rules.v4"
shuhari@debian:~$ 
```

# 9.APPLICATIONS

**Web Hosting and Public Services:** DMZs are commonly used to host public-facing websites, blogs, e-commerce platforms, and online applications that need to be accessible from the internet while isolating them from internal systems.

**Email Servers:** Placing email servers in a DMZ allows external users to send and receive emails while protecting the internal network from potential email-based threats like malware and phishing.

**Remote Access:** DMZs enable secure remote access for employees, partners, and vendors. VPN gateways or remote access servers can be placed in the DMZ, allowing external connections while maintaining internal network integrity.

**Application Hosting:** Hosting applications like online databases, collaboration tools, and customer portals in a DMZ allows controlled interaction from external users while preserving internal security.

**Secure File Sharing:** DMZs can host secure file transfer services that allow authorized external users to share files without exposing internal resources.

**Intrusion Detection and Prevention:** DMZs house intrusion detection and prevention systems (IDPS) that monitor network traffic for suspicious activities and help protect both internal and external assets.

**Honeypots and Threat Intelligence:** DMZs are ideal environments for deploying honeypots, which mimic vulnerable systems to attract and gather insights about attackers' tactics and intentions.

**Load Balancing and Redundancy:** DMZs can host load balancers that distribute traffic across multiple backend servers, enhancing service availability and performance.

**Testing and Development:** DMZs provide a controlled environment for testing and developing applications without affecting the core internal network.

**DDoS Mitigation:** DMZs can absorb and filter distributed denial of service (DDoSattacks, preventing them from directly affecting internal resources.

# 10.ADVANTAGES & DISADVANTAGES

- **Enhanced Security:** The primary advantage of a DMZ is the improved security it offers by isolating public-facing services from sensitive internal resources.

- **Isolation:** DMZs provide a buffer zone that separates external networks (e.g., internet) from internal networks, minimizing the potential for direct attacks.

- **Reduced Attack Surface:** By limiting exposure to external threats, DMZs reduce the points of entry for attackers and potential breaches.

- **Controlled Access:** DMZs enable controlled and regulated communication between external and internal networks, reducing the risk of unauthorized access.

- **Threat Mitigation:** If a service within the DMZ is compromised, attackers are contained within the DMZ, preventing easy access to internal resources

- **Compliance:** DMZs assist in meeting security compliance requirements by demonstrating separation between internal and external networks.

- **Public Service Hosting:** DMZs allow secure hosting of public-facing services like websites, email servers, and applications while safeguarding the core network.

- **Secure Collaboration:** DMZs facilitate secure interaction with external partners, vendors, and customers without compromising internal security.

- **Monitoring and Analysis:** DMZs enable focused monitoring and analysis of external traffic, aiding in the early detection of potential threats.

- **Testing Environment:** DMZs provide a controlled space for testing new services or applications without affecting the internal network..

- **Complex Configuration:** Setting up and maintaining DMZs can be complex, requiring careful design and configuration of firewalls, security policies, and routing.

- **Resource Allocation:** Hosting services within a DMZ may require additional hardware and resources, increasing operational costs.

- **Network Complexity:** The presence of a DMZ adds to the network's overall complexity, requiring expertise to manage effectively.

- **Maintenance Overhead:** Regular maintenance and updates are necessary to ensure the security and functionality of DMZ components.

- **Misconfiguration Risks:** Improperly configured DMZ components can inadvertently expose internal resources or disrupt services.

- **False Sense of Security:** Relying solely on a DMZ may create a false sense of security; other security measures are still needed within the internal network.

- **Resource Duplication:** Hosting duplicate services within the DMZ and internal network may require additional effort for synchronization and updates.

- **Intrusion Spread:** While containing attacks, a DMZ breach can still lead to unauthorized access within the DMZ itself.

- **Scalability Challenges:** As the network grows, scaling DMZ infrastructure to accommodate increased traffic and services can be challenging.

- **Single Point of Failure:** If not designed properly, a DMZ can become a single point of failure if compromised, impacting services.

# 11.CONCLUSION

In conclusion, the project "Demilitarized Zone Armor" underscores the paramount importance of robust network security through the strategic implementation of a Demilitarized Zone (DMZ) architecture. By crafting a secure intermediary between the external and internal networks, we have created a resilient defense mechanism that facilitates seamless communication while safeguarding sensitive resources from external threats.

The deployment of advanced tools within the DMZ, including the external firewall (pfSense) and internal firewall (iptables), establishes a comprehensive security framework that filters, controls, and monitors traffic. This multi-layered approach minimizes the risk of unauthorized access and malicious intrusions, exemplifying the essence of fortified network protection.

The integration of a honeypot (Cowrie) within the DMZ further enriches our defensive strategy by luring potential attackers and providing invaluable insights into their tactics. This knowledge empowers us to continuously refine our defenses, enhancing our ability to anticipate, prevent, and mitigate security risks.

As we host a public-facing Apache web server within the DMZ, we have showcased how external users can access services without compromising internal assets. This project illuminates the symbiotic relationship between accessibility and security, demonstrating that a well-architected DMZ is pivotal in striking this balance.

Our journey through this project reaffirms the significance of innovation and adaptability in the realm of cybersecurity. By implementing the DMZ architecture and its accompanying components, we have fortified our network against the dynamic landscape of cyber threats. This accomplishment underscores our commitment to safeguarding data, nurturing secure communication, and championing the evolution of network security in an increasingly interconnected world.

# 12.REFERENCES

*[1] "DMZ (computing)," Wikipedia. 12-Feb-2019*

*[2] "Configuring IP Access Lists," Cisco. [Online]. Available:*

*https://www.cisco.com/c/en/us/support/docs/security/ios-firewall/23602-confaccesslists.html.*

*[Accessed: 01-Jan-2020]*

*[3] "What Is a Distributed Denial-of-Service (DDoS) Attack?," Cloudflare. [Online]. Available:*

*https://www.cloudflare.com/en-ca/learning/ddos/what-is-a-ddos-attack/. [Accessed: 01-Jan-2020].*

*[4] "GNS3 | The software that empowers network professionals." [Online]. Available:*

*https://gns3.com/. [Accessed: 05-Oct-2019].*

*[5] "Download VMware Workstation Pro | CA," VMware. [Online]. Available:*

*https://www.vmware.com/ca/products/workstation-pro/workstation-pro-evaluation.html.*

*[Accessed: 05-Oct-2019]*

*[22] "Wireshark · Go Deep." [Online]. Available: https://www.wireshark.org/. [Accessed: 05-Oct-*

*2019].*

.