



CYSA PROJECT

Prof Drew Gainsmiller

ABSTRACT

Configure pfsense firewall rules, snort and other IDS/IPS software according to the given instructions. Respond to the questions asked, providing sufficient answers which can be implemented to resolve the issues. Also submit a video presentation of project.

Elisha Ngwana

CySA Final Project



TAMUK QuickStart CySA+ Final Project

Scenario

Based upon your previous work with your client, they have asked you to improve the security of their network. They specifically wish to implement a way to identify and log attacks against their web and production servers. After some research, you decide that installing an IDS/IPS would meet their needs.

Project Tasks

By successfully completing this project, you will have demonstrated your understanding of firewalls and your ability to install and configure an IDS/IPS.

Project tasks are as follows:

Download and install Snort on the pfSense firewall.

Configure Snort to log suspicious activity on the Untrusted and DMZ interfaces.

Configure Snort to alert to the types of exploits you performed in the previous ethical hacking project.

Create firewall rules to block FTP traffic from the Untrusted network to the DMZ

Deliverables

To successfully complete this project, you must perform the following:

Complete all project tasks.

Answer the 10 exam questions in written form and submit to your coach.

Provide an executive presentation to your coach via Zoom. The presentation will be done using PowerPoint (or similar product), followed by a live demonstration of your solutions. The PowerPoint presentation must have a design of your choice and have at least 7 slides to include:

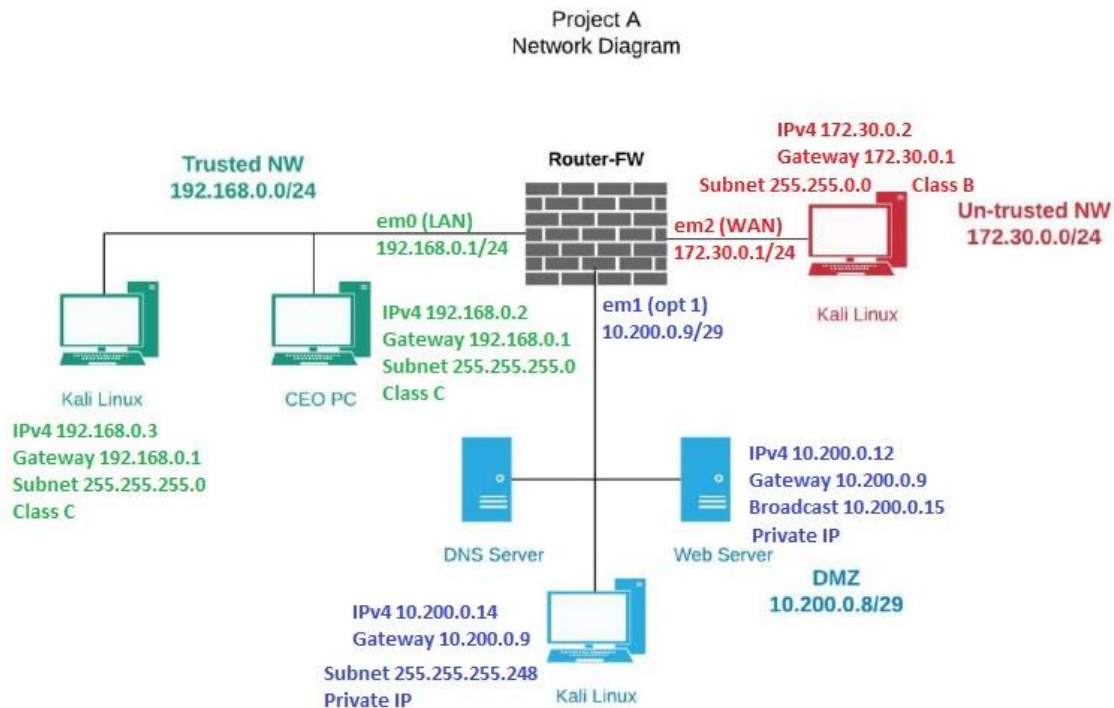
- a. Executive Summary at beginning.
- b. Description of vulnerabilities found.
- c. Description of how you used Snort.
- d. Recommendations and closing remarks at the end.

Exam Questions

1. Write an example of a firewall rule that will allow only HTTPS traffic to enter the DMZ.
2. Write an example of a firewall rule that will block FTP traffic originating from the Untrusted network.
3. Does the order of rule placement in a firewall matter and if so, why?
4. Which Linux distribution is pfSense based on?
5. Which logs would be useful in monitoring traffic on the firewall and why?
6. How could you block users from accessing inappropriate websites?
7. How would you implement secure file transfer between the Trusted network and the web server in the DMZ?
8. Given that you have an IT administrator workstation with an address of 192.168.0.30/24, write an example of a firewall rule to allow only that workstation to access the webserver using HTTPS and FTPS.
9. What would you recommend to improve the security of the DMZ and the Trusted network, and why?
10. List and briefly describe 3 commercial IDS/IPS products you would recommend to a client

Project Accounts

System	Username	Password
Production Server	msfadmin	msfadmin
Router-FW	admin	pfsense
DNS Server	root	password
Web Server	admin	\$eclab!2
	root	\$eclab!3
CEO PC	user	\$eclab!2
	root	\$eclab!3



ANSWERS

1. Write an example of a firewall rule that will allow only HTTPS traffic to enter the DMZ.

Policy: Permit SSH/HTTPS from Kali 10.200.0.14 and Web Server 10.200.0.12 to LAN

This feature is similar to object groups on the Cisco IOS, where we group similar objects together to make configuration simpler. With aliases, instead of specifying the individual objects, you just specify the alias name.

Therefore, let's configure two aliases: one for SSH and HTTPS and the second one for the hosts 10.200.0.12 and 10.200.0.14. To do this, we will navigate to **Firewall > Aliases**:

Firewall: Aliases



IP

Ports

URLs

All

Name	Values	Description
<p>Note:</p> <p>Aliases act as placeholders for real hosts, networks or ports. They can be used to minimize the number of changes that have to be made if a host, network or port changes. You can enter the name of an alias instead of the host, network or port in all fields that have a red background. The alias will be resolved according to the list above. If an alias cannot be resolved (e.g. because you deleted it), the corresponding element (e.g. filter/NAT/shaper rule) will be considered invalid and skipped.</p>		

As you can see, we can create aliases for IP, Ports, and URLs. We will start with the one for IP and then move to the one for ports.

Firewall: Aliases: Edit



Alias Edit

Name
The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".

Description
You may enter a description here for your reference (not parsed).

Type

Host(s)

Enter as many hosts as you would like. Hosts must be specified by their IP address or fully qualified domain name (FQDN). FQDN hostnames are periodically re-resolved and updated. If multiple IPs are returned by a DNS query, all are used. You may also enter an IP range such as 192.168.1.1-192.168.1.10 or a small subnet such as 192.168.1.16/28 and a list of individual IP addresses will be generated.

Description

IP	Ports	URLs	All
Name		Values	
DMZ_SERVERS		172.16.100.200, 172.16.100.201	

IP	Ports	URLs	All
Name		Values	
SSH_HTTPS		22, 443	

When you are done with your configuration, apply your changes and we can move on to creating the firewall rule itself. We will navigate to **Firewall > Rules** and then select the **DMZ** tab. The settings for my own rule are shown below:

Firewall: Rules: Edit



Edit Firewall rule	
Action	Pass ▼ Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
Disabled	<input type="checkbox"/> Disable this rule Set this option to disable this rule without removing it from the list.
Interface	DMZ ▼ Choose which interface packets must be sourced on to match this rule.
TCP/IP Version	IPv4 ▼ Select the Internet Protocol version this rule applies to
Protocol	<div style="border: 1px solid red; padding: 2px;">TCP ▼</div> Choose which IP protocol this rule should match. Hint: in most cases, you should specify TCP here.
Source	<input type="checkbox"/> not Use this option to invert the sense of the match. <div style="border: 1px solid red; padding: 2px;"> Type: Single host or alias ▼ Address: DMZ_SERVERS / ▼ </div> <input type="button" value="Advanced"/> - Show source port range
Destination	<input type="checkbox"/> not Use this option to invert the sense of the match. <div style="border: 1px solid red; padding: 2px;"> Type: LAN net ▼ Address: / ▼ </div>
Destination port range	<div style="border: 1px solid red; padding: 2px;"> from: (other) ▼ SSH_HTT to: (other) ▼ SSH_HTT </div> Specify the port or port range for the destination of the packet for this rule. Hint: you can leave the 'to' field empty if you only want to filter a single port.
Log	<input type="checkbox"/> Log packets that are handled by this rule Hint: the firewall has limited local log space. Don't turn on logging for everything. If you want to do a lot of logging, consider using a remote syslog server (see the Diagnostics: System logs: Settings page).
Description	<div style="border: 1px solid gray; padding: 2px;"> Allow SSH/HTTPS from DMZ hosts to LAN </div> You may enter a description here for your reference.

As you may have noticed when creating the port aliases, you don't specify the protocol. It is when we are creating the firewall rule that we specify the protocol, as shown above. Also notice how we specified the source as the alias we created—once you start typing the name, aliases that match that name show up. We also used the alias we created for the ports under the *Destination port range* field. Finally, there are some default names such as *LAN address* (i.e., LAN interface IP address of pfSense) and *LAN net* (i.e., LAN network and other static routes configured on that interface) that we can use when configuring rules. These make your life easier because, if an address/network changes, you won't have to alter the rule as the rule will be automatically updated to match the new address(es).

Policy #4: Allow DNS, HTTP, and HTTPS from DMZ to Internet

There are several ways you can configure this rule, depending on how restrictive you want your rule to be. DNS (not zone transfers) uses UDP port 53 by default, while HTTP and HTTPS use TCP port 80 and 443, respectively. If you create a port alias matching the

three protocols, you will have to use “TCP/UDP” in the Protocol field of the firewall rule. This means that TCP/UDP ports 53, 80 and 443 will be allowed which is more than you want.

Let’s practice the principle of least privilege and be as restrictive as possible. We will create a port alias for HTTP and HTTPS and then create a standalone rule for DNS.

Floating WAN LAN DMZ										
	ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
<input type="checkbox"/>	▶	IPv4 ICMP	*	*	*	*	*	none		Allow ICMP any any
<input type="checkbox"/>	▶	IPv4 TCP	DMZ_SERVERS	*	LAN net	SSH_HTTPS	*	none		Allow SSH/HTTPS from DMZ hosts to LAN
<input type="checkbox"/>	▶	IPv4 TCP	DMZ net	*	*	HTTP_S	*	none		Allow HTTP/S from DMZ to Internet
<input type="checkbox"/>	▶	IPv4 UDP	DMZ net	*	*	53 (DNS)	*	none		Allow DNS from DMZ to Internet

If you were able to identify a gap in this our configuration, I salute your observation skills. Because firewall rules apply to traffic coming into an interface and since we didn’t specify a destination network, it means this last rule we just created also allows hosts on the DMZ to open DNS, HTTP, and HTTPS connections to the LAN!

To remedy this situation, we need to add a rule that blocks traffic from the DMZ network to the LAN and place this rule between Policy #3 and Policy #4.

First, let’s create the rule: by default, new rules are added at the bottom.

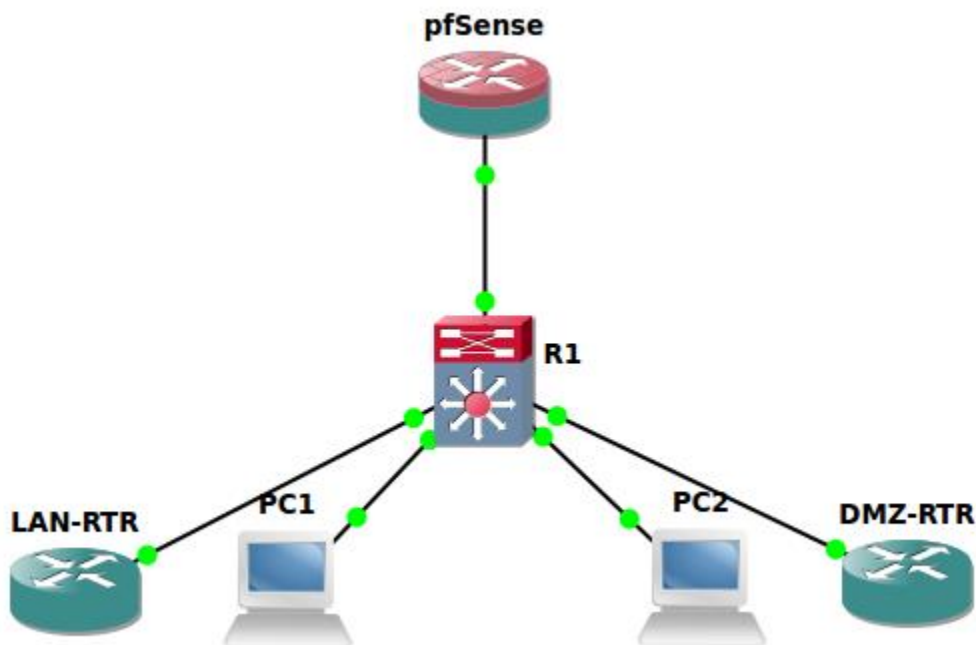
Floating WAN LAN DMZ										
	ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
<input type="checkbox"/>	▶	IPv4 ICMP	*	*	*	*	*	none		Allow ICMP any any
<input type="checkbox"/>	▶	IPv4 TCP	DMZ_SERVERS	*	LAN net	SSH_HTTPS	*	none		Allow SSH/HTTPS from DMZ hosts to LAN
<input type="checkbox"/>	▶	IPv4 TCP	DMZ net	*	*	HTTP_S	*	none		Allow HTTP/S from DMZ to Internet
<input type="checkbox"/>	▶	IPv4 UDP	DMZ net	*	*	53 (DNS)	*	none		Allow DNS from DMZ to Internet
<input checked="" type="checkbox"/>	✖	IPv4 *	*	*	LAN net	*	*	none		Block all traffic from DMZ to LAN

To move the rule to the correct position, we will select the checkbox in front of the rule and click the “Move selected rules before this rule” button for the rule which we want the selected rules to precede (highlighted above):

Floating WAN LAN DMZ										
	ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
<input type="checkbox"/>	▶	IPv4	ICMP	*	*	*	*	none		Allow ICMP any any
<input type="checkbox"/>	▶	IPv4	TCP	DMZ_SERVERS	*	LAN net	SSH HTTPS	none		Allow SSH/HTTPS from DMZ hosts to LAN
<input type="checkbox"/>	✖	IPv4	*	*	LAN net	*	*	none		Block all traffic from DMZ to LAN
<input type="checkbox"/>	▶	IPv4	TCP	DMZ net	*	*	HTTP_S	none		Allow HTTP/S from DMZ to Internet
<input type="checkbox"/>	▶	IPv4	UDP	DMZ net	*	*	53 (DNS)	none		Allow DNS from DMZ to Internet

With this, we have come to the end of our rules definition. The last policy says that everything else should be denied, but that is already implicit in the rules table (just like a Cisco ACL). Explicitly defining a “deny all” rule is useful when you want to log such traffic.

It is always advisable to test your firewall rules to make sure you have not accidentally permitted traffic that should be blocked or denied traffic that should be allowed. In our case, we may want to add some smarter devices (than VPCS) onto the LAN and DMZ that will allow us open SSH and HTTPS connections. Therefore, our GNS3 topology now looks like this:



Note: I have basic IP configuration on the routers. I have also enabled SSH on the LAN-RTR. Both routers are configured to use pfSense as their DNS server. Let's begin our test by checking that the LAN-RTR can ping an Internet URL (i.e., DNS and ICMP):


```
LAN-RTR#ping google.com
Translating "google.com"...domain server (172.16.215.100) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 197.253.18.123, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 46/67/81 ms
```

Next we will ping from a DMZ host to the LAN since ICMP from the DMZ is allowed to any destination (policy #2):

```
DMZ-RTR#ping 172.16.215.201
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.215.201, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/26/42 ms
```

To test the third policy, I will open an SSH connection from the DMZ-RTR to the LAN-RTR:

```
DMZ-RTR#ssh -l cisco 172.16.215.201
Password:
```

```
LAN-RTR>
```

For the fourth policy, I can ping from the DMZ-RTR to an Internet URL. Since this will involve DNS, we can confirm that our fourth policy works:

```
DMZ-RTR#ping google.com
Translating "google.com"...domain server (172.16.100.1) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 197.253.18.123, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 69/74/83 ms
```

Just to confirm that our deny rule works (the one denying DMZ from accessing the LAN), I will change the IP address of the DMZ-RTR from 172.16.100.201 to 172.16.100.220 and try to open SSH to LAN-RTR again. As shown below, it won't work:

```
DMZ-RTR(config)#interface e0/0
DMZ-RTR(config-if)#ip addr 172.16.100.220 255.255.255.0
DMZ-RTR(config-if)#
DMZ-RTR(config-if)#do ssh -l cisco 172.16.215.201
DMZ-RTR(config-if)#
```

Although the webGUI doesn't (yet) provide a way to check the counters on firewall rules, we can use the following command through the Shell: **pfctl -vvsr**:

1. @89(1456227294) block drop in quick on le0_vlan20 inet from any to 172.16.215.0/24 label "USER_RULE: Block all traffic from DMZ to LAN"
- 2.
3. [Evaluations: 11 Packets: 6 Bytes: 264 States: 0]

4.

5. [Inserted: pid 25519 State Creations: 18446735277677785112]

Note: To access the Shell, enter option 8 at the console of pfSense or via the terminal when connected via SSH.

```
*** Welcome to pfSense 2.2.6-RELEASE-pfSense (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 192.168.8.101/24
LAN (lan)      -> le0_vlan10 -> v4: 172.16.215.100/24
DMZ (opt1)     -> le0_vlan20 -> v4: 172.16.100.1/24

8) 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) Disable Secure Shell (sshd)
6) Halt system           15) Restore recent configuration
7) Ping host             16) Restart PHP-FPM
8) Shell

Enter an option: █
```

Option 2 - Snort HTTP Command

SSL/TLS

Encrypted traffic should be ignored by Snort for both performance reasons and to reduce false positives. The SSL Dynamic Preprocessor (SSLPP) decodes SSL and TLS traffic and optionally determines if and when Snort should stop inspection of it. Typically, SSL is used over port 443 as HTTPS. By enabling the SSLPP to inspect port 443 and enabling the noinspect encrypted option, only the SSL handshake of each connection will be inspected. Once the traffic is determined to be encrypted, no further inspection of the data on the connection is made.

By default, SSLPP looks for a handshake followed by encrypted traffic traveling to both sides. If one side responds with an indication that something has failed, such as the handshake, the session is not marked as encrypted. Verifying that faultless encrypted traffic is sent from both endpoints ensures two things: the last client-side handshake packet was not crafted to evade Snort, and that the traffic is legitimately encrypted.

In some cases, especially when packets may be missed, the only observed response from one endpoint will be TCP ACKs. Therefore, if a user knows that server-side encrypted data can be trusted to mark the session as encrypted, the user should use the 'trustservers' option, documented below.

Configuration

1. ports {<port> [<port>< ... >]}

This option specifies which ports SSLPP will inspect traffic on.

By default, SSLPP watches the following ports:

- 443 HTTPS
- 465 SMTPS, 563 NNTPS, 636 LDAPS, 989 FTPS, 992 TelnetS, 993 IMAPS, 994 IRCs, and 995 POPS
- 2. Noinspect - Disable inspection on traffic that is encrypted. Default is off.
- 3. max heartbeat length
- 4. trustservers

DMZ Configuration

Allow TCP/UDP from DMZ subnet to DMZ Address port 53 for DNS from the firewall

Allow TCP from DMZ subnet to DMZ address port 443 for accessing the GUI (optional)

Allow ICMP from DMZ subnet to DMZ address to ping the firewall from the DMZ

Allow any traffic required from DMZ to LAN (if any)

Reject Any from DMZ subnet to RFC1918 – Do not allow DMZ to reach LAN or other private networks

Allow Any from DMZ subnet to any – Internet access rule

2. Write an example of a firewall rule that will block FTP traffic originating from the Untrusted network.

Block "untrusted network" from communicating with DMZ


To prevent "untrusted network" from being able to communicating with DMZ networks, create rules that block connections via your WAN(s). For each alias URL and each WAN, create a firewall rule:


- in pfSense, visit the **Firewall** → **Rules** → **WAN** tab and press the upper-right **+** button
- for **Action**, select **Block**
- for **Interface**, select **WAN**
- for **TCP/IP Version**, select **IPv4**
- for **Protocol**, select **any**
- for **Source**:
 - for **Type**, select **Single host or alias**
 - for **Address**, enter the name of the **URL alias** which contains the IP addresses you want to block
- for **Destination** select **any**

- enter a **Description**
- press the **Save** button
- press the **Apply Changes** button

If you visit the **Firewall** → **Rules** → **WAN** tab, you should now have something like:

Firewall: Rules

WAN										
ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	
	*	RFC 1918 networks	*	*	*	*	*		Block private networks	
	*	Reserved/not assigned by IANA	*	*	*	*	*	*	Block bogon networks	
	IPv4 *	BadGuyDROPIlist	*	*	*	*	none		Block criminal-controlled systems from all	

If you press the  button to the right of the rule you just created, you should now have something like:

Firewall: Rules: Edit

Edit Firewall rule

Action

Block

Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

Disabled

☐ Disable this rule

Set this option to disable this rule without removing it from the list.

Interface

WAN

Choose on which interface packets must come in to match this rule.

TCP/IP Version

IPv4

Select the Internet Protocol version this rule applies to

Protocol

any

Choose which IP protocol this rule should match.
Hint: in most cases, you should specify *TCP* here.

Source

☐ not

Use this option to invert the sense of the match.

Type:

Single host or alias

Address:

BadGuyDROPIlist

 /

Destination

☐ not

Use this option to invert the sense of the match.

Type:

any


Address: /

Log

☐ Log packets that are handled by this rule

Hint: the firewall has limited local log space. Don't turn on logging for everything. If you want to do a lot of logging, consider using a remote syslog server (see the Diagnostics: System logs: Settings page).

Description

 Block criminal-controlled systems from all

You may enter a description here for your reference.

Option 2 – SNORT RULE

Examples/Default Configuration from snort.conf

```
preprocessor ftp_telnet: \  
global \  
encrypted_traffic yes \  
inspection_type stateful
```

```
preprocessor ftp_telnet_protocol:\  
telnet \  
normalize \  
ayt_attack_thresh 200
```

```
# This is consistent with the FTP rules as of 18 Sept 2004.  
# Set CWD to allow parameter length of 200  
# MODE has an additional mode of Z (compressed)  
# Check for string formats in USER & PASS commands  
# Check MDTM commands that set modification time on the file.
```

```
preprocessor ftp_telnet_protocol: \  
ftp server default \  
def_max_param_len 100 \  
alt_max_param_len 200 { CWD } \  
cmd_validity MODE < char ASBCZ > \  
cmd_validity MDTM < [ date nnnnnnnnnnnnn[n[n[n]]] ] string > \  
chk_str_fmt { USER PASS RNFR RNTD SITE MKD } \  
telnet_cmds yes \  
ignore_data_chan yes  
preprocessor ftp_telnet_protocol: \  

```

```
ftp client default \  
max_resp_len 256 \  
bounce yes \  
telnet_cmds yes  
ftpbounce
```

The ftpbounce keyword detects FTP bounce attacks.

Format

```
ftpbounce;
```

Example

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP PORT bounce attempt"; \  
flow:to_server,established; content:"PORT"; nocase; ftpbounce; pcre:"/^PORT/smi"; \  
classtype:misc-attack; sid:3441; rev:1;)
```

3. Does the order of rule placement in a firewall matter and if so, why?

Firewall rules have a priority order that determines the order in which the rules are applied to network traffic. Firewall rules are shown as a list on the Rules page. The rules are applied from top to bottom, and the first rule that matches the traffic overrides all the other rules below. The main principle is to allow only the needed traffic and block the rest. Therefore, the last rule of a security level is the Deny rest rule. It blocks all the traffic that the rules above it do not specifically allow.

An example of how the priority order works

Following examples clarify how you can control which rules are applied to a specific network traffic by changing the order of firewall rules.

- You have added a rule that denies all outbound FTP traffic. Above the rule in the rules list, you add another rule that allows an FTP connection to your Internet Service Provider's IP address. This rule allows you to create an FTP connection to that IP address.
- You have added a rule that allows you to create an FTP connection to your Internet Service Provider's IP address. Above the rule in the rules list, you add another rule that denies all FTP traffic. This rule prevents you from creating an FTP connection to your Internet Service Provider's IP address (or any other IP address).

4. Which Linux distribution is pfSense based on?

Pfsense 2.4.5 - Release (AMD64)

5. Which logs would be useful in monitoring traffic on the firewall and why?

Linux Firewall Logs

The Linux kernel has a packet filtering framework called **netfilter**. This framework lets you permit, drop, and modify the traffic that comes in and out of a system. A tool, **iptables** furthers this functionality with a firewall, which you can configure using rules. Additional programs, like fail2ban, also rely on iptables to block attackers.

How does iptables work?

Iptables is a command-line interface to the packet filtering capabilities in netfilter. However, we won't distinguish between iptables and netfilter for the purposes of this article. To keep things clear we will refer to the entire concept as iptables.

The packet filtering function offered by iptables is structured as tables, targets, and chains. Put simply, a table lets you process packets in a certain way. The filter table is the default table. Chains are connected to these tables. You can monitor traffic at different points, using these chains. You can see traffic, as it arrives on the network interface or before it is passed over to a process. You can also add rules to the chains so that they match certain packets, for example,

TCP packets going to port 70, and connect it to a target. A target will determine if the packet should be permitted or blocked.

When a packet enters or exits (according to the chain) iptables compares it against rules in these chains one at a time. When it identifies a match, it isolates the target and carries out the required action. If it doesn't identify a match with any of the rules it carries out the action specified by the default policy of the chain. The default policy also acts as a target. By default, all chains have a policy of permitting packets.

Working with and interpreting iptable firewall logs

To create firewall logs, the kernel needs to be firewall logging enabled. By default, matched packets are logged as kern.warn (priority 4) messages. You can change the log priority with the -log-level option to -j LOG.

The majority of the IP packet header fields are disclosed when a packet matches a rule with the LOG target. By default, firewall log messages are written to /var/log/messages.

System Monitoring

pfSense provides a wealth of information about the state of the firewall, its services, traffic flowing through the firewall, and log data.

Logs

Logs on pfSense contain recent events and messages from daemons. These messages can be stored locally on a limited basis, or forwarded to a central logging server for long-term storage, better reporting, alerting, and so on.

System Logs

Firewall Logs

Filter Log Format for pfSense 2.2

Gateway Logs

NTP Logs

Package Logs

PPP Logs

Resolver Logs

Routing Logs

Log Settings

Adjusting the Size of Log Files

Copying Logs to a Remote Host with Syslog

Working with Binary Circular Logs (clog)

Troubleshooting "login on console as root" Log Messages

Troubleshooting "promiscuous mode enabled" Log Messages

Status Information

These articles cover various ways to check the status of services or features of the firewall, or the firewall itself. They also cover simple network diagnostic tests.

Show States

Show Source Tracking

States Summary

Packet Filter Information

Gateway Status

Using an Alternate Monitor IP Address for Gateway Monitoring

CARP Status

Interface Status

Viewing Active Network Sockets

Services Status and Control

SMART Status

System Activity

ARP Table

Troubleshooting ARP Move Log Messages

NDP Table

Hardware Monitoring Support

Performing a Packet Capture

Ping Host

Test Port

Traceroute

Traffic Monitoring

These articles cover monitoring traffic on interfaces as well as using additional packages for more detailed monitoring of user throughput/usage.

Monitoring Graphs

Monitoring Bandwidth Usage

Viewing Real-Time Traffic Graphs

Exporting NetFlow with softflowd

6. How could you block users from accessing inappropriate websites?

Blocking Web Sites

There are several options for blocking websites with pfSense® software, some of which are described on this article. It's not an exact science, but these solutions typically function well enough for a majority of use cases.

a. Using DNS

If the built in DNS Resolver or Forwarder are active an override can be entered there to resolve the unwanted website to an invalid IP address such as 127.0.0.1. With the DNS Resolver, additional methods are possible via custom options. This first example will prevent any host under the given zone from being resolved by clients:

server:

local-zone: "movie.edu" static

When the firewall enforces DNS resolution in this way, the firewall must also force clients to resolve DNS using the firewall. Otherwise, clients could bypass the restrictions by using alternate DNS servers. See Redirecting Client DNS Requests for details.

b. Using Firewall Rules (not advisable for sites with low TTL across many servers)

If a website rarely changes IP addresses, access to it can be blocked using an alias containing its IP addresses and then using this alias in firewall rules.

c. Using a Proxy

If web traffic flows through a proxy server, that proxy server can likely be used to prevent access to such sites. For example, Squid has an add-on called SquidGuard which allows for blocking web sites by URL or other similar criteria. There is a very brief introduction to Squid and SquidGuard to be found in A Brief Introduction to Web Proxies and Reporting: Squid, SquidGuard, and Lightsquid.

In modern environments this is not effective as it works best on HTTP, and not HTTPS. HTTPS can sometimes be filtered via peek/splice to inspect SNI and similar aspects of connections, but even that fails with modern security practices like encrypted SNI. A non-transparent proxy setup is more likely to work correctly but is more complicated to setup and maintain.

d. Prevent Bypassing Restrictions

With any of the above methods, there are many ways to get around the defined blocks. The easiest and likely most prevalent is using any number of proxy websites. Finding and blocking all of these individually and keeping the list up to date is impossible. The best way to ensure these sites are not accessible is using an external proxy or content filtering capable of blocking by category.

To further maintain control, use a restrictive egress ruleset and only allow traffic out to specific services and/or hosts. For example, only allow DNS access to the firewall or the DNS servers specifically used for LAN clients (Redirecting Client DNS Requests). Also, if a proxy is in use on the network, make sure to disallow direct access to HTTP and HTTPS through the firewall and only allow traffic to and/or from the proxy server.

7. How would you implement secure file transfer between the Trusted network and the web server in the DMZ?

Secure FTP

Configuring Dual secured firewalls whereby the configuration between the DMZ and internal firewall is hardened, and the outer firewall like with trusted network required authentication. Firewalls are positioned on each side of the DMZ. The outer firewall is more of a “screening” firewall; in that it will block certain protocols but let others through that are allowed in the DMZ. The outer firewall is allowing FTP, port 25 traffic (as well as other protocols such as http, port 80) into the DMZ. The inner firewall, which is protecting the internal network, is denying these protocols from entering the internal network. Thus, any FTP transfer is secured and authenticated. The advantage to placing servers in this zone is to prevent any unauthorized traffic from entering the Internal Network. Once the DMZ is designed and built, the Intrusion detection system devices and the FTP servers can be positioned.

8. Given that you have an IT administrator workstation with an address of 192.168.0.30/24, write an example of a firewall rule to allow only that workstation to access the webserver using HTTPS and FTPS.

Server Behind pfSense

FTPS, or encrypted FTP, is not affected. The proxy could not have affected its traffic before.

A server behind pfSense would work fine with active mode, there would be no difference here. In active mode the server would make outbound connections back to the client, so as long as the firewall rules on the interface containing the server allow outbound connections, it will work.

A server behind pfSense running in Passive mode will function but requires a few items to be configured:

1. Port forwards or 1:1 NAT to forward not only port 21, but also the passive port range in to the server

2. The passive port range must be configured on the server, corresponding to the range of ports forwarded in the previous step.
3. The server may also need to be configured to account for NAT. Some clients will ignore private addresses in passive responses so this may not be necessary.

Sample Configuration for vsftpd

In vsftpd.conf:

```
# Do not allow the client to use PORT
port_enable=NO
# Use the hostname in the PASV response (DNS must be setup and match!)
pasv_addr_resolve=YES
# Enable Passive Mode
pasv_enable=YES
# Set the passive port range (1000 ports)
pasv_min_port=20000
pasv_max_port=20999
```

9. What would you recommend to improve the security of the DMZ and the Trusted network, and why?

There is the need to close unused or vulnerable ports, encrypt and harden data that flows from the DMZ to the trusted network. Traffic from the unsecured network should be blocked from reaching the trusted network. Here are four tips to help ensure that a DMZ is secure:

PRESERVE ISOLATION AS MUCH AS POSSIBLE.

Keep the rules that allow traffic between the DMZ and an internal network as tight as possible. Too often, administrators seeking to troubleshoot a problem create a rule allowing full access between a DMZ system and a back-end server on the internal network (or the entire internal network). This defeats the purpose of the DMZ and effectively merges it with the internal network. Instead, create specific firewall rules that allow communication only between specific servers on specific ports required to meet business requirements.

PRACTICE GOOD VULNERABILITY MANAGEMENT.

DMZ servers are exposed to the world, so take extra steps to ensure that they are fully patched to deal with the latest security vulnerabilities. Many security professionals recommend daily, automated vulnerability scans of DMZ systems that provide rapid alerts of newly detected vulnerabilities. In addition, consider patching DMZ systems on a much more frequent basis than protected systems to reduce the window of vulnerability between the time when a patch is released and its application to DMZ servers.

USE APPLICATION LAYER DEFENSES FOR EXPOSED SERVICES.

Choose a network firewall that has strong application layer protection, rather than just a port filter. A firewall should have the ability to inspect the content of traffic and block malicious requests. One common example of this is screening inbound web requests for signs of embedded SQL injection attacks, preventing them from even reaching the web server.

MONITOR, MONITOR, MONITOR.

The DMZ should be one of the major focuses of an organization's network monitoring efforts. Use intrusion detection systems, security incident and event management systems, log monitoring and other tools to remain vigilant for signs of an attack.

DMZ systems are at the pointy end of the network security spear and are subject to external attack on a daily basis. For this reason, it's important to take the time to ensure that they are among the most secure servers in an organization and are rigorously maintained.

10. List and briefly describe 3 commercial IDS/IPS products you would recommend to a client.

Snort - Packet sniffer, Packet logger, Threat intelligence, Signature blocking, Real-time updates for security signatures, In-depth reporting, Ability to detect a variety of events including OS fingerprinting, SMB probes, CGI attacks, buffer overflow attacks, and stealth port scans. Installs on Unix, Linux, and Windows.

Solar Winds - This powerful security tool uses both network-based and host-based intrusion detection methods and takes preventative action. Pre-installed presets will get you up and running in no time. Installs on Windows Server or via cloud.

Suricata - Collects data at the application layer, Ability to monitor protocol activity at lower levels such as TCP, IP, UDP, ICMP, and TLS, real-time tracking for network applications such as SMB, HTTP, and FTP, Integration with third-party tools such as Anaval, Squil, BASE, and Snorby, built-in scripting module, uses both signature and anomaly-based methods, Clever processing architecture. Works on Unix, Linux, Windows, and Mac-OS.

Splunk - Widely-used network analysis tools that has intrusion prevention features. Available for Windows, Linux, and in the Cloud.

Resources – Exam Answer Sources

1. [pfSense Series: Firewall Rules - Intense School](#)
2. [Setting Up Blocking Firewall Rules \(derman.com\)](#)
3. [How Does the Priority Order of Firewall Rules Work? | Linux Security | 11.00 | F-Secure User Guides \(f-secure.com\)](#)
4. <https://docs.netgate.com/pfsense/en/latest/config/console-menu.html>
5. [The Significance and Role of Firewall logs \(exabeam.com\)](#), [System Monitoring — pfSense Documentation \(pfsense-docs.readthedocs.io\)](#)
6. [pfSense Configuration Recipes — Blocking Web Sites | pfSense Documentation \(netgate.com\)](#)
7. [FTP Server Security Strategy for the DMZ \(giac.org\)](#)
8. [pfSense Configuration Recipes — Using NAT and FTP without a Proxy | pfSense Documentation \(netgate.com\)](#)
9. [Four Tips for Securing a Network DMZ | FedTech Magazine](#)
10. [Top 10 BEST Intrusion Detection Systems \(IDS\) \[2020 Rankings\] \(softwaretestinghelp.com\)](#)

THANK YOU!