CySA+ Lab Series

Lab 19: Creating New Rules and Testing IDS/IPS Using Snort

Document Version: 2022-10-10

Material in this Lab Aligns to the Following	
CompTIA CySA+ (CS0-002) Exam Objectives	1.3 - Given a scenario, perform vulnerability management activities 1.4 - Given a scenario, analyze the output from common vulnerability tools 1.7 - Given a scenario, implement controls to mitigate attacks and software vulnerabilities 3.1 - Given a scenario, analyze data as part of security monitoring activities 3.2 - Given a scenario, implement configuration changes to existing controls to improve security 3.4 - Compare and Contrast automation concepts and technologies 4.2 - Given a scenario, apply the appropriate incident response procedure 4.3 - Given an incident, analyze potential indicators of compromise 4.4 - Given a scenario, utilize basic digital forensics techniques 5.2 - Given a scenario, apply security concepts in support of organizational risk mitigation
All-In-One CompTIA CySA+ Second Edition ISBN-13: 978-1260464306 Chapters	3: Vulnerability Management Activities 4: Vulnerability Assessment Tools 7: Mitigating Controls for Attacks and Software Vulnerabilities 11: Data Analysis in Security Monitoring Activities 12: Implement Configuration Changes to Existing Controls to Improve Security 14: Automation Concepts and Technologies 16: Appropriate Incident Response Procedures 17: Analyze Potential Indicators of Compromise 18: Utilize Basic Digital Forensics Techniques 20: Security Concepts in Support of Organizational Risk Mitigation

Contents

Introduction

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are network security tools that continually monitor network traffic for attacks and, in the case of IPS, can take action to mitigate malicious activity, including reporting and/or blocking.

IDS/IPS can be implemented as a hardware appliance or can be implemented in software and added to firewalls. It is placed inline, within the flow of traffic, and is usually incorporated into or implemented just behind the firewall. They identify threats using the following techniques:

- **Signature-Based:** Matches the threat activity to well-known signatures. However, it can only detect and mitigate attacks that have been previously identified.

- **Anomaly-Based**: Monitors traffic looking for abnormal behavior by comparing network activity against baseline standards. It can identify attacks that have not been previously documented but can produce false positives. Artificial Intelligence and Machine Learning are being used to improve detection with fewer false positives.

- **Policy-Based**: Employs security policies and rules that have been identified and will detect and block security policies.

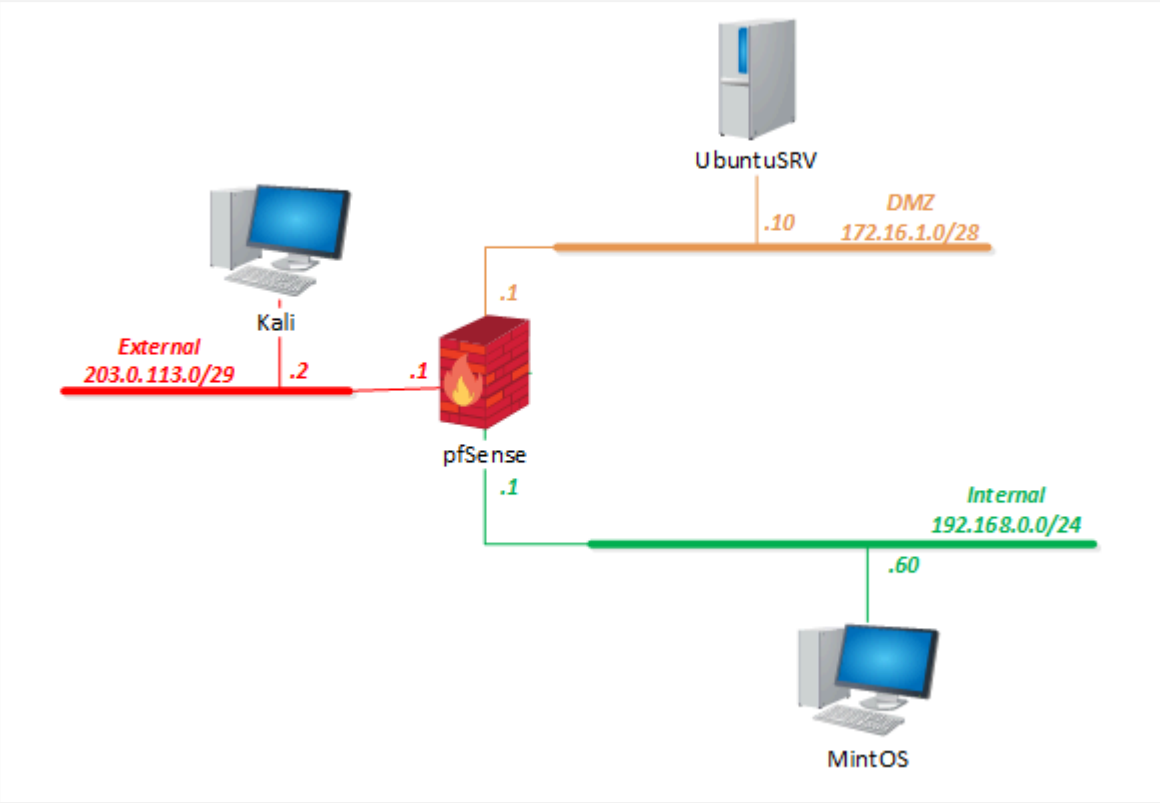
In this lab, you will explore the Snort IDS by viewing rules, creating new rules, and triggering and monitoring events based on the new rules.

> "*Snort* is the foremost Open Source Intrusion Prevention System (IPS) in the world."
>
> "*Snort* can be deployed inline to stop these packets, as well. Snort has three primary uses: As a packet sniffer like *tcpdump*, as a packet logger — which is useful for network traffic debugging, or it can be used as a full-blown network intrusion prevention system. *Snort* can be downloaded and configured for personal and business use alike."
>
> *https://www.snort.org/*

Objectives

- View Snort Rules
- Create New Snort Rules
- Test the New Rule and Trigger an Event

Lab Topology



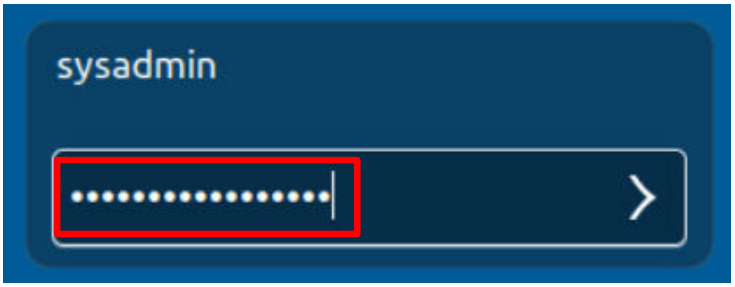
Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

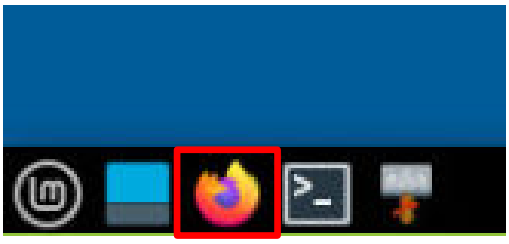
Virtual Machine	IP Address	Account	Password
WinOS (Server 2019)	192.168.0.50	Administrator	NDGlabpass123!
MintOS (Linux Mint)	192.168.0.60	sysadmin	NDGlabpass123!
OSSIM (AlienVault)	172.16.1.2	root	NDGlabpass123!
UbuntuSRV (Ubuntu Server)	172.16.1.10	sysadmin	NDGlabpass123!
Kali	203.0.113.2	sysadmin	NDGlabpass123!
pfSense	203.0.113.1 172.16.1.1 192.168.0.1	admin	NDGlabpass123!

1 Access Snort on the pfSense Firewall

1. Set the focus on the **MintOS** computer.
2. Log in to the sysadmin account using the password: `NDGlabpass123!`

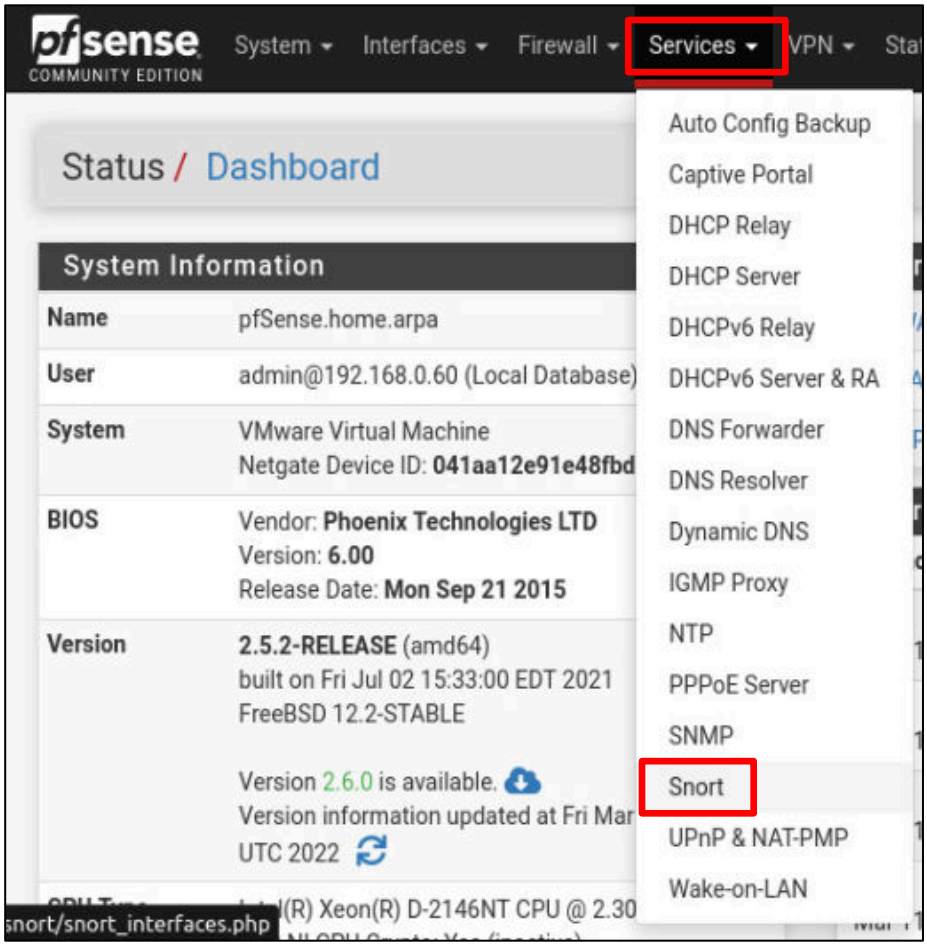


3. Open the browser by clicking on the **Firefox** icon located in the toolbar at the bottom of the window.

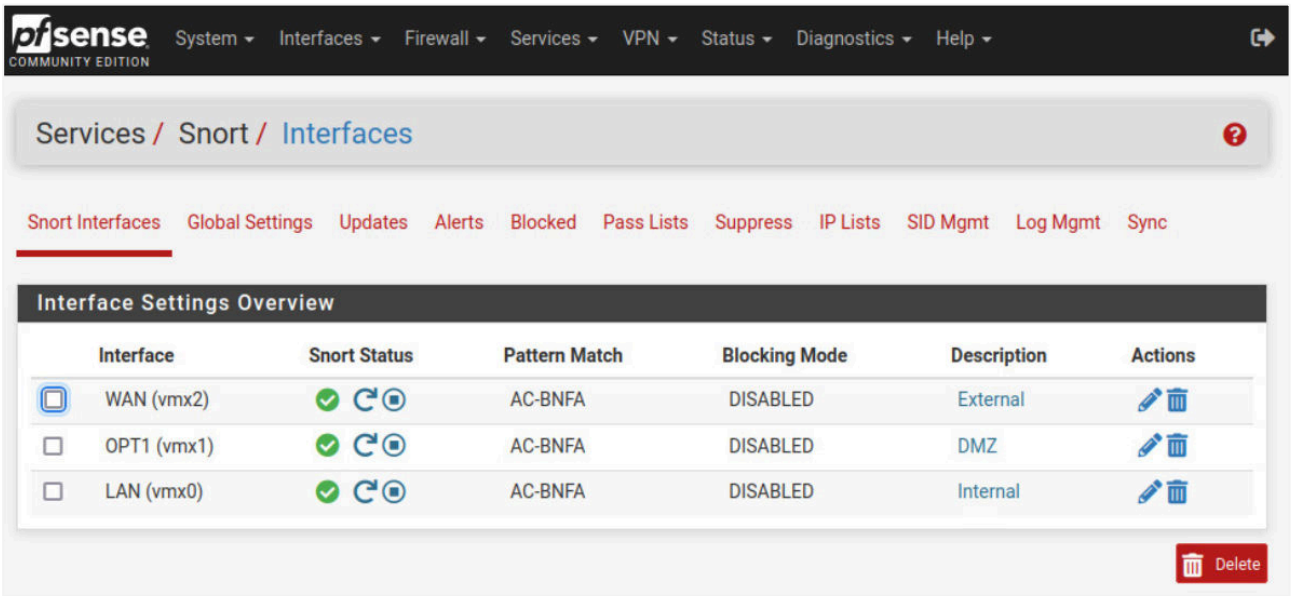


4. In the address bar of the browser, type `192.168.0.1`, which is the IP address of the *pfSense* server.
5. Log in as `admin` using the password `NDGlabpass123!` and click the **SIGN IN** button.

6. You will see the *Dashboard* for the *pfSense* firewall. Click on the **Services** menu item, then click on **Snort.**



You will see the *Services/Snort/Interfaces* page.



7. Remain on the **Snort Interfaces** page and continue to the next section.

2 Snort Rules

The "secret sauce" of *Snort* is the rules that are applied to traffic on the network. Rules are written to differentiate between normal traffic and malicious activities. Rules are designed to detect vulnerabilities. In order to create a rule, an understanding of how the vulnerability works is required.

Snort uses a series of rules to define malicious network activity and then applies those rules to find traffic that matches the rules and generates alerts. It can also block the attack.

Snort rules can identify attack methods, such as OS Fingerprinting, Denial of Service, Buffer Overflow, Port Scans, Server Message Block Probes, and more.

Snort keeps rules organized in collections of "Rulesets", which are the categories where rules can be grouped. *Snort* is preloaded with a Ruleset of Emerging Threat (ET) Open Rules. There are additional rules that can be downloaded from both Cisco Talos (the team that creates and publishes *Snort* rules) as well as *Snort GPLv2 Community* supplied and *Snort OPENAPPID* rules. In order to receive the Snort Subscriber Rules, you must register either for a free Registered User Rules account or a paid Snort Subscriber Rule Set account.

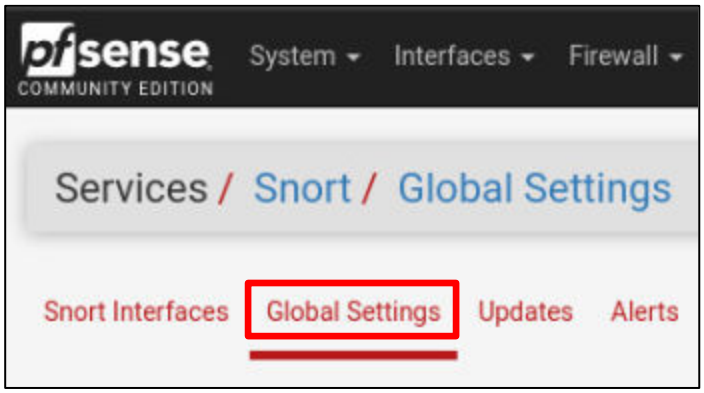
You can also create custom rules that can be installed and applied to your *Snort* installation.

2.1 Configure Snort Rules Updates

Since new vulnerabilities are discovered every day, *Snort* Rules are updated frequently.

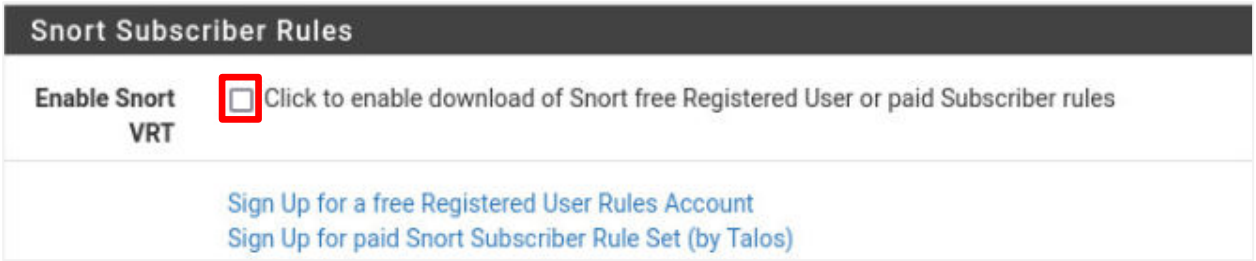
Snort rules are downloaded from the internet on either a scheduled or on-demand basis. Since the lab environment does not have internet access, you will not be able to download/update the rules. The following will show you how to configure *Snort* for configuring rule updates.

1. To enable downloading of *Snort* rules, click on the **Global Settings** menu item.



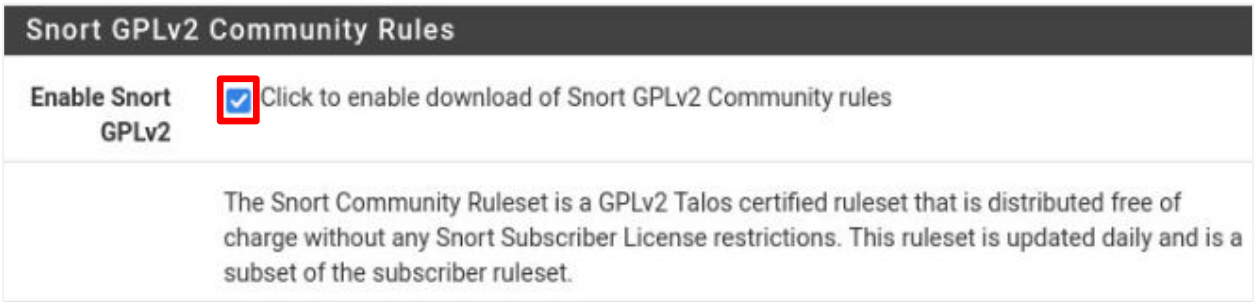
The Global Settings will allow you to enable the downloading of Snort Rules updates.

2. Under *Snort Subscriber Rules*, leave the **Enable Snort VRT** rules checkbox unchecked.



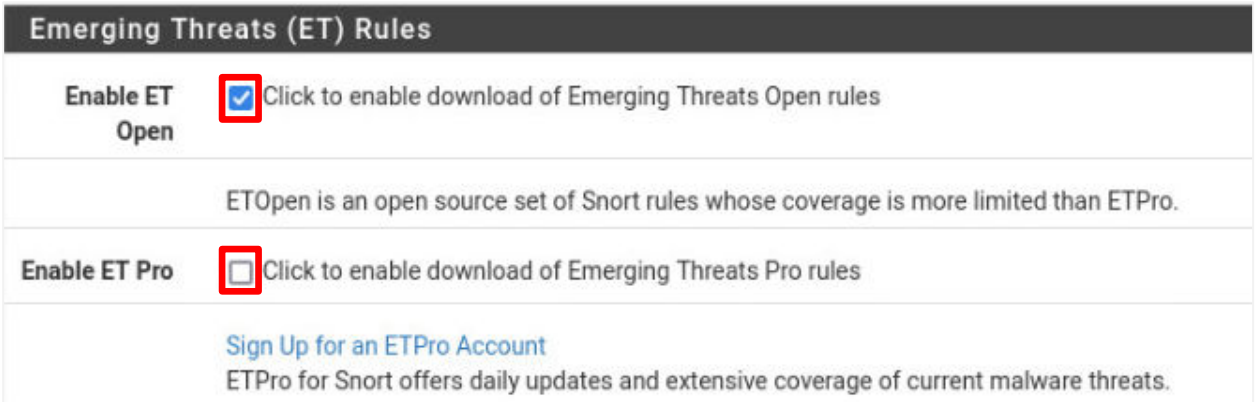
Selecting the *Snort Subscriber Rules* will enable the downloading of either the *Registered User Rules* (which is free) or the *Snort Subscriber Rule Set* (which is a paid subscription). When checked, an additional field will be displayed asking for an *Oinkmaster Code* which you will receive via email when you register.

3. Under the *Snort GPLv2 Community Rules* section, make sure **Enable Snort GPLv2 rules** is checked.



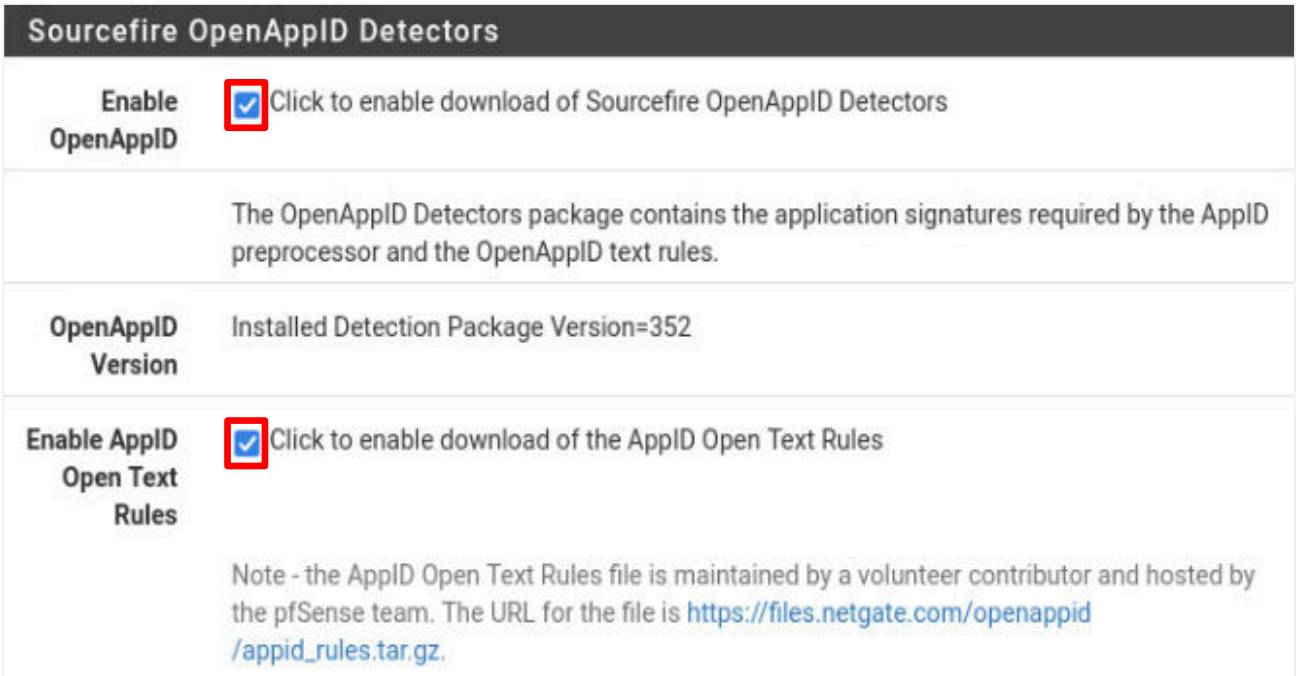
Selecting the *Snort GPLv2 Community Rules* will enable the downloading of the *GPLv2 Talos* certified ruleset. These rules are free of charge and do not need a subscription.

4. Under the **Emerging Threats (ET) Rules** section, make sure the **Enable ET Open** box is checked and then **Enable ET Pro** box is unchecked.



Emerging Threats Rules are sets of rules which significantly enhance malware detection. Selecting the *Enable ET Open* in the *Emerging Threat Rules* section will enable the downloading of the *Open ET* rules. These rules are free, do not require a subscription, and are maintained by the *Snort* community. The rules cover scanning activities, protocol attack patterns, blacklists, and more. There is an optional checkbox for enabling and downloading *Emerging Threats Pro* rules. These rules are not free and require a subscription.

5. Under *the Sourcefire OpenAppID Detectors* section, check the **Enable OpenAppID** and **Enable AppID Open Text Rules** checkboxes.

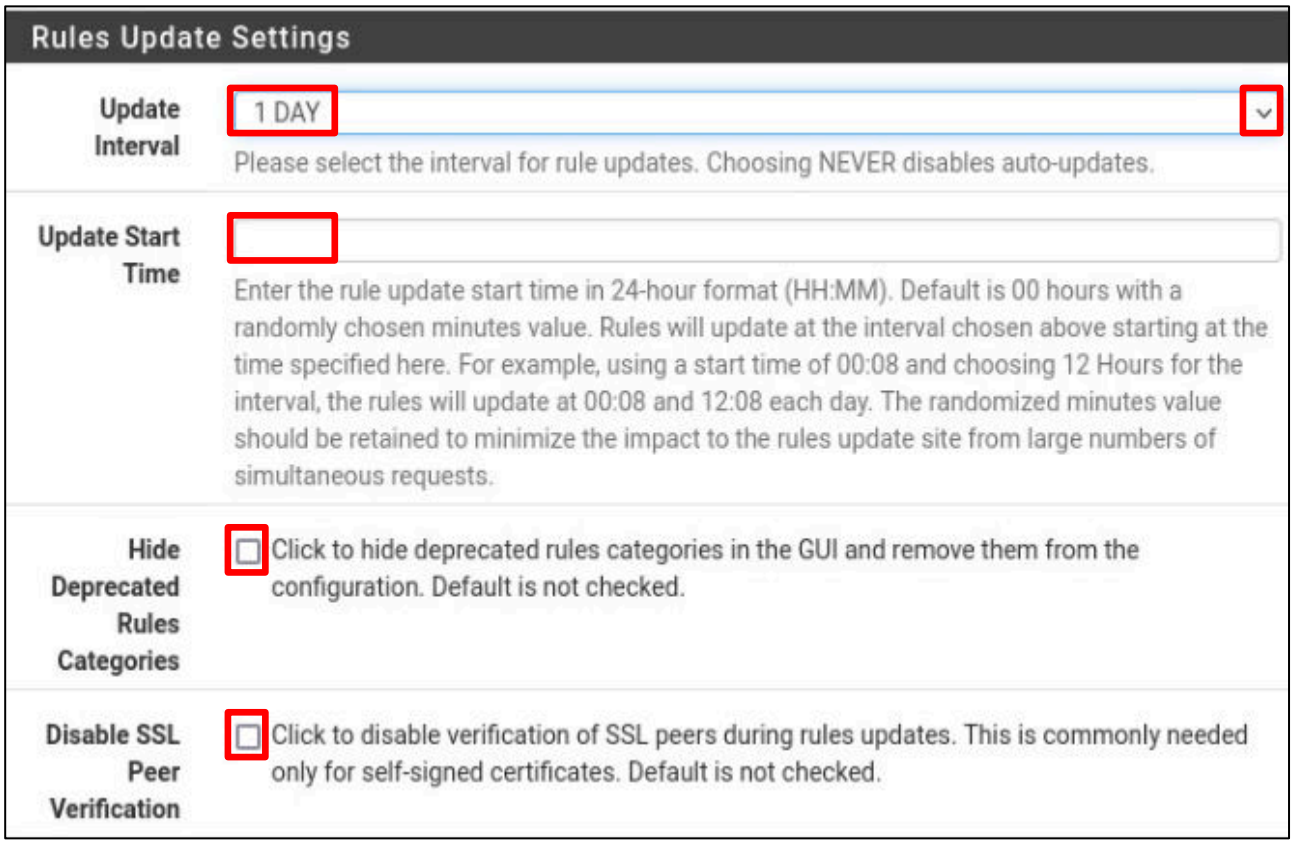


Selecting the *Sourcefile OpenAppID Detectors* will enable the downloading of the application-focused detection language *OpenAppID*. It is not threat detection; it instead is an application identification processing module. According to Martin Roesch, the author of *Snort*:

> *"OpenAppID puts control in the hands of users, allowing them to control application usage in their network environments and eliminating the risk that comes with waiting for vendors to issue updates. Practically speaking, we're making it possible for people to build their own open source Next-Generation Firewalls."*

OpenAppID can be used to block over 2600 services like Facebook, Twitter, Snapchat, Netflix, and Amazon.

6. Under the *Rules Update Settings* section, click the list arrow on the right side of the *Update Interval* entry and select **1 DAY**. Delete the *Update Start Time*, leaving the entry blank. Leave the *Hide Deprecated Rules Categories* and *Disable SSL Peer Verification* entries unchecked.



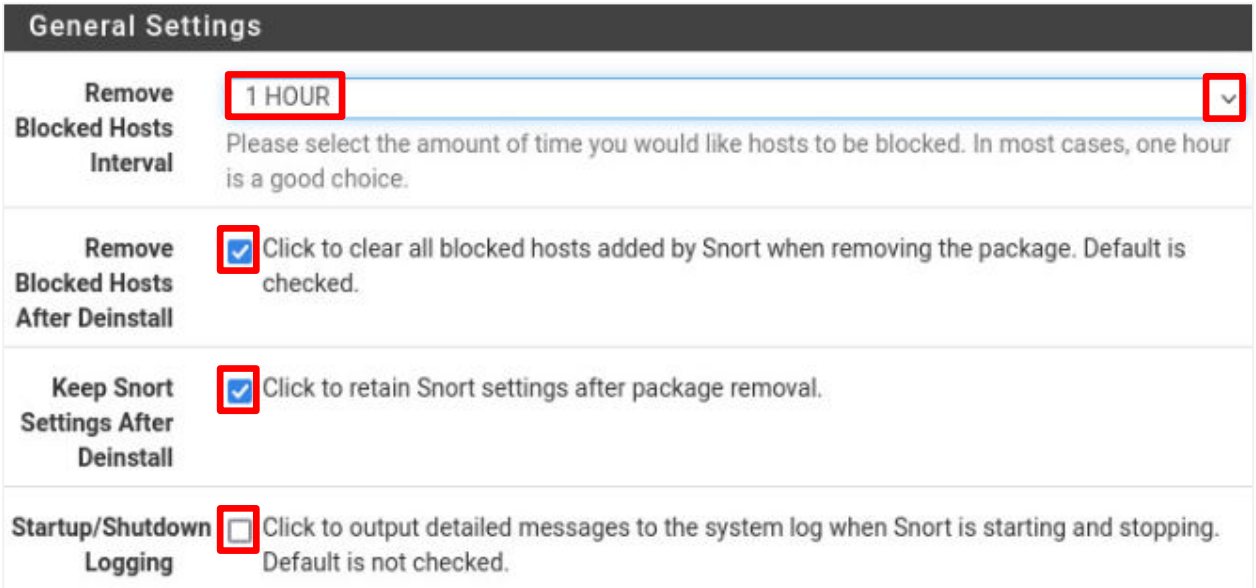
- **Update Interval:** Sets the frequency of updates. **Never**, means that updates will need to be done manually.

- **Update Start Time**: The start time (in 24-hour format) of the updates.

- **Hide Deprecated Rules Categories**: Hide deprecated rules and remove them from the Snort configuration.

- **Disable SSL Peer Verification**: Disable SSL verification of update peers. It needs to be checked if using self-signed certificates but is typically not checked.

7. Under the *General Settings* section, use the list arrow on the right side of the *Remove Blocked Hosts Interval* entry and change it to **1 HOUR**. Leave the **Remove Blocked Hosts After Deinstall** and **Keep Snort Settings After Deinstall** boxes checked and leave the **Startup/Shutdown Logging** box unchecked.



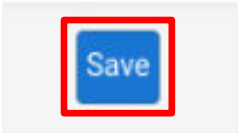
- **Remove Blocked Hosts Interval**: Allows the selection of the amount of time that a host will remain blocked when malicious activity is detected.

- **Remove Blocked Hosts After Deinstall**: When checked, hosts that have been previously blocked for malicious activity by Snort and have been detected will be cleared when Snort is removed from *pfSense*.

- **Keep Snort Settings After Deinstall:** When checked, the configuration settings will be retained after Snort is uninstalled.

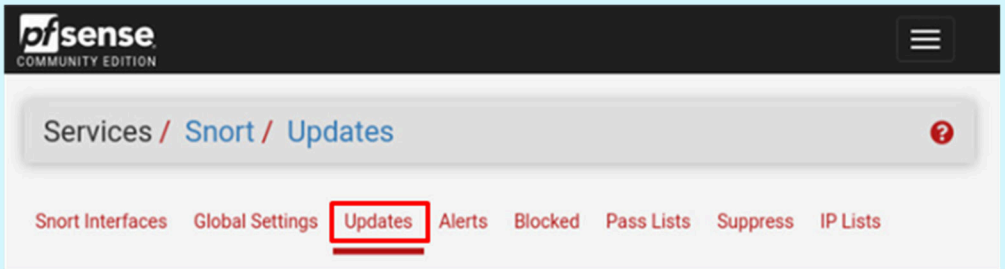
- **Startup/Shutdown Logging**: When checked, detailed messages will be sent to the system log when Snort is started and stopped.

8. Click on the **Save** button at the bottom of the page.

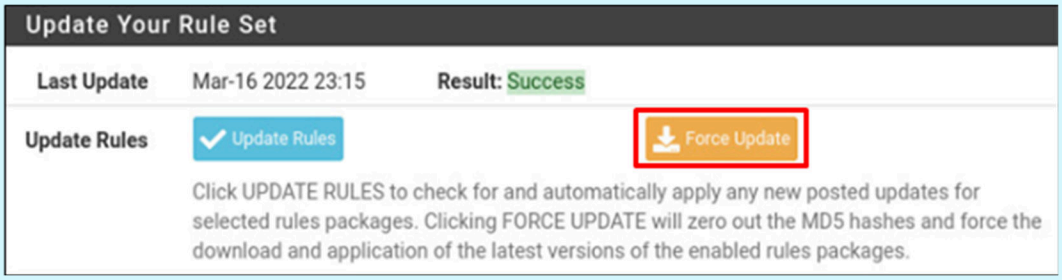


This lab environment does not have internet access, so the **Snort Rule** update cannot be done. In a production environment, the next two steps would be required.

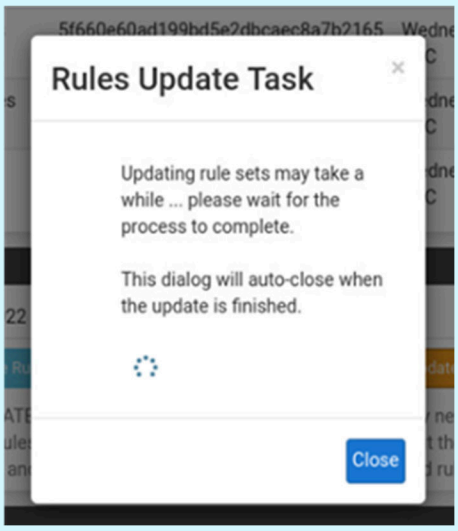
On the top of the window, click on the **Updates Menu** Item.



Scroll down to the **Update Your Rule Set** section and click on the **Force Update** button.



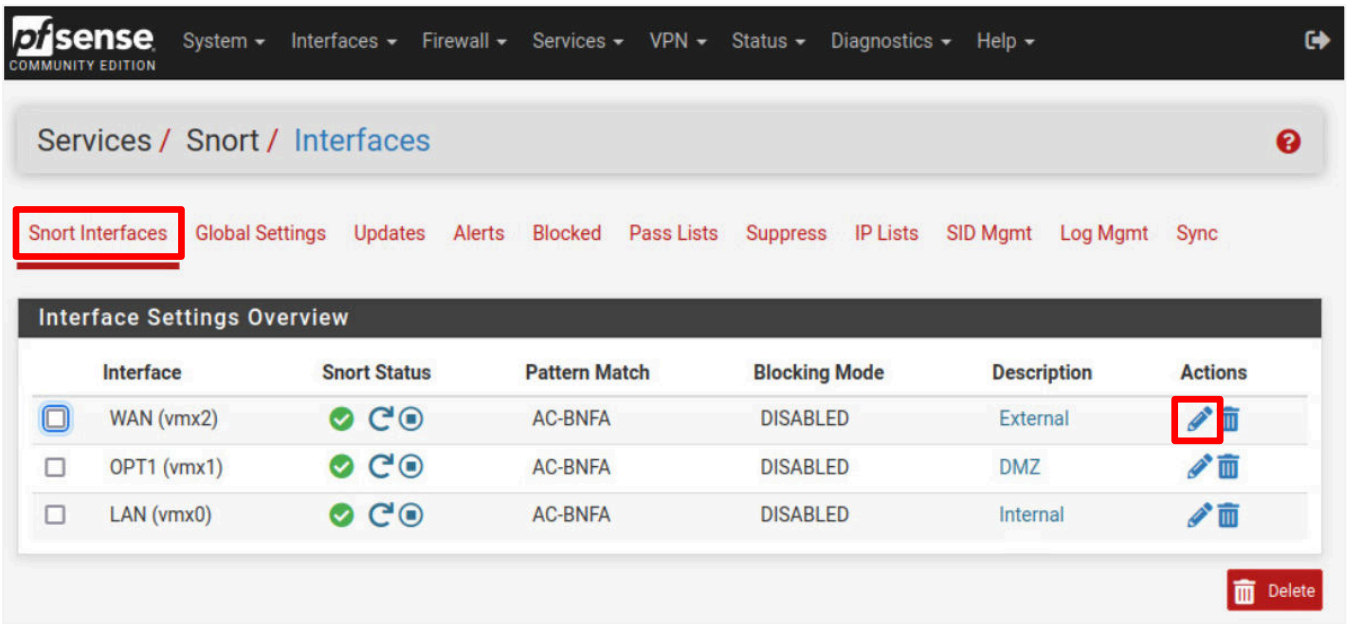
This will force a rule update. The Rule Update Task process window will be shown. When the task is complete, the window will close.



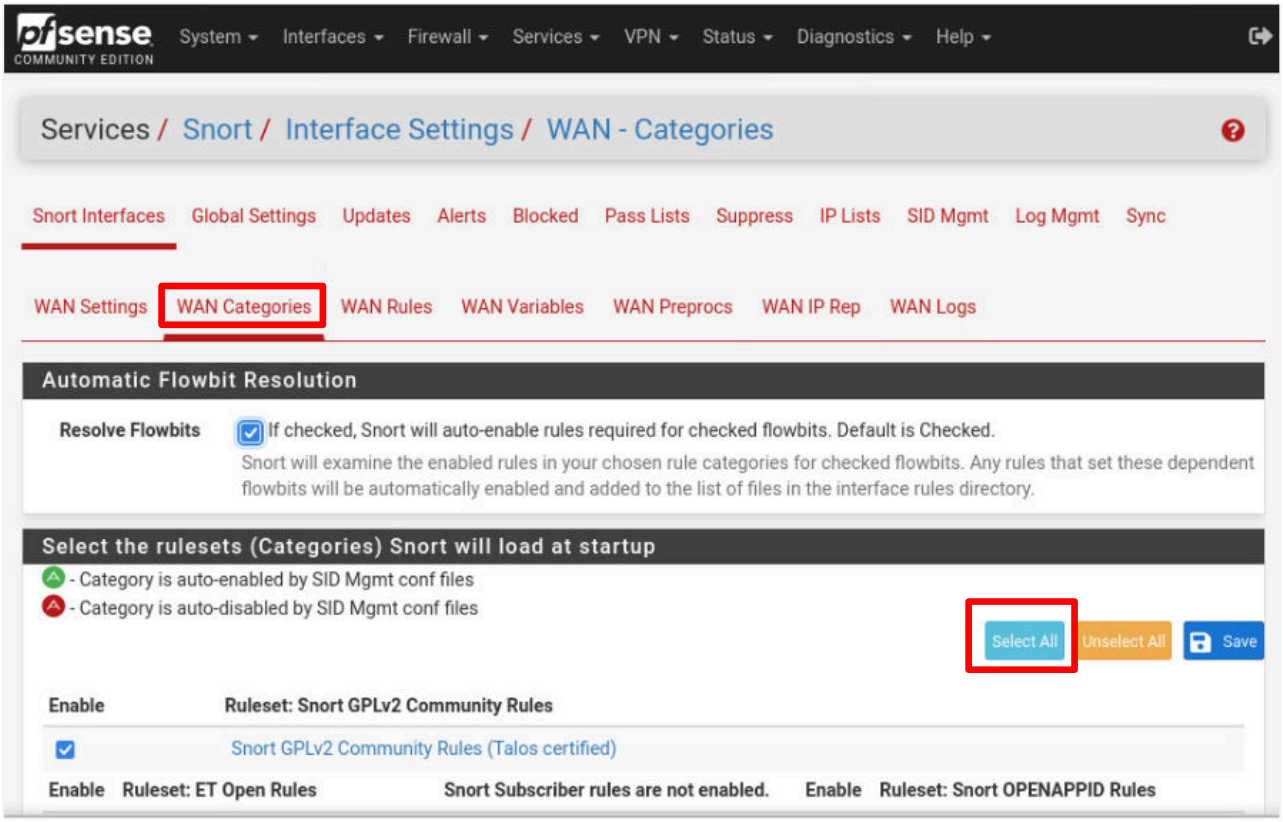
9. Remain on the *Snort* interface and continue to the next section.

2.2 Activate Snort Rules

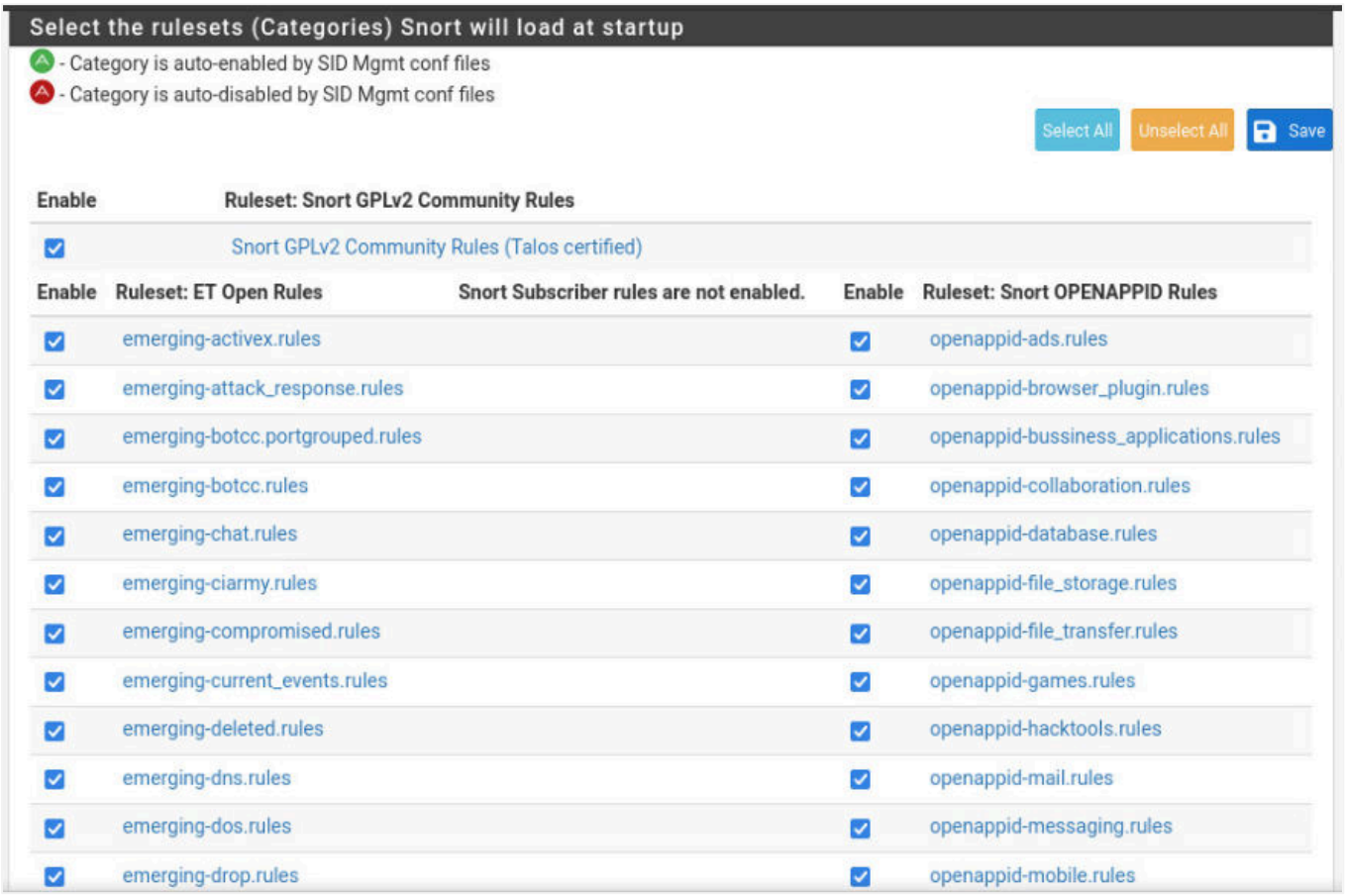
1. Go back to the top of the window and click on the **Snort Interfaces** menu option. To see the rules that have been enabled for the **WAN** interface, click on the **Edit Pencil Icon** under *Actions* for the WAN.



2. Click on the **WAN Categories** sub-menu, and you will see all of the *Rulesets*. Click on the **Select All** button to select all of the *Rulesets* (which then selects all of the rules).

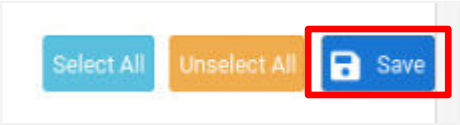


3. Scroll down, and you will see the **Rulesets for the ET Open Rules** and Snort **OPENAPPID Rules**.

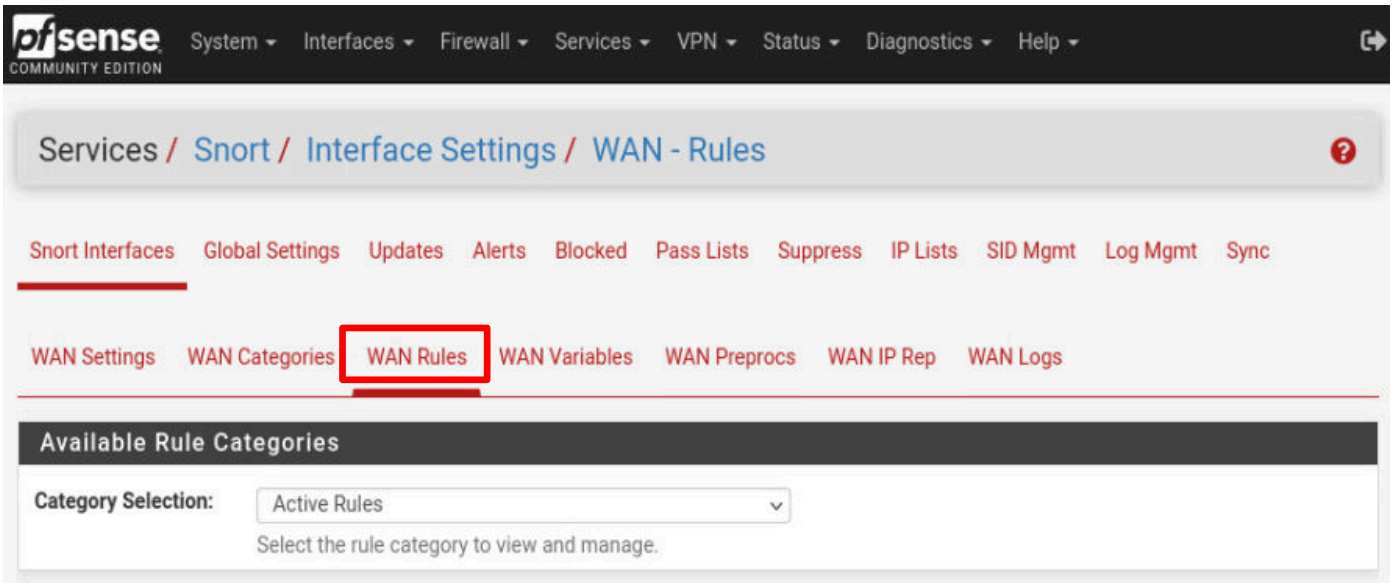


Snort Subscriber Rules have not been enabled, so they will not show up in the list

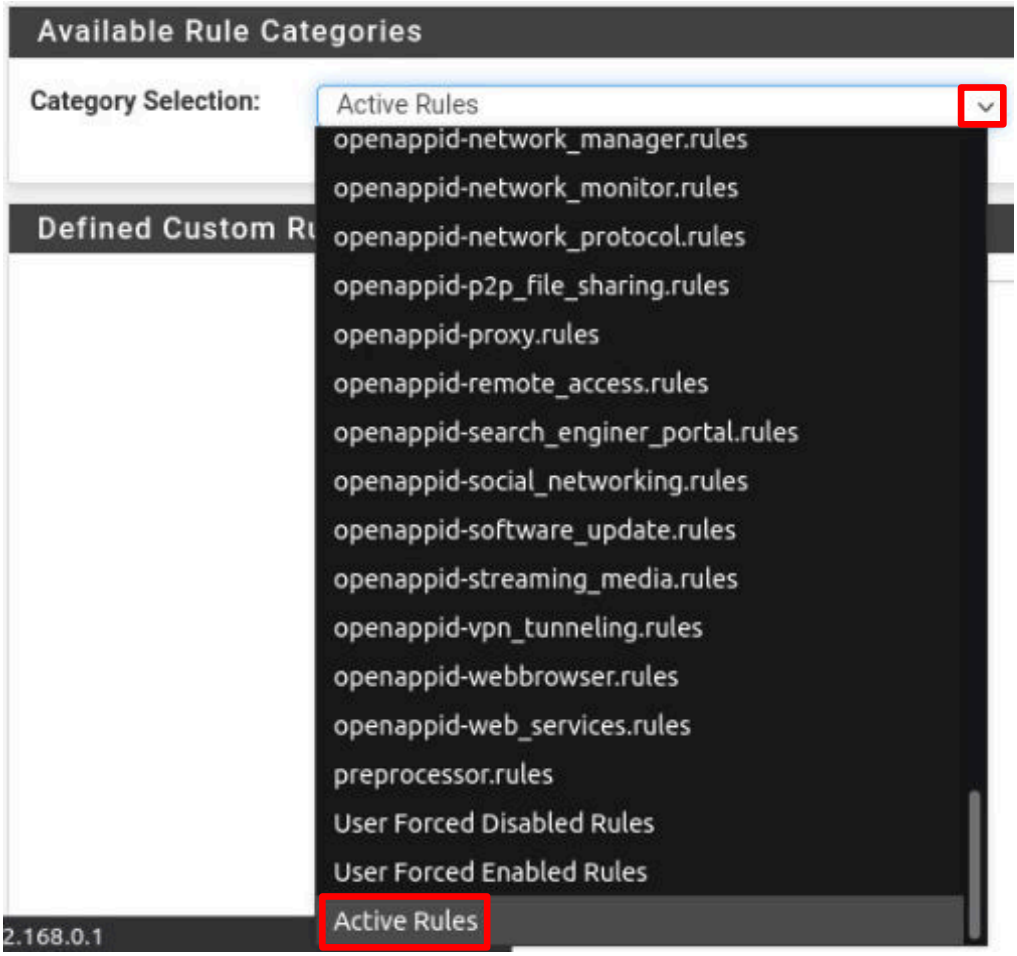
4. Click on the **Save** button at the top of the page.



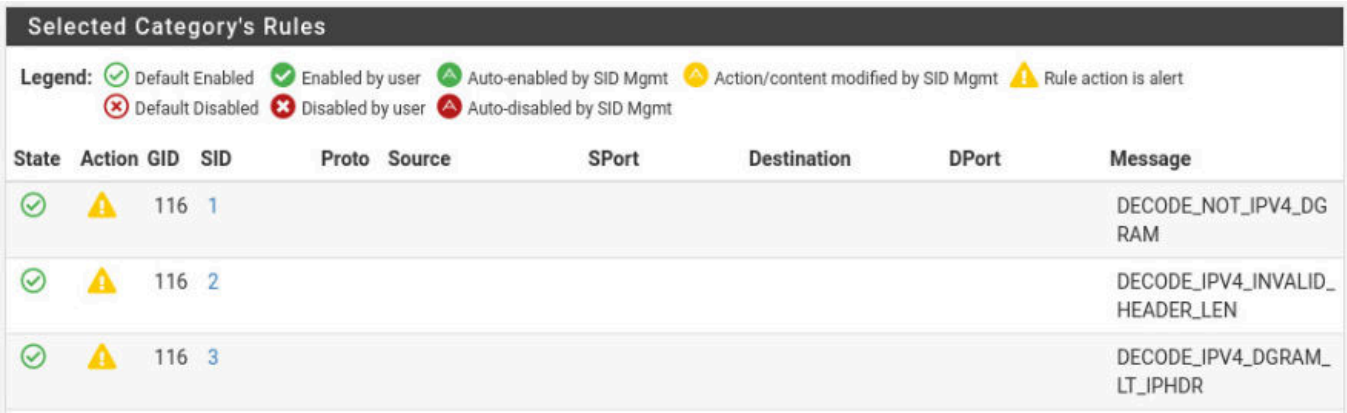
5. On the *Snort Interfaces* menu, click on the **WAN Rules** sub-menu.



6. Under the *Available Rules Categories*, there will be a set of *Category Selections* to choose from. The *Categories* are from the list of enabled **Rulesets** from **WAN Categories**. Click on the list arrow on the right side of the selection box, scroll down to the bottom of the list and select **Active Rules.**

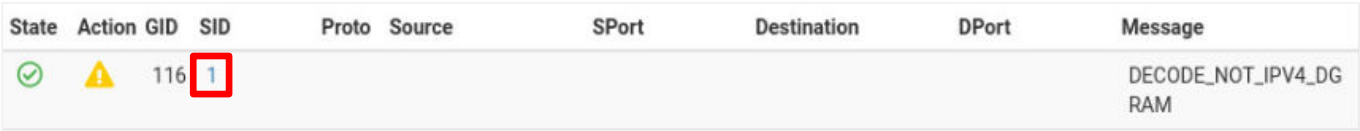


7. Scrolling down, under *Selected Category's Rules*, there will be a list of all of the rules that are active in all of the categories (there are currently over 28,000 rules, so the list will be very long), which is why the rules are broken down into Rulesets).



Make note of the **GID** (Generator ID) and **SID** (Snort Rule ID) columns. The GID identifies what part of *Snort* generates the event when a rule fires (*GID 1* is associated with the Snort Rules subsystem, and GIDs over 100 are for specific preprocessors and decoders). **SID** uniquely identifies Snort Rules. SIDs from 100-999,999 are rules that are included with *Snort*, <100 are reserved for future use and local, custom rules should use **SIDs > 1,000,000**).

8. To see a **Rule's Text**, click on the **SID** number for the first rule in the list.



9. This will open the *View Rules Text* window showing the code for the rule**.** Click the **Close** button to close the window.

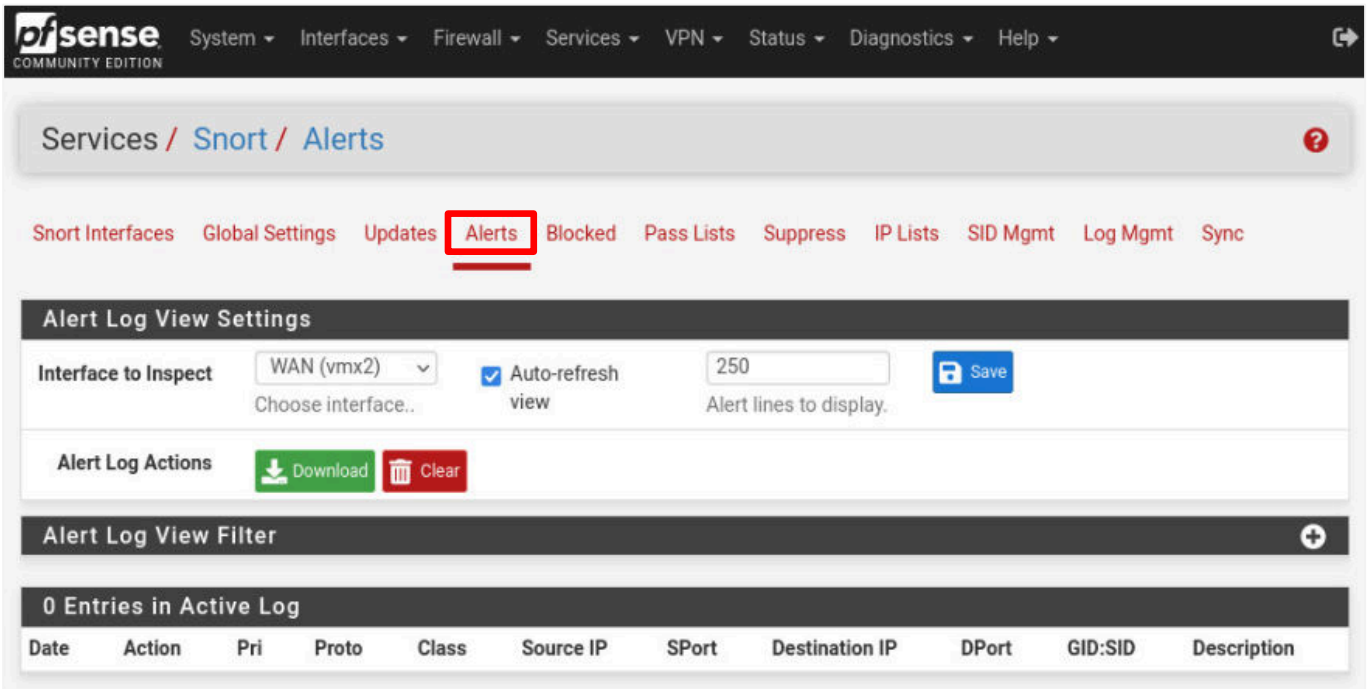
An examination of the Rule's Text will be discussed in a later section of the lab.

10. Remain on the *pfSense* interface and proceed to the next section to generate some malicious traffic.

2.3 Generate Malicious Traffic

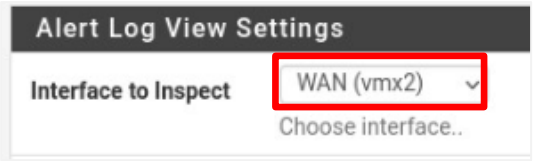
In *Lab 02: Web Application Scanning*, you discovered vulnerabilities on websites. One of the vulnerabilities that was uncovered was Cross-Site Scripting. Let's take a look at what an IPS/IDS does with this vulnerability.

1. Click on the **Alerts** menu option at the top of the page.

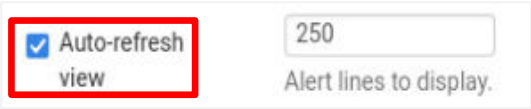


The *Alerts* page is where alerts will be displayed and can be used by a security analyst to discover traffic that has been able to get through the firewall and exploit the vulnerability on a website and develop a strategy to mitigate the problem.

2. Click on the list arrow in the *Interfaces to Inspect* selection and choose **WAN (vmx2)** if not already selected.

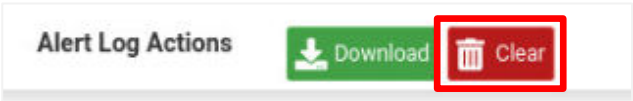


3. If not checked, click the **Auto-Refresh View** checkbox to automatically update the alert list.



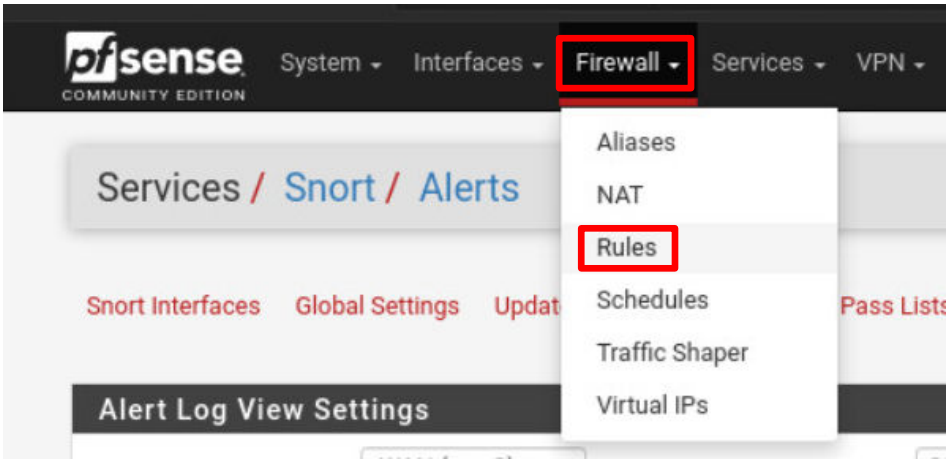
4. Click the **Save** button to save the alert log settings.

5. Click on the **Clear** button to delete all previous entries.

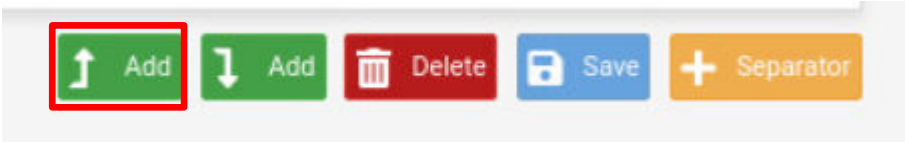


6. When asked, *Are you sure you wish to clear all interface log files?* click **OK**.

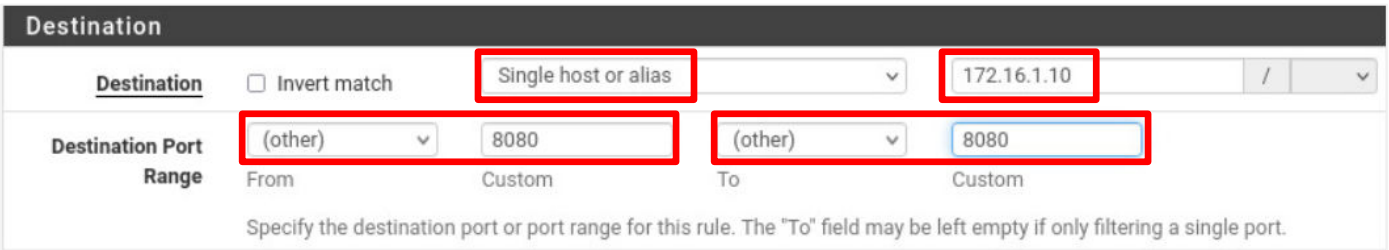
7. You will be spinning up the *Bodgeit* website on the *UbuntuSRV* computer on TCP Port 8080, which requires a firewall rule allowing Port 8080 traffic through. Click on **Firewall**, and then click on **Rules**.



8. At the bottom of the page, click on the **Add to Top** button:

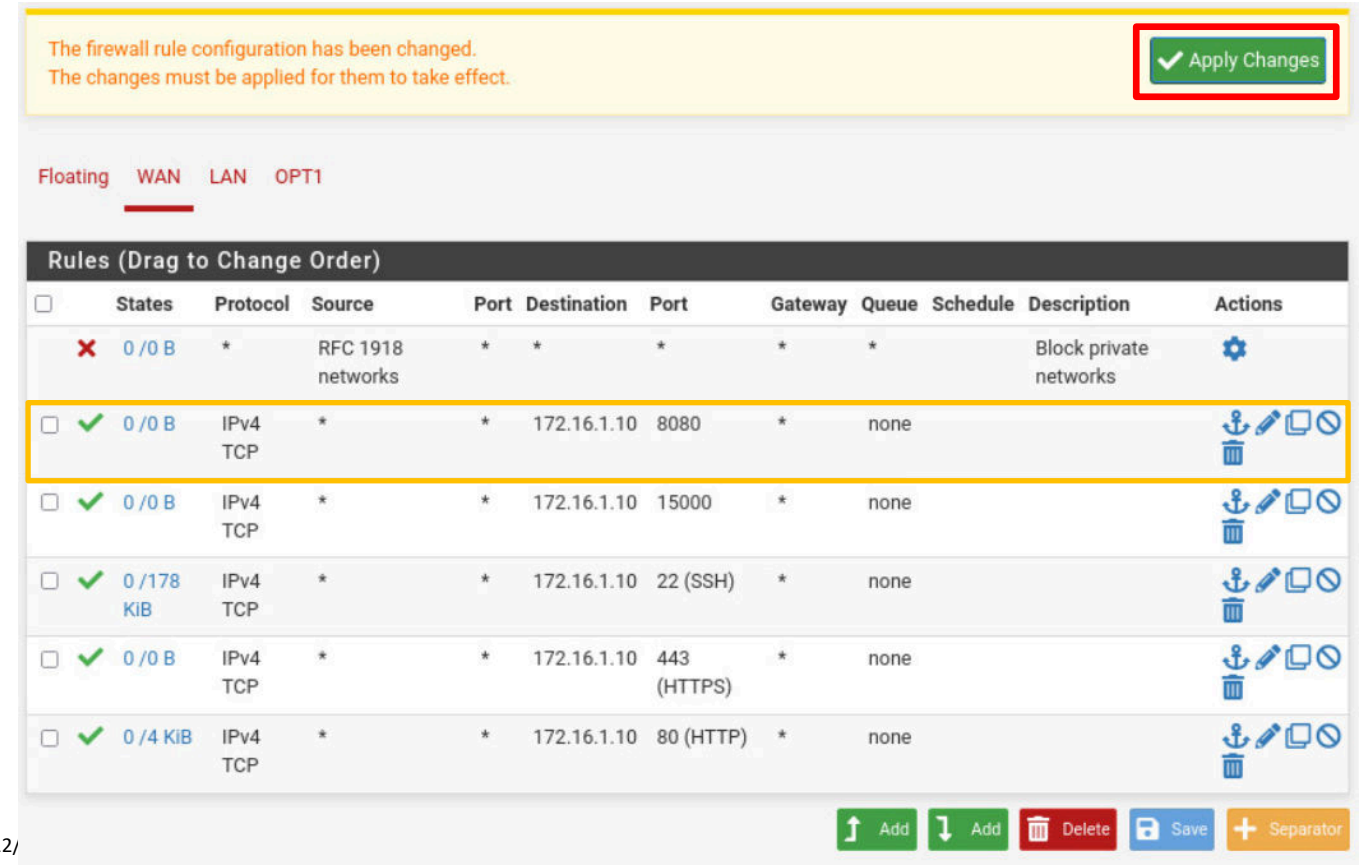


9. On the *Firewall/Rules/Edit* page, in the *Destination* section, click the list arrow and change the *Destination* from *any* to **Single host or alias**. Click on the **Destination Address** box and type `172.16.1.10`. For the *Destination Port Range*, leave the **From** list box **(other)** and type `8080` in the *Custom* box. In the *To* list box, leave it set to **(other)** and type `8080` in the *Custom* box.

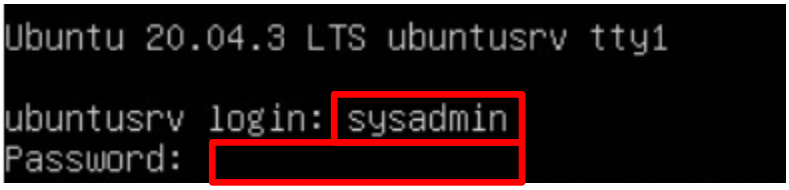


10. At the bottom of the page, click the **Save** button.

11. You should now see the new rule added to the top of the list. Click the **Apply Changes** button.

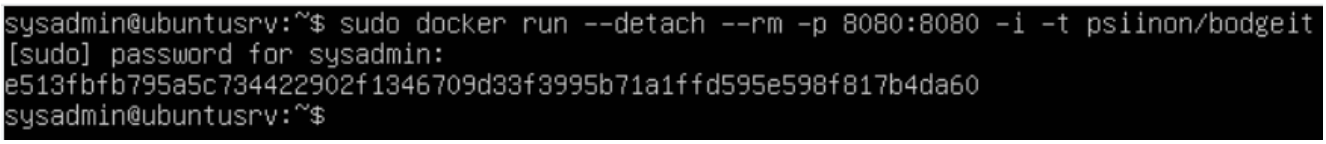


12. Set the focus to the **UbuntuSRV** computer.
13. Log in as `sysadmin` using the password: `NDGlabpass123!`

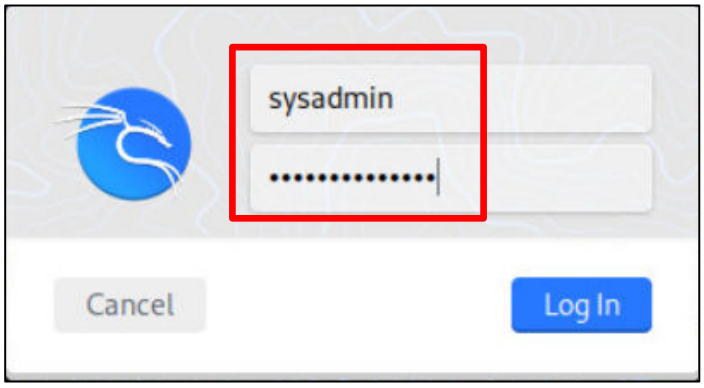


14. Start the *Bodgeit* website inside a *Docker* container. A Docker container is a form of virtualization that utilizes the OS in order to allow software to run inside of an isolated, virtual instance in any Linux environment. In order to start the *Bodgeit* docker container, type the following command, using the password `NDGlabpass123!` when prompted:

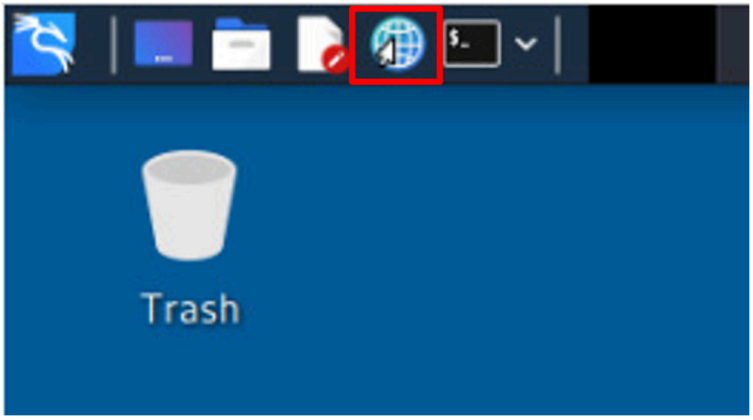
```
sudo docker run --detach --rm -p 8080:8080 -i -t psiinon/bodgeit
```



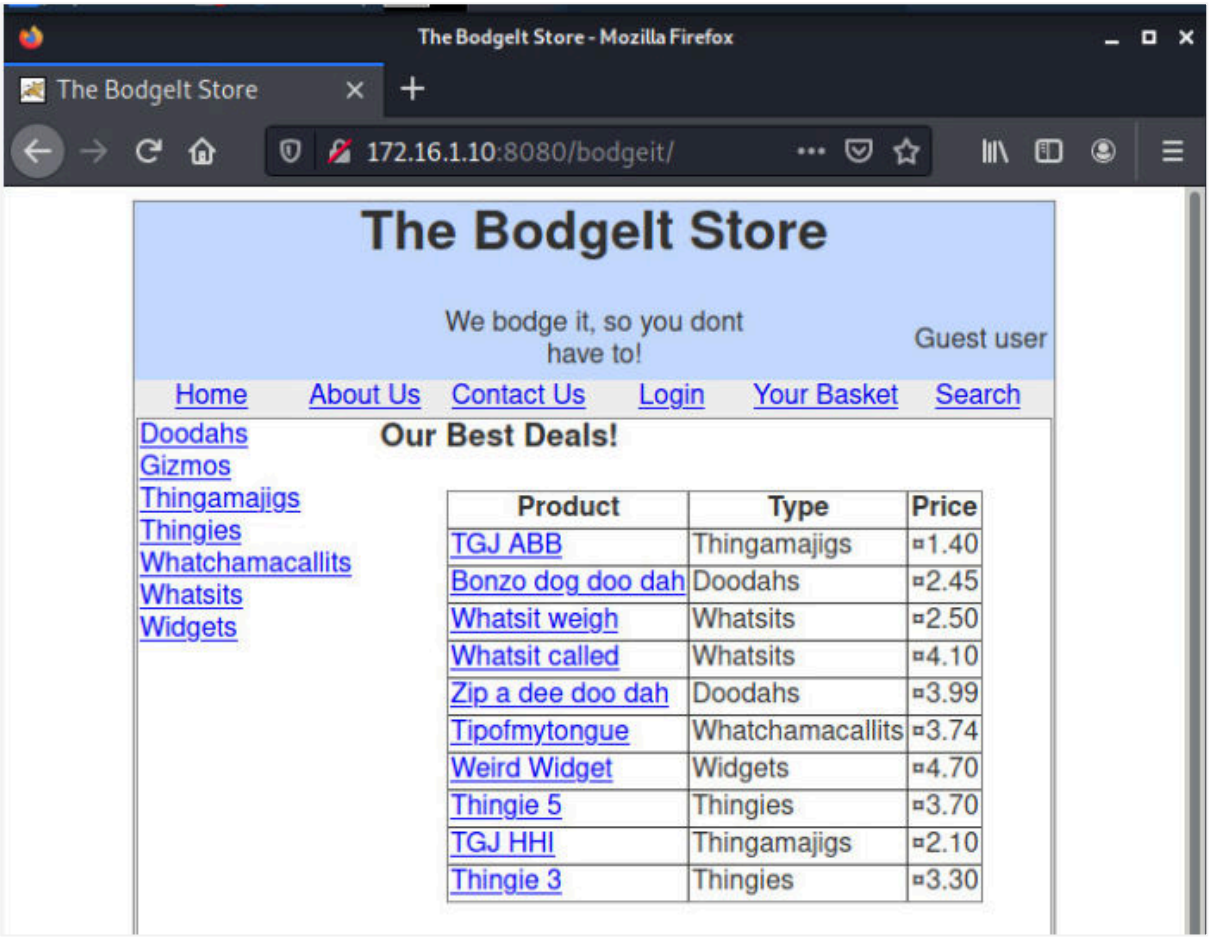
15. Set focus on the **Kali** computer.
16. Log in as `sysadmin` using the password: `NDGlabpass123!`



17. Open the **Web Browser** application on the taskbar.

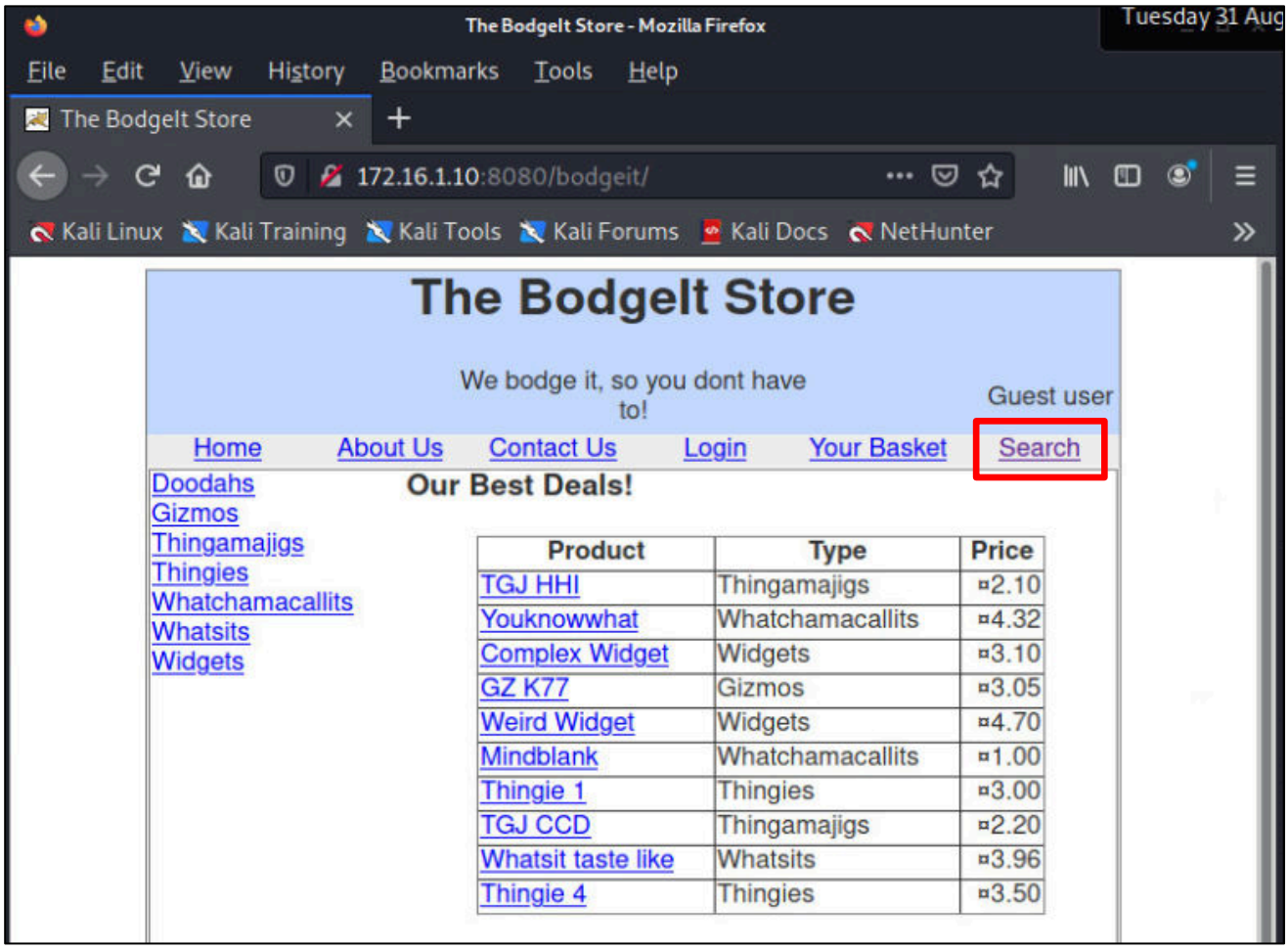


18. Type the address `http://172.16.1.10:8080/bodgeit/`. Confirm that the website has successfully loaded.



Let's see if we can exploit the problem by inserting a simple popup alert into the search function.

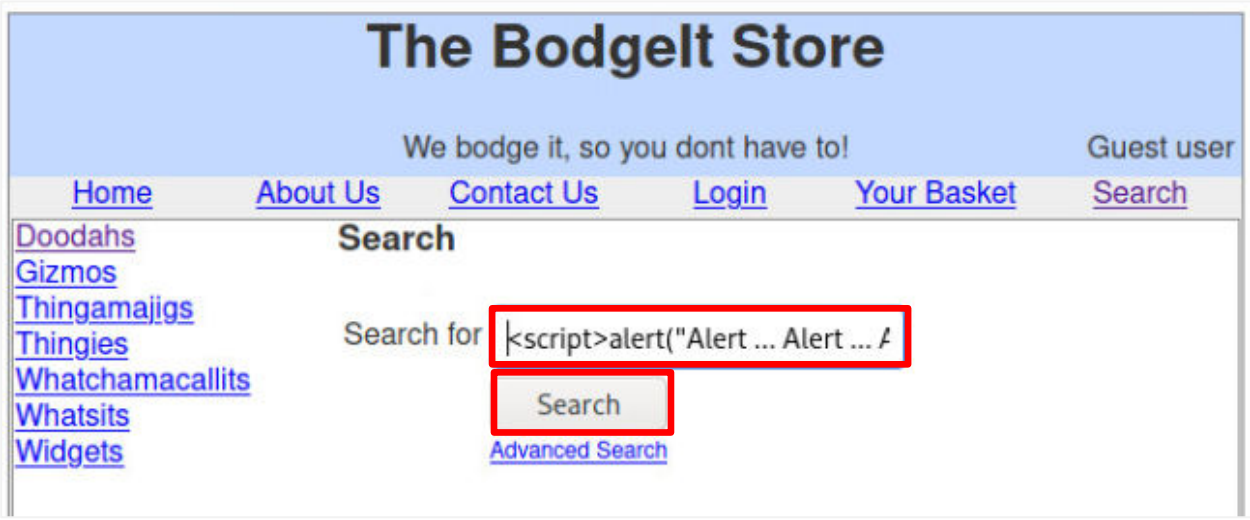
19. Click on the **Search** menu option.



20. To inject a **Cross-Site Script** that will exploit the vulnerability, type the following into the *Search for* entry field:

```
<script>alert("Alert … Alert … Alert")</script>
```

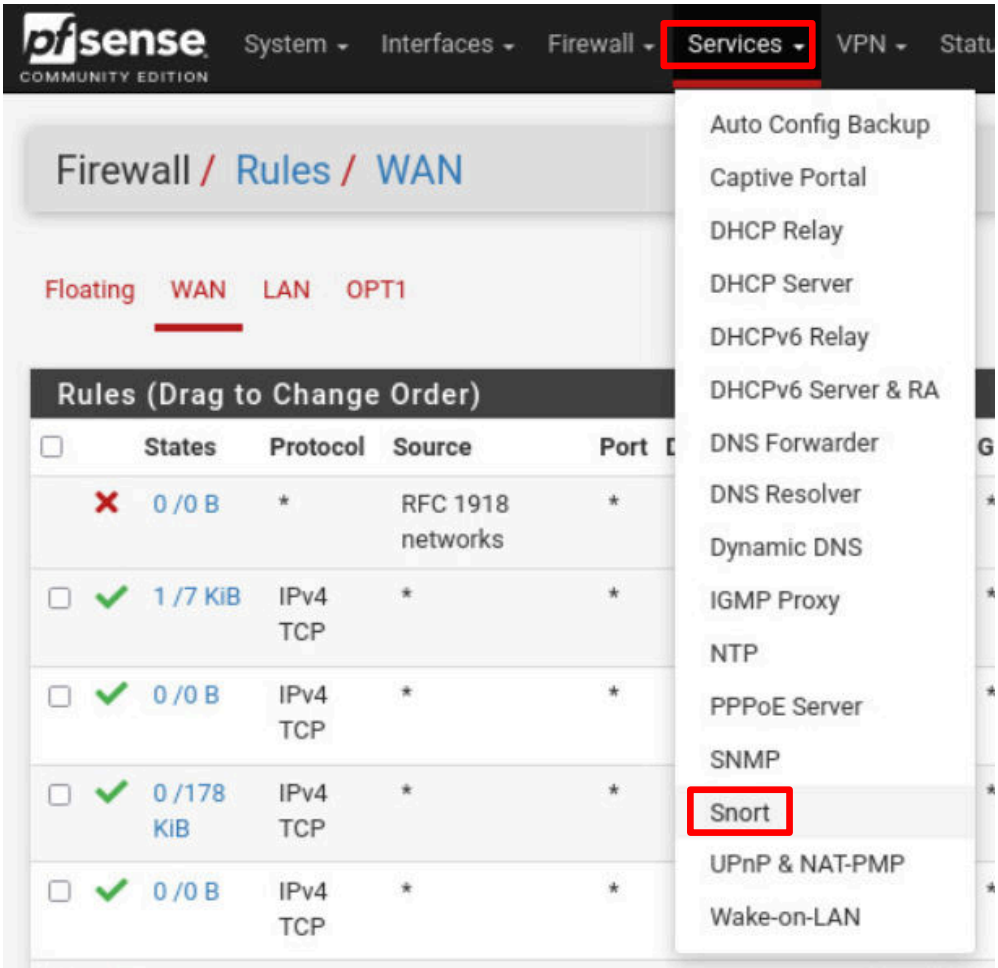
Click the **Search** button below the *Search For* box to execute the **Cross-Site Script,** which will pop up the alert message box.



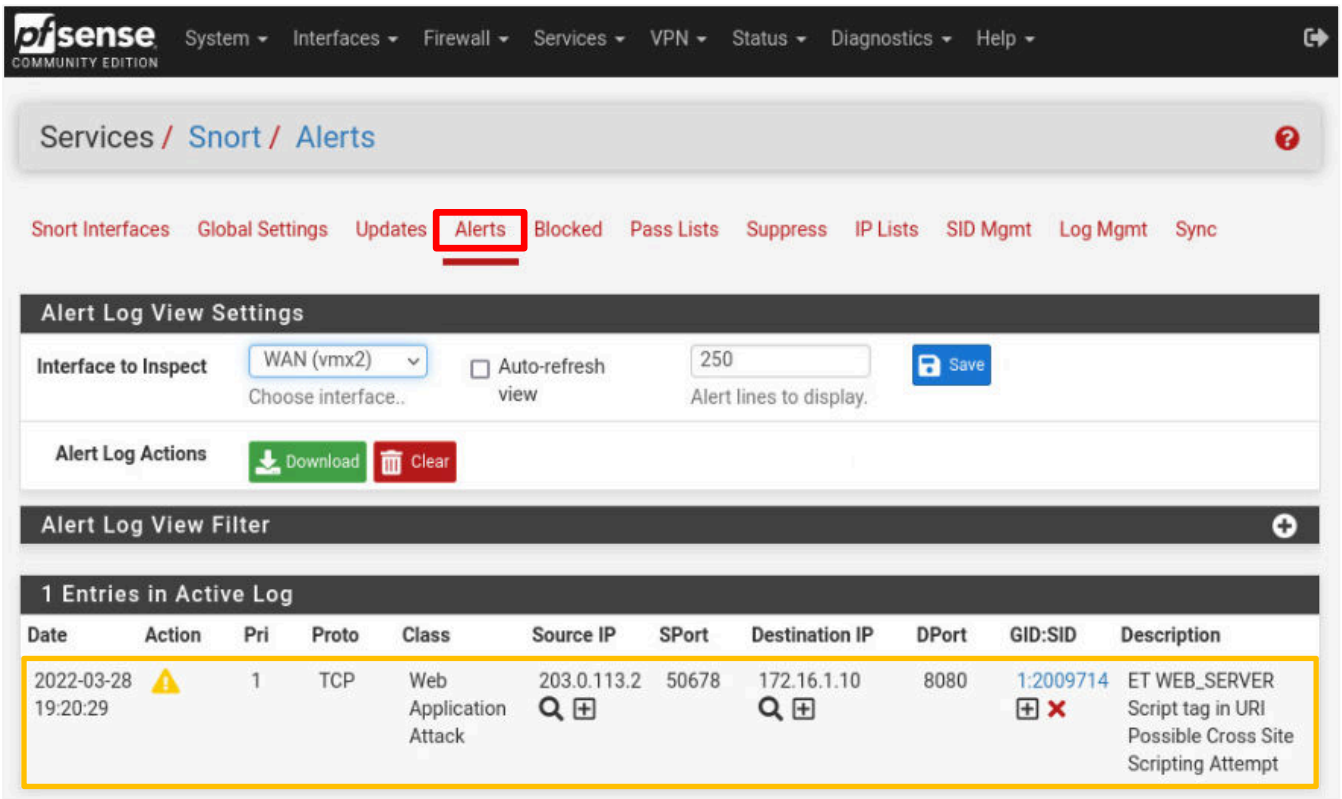
21. This time the response to the search is the **Cross-Site Script** popup box.

22. Click the **OK** button to close the *Alert* popup.
23. Close the *Firefox* web browser.

24. Return focus to the **MintOS** computer. On the *pfSense* page, click on **Services** and then click on **Snort**.



25. Click on the **Alerts** menu option, and you will see the alert indicating that a *Cross-Site Script* was attempted to the *Bodgeit* web server.



Make a note of the **GID:SID** of the Cross-Site Script alert. You will use it in the next section.

26. Remain on the *pfSense* interface and continue to the next section.

2.4 Examine Snort Rules

A complete discussion of Snort rules are beyond the scope of this lab. There are many excellent guides and tutorials available online. Here are just a couple that you may wish to consider:

Snort User's Manual
http://manual-snort-org.s3-website-us-east-1.amazonaws.com/snort_manual.html

Writing Snort Rules / Snort Cheat Sheet and Examples
https://cyvatar.ai/write-configure-snort-rules/

From the **Snort Rule Infographic***, here's the breakdown of **Snort Rules**:

BASIC OUTLINE OF A SNORT RULE

```
[action][protocol][sourceIP][sourceport] -> [destIP][destport] ( [Rule options] )
└──────────────────────────────────────────────────────────────┘
                          Rule Header
```

** https://www.snort.org/documents/snort-rule-infographic*

The details of the **Rule Header** are:

[**action**] The action to be taken when Snort matches a packet with the rule criteria
- Alert: generate an alert and then log the packet
- Log: log the packet
- Pass: ignore the packet
- Activate: alert and then turn on another dynamic rule
- Dynamic: remain idle until activated by an activate rule, then act as a log rule

[**protocol**] The type of traffic by protocol. The four protocols that Snort uses to detect suspicious behavior are TCP, UDP, ICMP, and IP

[**sourceIP**] The source address of the packet. It can be an IP address/CIDR or a variable defined in snort.conf. The *any* indicator will look at all source IPs. Typically it will use the variable $EXTERNAL_NET, which is defined in the snort.conf file.

[**sourceport**] The source port of the packet. It can be a port number or a variable defined in snort.conf. The *any* indicator will look at all ports.

[**->**] The direction of the packet, from Source to Destination

[**destIP**] The destination address of the packet. It can be an IP Address/CIDR or a variable defined in snort.conf. The *any* indicator will look at all destination IPs. Typically it will use the variable $HOME_NET, which is defined in the snort.conf file.

[**destport**] The destination port of the packet. It can be a port number or a variable defined in snort.conf. The *any* indicator will look at all ports.

Rule Options form the heart of Snort's intrusion detection engine, combining ease of use with power and flexibility. All Snort rule options are separated from each other using a semicolon (**;**).Rule option keywords are separated from their arguments with a colon (**:**).

General Rule Options:

Message A meaningful message typically includes what the rule is detecting. The msg rule option tells Snort what to output when the rule matches. It is a simple text string.

Flow For the rule to fire, specifies which direction the network traffic is going. The flow keyword is used in conjunction with TCP stream reassembly. It allows rules to only apply to certain directions of the traffic flow.

Reference The reference keyword allows rules to include references to external sources of information.

Classtype The classtype keyword is how Snort shares what the effect of a successful attack would be.

sid and rev The snort id is a unique identifier for each rule. This information allows output plugins to identify rules easily and should be used with the rev (revision) keyword.

Payload Detection Options:

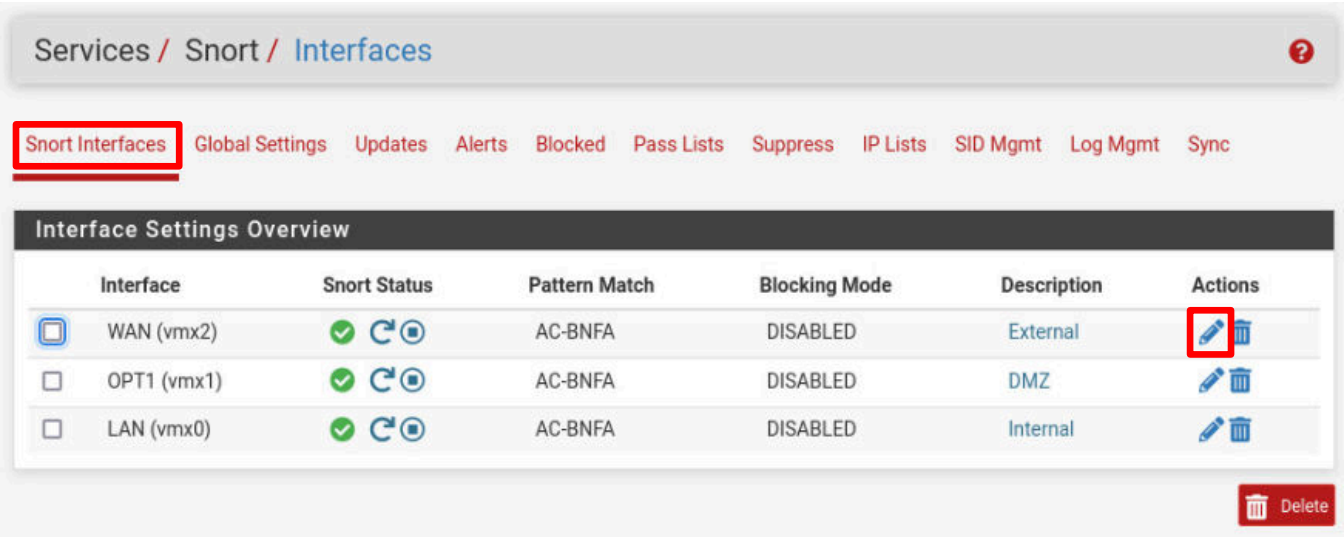
There are many Payload Detection Options. To see a complete detailed list with examples, refer to the Snort User's Manual, Section 3.5, Detection Rule Options, Content at:
http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node32.html

The most important option is Content:

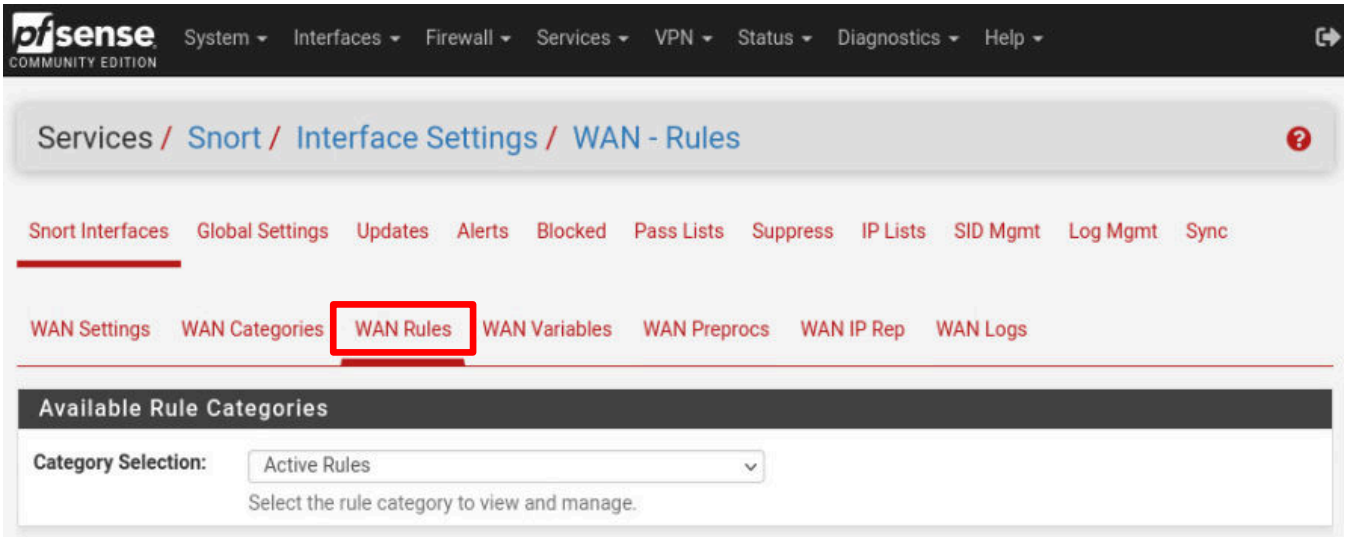
Content This important feature allows the user to set rules that search for specific content in the packet payload and trigger a response based on that data. The option data can contain mixed text and binary data.

There are seventeen different *Content Modifiers*. To see a complete detailed list with examples, refer to the Snort User's Manual, Section 3.5, Detection Rule Options, Content at:
http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node32.html#SECTION00451000000000000000

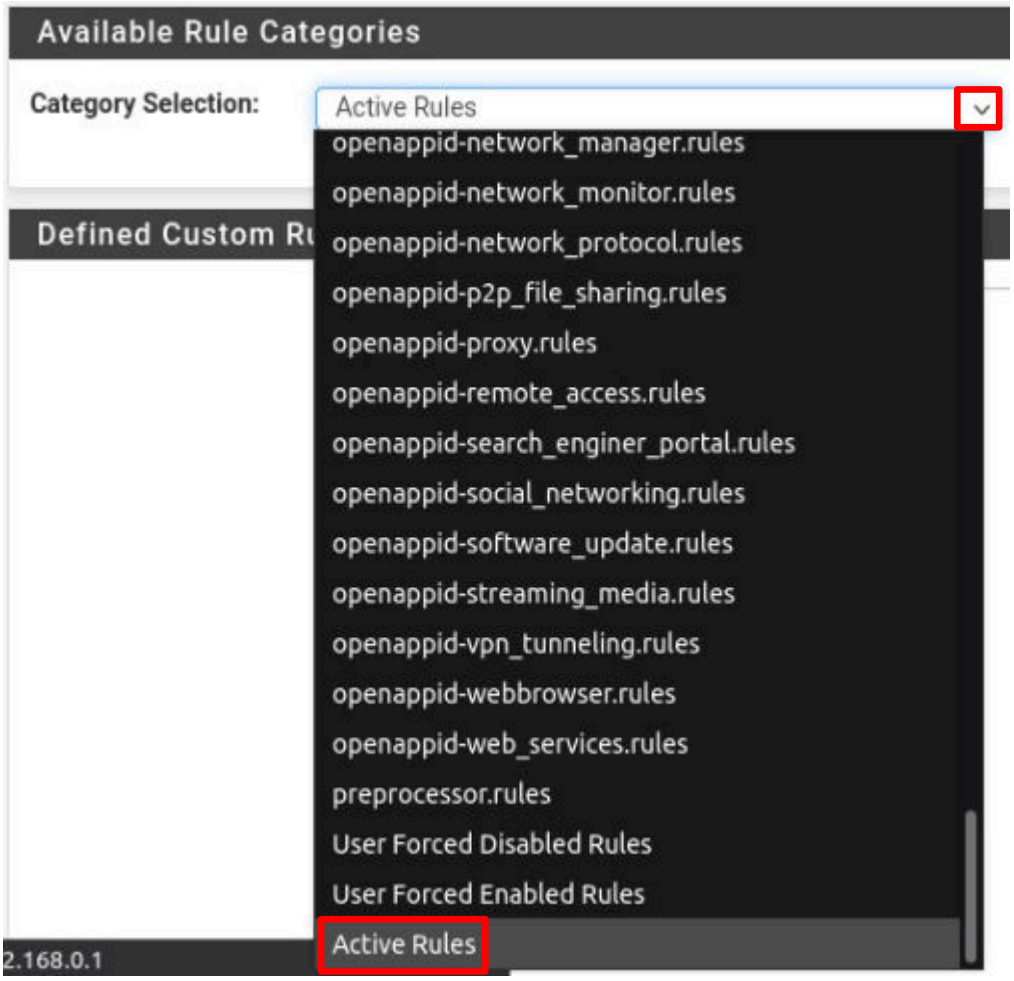
1. To view the details about the rule, click on the **Snort Interfaces** menu item, and then click on the **Edit Pencil** icon for the *WAN (vmx2)* interface.



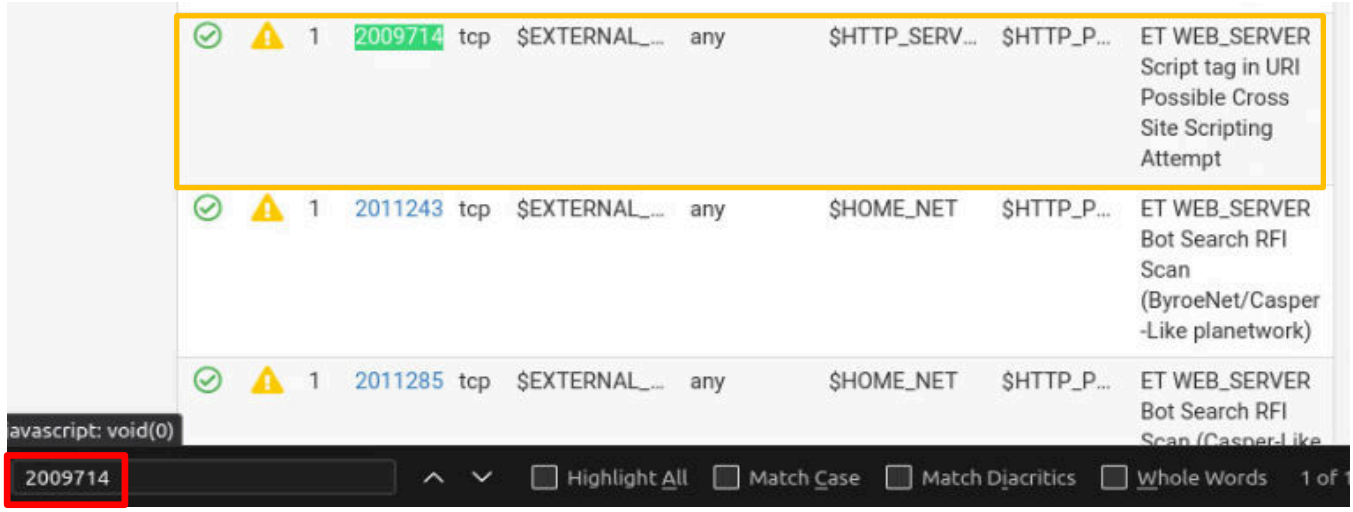
2. On the *Snort Interfaces* menu, click on the **WAN Rules** submenu.



3. Under the *Available Rules Categories*, there will be a set of *Category Selections* to choose from. The categories are from the list of enabled **Rulesets** from **WAN Categories**. Click on the list arrow on the right side of the selection box, scroll down to the bottom of the list and select **Active Rules.**

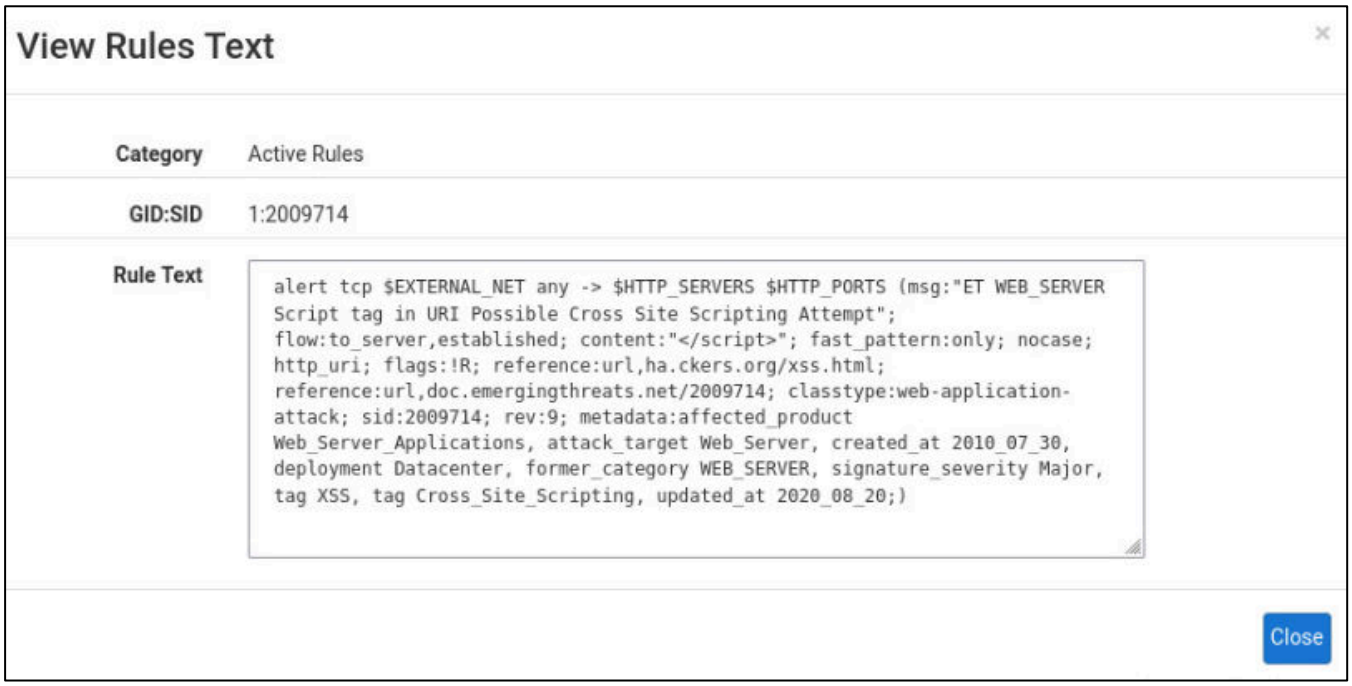


You could scroll down until you get to the **Rule's SID** (**2009714**), but it's easier to use **Firefox's Find on Page** function. Press **Ctrl+F,** which will open the *Find in Page* popup. Type `2009714` into the field, and the rule will be listed.



4. Click on the **SID** number for the rule.

This will open the rule.



Looking at the *Rule Text*, you can see that the GID is 1, which indicates the rule is part of the Snort rules subsystem, and the SID is assigned to 2009714.

Looking at the text of the rule, here's the Rule Header

[action]	**alert**
[protocol]	**tcp**
[sourceIP]	**$EXTERNAL_NET** (the variable containing all networks that ARE NOT on the internal IP network)
[sourceport]	**any**
[->]	**from $EXTERNAL_NET**
[destIP]	**$HTTP_SERVERS** (the variable containing the IP addresses of Web Servers)
[destport]	**$HTTP_PORTS** (the variable containing the TCP Ports of Web Servers)

… and here are the **Rule Options**:

General Rule Options:

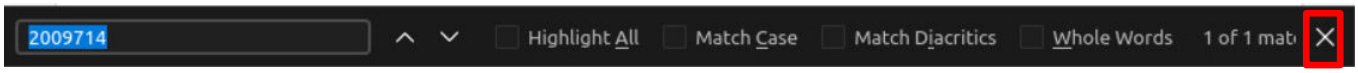
Message	**msg: "ET WEB_SERVER Script tag in URI Possible Cross Site Scripting Attempt"**
Flow	**flow: to_server, established** (established TCP connection)
Reference	**reference: url, ha.ckers.org/xss.html** **reference: url, doc.emergingthreats.net/2009714**
Class type	**classtype: web-application-attack**
SID	**sid: 2009714**
REV	**rev: 9**
Metadata	**metadata: affected_product Web_Server_Applications, attack_target Web_Server created at 2010_07_30, deployment Datacenter, former_category WEB_SERVER, signature severity Major, tag XSS, tag Cross_Site_Scripting, updated_at 2020_08_20**

Payload Detection Options:

Content	**"</script>"**	
	fast_pattern: only	
	nocase	<< Snort should look for the specific pattern, ignoring case >>
	http_uri	<< Restricts search to the NORMALIZED request URI field >>
	flags !R	<< Tests TCP flags for NOT RST >>

5. Click the **Close** button in the lower-right corner of the window to close the **View Rules Text** window.

6. At the bottom of the *Firefox* window, close the **Find** function bar by clicking the **X** at the right side of the window:



7. Remain on the *Snort* interface and continue to the next section.

2.5 Writing Custom Snort Rules

There will be times when the existing base of **Rules** does not meet the needs of the organization or when a **Rule** needs to be modified to accommodate unique conditions. When that time arrives, the Security Analyst will need to write **Custom Snort Rules**.

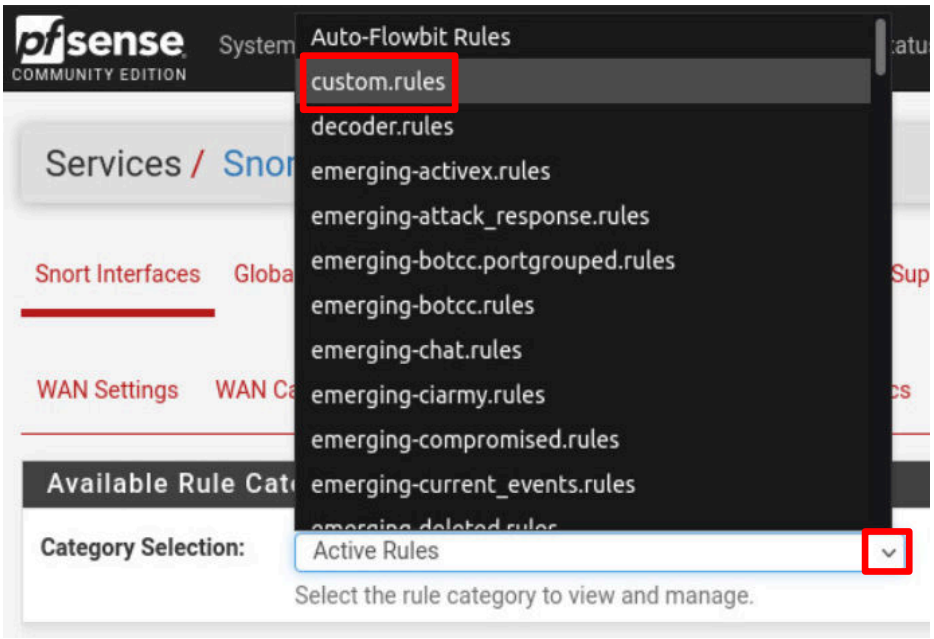
In the previous sections, **Snort Rules** were explained, and their structure dissected. In this section, you will create a custom rule that will detect **TCP SYN Flood DoS Attack**.

A SYN Flood is when a bad actor exploits the TCP 3-Way Handshake to execute a Denial of Service attack.

In a normal TCP 3-Way Handshake, a client sends a SYN packet to the server, which returns an SYN-ACK packet back to the client's IP address. When the client receives the SYN-ACK, it replies with an ACK and the connection is established.

A SYN Flood is triggered when the bad actor sends a huge amount of SYN packets to the server, each one using a random, non-existing client IP address. The server will then attempt to send its SYN-ACK packets to the fake hosts awaiting ACK that will never arrive, leaving the server's ports open. As the number of unacknowledged packets accumulates, CPU and memory usage will increase to the point where the server will hang or crash.

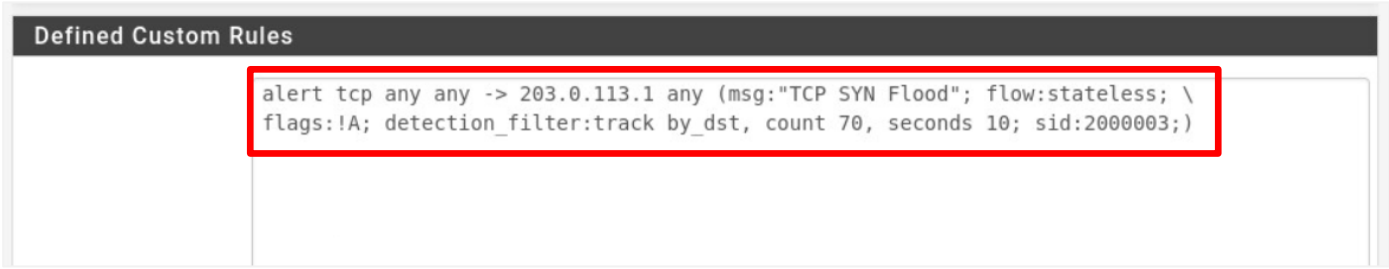
1. Scroll back to the top of the page and in the **Available Rule Category** section, under **Category Selection**, click on the list arrow and select **custom.rules** from the list.



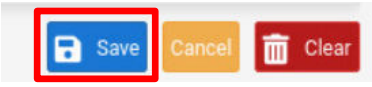
2. In the *Defined Custom Rules* section, type the following **Rule**:

```
alert tcp any any -> 203.0.113.1 any (msg: TCP SYN flood”; flow:stateless; \
flags:!A; detection_filter:track by_dst, count 70, seconds 10; sid:2000003;)
```

The backslash (****) is used to continue the rule to the next line.



3. Click the **Save** button at the bottom-right of the page.



Let’s take a look at the Rule:

<u>Rule Header:</u>

[action]	**alert**
[protocol]	**tcp**
[sourceIP]	**any** (any IP address)
[sourceport]	**any** (any Port number)
[->]	
[destIP]	**203.0.113.1** (the IP address of the WAN interface on the *pfSense* firewall)
[destport]	**any** (any Port)

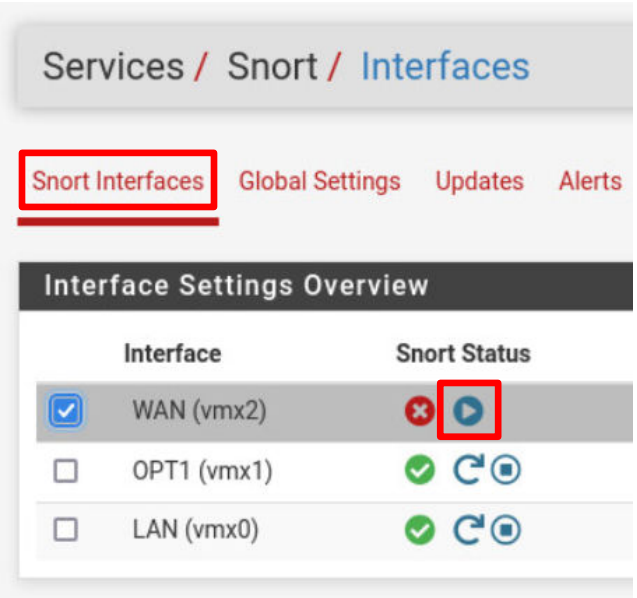
<u>Rule Options:</u>

Message	**msg: ”TCP Syn Flood”**	
Flow	**flow: stateless** << trigger regardless of the state of the stream >>	
SID	**sid: 2000003**	
Flag	**flags !A**	<< tests TCP flags for NOT ACK >>

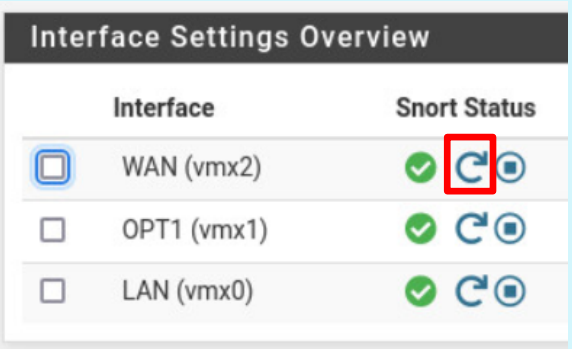
<u>Post Detection Rule Options:</u>

detection_filter	<< defines a minimum traffic rate before generating an alert >>
track by_dst	<< traffic rate tracked by destination address >>
count 70	<< maximum number of rule matches in “s” seconds before detection filter limit is exceeded >>
seconds 10	<< time in which count is accrued >>

4. Click on the *Snort Interfaces* menu item. Then click on the **Start icon** under *Snort Status* to enable the interface.

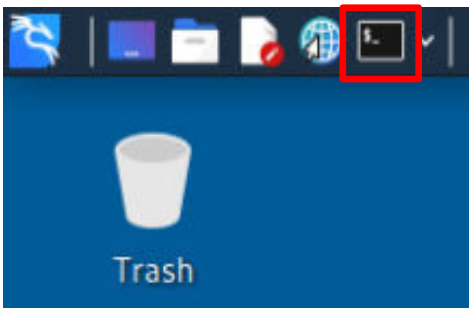


If the interface is already enabled, click the **Restart** icon.



To test the Rule:

5. Return to the **Kali** computer.
6. Click the **Terminal** icon to open a terminal window.



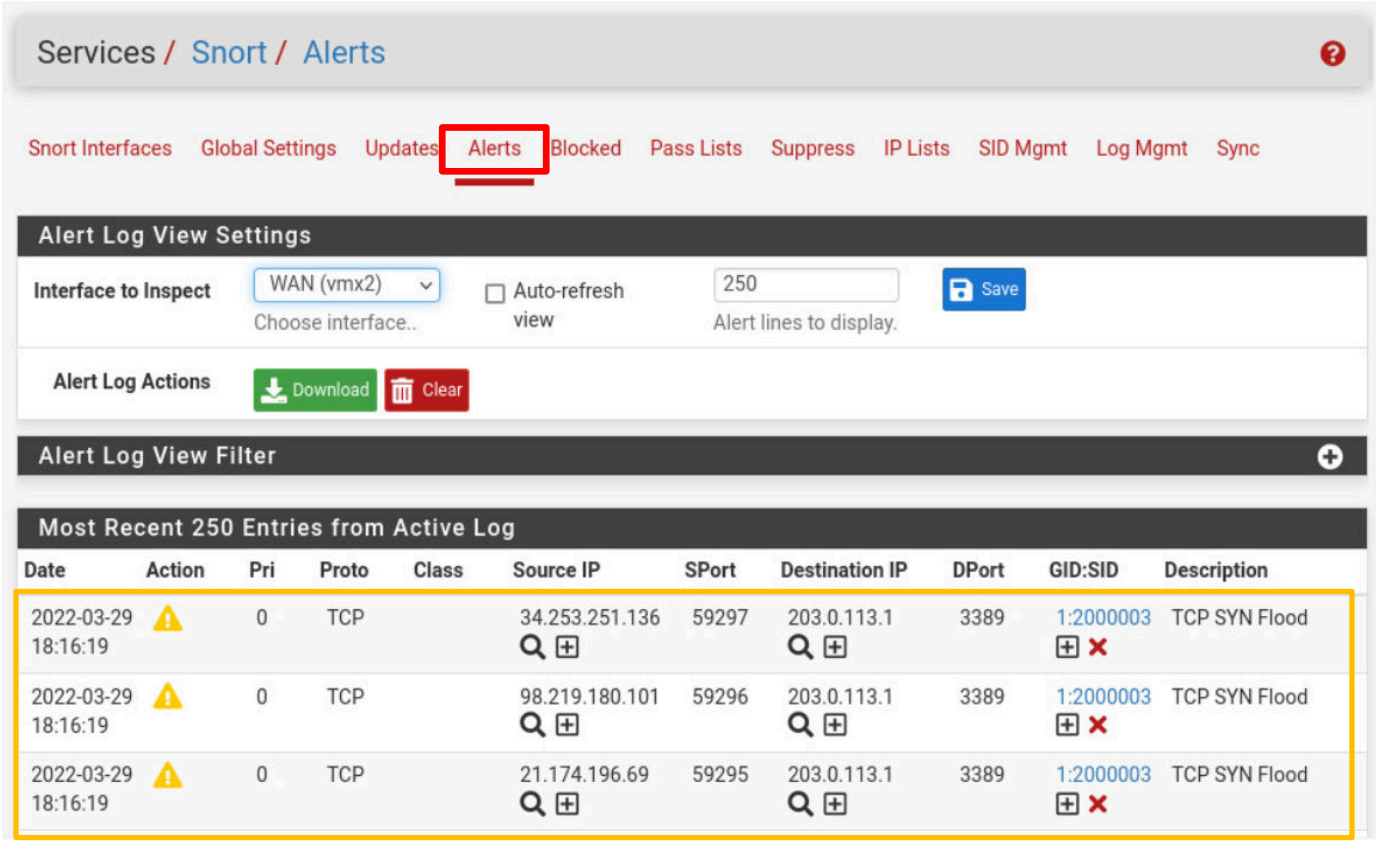
7. Type the following command to execute the **SYN Flood**

```
sudo hping3 -c 15000 -d 120 -S -w 64 -p 3389 --flood --rand-source 203.0.113.1
```

If asked for the **[sudo] password for sysadmin,** type: `NDGlabpass123!`

```
┌──(sysadmin㉿kali)-[~]
└─$ sudo hping3 -c 1000 -d 120 -S -w 64 -p 3389 --flood --rand-source 203.0.113.1
[sudo] password for sysadmin:
HPING 203.0.113.1 (eth0 203.0.113.1): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
```

8. Return to the **MintOS** computer.
9. Click on **Alerts** on the *Snort* page.



You will see the rule, **SID:20000003** you created has detected all of the **SYNs** in the flood. Notice that for each **SYN**, there is a different **Source IP** and **SPort**.

10. Return to the **Kali** computer and in the terminal window, type **Ctrl+C** to stop the *hping* flood.
11. The lab is now complete; you may now end the reservation.