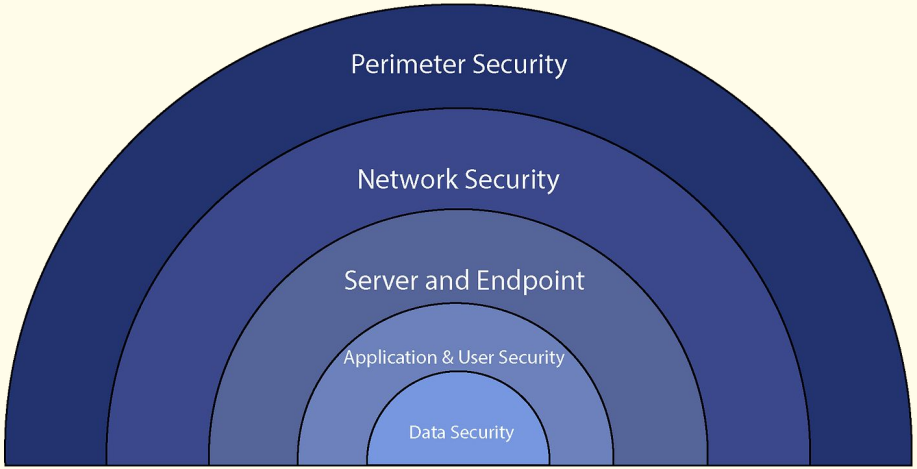
Intrusion Detection & Prevention

Trey Atwood

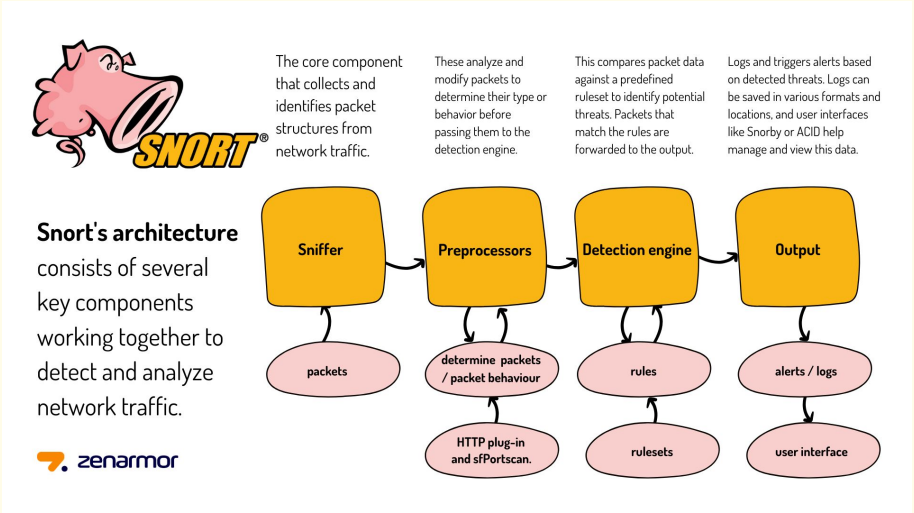
Concept of defense in depth

The concept of defense in depth is a Cybersecurity strategy that uses multiple layers of security controls to protect systems and data within them. Doing so creates redundancy so if one of the layers fail, others are in place to stop an attack. An example of this is having a firewall, an IDS or IPS, an antivirus software, and encryption all on the network.



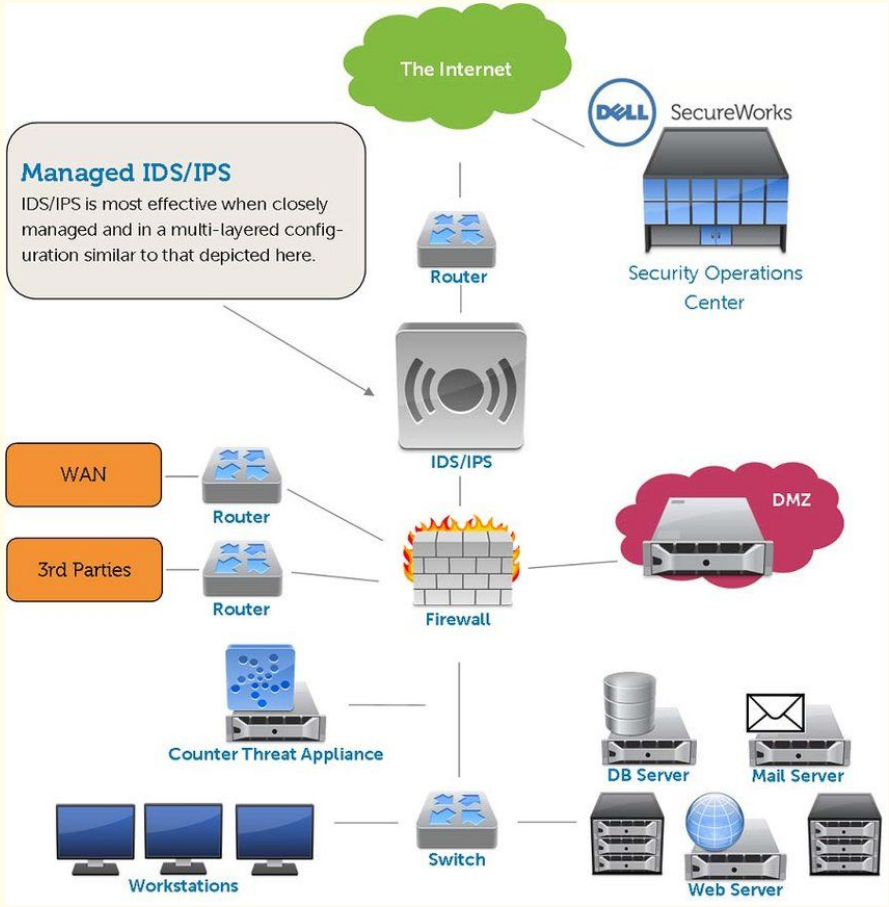
How an IDS/IPS adds to or diminishes the defense of a network

An IDS/IPS enhances the network defense by detecting or preventing malicious activity. An IDS (intrusion detection system) adds value by providing alerts without slowing down the network traffic, while the IPS (intrusion prevention system) adds value by blocking the threats as they transpire. There is a drawback of using IDS/IPS in the potential for false positives, where legitimate network traffic can be blocked and interrupt business operations when it shouldn't be.



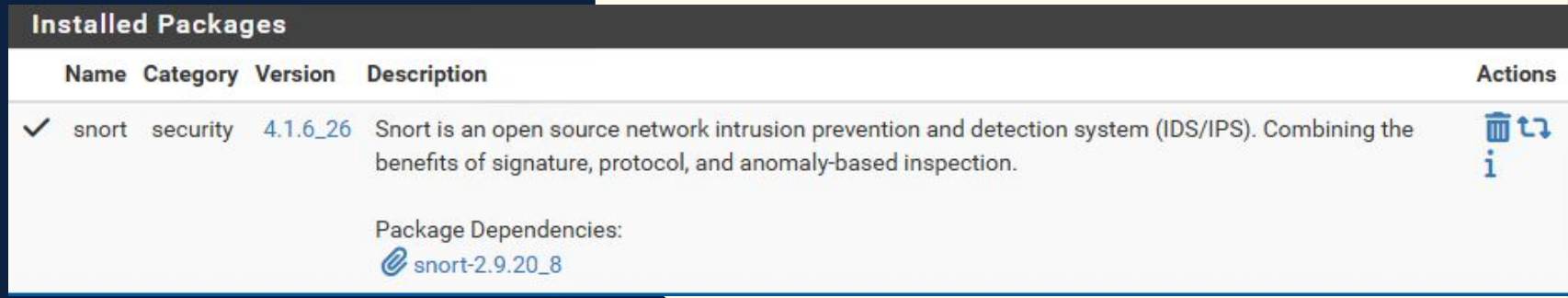
What is the role of the proxy server on a network protected by an IDS/IPS

A proxy server on a network acts as a central point for traffic inspection, security control, and anonymity. It intercepts user requests, uses its own IP address to communicate with the internet and can enforce access controls as to what users can access and see.



Today we are working with Snort, to practice intrusion detection and prevention, to see what both sides of it look like, as the attacker and as the person getting attacked.

To start we first need to install Snort on to our PFSense.

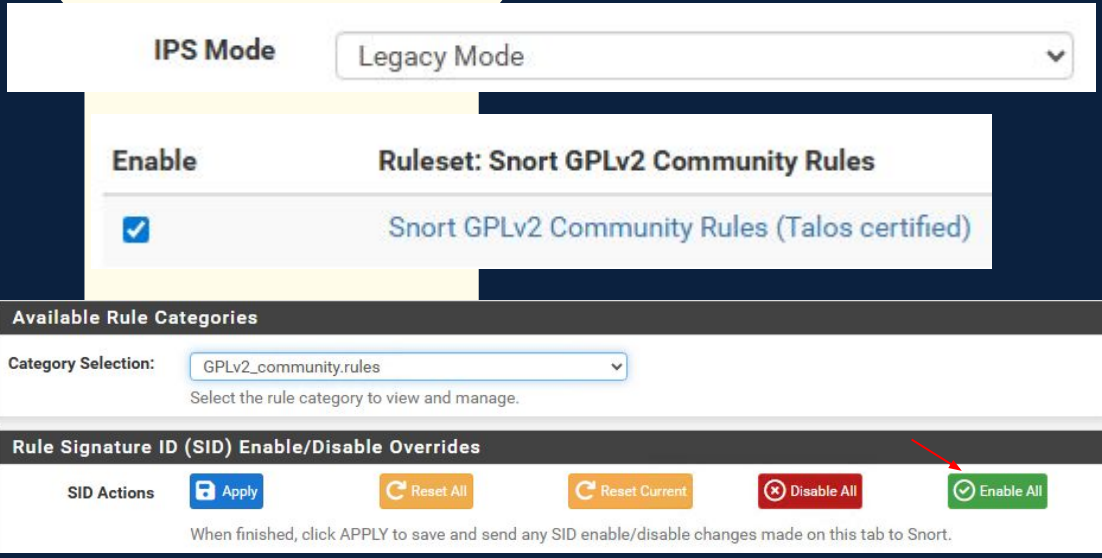


Now that Snort is installed we will need to change a couple of settings around.

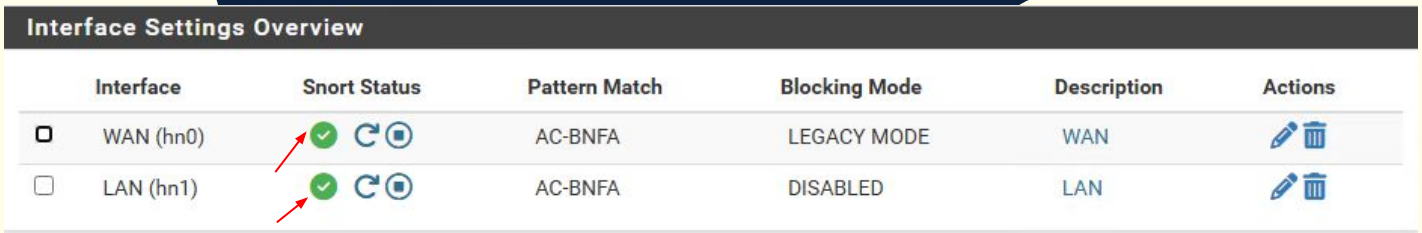
After downloading Snort and changing some settings around it's time to set up rules for our LAN and WAN networks.

Starting with the WAN interface, we keep it on Legacy Mode since we're running this on an Azure Machine, make sure to enable the GPLv2 community rules, and Enable all the rules in the set.

Next when configuring the LAN network, we keep all the same settings that are defaulted. The only things that we change is the same we did for the WAN, just enabling all the GPLv2 rules.

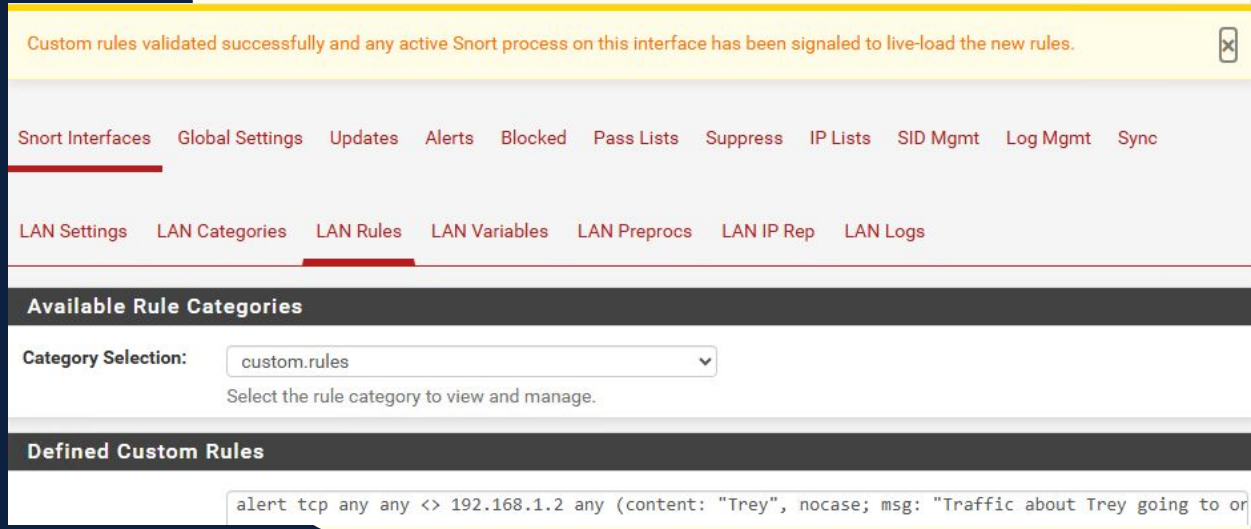


Once you configure both LAN and WANs, we will enable both in the Snort interfaces tab and we are working!!



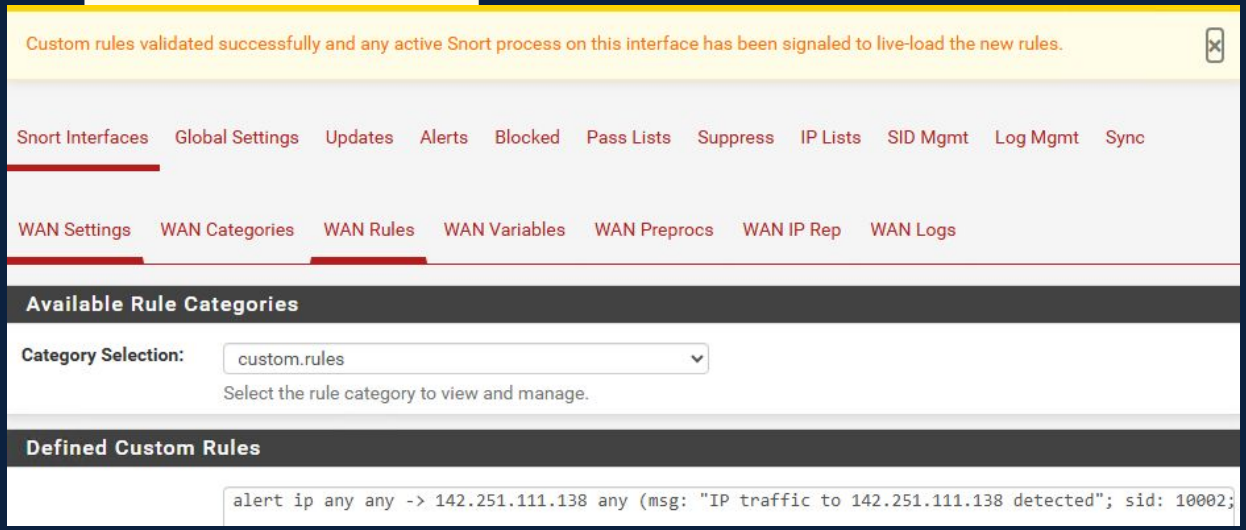
We will now configure a couple custom rules to our LAN network, to do so edit the LAN rules and select “custom rules”

Breaking down the custom rule the “alert” is the command word, which in this case is just alerting us if something comes in, next is the protocol for this we’re using “tcp”, the next two “anys” are the source IP and the source port. Meaning traffic coming from any source IP and source port is flagged, next “<>” means traffic coming or going to the ip “192.168.1.2” which is our web server. The last “any” just means any traffic going into our web server. Inside the “content” when Snort is inspecting packets it’ll flag if there’s any mentions of my name “Trey”, then the “msg” is the message that will be in the alert.



When accessing my web server from an external user, we see an alert with the description saying “Traffic about Trey …” so our rule works!

This time we are working with the WAN side, setting up another custom rule but this time I make the rule alert us if there is traffic to the IP address “142.251.111.138” which is the IP for google.com



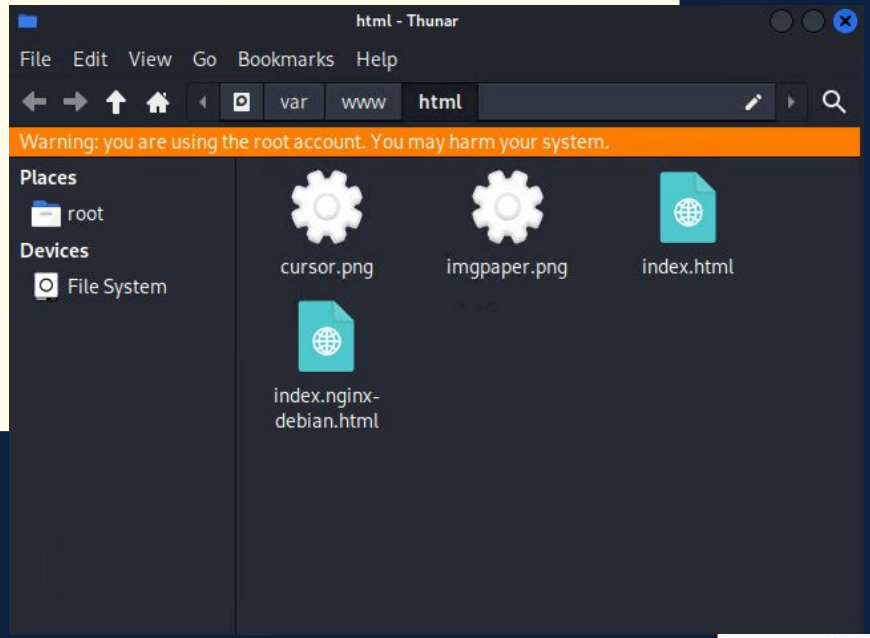
Here we see that when we try to access Google, from the IP “142.251.111.138”, we are given an alert saying there was traffic detected to that IP.

14 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-11-18 17:18:30	⚠	0	TCP		192.168.1.2	50191	142.251.111.138	80	1:10002	IP traffic to 142.251.111.138 detected

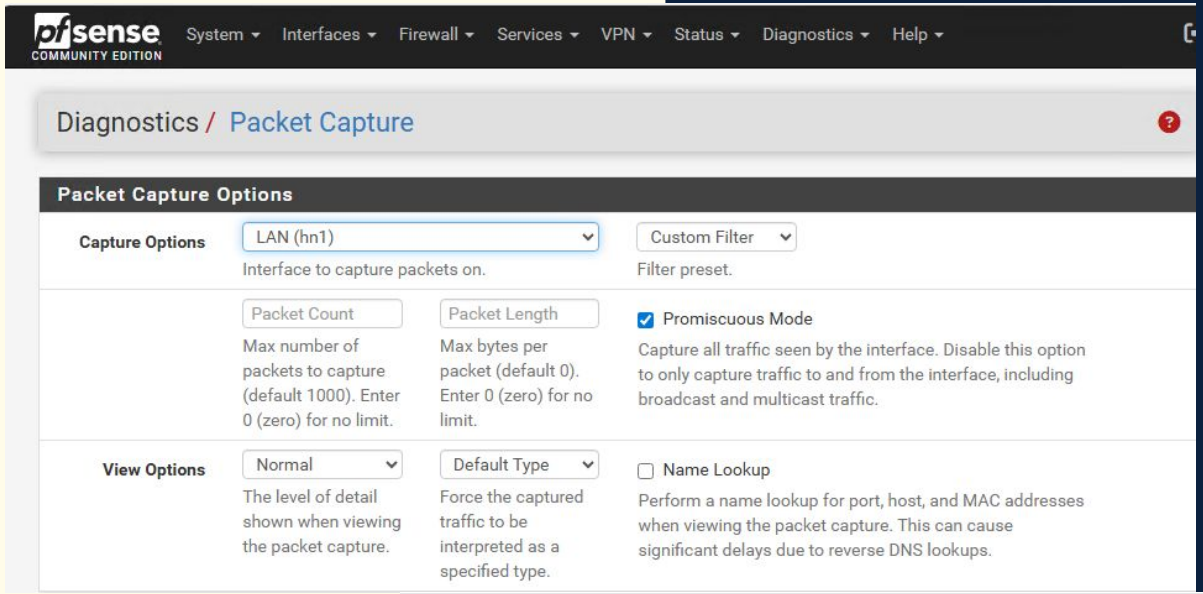
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start apache2
[sudo] password for kali:
```

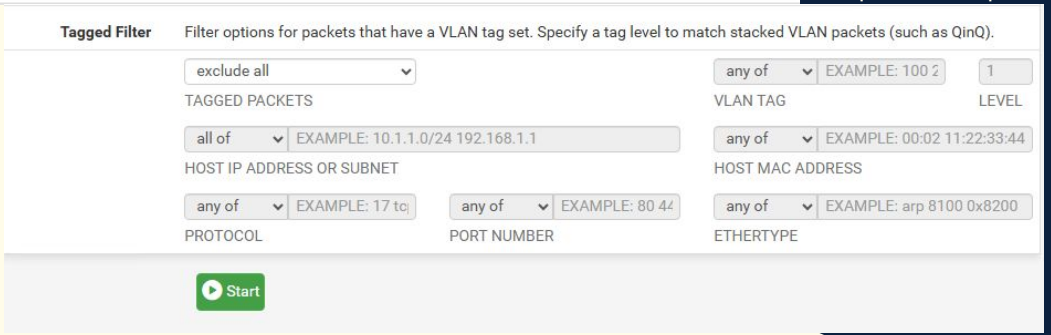
Next we are going to mess around with a virus. Using Kali, we need to set up and Apache server.



Here in our wwwroot folder on the Apache2 web server we have added our two virus files “cursor.png” and “imgpaper.png”. These are both png files with hidden executables in them.

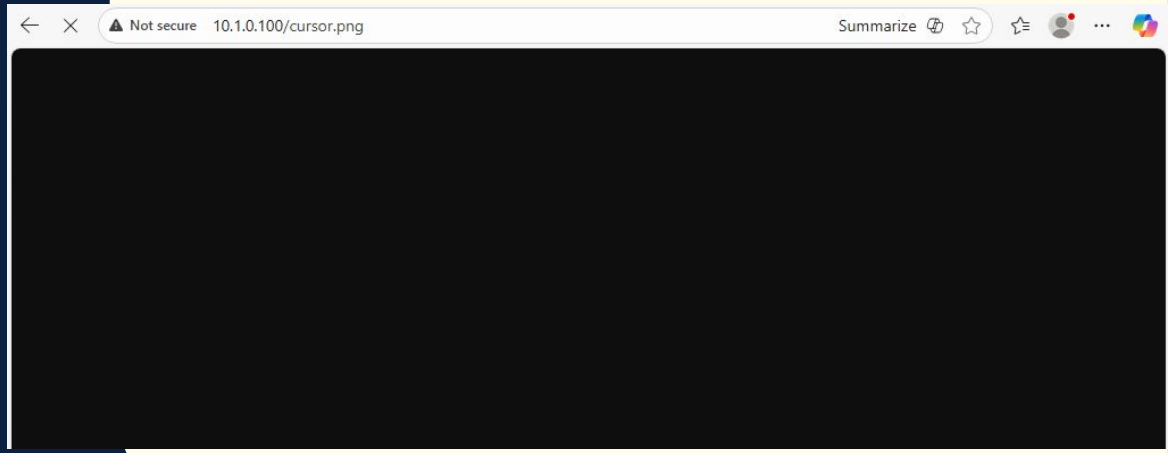
Before we start this in PFSense we will enable packet tracer

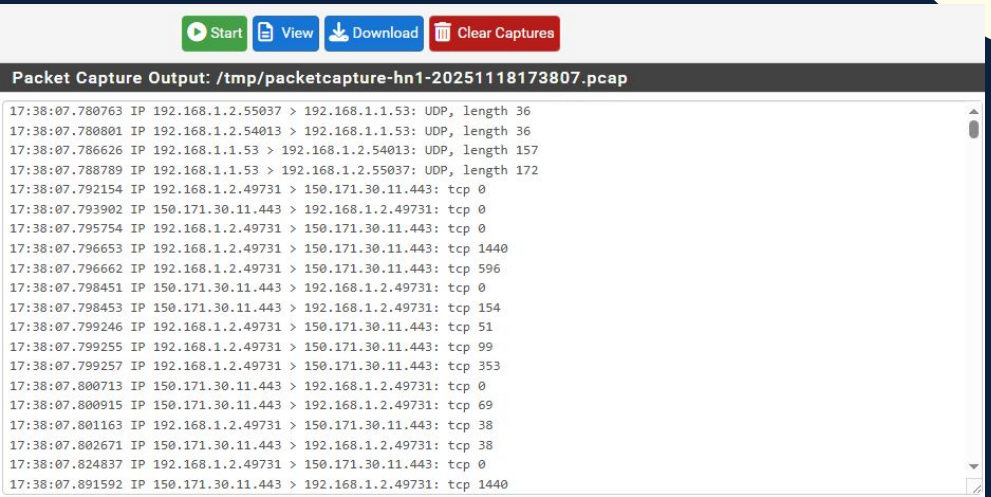




This doesn't track for very long so we need to be quick when doing this next step with the viruses. We want to make sure that we capture the packets with the virus in them.

Quickly start the packet capture on PFSense, then enter the address “10.1.0.100/cursor.png” into the URL and then stop the packet capture. The “cursor.png” is the virus file that we added to our Apache web server.





Inside PFSense this is what we see after stopping the packet capture. We will download this as a .pcap file and inspect it with Wireshark.

Inside the .pcap file there are a bunch of entries, so we will filter them by “http” and it will display our web search to “.../cursor.png”

http

No.	Time	Source	Destination	Protocol	Lengtl	Info
209	5.663281	192.168.1.2	10.1.0.100	HTTP	503	GET /cursor.png HTTP/1.1

Wireshark · Follow TCP Stream (tcp.stream eq 5) · packetcapture-hn1-202511181738...

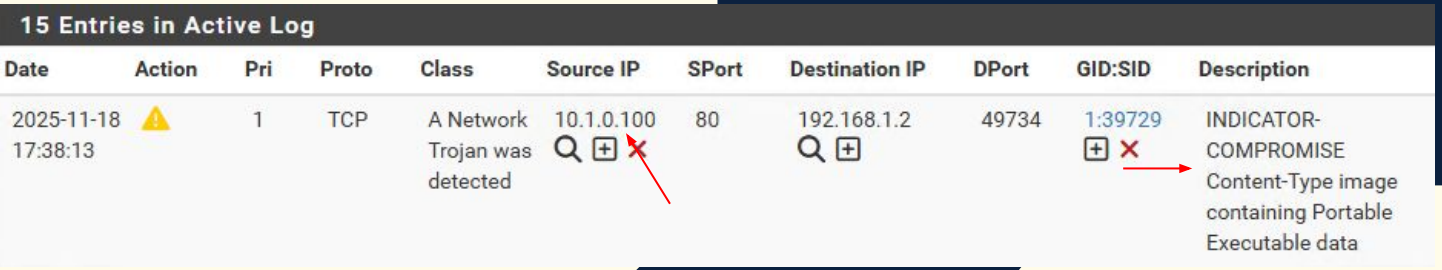
```
GET /cursor.png HTTP/1.1
Host: 10.1.0.100
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/142.0.0.0 Safari/537.36 Edg/142.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9


HTTP/1.1 200 OK
Date: Tue, 18 Nov 2025 17:38:13 GMT
Server: Apache/2.4.63 (Debian)
Last-Modified: Thu, 13 Nov 2025 17:03:46 GMT
ETag: "7b000-6437cdd349ac8"
Accept-Ranges: bytes
Content-Length: 503808
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: image/png

MZ.....................@.....................................          .
!..L.!This program cannot be run in DOS mode.
```

We inspect the TCP Stream of the packet and we can see here that it’s a virus file because at the bottom it says “This program cannot be run in DOS mode.”

Looking back into the alerts on Snort we can see that this was flagged.

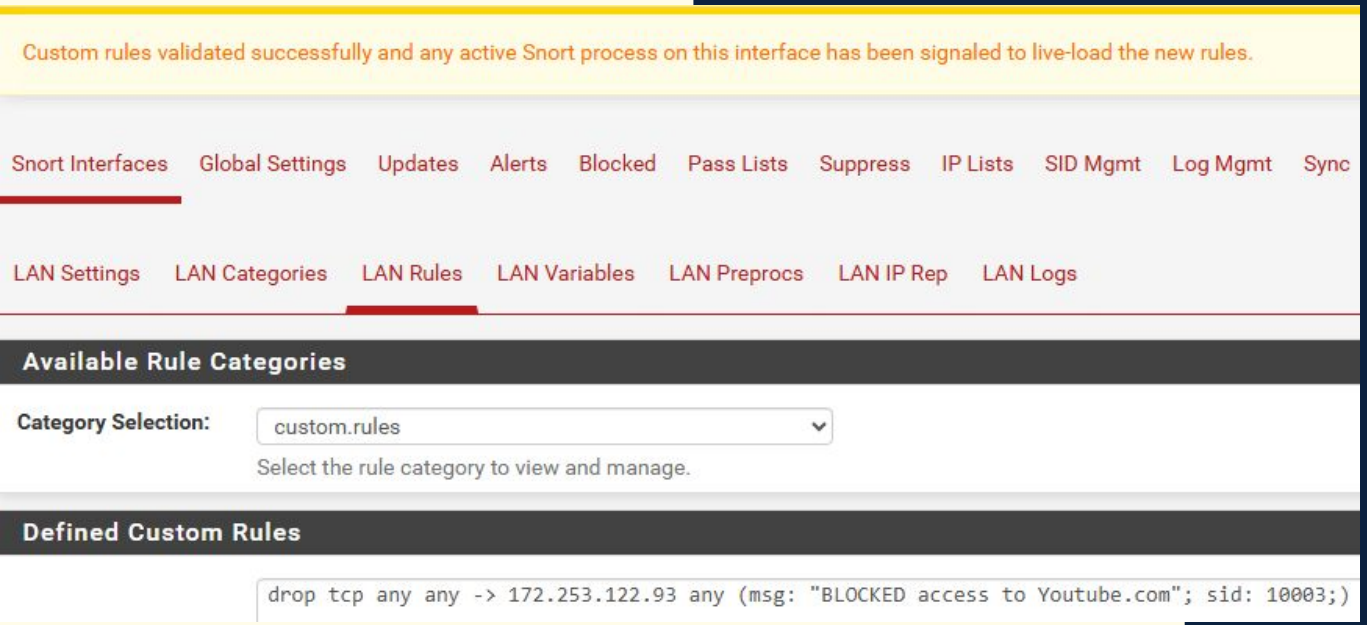


15 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-11-18 17:38:13	⚠	1	TCP	A Network Trojan was detected	10.1.0.100	80	192.168.1.2	49734	1:39729	INDICATOR-COMPROMISE Content-Type image containing Portable Executable data

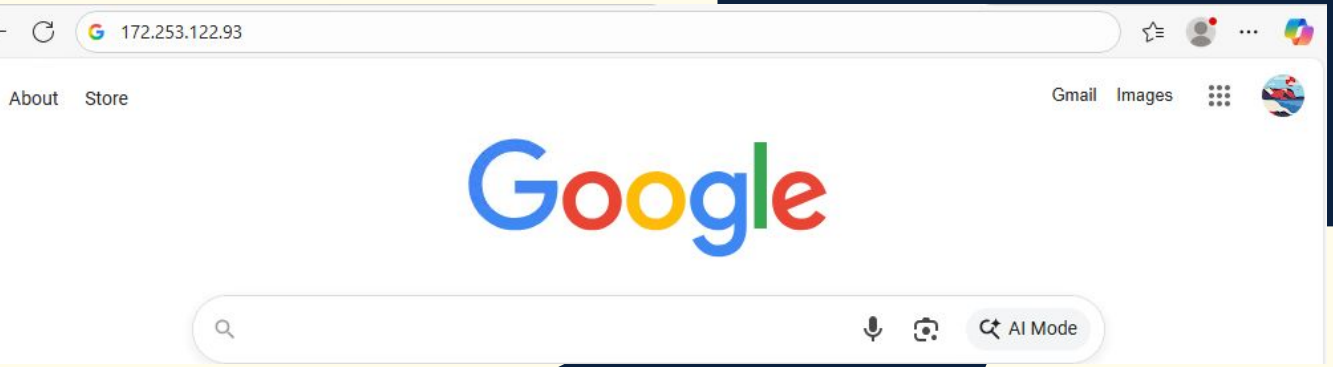
We can see it blocked the Source IP which was “10.1.0.100” which is our Apache server, and described it as “Content-Type image containing portable Executable data”. It found and flagged the virus inside the web server and blocked it from executing on our systems. Now at this point we would need to inspect the virus and check everything out to make sure that everything is how it should be and no viruses were actually downloaded onto the system.

Now for fun let's block access to "youtube.com" on our LAN network so people aren't able to get sidetracked while working.



Here for the command instead of doing "alert" we are doing a "drop" command. With this command we won't be able to access youtube.com at all on the network, it will just drop our packet when we try to connect to the website. We won't get an error, or a message saying the sites blocked it just won't connect.

As you can see here when attempting to connect to youtube.com from the IP “172.253.122.93” which is one of their many IPs we can’t connect and are sent to Google’s homepage.



24 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-11-18 17:53:14	⚠	0	TCP		192.168.1.2	49870	172.253.122.93	80	1:10003	BLOCKED access to Youtube.com

Inside of our alert page in Snort we can see here the entry of the attempt to connect to youtube.com. And our message saying it blocked access to the site.