COMP232 - A3

Adam Cain

https://www.isi.deterlab.net/file.php?file=/share/shared/SecuringlegacysystemswithSnort/index.html#basictasks

Start Snort Without Rules

This experiment involves using an SSH client to connect to users.deterlab.net, with a Deterlab username and password, and then connecting to the Snort node. I then start the Snort application. In another terminal, tcpdump is used to capture data going to and from client1. After 30 seconds, tcpdump is stopped and a command is run to produce a set of x,y coordinates representing time and the number of packets per second.

1. What happens to the traffic to client1 when Snort is not running?

   If snort is not running the traffic that passes through the snort node to client1 will be unmonitored. The traffic will flow to the client without any problem however no alerts or notifications would be generated.

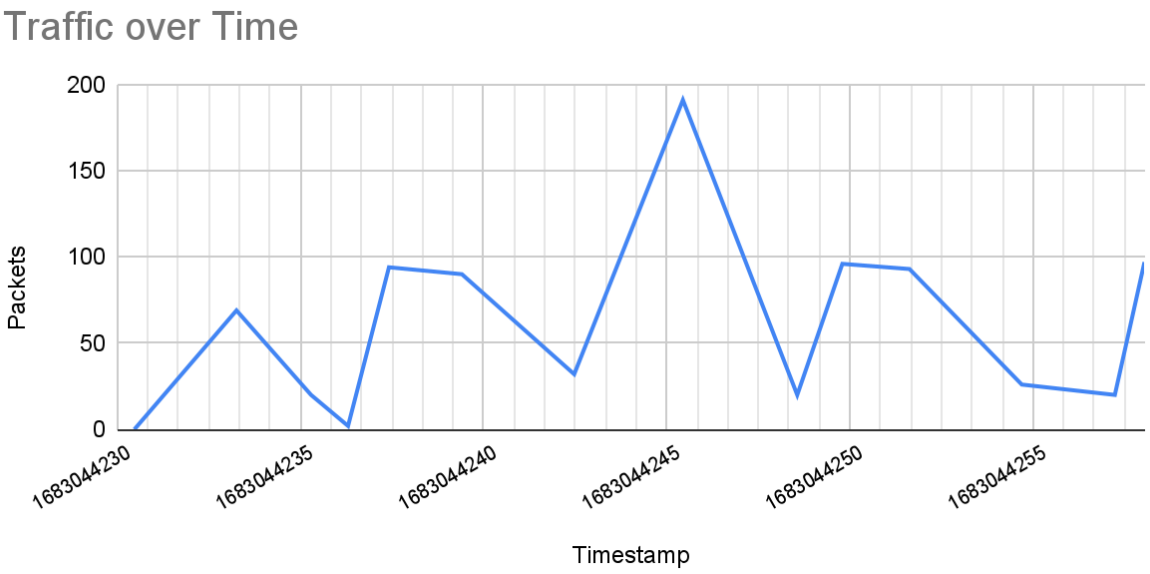
2. Is this a good thing?

   Without snort running there will not be any intrusion detection or prevention, leaving any malicious activity going unnoticed.

3. Based on Snort's output, what can you say about the application? What port does it connect to?

   Based on Snorts output you can see that it uses a TCP protocol,uses TCP SYN flag to establish the connection and receives ACK, acknowledgments, flag as a response. You can see that it connects on IP address 100.1.10.10 on port 7777.

4. Please attach a graph of the traffic over time to your answers



5. What does the "-Q" option do in Snort?

   The -Q option in the snort command enables quiet mode. This suppresses normal startup messages and only prints alerts and error messages.

6. What does the "--daq nfq" option do in Snort?

   The –daq nfq option sets the data acquisition(daq) module to use the Linux netfilter queue(nfq) daq. This allows Snort to capture network traffic that is being processed by the Linux kernels net filter.

Analyze Network Traffic

This experiment involves connecting to the router node and using ifconfig to determine its network interfaces. Then using tcpdump again to capture data going to the server. After capturing data for a minute the tcpdump is terminated. The data is then copied to my local machine using scp and opened using Wireshark to analyse the tcp traffic.

5. The request that the client sends the server is broken into four parts. What are these parts and what order do they appear in? How are these parts separated in the request?

   The request contains Ethernet, IP and TCP headers along with an optional Data body, in that order. They are separated through encapsulation by placing the data in an Ethernet frame, which is encapsulated in an IP packet, and then encapsulated in a TCP segment which includes the data sent by the client.

6. Is this a secure way for the client to send requests to the server? Explain your answer.

The client is sending information unencrypted as it can be seen clearly by the router during the tcpdump. This means a bad actor could theoretically intercept the tcp request, such as in a man in the middle attack, and expose sensitive information that client would wish to keep private.

7. Can you recover one of the files sent by the server to a client? If so, attach the file, a pcap the relevant packets and indicate which client this was sent to.

   Dump.pcap, packets.pcap and file.txt attached to submission
   The File was sent from the server(100.1.10.10:7777) to client2(100.1.5.11:37244)

Start Snort Without Rules

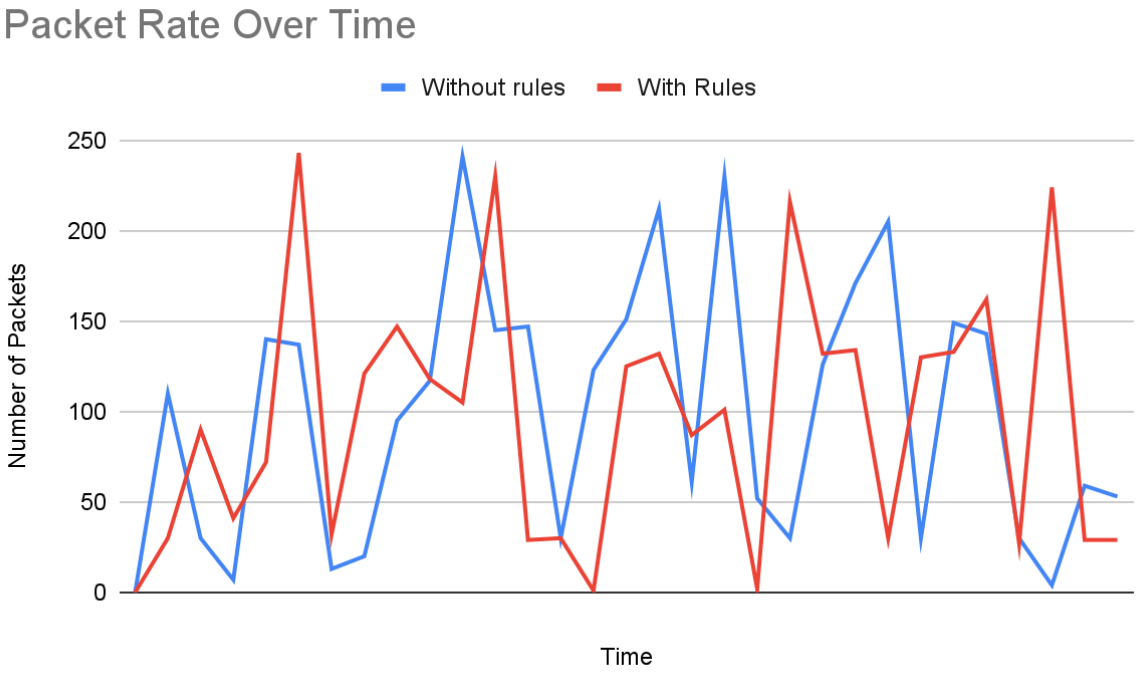
This experiment involves writing rules to protect against simple requests using Snort. Start by stopping the Snort application, then writing a configuration file using nano, and add a sample Snort rule that blocks requests from outgoing requests for .xml files. Then creating a folder for Snort alert logs and starting Snort with the new rule and alerts directed to the new folder. Then verifying the rule works by checking that no.xml files have been recreated on client1, client2 or the outsider nodes.

8. What rule did you use to secure the "classified" file?

reject tcp 100.1.200.10 ANY -> 100.1.10.10 7777 (msg: "Classified data exfiltration detected"; sid:2; content:"classified.txt";)

Assuming that the “classified” file meant the classified.txt file on the server.

9. Capture and compare the network traffic for the server when filtering these results using your configuration file and when no file is used. Attach the graph showing packet rate over time for both of these cases to your submission.



10. Can you think of any other files or extensions that should be filtered against?

    Server files such as config files or the actual server application could allow bad actors to find vulnerabilities in the server that may allow them access to the system. Blocking these from being downloaded by any user would prevent this from ever happening. If the server has the ability for files to be uploaded to it, files that may enable remote code execution such as .exe, .dll, .bat, etc should be restricted.