DETECTING SYN FLOOD ATTACK

Report (Fall 2020)

Ajinkya Mukherjee, M.S Cybersecurity

New York Institute of Technology | College of Engineering and Computer Science

Introduction

In this assignment, I was expected to write two rules to detect the SYN Flooding attack. I used a file called **"outside.tcpdump"** which attacks a machine with IP address **172.16.112.50.** There was a total of two signatures to be implemented by me.

While doing this project I ran into an anomaly, I got two different attack packet values using Wireshark and Snort.
While using **Wireshark** as an analysis tool I got the following values:

Total Packets: 820820
Total Number of Attack packets detected(SYNFlood Packets): 204800
Total Number of Normal Packets: 616020

While using **SNORT IDS** as an analysis tool I got the following values:

Total Packets: 820820
Total Number of Attack packets detected(SYNFlood Packets): 205410
Total Number of Normal Packets: 615020

Through graph 1.1 I have showed how the values for True positive rate are above the value of 1 and hence give us the wrong ROC Curve which is why I have not used values arising from Wireshark for any other graphs.

After Careful evaluation on both SNORT and Wireshark I noticed that the attack taking place on Pascal machine with IP address : *172.16.112.50* is coming from one specific IP address i.e. *10.20.30,40.* Even though this does not happen in real life it is the case for this specific dump file.

Formula's Used to Calculate the Values:

- **Detection Accuracy / True Positive Rate(TPR)**: $\frac{True\ Postives}{Normal\ Packets.}$
- **False Positive Rate(FPR)** : $\frac{False\ Positives}{Attack\ Packets}$

Tools:

1. Wireshark : Network Analysis
2. SNORT IDS: Network Analysis
3. VIM: Writing rules on Local.rules

System:

1. Ubuntu 18.04.5 (Linux)

Steps taken

1. Import the TCP dump file to Wireshark.
2. The local **snort.conf** file was changed by making the $HOME_NET var as a list of Destination IP addresses found in the file using Wireshark.
3. Commands to run the rules along with
    a. Edit the local.rules file: *sudo vim /etc/snort/rules/local.rules*
    b. Validate the rule configuration: *sudo snort -T -c /etc/snort/snort.conf*
    c. Run the tcpdump file with our custom rules: *sudo snort -r /home/Ajinkya/Desktop/Assignment1/outside.tcpdump -c /etc/snort/snort.conf -A console*.

Rule Parameters

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 1;sid: 100002;rev:1;)

- *Alert* : rule action
- *Any* : source IP
- *Any*: source port
- -> : direction from source to destination
- *$HOME_NET* : Destination IP. We are using the HOME_NET value from the snort.conf file.
- *Any* : Destination Port
- *Msg*: SNORT will include this message as the alert message.
- *Detection_filter*: Detects detection filter using either the destination IP (*track by_dst*) or source IP (*track by_src*).
- *Count* : number of requests during sampling period.
- *Seconds*: sampling period
- *Sid*: SNORT Rule ID
- *Rev*: Revision Number

ROC Analysis

The following criteria were kept in mind while doing the ROC analysis.

- ***ROC Curve has True Positive Rate(TPR) or Detection Accuracy in the Y-axis or vertical axis.***
- ***ROC Curve has False Positive Rate(FPR) in the X-axis or horizontal axis.***
- ***The value closest to the top left corner of the graph i.e., Higher True Positive Rate and Low False Positive Rate is the ideal Threshold value.***
- ***In case of similar distance from the top left corner of the graph, the value with lower False Positive Rate is chosen as the True Positive Rate values differ by a very small number.***
-

NOTE: *Lower Thresholds lead to a lot of False Positive Values which in turn leads to admin losing trust in the system as False Alarm Rate are very high, i.e., a lot of normal packets are flagged as attack packets.*
High Threshold leads to low False alarms and Higher True Negative i.e., a lot of attack packets are considered normal packets leading to a poor Detection accuracy and failing the system.

Rules & Signatures used

Rule 1

This rule analyzes the total number of SYN packets entering per unit time by tracking the number of SYN packets entering the destination IP at different thresholds and different times. (tracking by Destination).
The Thresholds used in the following rules are 5,9,15,25,29,61 with different sampling periods 1s,2s,5s, and 10s.
Even though the FPR and TPR are calculated for threshold 61, it is not shown in the Graphs below as the Detection accuracy for that threshold is extremely low as compared to other thresholds. The value has calculated the values for that threshold to show High Thresholds' effect on the Detection accuracy and True Misses.
The ROC analysis for every sampling period is below.

Table 1: Time 1s

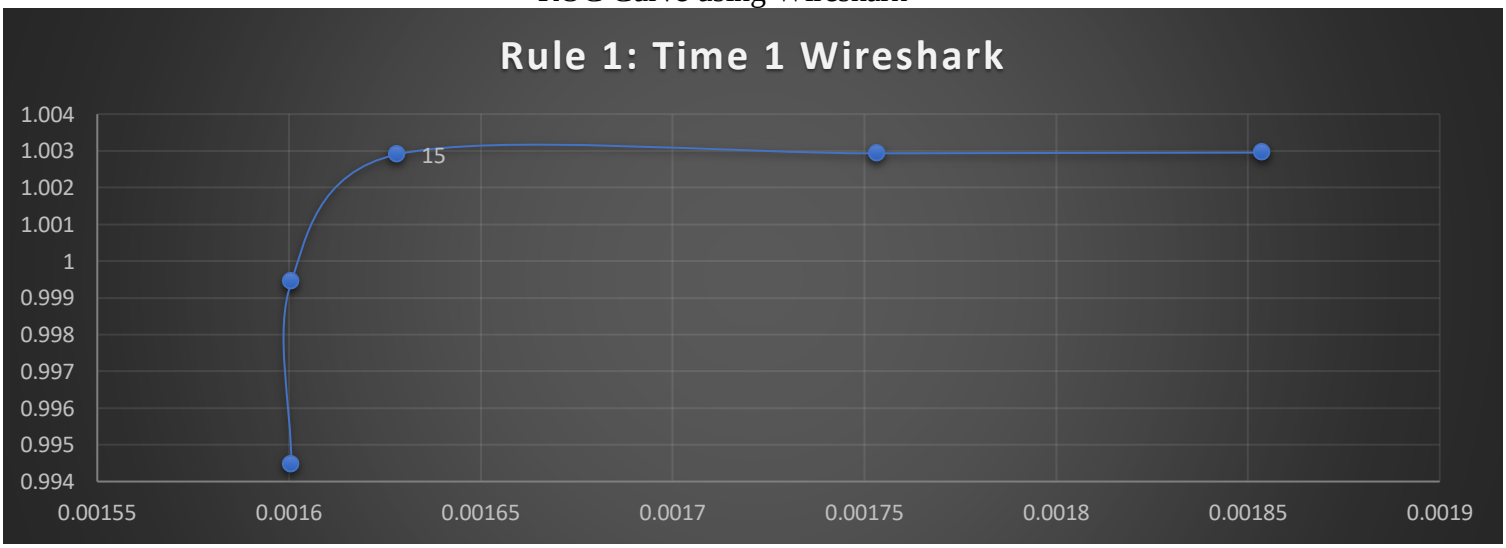
In the below table there are two False Positive Rate values(FPR) and True Positive Rate Value(TPR), one belongs to Analysis using Wireshark and the other using Snort. The Sampling period/ Time window is 1s

Threshold	Alerts generated	True +ves	False +ves	True -ves (WireShark)	True -ves(SNORT)	FPR(Wireshark)	TPR(Wireshark)	FPR(SNORT)	TPR(SNORT)
5	206547	205405	1142	-605	5	0.001853836	1.002954102	0.00185567	0.99997566
9	206481	205401	1080	-601	9	0.00175319	1.00293457	0.00175493	0.99995619
15	206398	205395	1003	-595	15	0.001628194	1.002905273	0.00162981	0.99992698
25	205671	204685	986	115	725	0.001600597	0.999438477	0.00160218	0.99647047
29	204654	203668	986	1132	1742	0.001600597	0.994472656	0.00160218	0.9915194
61	116728	115750	978	89050	89660	0.001587611	0.565185547	0.00158918	0.56350713

Graph 1: Analysis using Wireshark

Through the graph below, we notice that there are TPR values that are above 1, which is not possible. It is due to the irregularity of packet detection by Wireshark vs. snort. Since all the alerts used are used according to alerts generated by snort.

ROC Curve using Wireshark



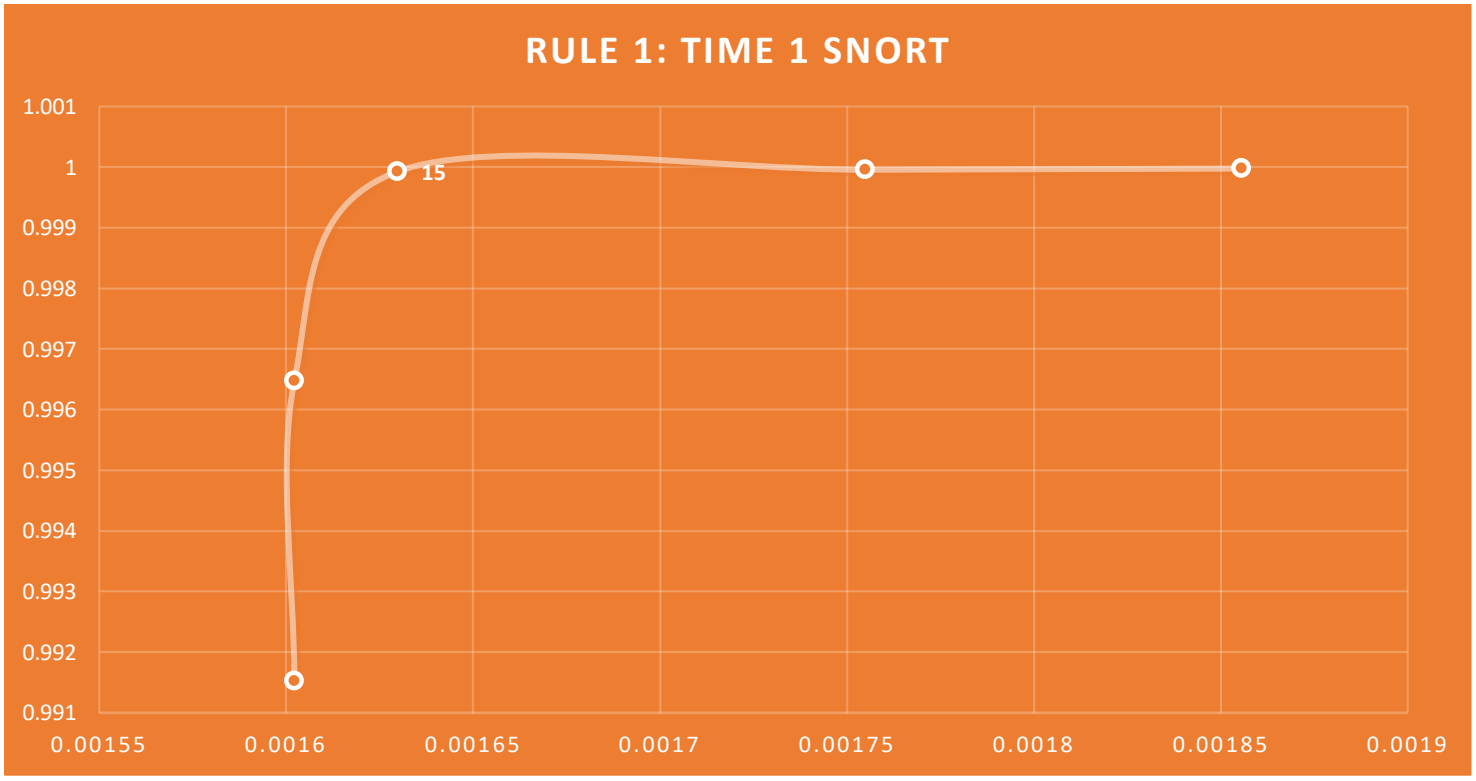
Graph 1.1

This graph shows that our value for True Positive Rate is going above 1, which should not be the case in an ideal ROC curve. It is due to the discrepancies in the analysis between Wireshark and SNORT.

According to ROC analysis, the best threshold to be used is the point closest to the top left corner of the graph, which would be the rule with threshold 15.

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 1;sid: 100002;rev:1;)

ROC Curve using SNORT IDS



Graph 1.2

This graph shows that all the values are under 1, since all the values are calculated based on packet generation from SNORT IDS.
According to ROC analysis, the best threshold to be used is the point closest to the top left corner of the graph, which would be the rule with threshold 15.

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 1;sid: 100002;rev:1;)

Overall, through both the graphs and the table, we see that with Thresholds lower than 15 (5 and 9), we do get higher Detection accuracy, but we also have a higher False Positive Rate, and that is why we cannot use those thresholds as this will lead to the admin lose trust in our system.
With Thresholds higher than 15, i.e.(25,29 and 61), we get a low False Positive rate and get a detection accuracy(True Positive Rate), which is lower than previous thresholds.

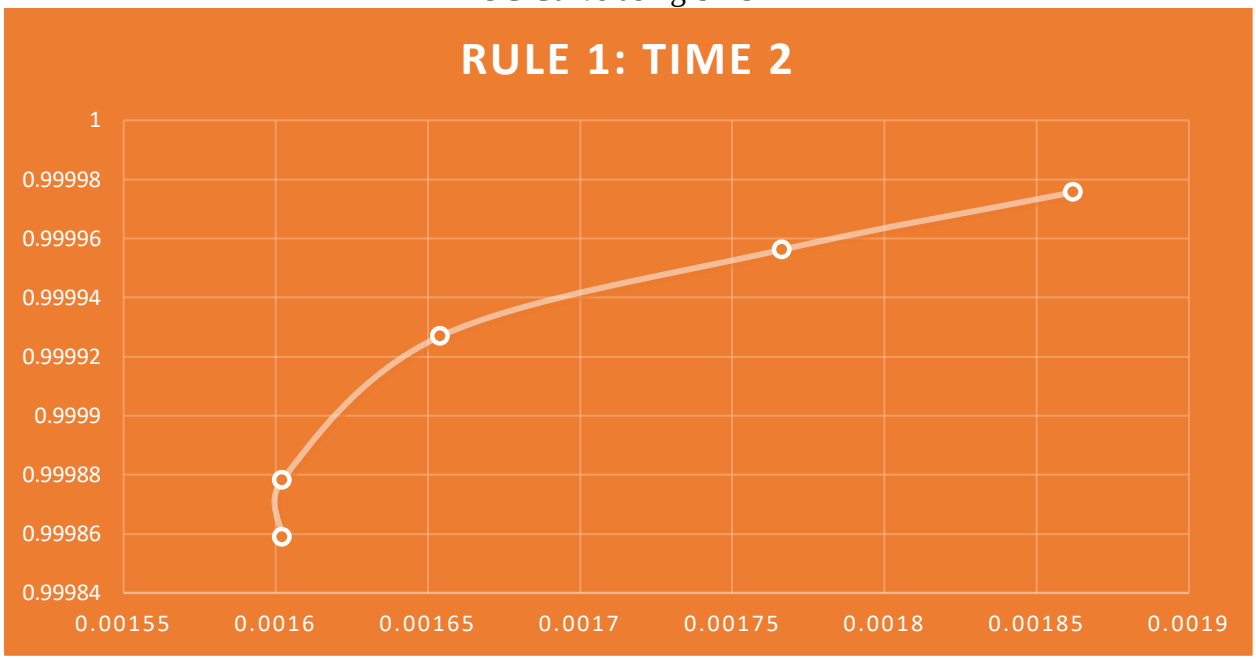
Hence Threshold of 15 is the ideal Threshold to use in this scenario.

Table 2: Time 2s

In the below table there are two False Positive Rate values(FPR) and True Positive Rate Value(TPR), one belongs to Analysis using Wireshark and the other using Snort. The Sampling period/ Time window is 2s

Threshold	Alerts generated	True +ves	False +ves	True -ves (WireShark)	True -ves (SNORT)	FPR(Wireshark)	TPR(Wireshark)	FPR(SNORT)	TPR(SNORT)
5	206551	205405	1146	-605	5	0.001860329	1.002954102	0.00186217	0.99997566
9	206488	205401	1087	-601	9	0.001764553	1.00293457	0.0017663	0.99995619
15	206413	205395	1018	-595	15	0.001652544	1.002905273	0.00165418	0.99992698
25	206371	205385	986	-585	25	0.001600597	1.002856445	0.00160218	0.99987829
29	206367	205381	986	-581	29	0.001600597	1.002836914	0.00160218	0.99985882
61	204797	203811	986	989	1599	0.001600597	0.995170898	0.00160218	0.99221557

ROC Curve using SNORT



Graph 1.3

According to ROC analysis, the best threshold to be used is the point closest to the top left corner of the graph, which would be the rule with threshold 15.

Since Rule with Threshold 9 has a higher False positive rate and a slightly better True Positive Rate, we should choose the Threshold with 15.

We see that with Thresholds lower than 15 (5 and 9), we do get higher Detection accuracy, but we also have a higher False Positive Rate, and that is why we cannot use those thresholds as this will lead to the admin lose trust in our system.

With Thresholds higher than 15, i.e.(25,29 and 61), we get a low False Positive rate and get a detection accuracy(True Positive Rate), which is lower than previous thresholds.

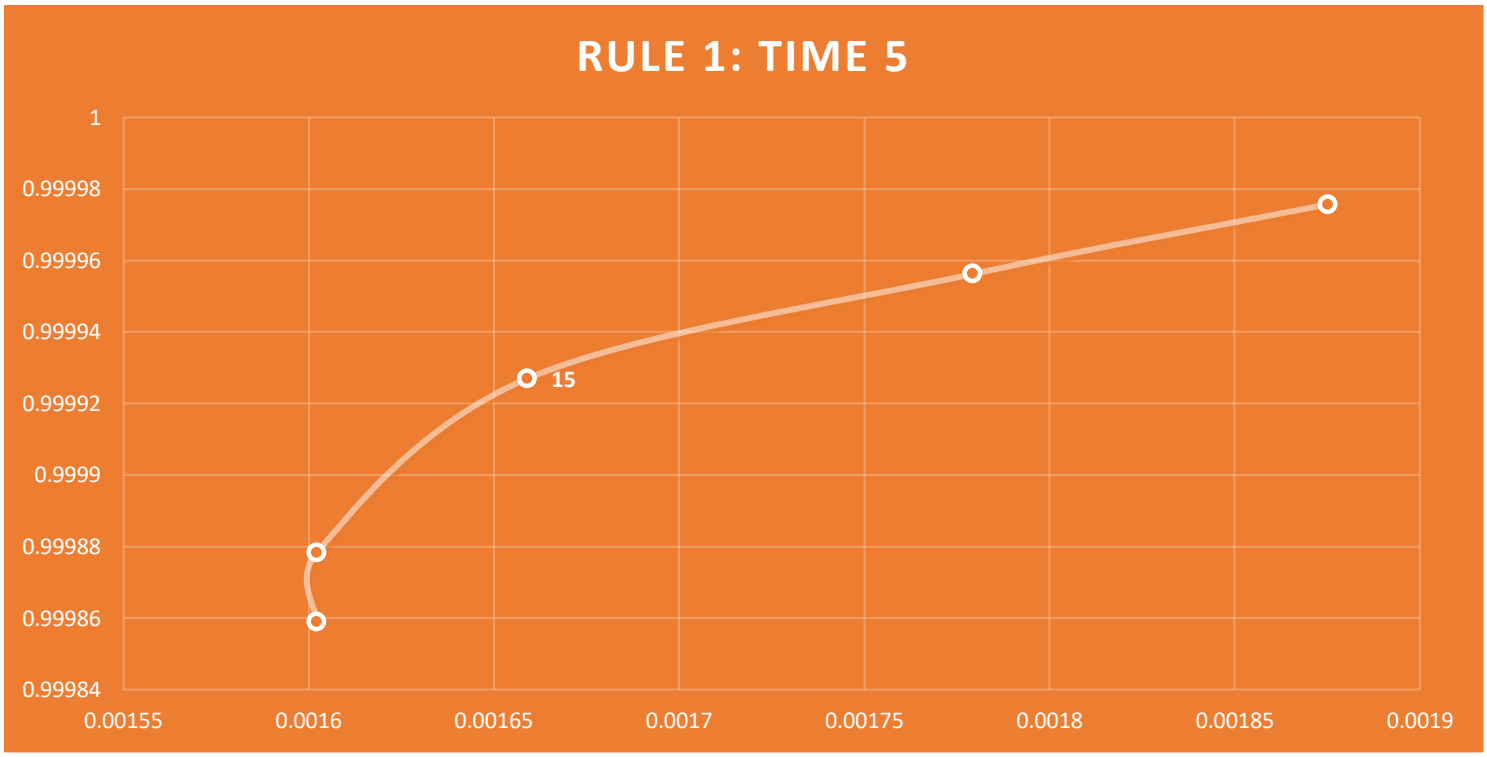
Hence Threshold of 15 is the ideal Threshold to use in this scenario.

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 2;sid: 100002;rev:1;)

Table 3: Time 5s

In the below table there are two False Positive Rate values(FPR) and True Positive Rate Value(TPR), one belongs to Analysis using Wireshark and the other using Snort. The Sampling period/ Time window is 5s

Threshold	Alerts generated	True +ves	False +ves	True -ves (WireShark)	True -ves (SNORT)	FPR(Wireshark)	TPR(Wireshark)	FPR(SNORT)	TPR(SNORT)
5	206559	205405	1154	-605	5	0.001873316	1.002954102	0.00187517	0.99997566
9	206496	205401	1095	-601	9	0.00177754	1.00293457	0.0017793	0.99995619
15	206416	205395	1021	-595	15	0.001657414	1.002905273	0.00165906	0.99992698
25	206371	205385	986	-585	25	0.001600597	1.002856445	0.00160218	0.99987829
29	206367	205381	986	-581	29	0.001600597	1.002836914	0.00160218	0.99985882
61	206335	205349	986	-549	61	0.001600597	1.002680664	0.00160218	0.99970303



Graph 1.4

According to ROC analysis, the best threshold to be used is the point closest to the top left corner of the graph, which would be the rule with threshold 15.

Since Rule with Threshold 9 has a higher False positive rate and a slightly better True Positive Rate, we should choose the Threshold with 15.

We see that with Thresholds lower than 15 (5 and 9), we do get higher Detection accuracy, but we also have a higher False Positive Rate, and that is why we cannot use those thresholds as this will lead to the admin lose trust in our system.

With Thresholds higher than 15, i.e.(25,29 and 61), we get a low False Positive rate and get a detection accuracy(True Positive Rate), which is lower than previous thresholds.

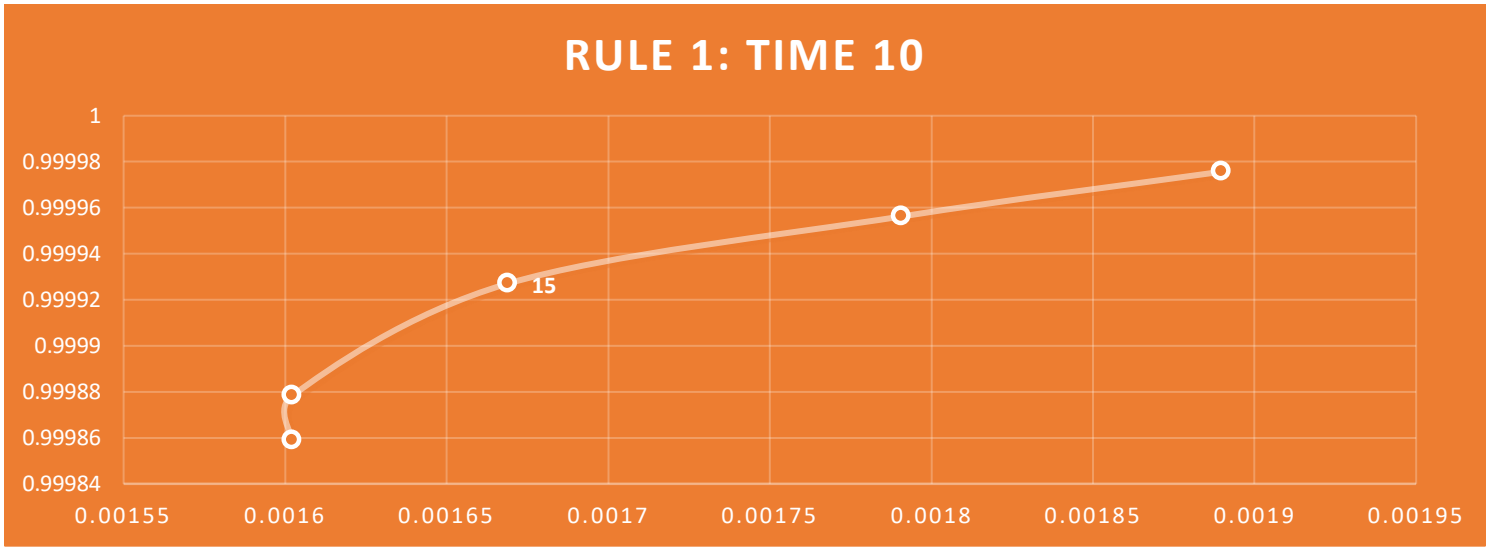
Hence Threshold of 15 is the ideal Threshold to use in this scenario.

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 5;sid: 100002;rev:1;)

Table 4: Time 10s

In the below table there are two False Positive Rate values(FPR) and True Positive Rate Value(TPR), one belongs to Analysis using Wireshark and the other using Snort. The Sampling period/ Time window is 10s

Threshold	**Alerts generated**	**True +ves**	**False +ves**	**True -ves (WireShark)**	**True -ves (SNORT)**	**FPR(Wireshark)**	**TPR(Wireshark)**	**FPR(SNORT)**	**TPR(SNORT)**
5	206568	205405	1163	-605	5	0.001887926	1.002954102	0.0018898	0.99997566
9	206503	205401	1102	-601	9	0.001788903	1.00293457	0.00179068	0.99995619
15	206422	205395	1027	-595	15	0.001667154	1.002905273	0.00166881	0.99992698
25	206371	205385	986	-585	25	0.001600597	1.002856445	0.00160218	0.99987829
29	206367	205381	986	-581	29	0.001600597	1.002836914	0.00160218	0.99985882
61	206335	205349	986	-549	61	0.001600597	1.002680664	0.00160218	0.99970303



Graph 1.5

According to ROC analysis, the best threshold to be used is the point closest to the top left corner of the graph, which would be the rule with threshold 15.

Since Rule with Threshold 9 has a higher False positive rate and a slightly better True Positive Rate, we should choose the Threshold with 15.

We see that with Thresholds lower than 15 (5 and 9), we do get higher Detection accuracy, but we also have a higher False Positive Rate, and that is why we cannot use those thresholds as this will lead to the admin lose trust in our system.

With Thresholds higher than 15, i.e.(25,29 and 61), we get a low False Positive rate and get a detection accuracy(True Positive Rate), which is lower than previous thresholds.

Hence Threshold of 15 is the ideal Threshold to use in this scenario.

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 10 ;sid: 100002;rev:1;)

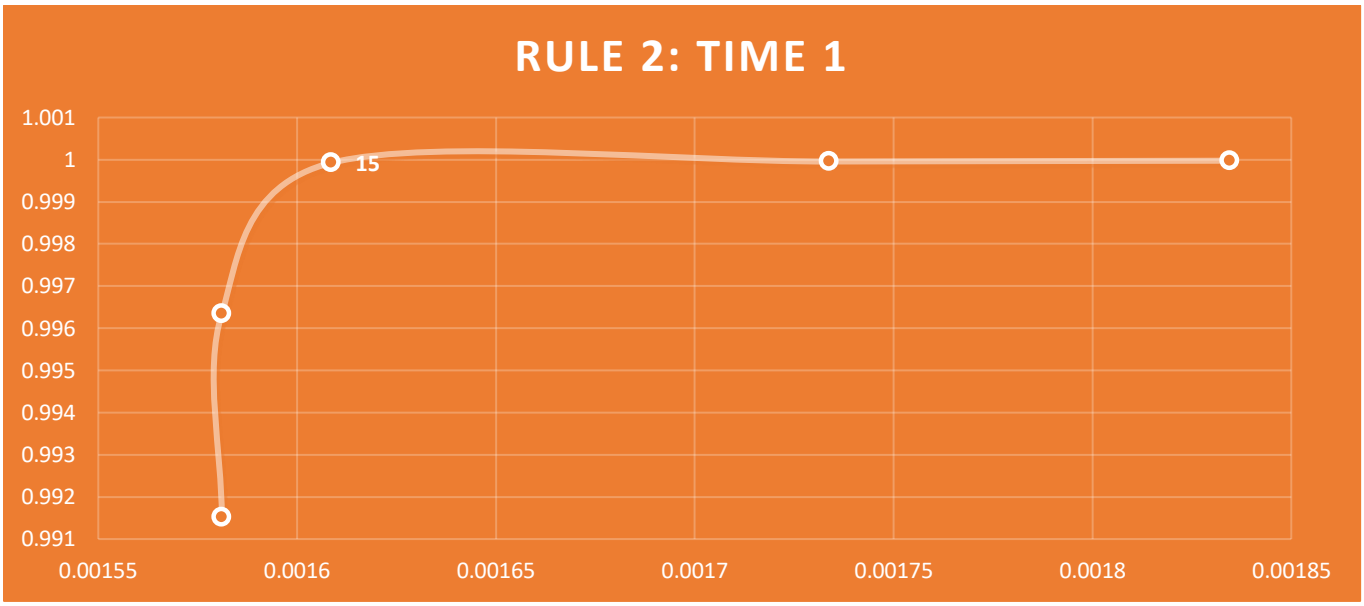
Rule 2

This rule analyzes the total number of SYN packets leaving per unit time by tracking the number of SYN packets leaving the source IP at different thresholds and different times. (tracking by Source).
The Thresholds used in the following rules are 5,9,15,25,29,61 with different sampling periods 1s,2s,5s, and 10s.
Even though the FPR and TPR are calculated for threshold 61, it is not shown in the Graphs below as the Detection accuracy for that threshold is extremely low as compared to other thresholds. The value has calculated the values for that threshold to show High Thresholds' effect on the Detection accuracy and True Misses.
The ROC analysis for every sampling period is below.

Table 1: Time 1s

In the below table there are two False Positive Rate values(FPR) and True Positive Rate Value(TPR), one belongs to Analysis using Wireshark and the other using Snort. The Sampling period/ Time window is 1s

Threshold	Alerts generated	True +ves	False +ves	True -ves (WireShark)	True -ves (SNORT)	FPR(Wireshark)	TPR(Wireshark)	FPR(SNORT)	TPR(SNORT)
5	206534	205405	1129	-605	5	0.001832733	1.002954102	0.00183455	0.99997566
9	206468	205401	1067	-601	9	0.001732087	1.00293457	0.0017338	0.99995619
15	206385	205395	990	-595	15	0.001607091	1.002905273	0.00160868	0.99992698
25	205633	204660	973	140	750	0.001579494	0.999316406	0.00158106	0.99634877
29	204641	203668	973	1132	1742	0.001579494	0.994472656	0.00158106	0.9915194
61	116721	115748	973	89052	89662	0.001579494	0.565175781	0.00158106	0.5634974



Graph 2.1

According to ROC analysis, the best threshold to be used is the point closest to the top left corner of the graph, which would be the rule with threshold 15.

Since Rule with Threshold 9 has a higher False positive rate and a slightly better True Positive Rate, we should choose the Threshold with 15.

We see that with Thresholds lower than 15 (5 and 9), we do get higher Detection accuracy, but we also have a higher False Positive Rate, and that is why we cannot use those thresholds as this will lead to the admin lose trust in our system.

With Thresholds higher than 15, i.e.(25,29 and 61), we get a low False Positive rate and get a detection accuracy(True Positive Rate), which is lower than previous thresholds.

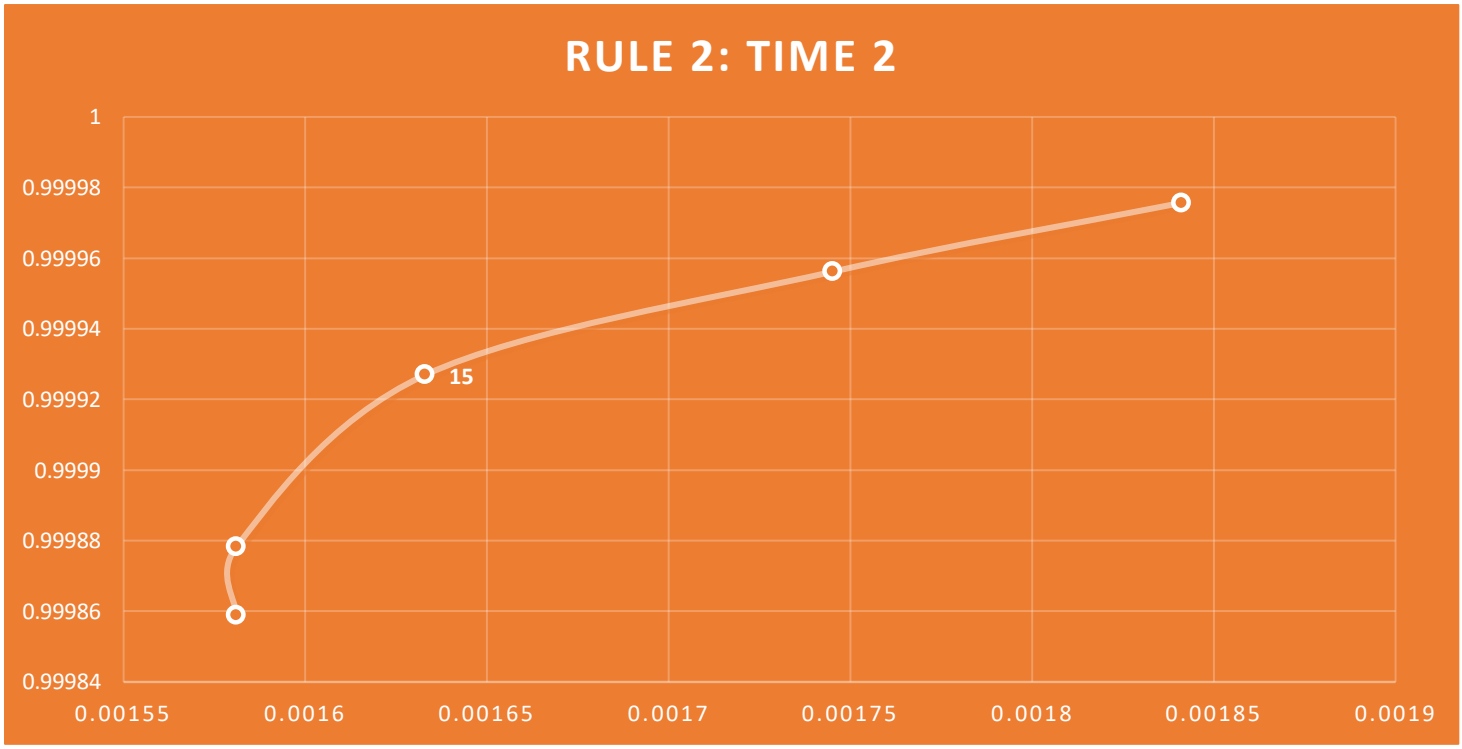
Hence Threshold of 15 is the ideal Threshold to use in this scenario.

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 2";flow:stateless;flags:S;detection_filter:track by_src,count 15, seconds 1 ;sid: 100002;rev:1;)

Table 2: Time 2s

In the below table there are two False Positive Rate values(FPR) and True Positive Rate Value(TPR), one belongs to Analysis using Wireshark and the other using Snort. The Sampling period/ Time window is 2s

Threshold	Alerts generated	True +ves	False +ves	True -ves (WireShark)	True -ves (SNORT)	FPR(Wireshark)	TPR(Wireshark)	FPR(SNORT)	TPR(SNORT)
5	206538	205405	1133	-605	5	0.001839226	1.002954102	0.00184105	0.99997566
9	206475	205401	1074	-601	9	0.00174345	1.00293457	0.00174518	0.99995619
15	206400	205395	1005	-595	15	0.001631441	1.002905273	0.00163306	0.99992698
25	206358	205385	973	-585	25	0.001579494	1.002856445	0.00158106	0.99987829
29	206354	205381	973	-581	29	0.001579494	1.002836914	0.00158106	0.99985882
61	204784	203811	973	989	1599	0.001579494	0.995170898	0.00158106	0.99221557



Graph 2.2

According to ROC analysis, the best threshold to be used is the point closest to the top left corner of the graph, which would be the rule with threshold 15.

Since Rule with Threshold 9 has a higher False positive rate and a slightly better True Positive Rate, we should choose the Threshold with 15.

We see that with Thresholds lower than 15 (5 and 9), we do get higher Detection accuracy, but we also have a higher False Positive Rate, and that is why we cannot use those thresholds as this will lead to the admin lose trust in our system.

With Thresholds higher than 15, i.e.(25,29 and 61), we get a low False Positive rate and get a detection accuracy(True Positive Rate), which is lower than previous thresholds.

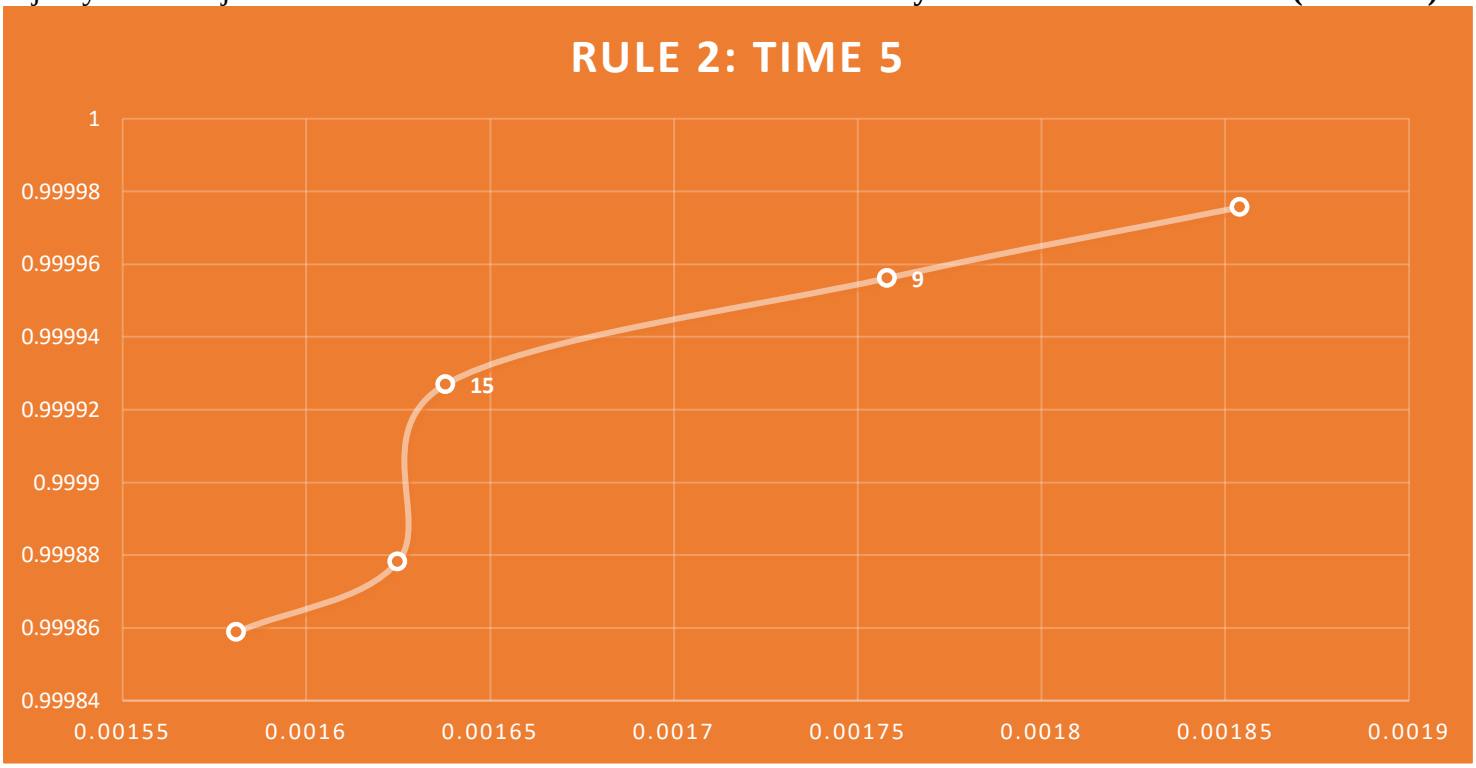
Hence Threshold of 15 is the ideal Threshold to use in this scenario.

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 2";flow:stateless;flags:S;detection_filter:track by_src,count 15, seconds 2 ;sid: 100002;rev:1;)

Table 3:Time 5s

In the below table there are two False Positive Rate values(FPR) and True Positive Rate Value(TPR), one belongs to Analysis using Wireshark and the other using Snort. The Sampling period/ Time window is 5s

Threshold	Alerts generated	True +ves	False +ves	True -ves (WireShark)	True -ves (SNORT)	FPR(Wireshark)	TPR(Wireshark)	FPR(SNORT)	TPR(SNORT)
5	206546	205405	1141	-605	5	0.001852213	1.002954102	0.00185405	0.99997566
9	206483	205401	1082	-601	9	0.001756436	1.00293457	0.00175818	0.99995619
15	206403	205395	1008	-595	15	0.001636311	1.002905273	0.00163793	0.99992698
25	206385	205385	1000	-585	25	0.001623324	1.002856445	0.00162493	0.99987829
29	206354	205381	973	-581	29	0.001579494	1.002836914	0.00158106	0.99985882
61	206322	205349	973	-549	61	0.001579494	1.002680664	0.00158106	0.99970303



Graph 2.3

According to ROC analysis, the best threshold to be used is the point closest to the top left corner of the graph, which would be the rule with threshold 15.

Since Rule with Threshold 9 has a higher False positive rate and a slightly better True Positive Rate, we should choose the Threshold with 15.

We see that with Thresholds lower than 15 (5 and 9), we do get higher Detection accuracy, but we also have a higher False Positive Rate, and that is why we cannot use those thresholds as this will lead to the admin lose trust in our system.

With Thresholds higher than 15, i.e.(25,29 and 61), we get a low False Positive rate and get a detection accuracy(True Positive Rate), which is lower than previous thresholds.

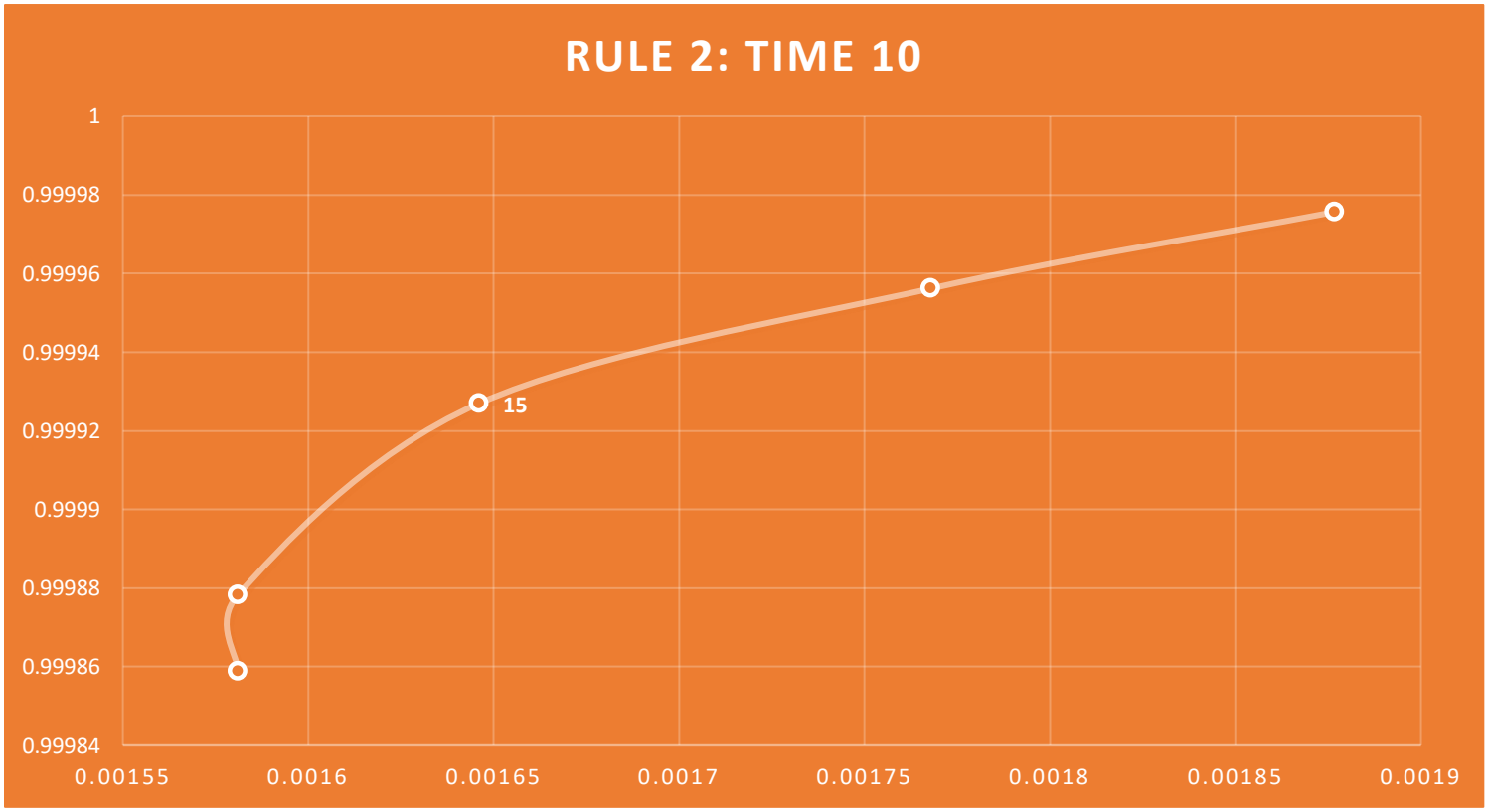
Hence Threshold of 15 is the ideal Threshold to use in this scenario.

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 2";flow:stateless;flags:S;detection_filter:track by_src,count 15, seconds 5 ;sid: 100002;rev:1;)

Table 4: 10s

In the below table there are two False Positive Rate values(FPR) and True Positive Rate Value(TPR), one belongs to Analysis using Wireshark and the other using Snort. The Sampling period/ Time window is 10s

Threshold	Alerts generated	True +ves	False +ves	True -ves (WireShark)	True -ves (SNORT)	FPR(Wireshark)	TPR(Wireshark)	FPR(SNORT)	TPR(SNORT)
5	206560	205405	1155	-605	5	0.001874939	1.002954102	0.0018768	0.99997566
9	206489	205401	1088	-601	9	0.001766176	1.00293457	0.00176793	0.99995619
15	206408	205395	1013	-595	15	0.001644427	1.002905273	0.00164606	0.99992698
25	206358	205385	973	-585	25	0.001579494	1.002856445	0.00158106	0.99987829
29	206354	205381	973	-581	29	0.001579494	1.002836914	0.00158106	0.99985882
61	206322	205349	973	-549	61	0.001579494	1.002680664	0.00158106	0.99970303



Graph 2.4

According to ROC analysis, the best threshold to be used is the point closest to the top left corner of the graph, which would be the rule with threshold 15.

Since Rule with Threshold 9 has a higher False positive rate and a slightly better True Positive Rate, we should choose the Threshold with 15.

We see that with Thresholds lower than 15 (5 and 9), we do get higher Detection accuracy, but we also have a higher False Positive Rate, and that is why we cannot use those thresholds as this will lead to the admin lose trust in our system.

With Thresholds higher than 15, i.e.(25,29 and 61), we get a low False Positive rate and get a detection accuracy(True Positive Rate), which is lower than previous thresholds.

Hence Threshold of 15 is the ideal Threshold to use in this scenario.

alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 2";flow:stateless;flags:S;detection_filter:track by_src, count 15, seconds 10 ;sid: 100002;rev:1;)

Comparison between Rule 1 and Rule 2:

Rule 1 calculates all the SYN packets entering the Destination IP per unit time by using the detection filter track by_dst, while Rule 2 is calculating all SYN Packets leaving from any Source IP per unit time using the detection filter, track by_src. The Detection Accuracy or True Positive Rate for both rules for all Thresholds at all sampling periods is the same. However, the False Positive Rate generated by Rule 2 is less than Rule 1 for all Thresholds of all Sampling Periods.
This factor makes it a better and efficient rule to detect an attack like SYNFlood since we need to have a high Detection Accuracy and low False Positive Rate.

Sampling Period of 1 is the most ideal sampling period out of all the sampling period as with the Threshold of 15 it generates the best True Positive Rate with the least False Positive Rate.

Hence, Selected Rules after ROC analysis:-

Rule 1:
alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 1;sid: 100002;rev:1;)

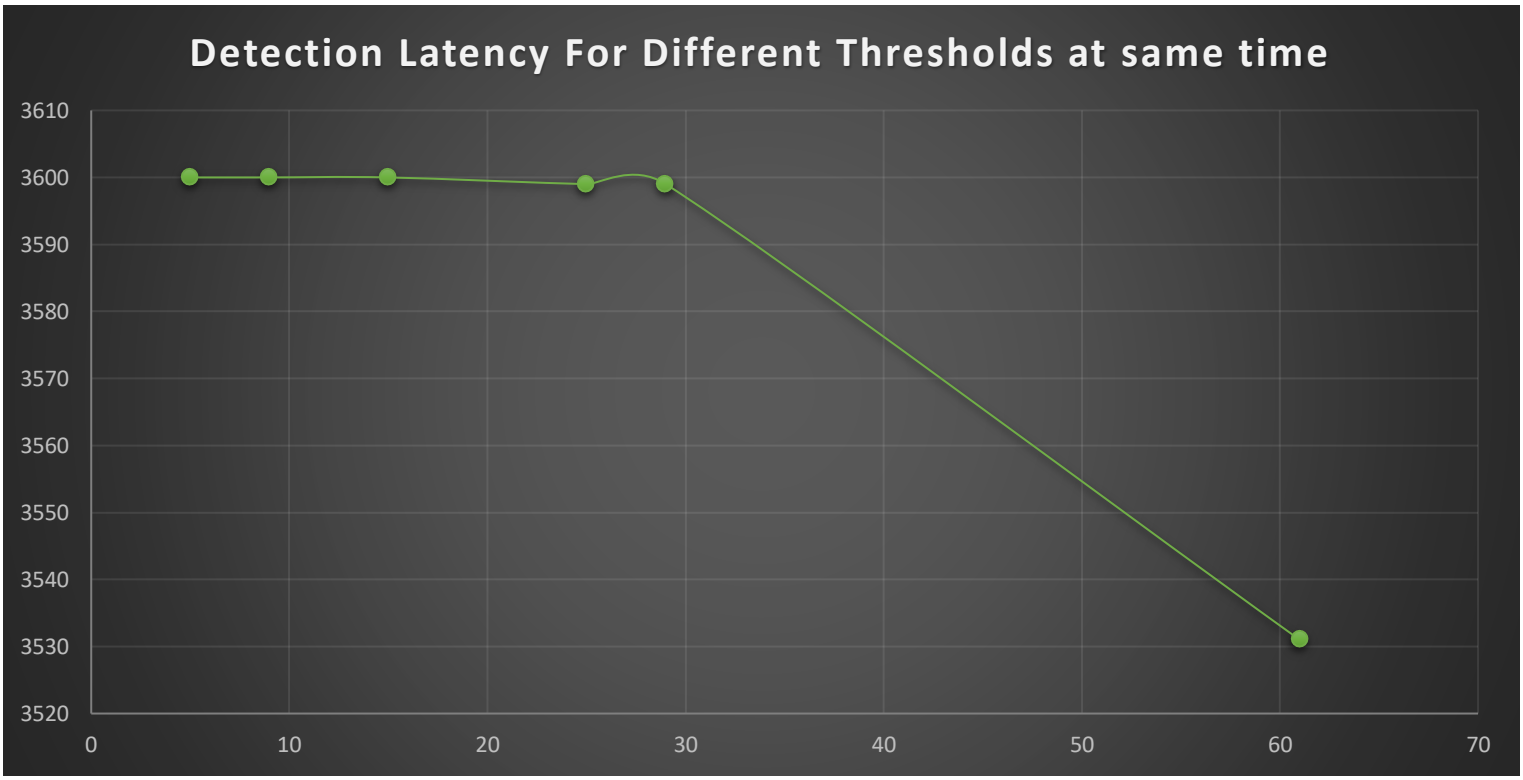
Rule 2:
alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule 2";flow:stateless;flags:S;detection_filter:track by_src,count 15, seconds 1 ;sid: 100002;rev:1;)

Detection Latency with Different Thresholds

The table shows the time between the actual start of the attack vs start time identified by the program at sampling period 1s. All times are in Eastern Time (EST).

Rule 1

Threshold	Time Identified by the program	Actual Time of the attack	Detection Latency(s)
5	14:10:26 = 51026s	15:10:26 = 54626s	3600
9	14:10:26 = 51026s	15:10:26 = 54626s	3600
15	14:10:26 = 51026s	15:10:26 = 54626s	3600
25	14:10:27 = 51027s	15:10:26 = 54626s	3599
29	14:10:27 = 51027s	15:10:26 = 54626s	3599
61	14:11:35 = 51095s	15:10:26 = 54626s	3531

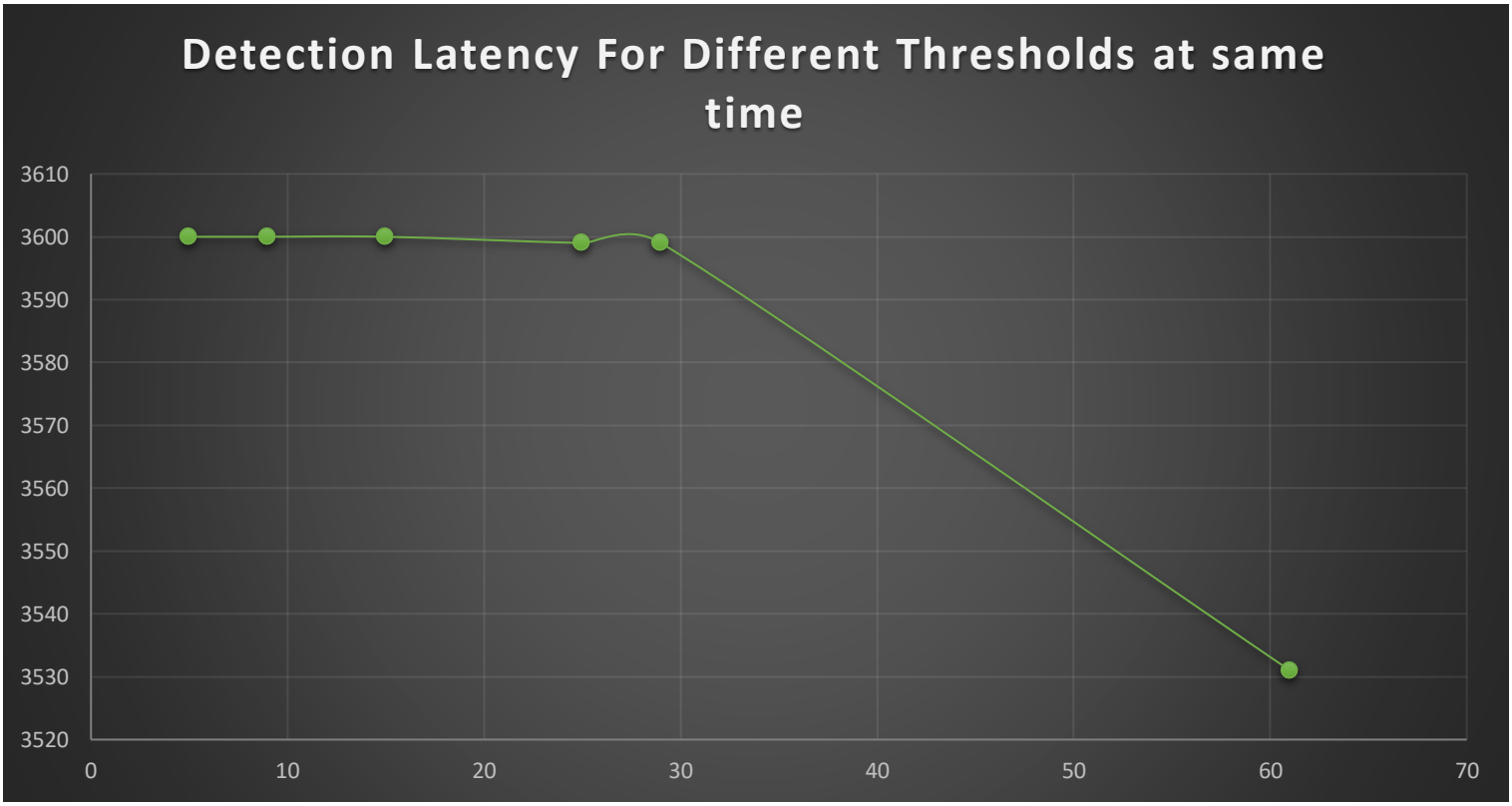


Graph 3.1

In this graph the Thresholds are on the X-axis while the Detection latency(Calculated in Seconds) is on the Y-Axis.
We notice that as the threshold increases the Detection Latency decreases which means that with higher thresholds our attack detection time of our program is also delayed.

Rule 2

Threshold	Time Identified by the program	Actual Time of the attack	Detection Latency(s)
5	14:10:26 = 51026s	15:10:26 = 54626s	3600
9	14:10:26 = 51026s	15:10:26 = 54626s	3600
15	14:10:26 = 51026s	15:10:26 = 54626s	3600
25	14:10:27 = 51027s	15:10:26 = 54626s	3599
29	14:10:27 = 51027s	15:10:26 = 54626s	3599
61	14:11:35 = 51095s	15:10:26 = 54626s	3531



Graph 3.2

In this graph the Thresholds are on the X-axis while the Detection latency(Calculated in Seconds) is on the Y-Axis.
We notice that as there is a significant increase in the threshold the Detection Latency decreases which means that with higher thresholds our attack detection time of our program is also delayed.

Comparing Rule 1 and 2

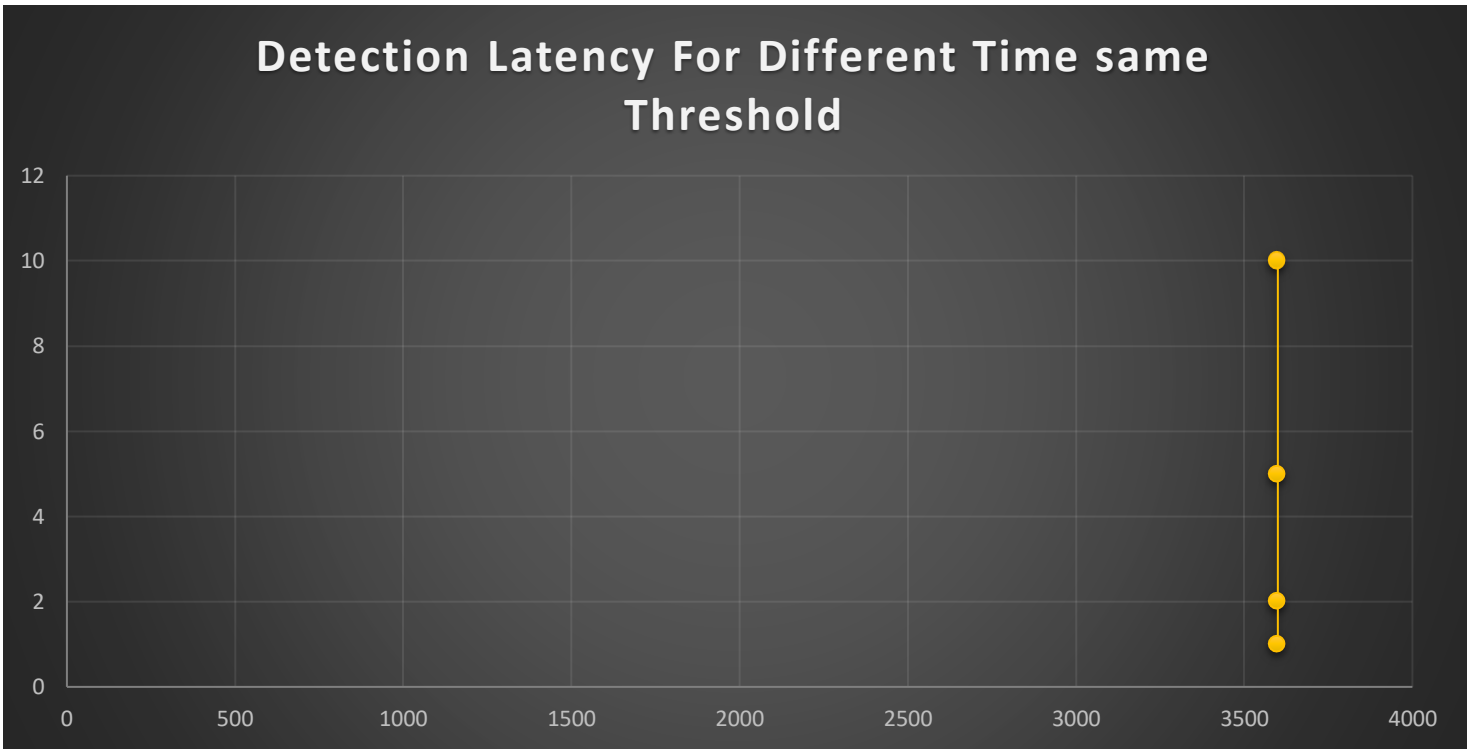
Through the tables and Graph 3.1 and 3.2, it is evident that type of the rule used has no effect on the Detection Latency of the program, i.e., the time taken by the program to detect the attack by Rule 1 or Rule 2 is the same.

Detection Latency with Different Time period

Through the ROC analysis it was concluded that Threshold of 15 at any chosen Sampling period was ideal to detect the attack.
The table shows the time between the actual start of the attack vs start time identified by the program at different sampling periods at Threshold 15. All times are in Eastern Time (EST).

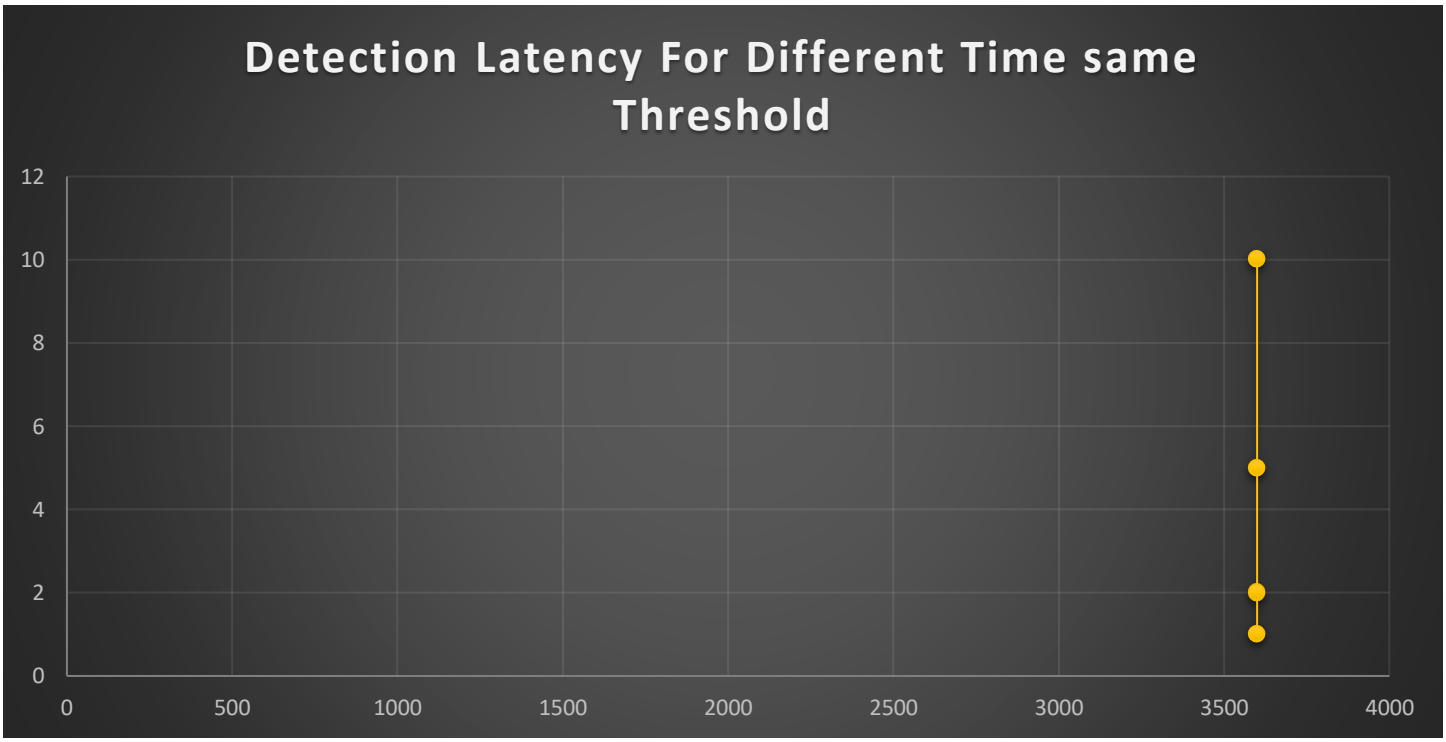
Rule 1

Time	Time Identified by the program	Actual Time of the attack	Detection Latency(s)
1	14:10:26 = 51026s	15:10:26 = 54626s	3600
2	14:10:26 = 51026s	15:10:26 = 54626s	3600
5	14:10:26 = 51026s	15:10:26 = 54626s	3600
10	14:10:26 = 51026s	15:10:26 = 54626s	3600



In this graph the Sampling time are on the X-axis while the Detection latency(Calculated in Seconds) is on the Y- Axis.
We notice that as there is no change in the detection latency if that sampling period is the different for same Threshold

Time	Time Identified by the program	Actual Time of the attack	Detection Latency(s)
1	14:10:26 = 51026s	15:10:26 = 54626s	3600
2	14:10:26 = 51026s	15:10:26 = 54626s	3600
5	14:10:26 = 51026s	15:10:26 = 54626s	3600
10	14:10:26 = 51026s	15:10:26 = 54626s	3600



In this graph the Sampling time are on the X-axis while the Detection latency(Calculated in Seconds) is on the Y- Axis.
We notice that as there is no change in the detection latency if that sampling period is the different for same Threshold

APENDIX:

All Rules used in this Project

```
****************************  SIGNATURE 1 ****************************
#         TIME(p) = 1s
#
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 5, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 9, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 20, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 21, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 25, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 28, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 29, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 60, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 61, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 62, seconds 1;sid: 100002;rev:1;)

#         Time(p) = 2s
#
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 5, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 9, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 2;sid: 100002;rev:1;)
```

```
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 20, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 21, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 25, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 28, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 29, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 60, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 61, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 62, seconds 2;sid: 100002;rev:1;)


#       Time(p) = 5s
#
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 5, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 9, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 20, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 21, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 25, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 28, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 29, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 60, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 61, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 62, seconds 5;sid: 100002;rev:1;)
#

#       Time(p) = 10s
#
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 5, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 9, seconds 10;sid: 100002;rev:1;)
```

```
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 15, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 20, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 21, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 25, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 28, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 29, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 60, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 61, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
1";flow:stateless;flags:S;detection_filter:track by_dst,count 62, seconds 10;sid: 100002;rev:1;)


****************************** SIGNATURE 2 ******************************

#       TIME(p) = 1s

#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 5, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 9, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 15, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 20, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 21, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 25, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 28, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 29, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 60, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 61, seconds 1;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 62, seconds 1;sid: 100002;rev:1;)

#       Time(p) = 2s
#
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 5, seconds 2;sid: 100002;rev:1;)
```

```
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 9, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 15, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 20, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 21, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 25, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 28, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 29, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 60, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 61, seconds 2;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 62, seconds 2;sid: 100002;rev:1;)


#       Time(p) = 5s
#
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 5, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 9, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 15, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 20, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 21, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 25, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 28, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 29, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 60, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 61, seconds 5;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 62, seconds 5;sid: 100002;rev:1;)
#

#       Time(p) = 10s
#
```

```
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 5, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 9, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 15, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 20, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 21, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 25, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 28, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 29, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 60, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 61, seconds 10;sid: 100002;rev:1;)
#alert tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Detected Rule
2";flow:stateless;flags:S;detection_filter:track by_src,count 62, seconds 10;sid: 100002;rev:1;)
```