519 - Network Security - Term Project Phase 2

e2237824 - Ahmet Kürşad Şaşmaz

1 ICMP Sequence Number as Covert Channel

The problem is using sequence numbers in ICMP packets as covert channel. In order not to be detected, our design should provide *normal-like behavior* of any ICMP packet sequence (which is normally sent by **ping** command).

1.1 Requirements

- Ping flooding is prohibited.
- Ping sequence numbers must in the increasing order.
- Due to possibility of monitoring by MITM at any level of routing, not sending any packet in the sequence is not possible.
- For successfully replied sequences, it is abnormal to wait until high sequence numbers.
- Consecutive **ping** commands shouldn't give any information about covert channel, so maximum sequence number in a group of **ping** session must be pseudo-random.

1.2 Design

1.2.1 Sending Bit 0

N : Number of packets to sent

K : Maximum value of k

k : A uniform random value

$N = 2k, k > 1, k < K$

- ping seq = 1, do not care success
- ping seq = 2, do not care success
- ping seq = 3, if fails and N = 4, then N $\leftarrow$ N+2

- ping seq = 4, if N = 4 and fails, then N ← N+2, else halt.
- ping seq = 5, if fails and N = 6, then N ← N+2
- ping seq = 6, if N = 6 and fails, then N ← N+2, else halt.
- ...

1.2.2 Sending Bit 1

N : Number of packets to sent

K : Maximum value of k

k : A uniform random value

$N = 2k + 1, k \geq 1, k < K$

- ping seq = 1, do not care success
- ping seq = 2, if fails and N = 3, then N ← N+2
- ping seq = 3, if N = 3 and fails, then N ← N+2, else halt.
- ping seq = 4, if fails and N = 5, then N ← N+2
- ping seq = 5, if N = 5 and fails, then N ← N+2, else halt.
- ...

1.2.3 Sending Empty Information

$N = 1|2$

- do:
- ping seq = 1, if N=1, halt
- ping seq = 2, if N=2, halt
- while: at least one of the packet successes;

1.2.4 Receiving

- If source IP does not exist, create record for session
- If latest maximum sequence number is greater than current ICMP sequence number:
- If latest maximum sequence number is greater or equal than 3:
- Add element (latest maximum sequence number mod(2)) to bit array

- latest maximum sequence number ← current ICMP sequence number
- Else if latest maximum sequence number is less than current ICMP sequence number:
- latest maximum sequence number ← current ICMP sequence number

2 Implementation

2.1 Sender

The original repository of **ping** from **iputils** is updated for covert channel.

2.2 Receiver

The command below prints all ICMP echo packets arrived.

tcpdump -i eth0 icmp and icmp[icmptype]=icmp-echo -n -l

Above command piped to our python script that parses the output of the command.

It parses and creates session if needed. Then it parses bit and adds it to the array, writes to file if file output is chosen.

3 Experimentation

3.1 Scenarios

- Send bit-0 N times
- Send bit-1 N times
- Send k random bits N times

3.2 Design & Implementation

A manager experimentation bash script is written to control sequence of starts and stops of docker scripts and gathering and comparing the results.

3.3 Parsing & Plotting

Parsing is done by human hands using Visual Code shortcuts. (Sometimes faster than parsing text file in Python). Plotting is done by online Python plotter.

4 Results

4.1 Scenario 1 : Send bit-0 N times

	TX	RX
Total	780	596
Mean	7.8	5.96
Std	2.04	1.61
%95	0.4	0.32

4.2 Scenario 2 : Send bit-1 N times

	TX	RX
Total	854	634
Mean	8.54	6.34
Std	2.77	1.72
%95	0.54	0.37

4.3 Scenario 3 : Send k random bits N times

NaN values will be added soon.

	TX 2	RX 2	TX 4	RX 4	TX 8	RX 8
Total	1372	1032	2639	1975	-	-
Mean	13.72	10.32	26.39	19.75	-	-
Std	2.85	1.89	4.70	3.13	-	-
%95	0.56	0.37	0.92	0.61	-	-

5 Conclusions

- Drop probability of **MITM** is 0.05. So that probability of arrival of reply packet is 0.95*0.95 = **0.9025**. Expected received packet count is one tenth of transmitted packet count. We didn't achieved this expectancy so well.
- All the data is transmitted correctly even in the packet drop scenario.
- It looks like sending k bits requires approximately 7.5 transmitted packets.