Háskólinn í Reykjavík
Tölvusamskipti

6. Október 2024
T-409-TSAM

Assignment 5:

The Botnet Saves the World

Nemendur:
Ágúst Máni Þorsteinsson
Hermann Helgi Þrastarson

Kennarar:
Stephan Schiffel

Assignment

This is our project report for Assignment 5: The Botnet Saves the World and will go over all our points for grading. As stated in the README.md file, all references to files can be found within our SavedFiles folder within the .Zip.

Wireshark trace

For this part of the assignment, we have added a file called Wireshark_trace.pcapng which, when opened with wireshark, will show a wireshark trace between our client and server demonstrating every command implemented between the client and server.

Connected to Instructor server

Here are the logs of us jumpstarting the server by using our CONNECTSERVER command from our client. We connect to the instructor and take in the SERVERS command. From there we ask for a list from the instructor and document the servers listed. This can be found in ExampleLog.txt.

```
[2024-10-28 14:12:11] // CLIENT // New command from Client: 4
[2024-10-28 14:12:11] // CLIENT // Attempting to connect to server specified by client.
[2024-10-28 14:12:11] // CONNECT // New server connected on: 130.208.246.249 : 5
[2024-10-28 14:12:11] // SENDING // SOHHELO,A5_23EOT
[2024-10-28 14:12:11] // CLIENT // Successfully connected to specified server.
[2024-10-28 14:12:11] // SENDING // Successfully connected to specified server.
[2024-10-28 14:12:11] // MESSAGE // New message received from:  : 5
[2024-10-28 14:12:11] // COMMAND // New command from Server:  : 5
[2024-10-28 14:12:11] // COMMAND // SOHHELO,Instr_1EOTSOHSERVERS,Instr_1,130.208.246.249,5001;Instr_2,130.208.246.249,5
[2024-10-28 14:12:11] // COMMAND // New command: HELO
[2024-10-28 14:12:11] // COMMAND // HELO detected. Taking in data.
[2024-10-28 14:12:11] // COMMAND // Group Instr_1 has been tied to: 130.208.246.249
[2024-10-28 14:12:11] // SENDING // SOHSERVERS,A5_23,130.208.246.249,4123;Instr_1,130.208.246.249,5001;EOT
[2024-10-28 14:12:11] // COMMAND // Succeeded in sending list of servers
[2024-10-28 14:12:11] // COMMAND // New command: SERVERS
[2024-10-28 14:12:11] // COMMAND // SERVERS detected. Taking in data.
[2024-10-28 14:12:11] // COMMAND // New server: Instr_1 IP: 130.208.246.249 Port: 5001 documented.
[2024-10-28 14:12:11] // COMMAND // New server: Instr_2 IP: 130.208.246.249 Port: 5002 documented.
[2024-10-28 14:12:11] // COMMAND // New server: A5_28 IP: 172.31.198.141 Port: 4005 documented.
[2024-10-28 14:12:11] // COMMAND // New server: A5_3 IP: 130.208.246.249 Port: 4003 documented.
[2024-10-28 14:12:11] // COMMAND // New server: NUMBER IP: 130.208.246.249 Port: 5005 documented.
[2024-10-28 14:12:11] // COMMAND // New server: A5_2 IP: 130.208.246.249 Port: 4002 documented.
[2024-10-28 14:12:11] // COMMAND // New server: A5_9 IP: 130.208.246.249 Port: 4009 documented.
[2024-10-28 14:12:11] // COMMAND // New server: A5_34 IP: 130.208.246.249 Port: 4034 documented.
[2024-10-28 14:12:11] // COMMAND // Finished commands list.
```

Send Messages

Here are logs showing us sending a message from us to other servers. This can be found in ExampleLog.txt. One message is sent to the Instr_1 server and one message to group A5_3.

Instr_1:

```
SENDMSG,Instr_1,Hi there! this is for the assignment.
Sending message: SENDMSG,Instr_1,Hi there! this is for the assignment.
Waiting for response.
Message received, size: 25
[2024-10-28 14:12:44] Message is in the botnet.
```

```
130   [2024-10-28 14:12:38] // MESSAGE // New message received from: CLIENT : 4
131   [2024-10-28 14:12:38] // CLIENT // New command from Client: 4
132   [2024-10-28 14:12:38] // COMMAND // Attempting to send a message
133   [2024-10-28 14:12:38] // COMMAND // Connected to group: Instr_1 attempting to send message
134   [2024-10-28 14:12:38] // SENDING // SOHSENDMSG,Instr_1,A5_23,Hi there! this is for the assignment.EOT
135   [2024-10-28 14:12:38] // COMMAND // Succeeded sending message to group: Instr_1
136   [2024-10-28 14:12:38] // SENDING // Message is in the botnet.
```

A5_3:

```
SENDMSG,A5_3,Hihi! this is group A5_23.
Sending message: SENDMSG,A5_3,Hihi! this is group A5_23.
Waiting for response.
Message received, size: 25
[2024-10-28 14:13:52] Message is in the botnet.
```

```
180   [2024-10-28 14:13:46] // MESSAGE // New message received from: CLIENT : 4
181   [2024-10-28 14:13:46] // CLIENT // New command from Client: 4
182   [2024-10-28 14:13:46] // COMMAND // Attempting to send a message
183   [2024-10-28 14:13:46] // COMMAND // Connected to group: A5_3 attempting to send message
184   [2024-10-28 14:13:46] // SENDING // SOHSENDMSG,A5_3,A5_23,Hihi! this is group A5_23.EOT
185   [2024-10-28 14:13:46] // COMMAND // Succeeded sending message to group: A5_3
186   [2024-10-28 14:13:46] // SENDING // Message is in the botnet.
```

Receiving messages

Here are logs showing us receiving messages to our server. This can be found in ExampleLog.txt. One message from group A5_3, and one from group A5_14.

We start by running the MESSAGEBUFFER command in our client to see who has messages to us. We then run GETMSG to see what message A5_3 has for us.

```
MESSAGEBUFFER
Sending message: MESSAGEBUFFER
Waiting for response.
Message received, size: 41
[2024-10-28 14:21:34] MESSAGEBUFFER, A5_2, 3, A5_3, 1, A5_53, 1
GETMSG,A5_3
Sending message: GETMSG,A5_3
Waiting for response.
Message received, size: 18
[2024-10-28 14:21:50] Hello from A5_3!!!
```

```
77  [2024-10-28 14:12:11] // MESSAGE // New message received from:  : 8
78  [2024-10-28 14:12:11] // COMMAND // New command from Server:  : 8
79  [2024-10-28 14:12:11] // COMMAND // SOHHELO,A5_3EOTSOHSENDMSG,A5_23,A5_3,Hello from A5_3!!!EOTSOHSERVERS,A5_3,130.208.246.
80  [2024-10-28 14:12:11] // COMMAND // New command: HELO
81  [2024-10-28 14:12:11] // COMMAND // HELO detected. Taking in data.
82  [2024-10-28 14:12:11] // COMMAND // Group A5_3 has been tied to: 130.208.246.249
83  [2024-10-28 14:12:11] // SENDING // SOHSERVERS,A5_23,130.208.246.249,4123;A5_2,130.208.246.249,4002;A5_3,130.208.246.
84  [2024-10-28 14:12:11] // COMMAND // Succeeded in sending list of servers
85  [2024-10-28 14:12:11] // COMMAND // New command: SENDMSG
86  [2024-10-28 14:12:11] // COMMAND // SENDMSG detected. Sending data
87  [2024-10-28 14:12:11] // COMMAND // SENDMSG has correct amount of variables, attempting to send message.
88  [2024-10-28 14:12:11] // COMMAND // Message is addressed to us. Storing message.
89  [2024-10-28 14:12:11] // COMMAND // New command: SERVERS
90  [2024-10-28 14:12:11] // COMMAND // SERVERS detected. Taking in data.
```

Receiving a message from group A5_14:

```
1518  [2024-10-28 14:25:41] // MESSAGE // New message received from: A5_14 : 12
1519  [2024-10-28 14:25:41] // COMMAND // New command from Server: A5_14 : 12
1520  [2024-10-28 14:25:41] // COMMAND // SOHSENDMSG,A5_23,A5_14,Hello back!EOT
1521  [2024-10-28 14:25:41] // COMMAND // New command: SENDMSG
1522  [2024-10-28 14:25:41] // COMMAND // SENDMSG detected. Sending data
1523  [2024-10-28 14:25:41] // COMMAND // SENDMSG has correct amount of variables, attempting to send message.
1524  [2024-10-28 14:25:41] // COMMAND // Message is addressed to us. Storing message.
```

Running GETMSG on A5_14:

```
1663  [2024-10-28 14:26:17] // MESSAGE // New message received from: CLIENT : 4
1664  [2024-10-28 14:26:17] // CLIENT // New command from Client: 4
1665  [2024-10-28 14:26:17] // COMMAND // GETMSG is correctly formated. Checking for messages
1666  [2024-10-28 14:26:17] // CLIENT // Group: A5_14 Has a messages for client. Responding to client
1667  [2024-10-28 14:26:17] // SENDING // Hello back!
1668  [2024-10-28 14:26:17] // CLIENT // Group: A5_14 has a message for client. Replied to Client Suceeded.
```

```
GETMSG,A5_14
Sending message: GETMSG,A5_14
Waiting for response.
Message received, size: 11
[2024-10-28 14:26:23] Hello back!
```

```
SENDMSG,A5_14,Hihi from group A5_23!
Sending message: SENDMSG,A5_14,Hihi from group A5_23!
Waiting for response.
Message received, size: 25
[2024-10-28 14:25:06] Message is in the botnet.
GETMSG,A5_14
Sending message: GETMSG,A5_14
Waiting for response.
Message received, size: 11
[2024-10-28 14:26:23] Hello back!
```

Bonus Points

The security issues of the botnet.

There are a couple of issues that come up with the botnet. First of which is that anyone can pretend to be anyone. No bot has any way of being sure that the group they are connecting to is the actual group. It could simply be someone running their own server and calling themselves, for example A5_23, whilst not being the actual group. With that said, bots could then be sending sensitive data to this fake A5_23 server, which is supposed to be intended for the real group 23.

Another issue is the concept of data scraping and hoarding. Any server on the botnet can message to all other connected bots STATUSREQ, and then call GETMSG for every single message each server has. This allows them to gain access to literally any message they can get their hands on, potentially revealing sensitive data or information intended for other groups. An additional concern in a similar vein is the possibility for a malicious bot to call GETMSG on any other bots messages, and change the data or information as they wish. This can have severe consequences as any malicious attacker can perform a ‘man-in-the-middle’ attack onto literally any connections that are not directly between two other bots. This is only exacerbated by the most obvious issue, none of the messages are encoded.

All messages are just raw data. Anyone can listen to what is being sent or received to anyone and see the messages. Making scraping and collecting data even easier. They don’t even need to be on the botnet to do this.

With all these points in mind, the botnet is incredibly insecure. It practically demands that all bots play nicely on the botnet and not do anything malicious. Even when all bots are playing nice, this doesn’t mean the botnet could be secure or work as intended either, since one glitched bot which can only receive messages and not forward them to another bot could easily hoard any messages sent through it by accident.

Potential Malicious Attack

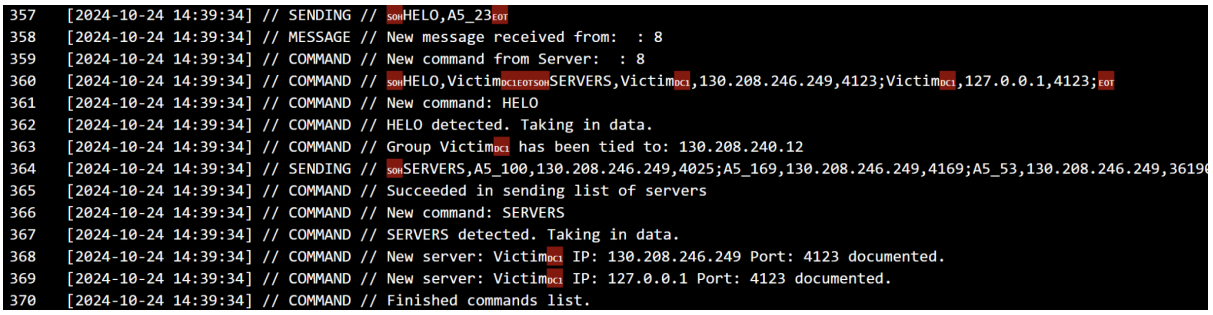
Within the saved files folder is a log file called 'MaliciousAttack.txt' which shows the communication log with what we believe to be an “enemy server”.
The log shows how the enemy server is trying to perform a DDOS on our server by sending a long line of repeating SOH and EOT within the HELO command. Our server handled this, as our server immediately stops computing any additional commands from a server once it fails once, but this could potentially force a bot into performing hundreds of operations with the intent of slowing it down. We named this bot A5_. After this, a new server attempts to connect to us within a very short timespan, which we call VictimDC1. This server maliciously sends in its servers list two servers with its own name, but with our IP address and port instead of its own information. This could potentially pass itself off as our socket, try to make us connect to ourselves, or could potentially make us disconnect from our own listening socket, which is what happens in the end. This could lead to major bugs and errors depending on the code and bot implementation, and thus we think it's a malicious attack.

This attack is performed twice, exactly the same in both cases. The reason why we think these instances are the same people is because when VictimDC1 and the unnamed A5_ disconnect, they both do it at practically the exact same time in both instances.

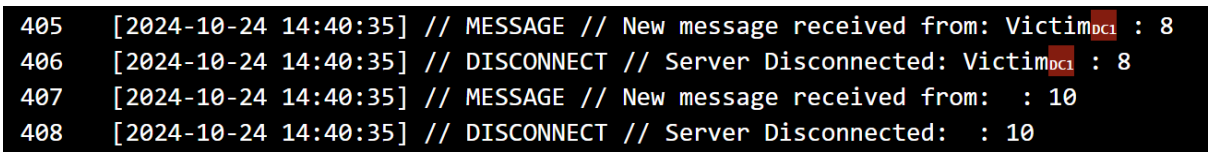
This attack actually caused our server to attempt to close our own listen socket after not receiving any keepalives from it. Effectively ruining our own socket. We have since fixed this issue and don't allow others to connect to us if they are trying to pretend to be us.
This can be seen in the MaliciousAttack logs where servers start to disconnect from us slowly one after another. At the end of the log, the final server disconnects and our server effectively freezes and nothing more happens even after we waited for 20+ minutes. Below are screenshots of the log referencing the attack.

The first attack. A5_ attempts to connect and sends a long list of empty commands.

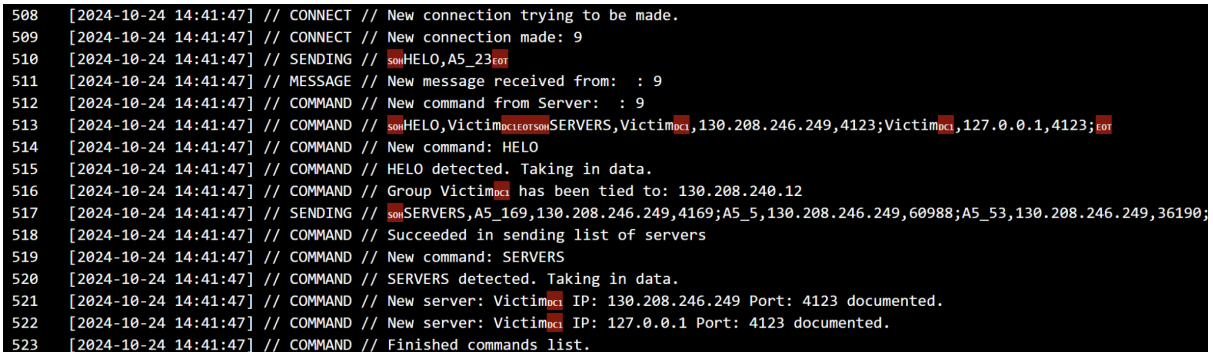
VictimDC1 connects for the first time. The victim server sends a HELO command.
Here the attack succeeds, we detect new servers that the victim is connected to which is our own server.



Both servers disconnect at the same time.



The Victim server reconnects.



The A5_ reconnects and attempts the same attack again.

Servers start to slowly disconnect

Here we are possibly connected to ourselves as VictimDC1 and we effectively ruin our socket by disconnecting.

Here is the final disconnect. After this our server freezes and nothing more happens after 20+ minutes of waiting.