Malware Forensics (MalFor)

Coursework

REPORT

Student No.: UP2009045
Word Count: 2042

Disclaimer
By submitting this report for marking, **I fully understand** that and **I agree**:
- This is an independent piece of coursework, and it is expected that I have taken responsibility for all the design, implementation, analysis of results and writing of the report. And, it is 2000 words (+/-10%).
- For the Turnitin plagiarism software, the match score of the report must be below 15% overall and less than 5% to an individual source (excluding appendices). Marks will be investigated accordingly. if appropriate.
- This marking scheme sets out what would be expected for each marking band for this module's coursework.
- The University "general criteria applicable to essays, reports and aspects of projects and dissertations" applies (on Moodle).
- It is blind marked.
- Any technical help, high-level advice and suggestions which may be provided in-person (e.g., labs) and electronically, has no contribution to or an indication of my final mark.
- Knowledge of a peer's work and their final mark is not justification for what my own mark should be.
- Any examples provided were used for ideas only, and "having followed an example" is not justification for what my mark should be.
- The coursework malware was used for analysis only and was isolated to a safe working
environment (a virtual machine), to prevent the malware from infecting the host.

Summary

Longtext who have been asked to investigate data exfiltration regarding Pavel Checkov and core IP. Longtext have identified a single server in relation to the malware which was executed by the user along with various commands. The server identified is a command and control (C2) server for an internet relay chat (IRC) botnet. Once the malware is executed on the system, it seems as if it is no longer running however, continues to run in the background; for a standard system user this would be hard to spot and report.

Further investigation revealed the C2 server was communicating over a default port (6667) for IRC traffic and naming conventions were also suspicious with names being ‘bot’ followed by random numbers. Following on from this, a ‘#malfor-woods’ channel was found which is password protected and is where the bot-herder conducts commands.

Two passwords were discovered during both static and dynamic analysis, ‘fancybear’ a static un-encrypted password assigned to each bot which provided us with access to the IRC server and ‘richmond’ which provided access to the channel for controlling the botnet.

During further analysis and debugging using IDA, we found there were various commands expecting to be received by the malware listed in Table 1. During creation, the author, who we believe to be Julian Murphy also implemented anti-debugging techniques to prevent disassembly of this executable using programs such as IDA.

Finally, we have included recommendations for remediation along with future prevention within the body of this report.

Introduction

Following a recent security incident involving a Gazprom employee exfiltrating company plans for a new gas pipeline, it has been revealed that this employee infected company systems with malware which requires further investigation. Gazprom has provided Longtext with the malware in question for us to conduct the investigation of which the details, steps and the outcome will be documented here.

This malware analysis will review how the malware functionality and effectiveness in exfiltrating said information via static and dynamic analysis, analysis of any associated infrastructure and dis-assembly of the malware functionality.

Following the analysis of the malware, we Longtext will provide some remedial steps which can be used to remove this risk and recommendations to prevent this risk in future.

Malware Analysis

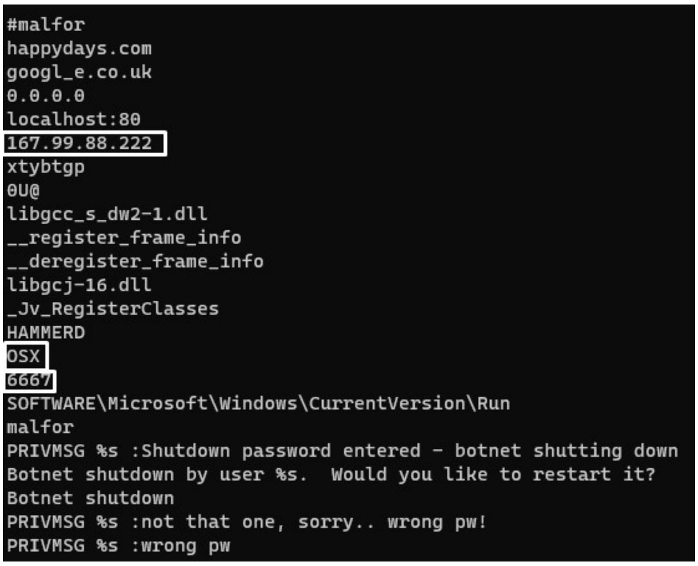
Static analysis

Beginning with static analysis, we firstly analysed the malware file provided by Gazprom using a combination of:

- Strings - This gathers whole text and numerical information from the malware files.
- PEiD – This allows us to identify if the file is packed or not along with displaying any dynamic link libraries (DLL) it imports/exports.
- PEView – Allows for analysis of the file structure viewing file properties, headers etc.
  VirusTotal – Usage of 72 well known and accurate malware engines to scan and provide information on the file along with gathering information such as previous community comments on the file.

This is done without executing the malware, however, is still conducted within a safe environment such as a virtual machine where the malware would not affect the host and external domain.

Strings



```
#malfor
happydays.com
googl_e.co.uk
0.0.0.0
localhost:80
167.99.88.222
xtybtgp
0U@
libgcc_s_dw2-1.dll
__register_frame_info
__deregister_frame_info
libgcj-16.dll
_Jv_RegisterClasses
HAMMERD
OSX
6667
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
malfor
PRIVMSG %s :Shutdown password entered - botnet shutting down
Botnet shutdown by user %s.  Would you like to restart it?
Botnet shutdown
PRIVMSG %s :not that one, sorry.. wrong pw!
PRIVMSG %s :wrong pw
```

Figure 1 - Strings output identifying IP address, operating system and ports 80 & 6667

Beginning with Strings, Figure 1 displays the output after running strings on the malware file Gazprom provided. Looking over this output, we noticed a few pieces of information:

Firstly, the block of text at the bottom of Figure 1 identifies an internet relay chat botnet which would go hand in hand with “6667”. This is the default port for IRC traffic (Wireshark, 2020), we can also see a public IP address 167.99.88.222 and a potential operating system OSX, which could mean this is being hosted on a Mac and using a bash terminal.

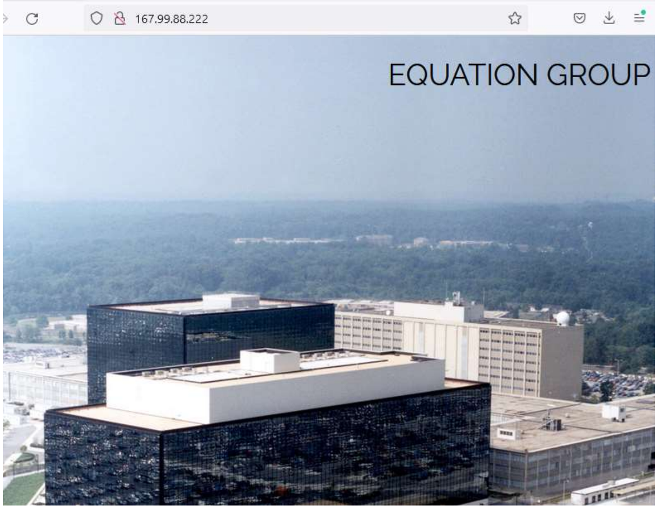
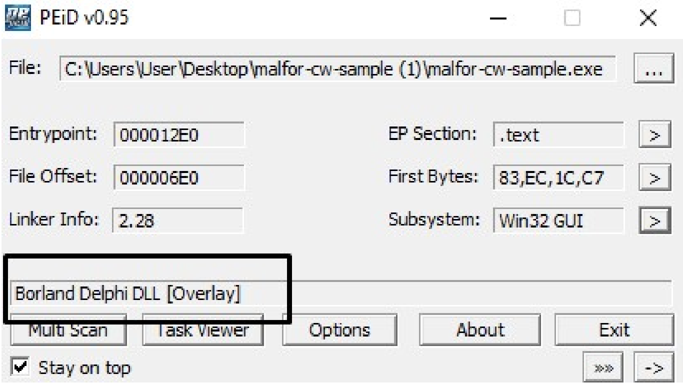


Figure 2 - Equation Group website linked to discovered IP

As displayed in Figure 2, after visiting the IP address found in the malware this displayed a standard webpage with "EQUATION GROUP". Further research into this group shows the threat actor group has an expansive past in cyber attacks (Checkpoint Research Team, 2015); further analysis did not reveal anything, for example looking up robots.txt led to a link which would not load. However, here we can see the site does not have SSL which explains the localhost:80 found previously in strings.

PEiD



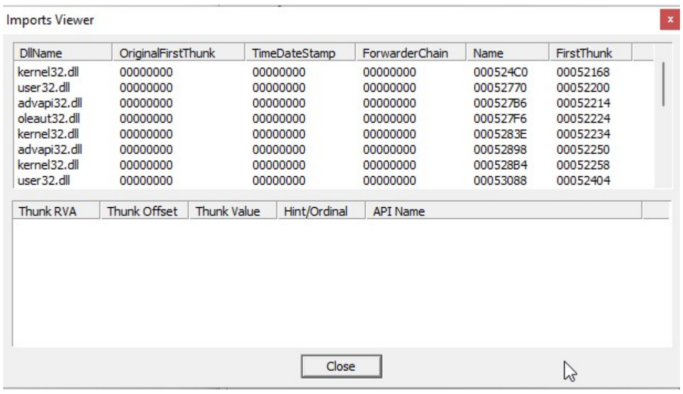
File: C:\Users\User\Desktop\malfor-cw-sample (1)\malfor-cw-sample.exe

Entrypoint: 000012E0
EP Section: .text
File Offset: 000006E0
First Bytes: 83,EC,1C,C7
Linker Info: 2.28
Subsystem: Win32 GUI

Borland Delphi DLL [Overlay]

Figure 3 - PEiD output displaying unpacked .exe file

Moving over to PEiD, after performing a hardcore scan on the malware file provided was built in Boreland Delphi shown in Figure 3. Delphi is no longer a commonly used program and although this file uses Delphi, the file was never packed during creation.



Imports Viewer

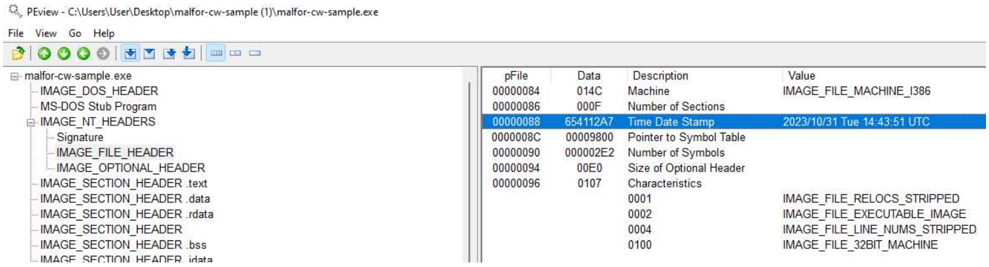
DllName	OriginalFirstThunk	TimeDateStamp	ForwarderChain	Name	FirstThunk
kernel32.dll	00000000	00000000	00000000	000524C0	00052168
user32.dll	00000000	00000000	00000000	00052770	00052200
advapi32.dll	00000000	00000000	00000000	000527B6	00052214
oleaut32.dll	00000000	00000000	00000000	000527F6	00052224
kernel32.dll	00000000	00000000	00000000	0005283E	00052234
advapi32.dll	00000000	00000000	00000000	00052898	00052250
kernel32.dll	00000000	00000000	00000000	000528B4	00052258
user32.dll	00000000	00000000	00000000	00053088	00052404

Thunk RVA	Thunk Offset	Thunk Value	Hint/Ordinal	API Name

Figure 4 – List of imports

The hardcore scan also reveals a list of DLLs that are used within the malicious executable and reviewing these does not reveal any DLL in relation with the previously found botnet.

PEView



PEview - C:\Users\User\Desktop\malfor-cw-sample (1)\malfor-cw-sample.exe

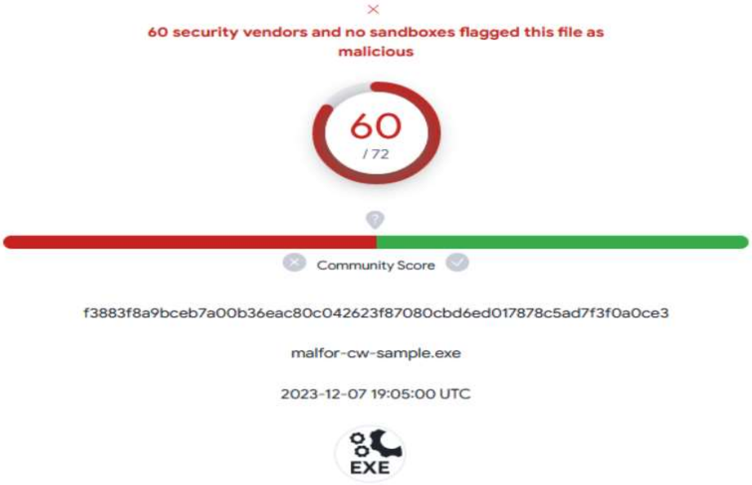
```
malfor-cw-sample.exe
  IMAGE_DOS_HEADER
  MS-DOS Stub Program
  IMAGE_NT_HEADERS
    Signature
    IMAGE_FILE_HEADER
    IMAGE_OPTIONAL_HEADER
  IMAGE_SECTION_HEADER .text
  IMAGE_SECTION_HEADER .data
  IMAGE_SECTION_HEADER .rdata
  IMAGE_SECTION_HEADER
  IMAGE_SECTION_HEADER .bss
  IMAGE_SECTION_HEADER .idata
```

pFile	Data	Description	Value
00000084	014C	Machine	IMAGE_FILE_MACHINE_I386
00000086	000F	Number of Sections	
00000088	654112A7	Time Date Stamp	2023/10/31 Tue 14:43:51 UTC
0000008C	00009800	Pointer to Symbol Table	
00000090	000002E2	Number of Symbols	
00000094	00E0	Size of Optional Header	
00000096	0107	Characteristics	
		0001	IMAGE_FILE_RELOCS_STRIPPED
		0002	IMAGE_FILE_EXECUTABLE_IMAGE
		0004	IMAGE_FILE_LINE_NUMS_STRIPPED
		0100	IMAGE_FILE_32BIT_MACHINE

Figure 5 - File header information

Reviewing the malware file header in PEView shows when the malware was created, in this case October 31st 2023 and was intended for 32-bit systems, however, this could be to maximise the attack surface as 32-bit programs will work on 64-bit systems but not the other way around.

VirusTotal



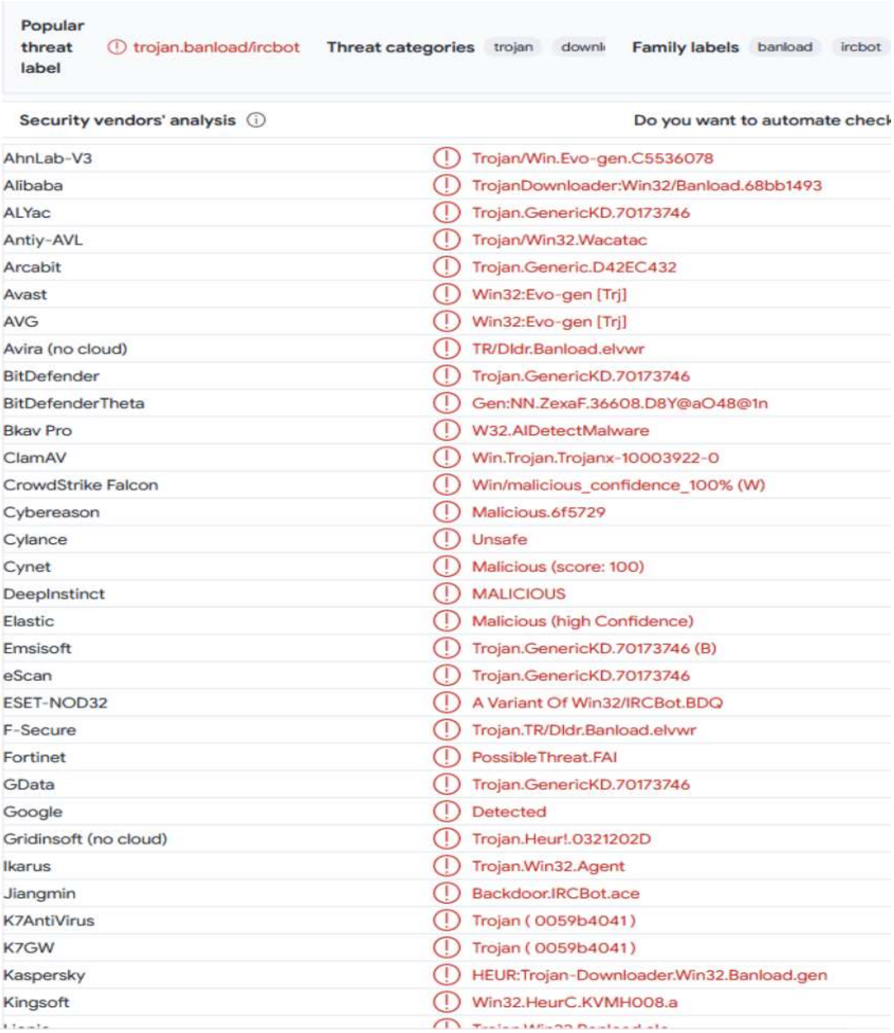
60 security vendors and no sandboxes flagged this file as malicious

f3883f8a9bceb7a00b36eac80c042623f87080cbd6ed017878c5ad7f3f0a0ce3

malfor-cw-sample.exe

2023-12-07 19:05:00 UTC

Figure 6 - VirusTotal scan result



Popular threat label trojan.banload/ircbot Threat categories trojan downl Family labels banload ircbot

Security vendors' analysis Do you want to automate check

AhnLab-V3	Trojan/Win.Evo-gen.C5536078
Alibaba	TrojanDownloader:Win32/Banload.68bb1493
ALYac	Trojan.GenericKD.70173746
Antiy-AVL	Trojan/Win32.Wacatac
Arcabit	Trojan.Generic.D42EC432
Avast	Win32:Evo-gen [Trj]
AVG	Win32:Evo-gen [Trj]
Avira (no cloud)	TR/Dldr.Banload.elvwr
BitDefender	Trojan.GenericKD.70173746
BitDefenderTheta	Gen:NN.ZexaF.36608.D8Y@aO48@1n
Bkav Pro	W32.AIDetectMalware
ClamAV	Win.Trojan.Trojanx-10003922-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)
Cybereason	Malicious.6f5729
Cylance	Unsafe
Cynet	Malicious (score: 100)
DeepInstinct	MALICIOUS
Elastic	Malicious (high Confidence)
Emsisoft	Trojan.GenericKD.70173746 (B)
eScan	Trojan.GenericKD.70173746
ESET-NOD32	A Variant Of Win32/IRCBot.BDQ
F-Secure	Trojan.TR/Dldr.Banload.elvwr
Fortinet	PossibleThreat.FAI
GData	Trojan.GenericKD.70173746
Google	Detected
Gridinsoft (no cloud)	Trojan.Heur!.0321202D
Ikarus	Trojan.Win32.Agent
Jiangmin	Backdoor.IRCBot.ace
K7AntiVirus	Trojan (0059b4041)
K7GW	Trojan (0059b4041)
Kaspersky	HEUR:Trojan-Downloader.Win32.Banload.gen
Kingsoft	Win32.HeurC.KVMH008.a

Figure 7 - Vendor results

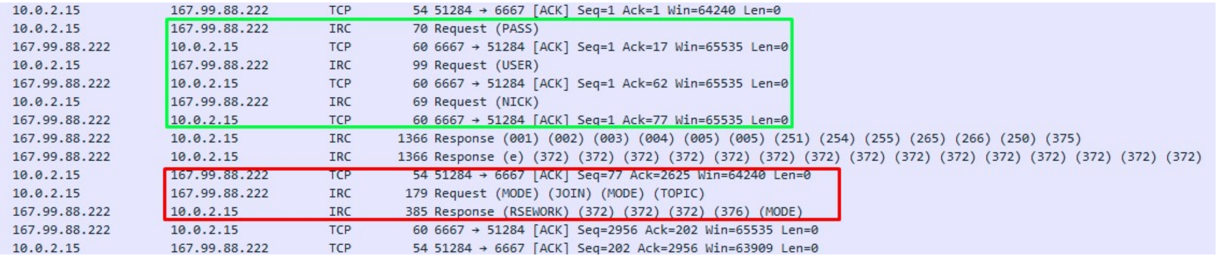
Scanning the file in through virus total, Figure 6, shows 60 out of 72 vendors detected this file was malicious and in Figure 7 we can see this was detected as an IRC botnet/banload by most vendors

Dynamic analysis

Following on from and using the information gathered during static analysis, dynamic analysis will take a deeper dive into the workings of the executable. During static analysis we gathered this was a potential IRC botnet, therefore we would expect to see some sort of network traffic in relation to this and will be using Wireshark to trace packets sent back and forth. This section is conducted within a secure environment.

Wireshark

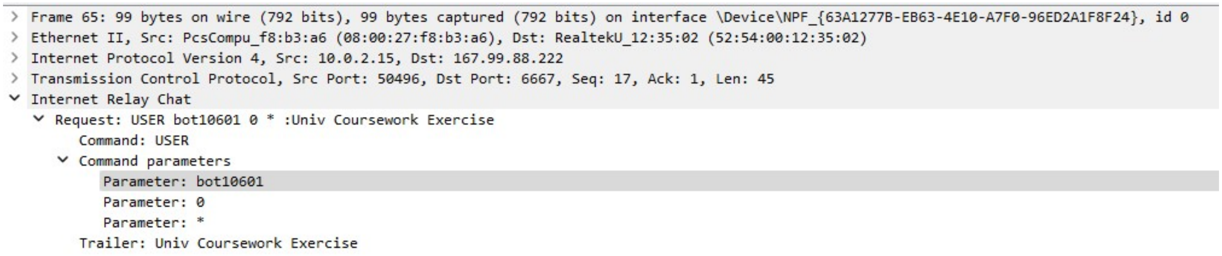
Frame analysis



```
10.0.2.15       167.99.88.222   TCP     54 51284 → 6667 [ACK] Seq=1 Ack=1 Win=64240 Len=0
10.0.2.15       167.99.88.222   IRC     70 Request (PASS)
167.99.88.222   10.0.2.15       TCP     60 6667 → 51284 [ACK] Seq=1 Ack=17 Win=65535 Len=0
10.0.2.15       167.99.88.222   IRC     99 Request (USER)
167.99.88.222   10.0.2.15       TCP     60 6667 → 51284 [ACK] Seq=1 Ack=62 Win=65535 Len=0
10.0.2.15       167.99.88.222   IRC     69 Request (NICK)
167.99.88.222   10.0.2.15       TCP     60 6667 → 51284 [ACK] Seq=1 Ack=77 Win=65535 Len=0
167.99.88.222   10.0.2.15       IRC   1366 Response (001) (002) (003) (004) (005) (005) (251) (254) (255) (265) (266) (250) (375)
167.99.88.222   10.0.2.15       IRC   1366 Response (e) (372) (372) (372) (372) (372) (372) (372) (372) (372) (372) (372) (372) (372) (372) (372)
10.0.2.15       167.99.88.222   TCP     54 51284 → 6667 [ACK] Seq=77 Ack=2625 Win=64240 Len=0
10.0.2.15       167.99.88.222   IRC    179 Request (MODE) (JOIN) (MODE) (TOPIC)
167.99.88.222   10.0.2.15       IRC    385 Response (RSEWORK) (372) (372) (372) (376) (MODE)
167.99.88.222   10.0.2.15       TCP     60 6667 → 51284 [ACK] Seq=2956 Ack=202 Win=65535 Len=0
10.0.2.15       167.99.88.222   TCP     54 51284 → 6667 [ACK] Seq=202 Ack=2956 Win=63909 Len=0
```

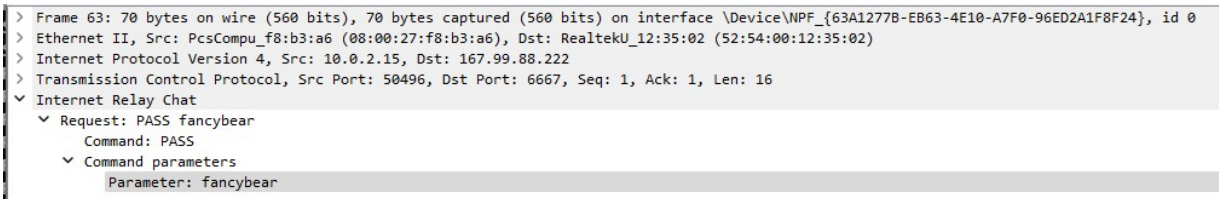
Figure 8 - Traffic sent after running .exe

After running the executable Wireshark detected both inbound and outbound TCP/IRC traffic between the Equation Group address and our machine requesting a password (PASS), username (USER) and nickname (NICK) showing a logon event, this is displayed in the green box of Figure 6. After this in the red box, a channel join.



```
> Frame 65: 99 bytes on wire (792 bits), 99 bytes captured (792 bits) on interface \Device\NPF_{63A1277B-EB63-4E10-A7F0-96ED2A1F8F24}, id 0
> Ethernet II, Src: PcsCompu_f8:b3:a6 (08:00:27:f8:b3:a6), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
> Internet Protocol Version 4, Src: 10.0.2.15, Dst: 167.99.88.222
> Transmission Control Protocol, Src Port: 50496, Dst Port: 6667, Seq: 17, Ack: 1, Len: 45
v Internet Relay Chat
  v Request: USER bot10601 0 * :Univ Coursework Exercise
      Command: USER
    v Command parameters
        Parameter: bot10601
        Parameter: 0
        Parameter: *
      Trailer: Univ Coursework Exercise
```

Figure 9 - USER command and username bot10601 detected in Wireshark

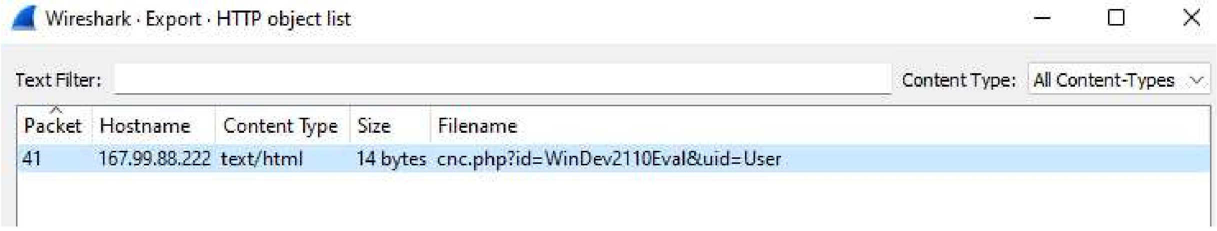


```
> Frame 63: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{63A1277B-EB63-4E10-A7F0-96ED2A1F8F24}, id 0
> Ethernet II, Src: PcsCompu_f8:b3:a6 (08:00:27:f8:b3:a6), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
> Internet Protocol Version 4, Src: 10.0.2.15, Dst: 167.99.88.222
> Transmission Control Protocol, Src Port: 50496, Dst Port: 6667, Seq: 1, Ack: 1, Len: 16
v Internet Relay Chat
  v Request: PASS fancybear
      Command: PASS
    v Command parameters
        Parameter: fancybear
```

Figure 10 - PASS command and password fancybear detected in Wireshark

Taking a deeper look at these frames, we can see the username and password which we can use to log into the IRC server as a bot. We wouldn't need the nickname as that is simply a display name and does not provide any authentication. Figure 7 shows us the user is "bot" followed by a random selection of numbers and Figure 8 shows the password is "fancybear".

HTTP Export List

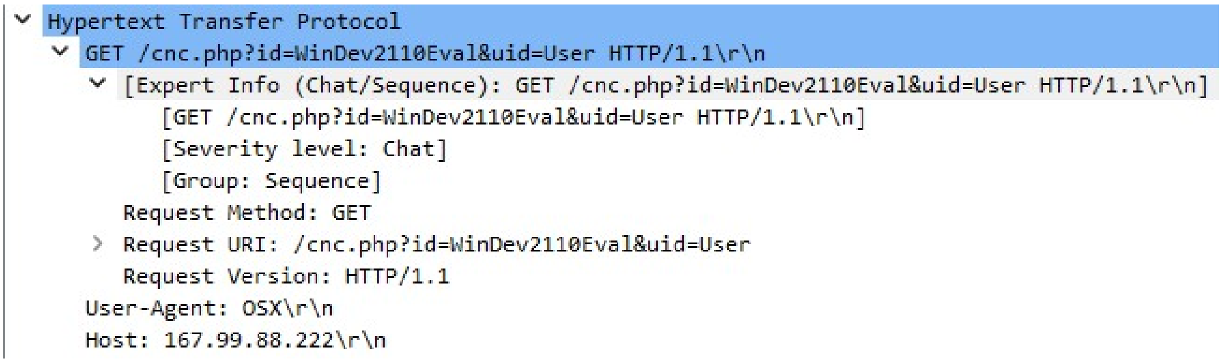


Text Filter:

Content Type: All Content-Types

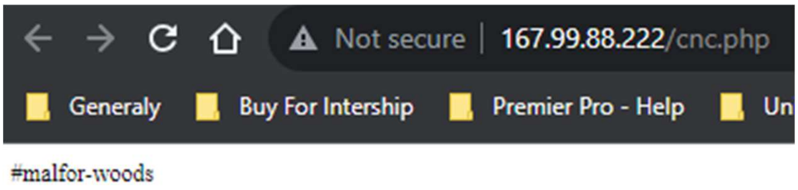
Packet	Hostname	Content Type	Size	Filename
41	167.99.88.222	text/html	14 bytes	cnc.php?id=WinDev2110Eval&uid=User

Figure 11 - HTTP exports after running malware



```
Hypertext Transfer Protocol
  GET /cnc.php?id=WinDev2110Eval&uid=User HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /cnc.php?id=WinDev2110Eval&uid=User HTTP/1.1\r\n]
      [GET /cnc.php?id=WinDev2110Eval&uid=User HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
  > Request URI: /cnc.php?id=WinDev2110Eval&uid=User
    Request Version: HTTP/1.1
  User-Agent: OSX\r\n
  Host: 167.99.88.222\r\n
```

Figure 12 - Frame information of identified HTTP export

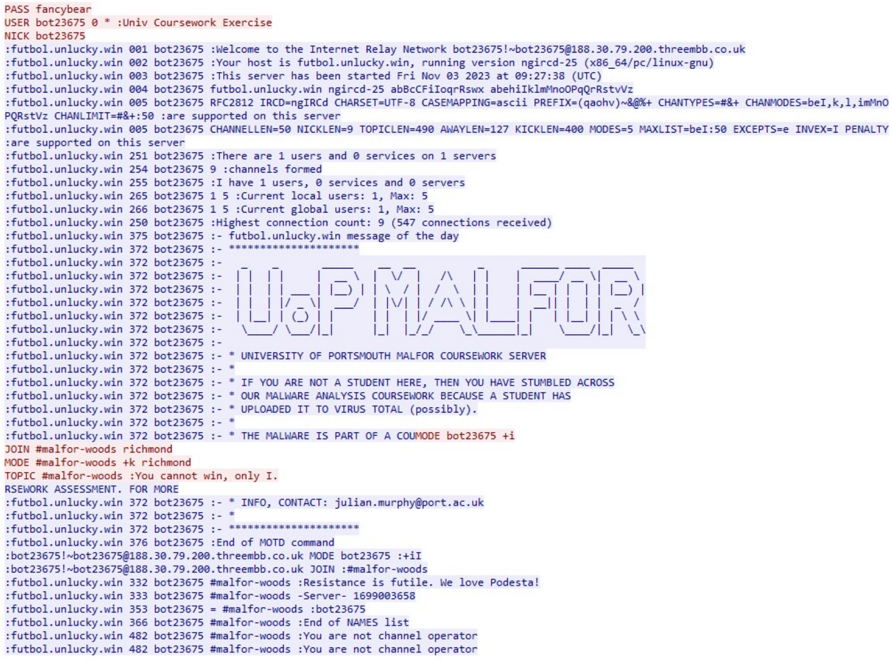


#malfor-woods

Figure 13 - #malfor-woods text can also be seen on the C2 link confirming this is a C2 channel

Reviewing HTTP exports Figure 9, shows a singular export to the IP address identified during static analysis and reviewing the frame for this, Figure 10, further confirms that the host is operating on OSX. A cnc.php link can be seen under 'Request URI' of Figure 10, cnc commonly meaning command and control which would coincide with our findings of there being an IRC command and control server.

TCP Stream



```
PASS fancybear
USER bot23675 0 * :Univ Coursework Exercise
NICK bot23675
:futbol.unlucky.win 001 bot23675 :Welcome to the Internet Relay Network bot23675!~bot23675@188.30.79.200.threembb.co.uk
:futbol.unlucky.win 002 bot23675 :Your host is futbol.unlucky.win, running version ngircd-25 (x86_64/pc/linux-gnu)
:futbol.unlucky.win 003 bot23675 :This server has been started Fri Nov 03 2023 at 09:27:38 (UTC)
:futbol.unlucky.win 004 bot23675 futbol.unlucky.win ngircd-25 abBcCFiIoqrRswx abehiIklmMnoOPqQrRstvVz
:futbol.unlucky.win 005 bot23675 RFC2812 IRCD=ngIRCd CHARSET=UTF-8 CASEMAPPING=ascii PREFIX=(qaohv)~&@%+ CHANTYPES=#&+ CHANMODES=beI,k,l,imMnO
PQRstVz CHANLIMIT=#&+:50 :are supported on this server
:futbol.unlucky.win 005 bot23675 CHANNELLEN=50 NICKLEN=9 TOPICLEN=490 AWAYLEN=127 KICKLEN=400 MODES=5 MAXLIST=beI:50 EXCEPTS=e INVEX=I PENALTY
:are supported on this server
:futbol.unlucky.win 251 bot23675 :There are 1 users and 0 services on 1 servers
:futbol.unlucky.win 254 bot23675 9 :channels formed
:futbol.unlucky.win 255 bot23675 :I have 1 users, 0 services and 0 servers
:futbol.unlucky.win 265 bot23675 1 5 :Current local users: 1, Max: 5
:futbol.unlucky.win 266 bot23675 1 5 :Current global users: 1, Max: 5
:futbol.unlucky.win 250 bot23675 :Highest connection count: 9 (547 connections received)
:futbol.unlucky.win 375 bot23675 :- futbol.unlucky.win message of the day
:futbol.unlucky.win 372 bot23675 :- ********************
:futbol.unlucky.win 372 bot23675 :-   _    _       _____    __  __          _      ______ ____  _____
:futbol.unlucky.win 372 bot23675 :-  | |  | |     |  __ \  |  \/  |   /\   | |    |  ____/ __ \|  __ \
:futbol.unlucky.win 372 bot23675 :-  | |  | | ___ | |__) | | \  / |  /  \  | |    | |__ | |  | | |__) |
:futbol.unlucky.win 372 bot23675 :-  | |  | |/ _ \|  ___/  | |\/| | / /\ \ | |    |  __|| |  | |  _  /
:futbol.unlucky.win 372 bot23675 :-  | |__| | (_) | |      | |  | |/ ____ \| |____| |   | |__| | | \ \
:futbol.unlucky.win 372 bot23675 :-   \____/ \___/|_|      |_|  |_/_/    \_\______|_|    \____/|_|  \_\
:futbol.unlucky.win 372 bot23675 :-
:futbol.unlucky.win 372 bot23675 :- * UNIVERSITY OF PORTSMOUTH MALFOR COURSEWORK SERVER
:futbol.unlucky.win 372 bot23675 :- *
:futbol.unlucky.win 372 bot23675 :- * IF YOU ARE NOT A STUDENT HERE, THEN YOU HAVE STUMBLED ACROSS
:futbol.unlucky.win 372 bot23675 :- * OUR MALWARE ANALYSIS COURSEWORK BECAUSE A STUDENT HAS
:futbol.unlucky.win 372 bot23675 :- * UPLOADED IT TO VIRUS TOTAL (possibly).
:futbol.unlucky.win 372 bot23675 :- *
:futbol.unlucky.win 372 bot23675 :- * THE MALWARE IS PART OF A COUMODE bot23675 +i
JOIN #malfor-woods richmond
MODE #malfor-woods +k richmond
TOPIC #malfor-woods :You cannot win, only I.
RSEWORK ASSESSMENT. FOR MORE
:futbol.unlucky.win 372 bot23675 :- * INFO, CONTACT: julian.murphy@port.ac.uk
:futbol.unlucky.win 372 bot23675 :- *
:futbol.unlucky.win 372 bot23675 :- ********************
:futbol.unlucky.win 376 bot23675 :End of MOTD command
:bot23675!~bot23675@188.30.79.200.threembb.co.uk MODE bot23675 :+iI
:bot23675!~bot23675@188.30.79.200.threembb.co.uk JOIN :#malfor-woods
:futbol.unlucky.win 332 bot23675 #malfor-woods :Resistance is futile. We love Podesta!
:futbol.unlucky.win 333 bot23675 #malfor-woods -Server- 1699003658
:futbol.unlucky.win 353 bot23675 = #malfor-woods :bot23675
:futbol.unlucky.win 366 bot23675 #malfor-woods :End of NAMES list
:futbol.unlucky.win 482 bot23675 #malfor-woods :You are not channel operator
:futbol.unlucky.win 482 bot23675 #malfor-woods :You are not channel operator
```

Figure 14 - TCP stream displaying IRC connection to server and #malfor-woods channel

After re-executing the malware and again tracking this in Wireshark to see if there were any further changes, following the TCP stream, Figure 12, to display the full conversation between the malware and the command and control server does not show any significant changes.

However, each time the malware is executed a new Bot name is assigned, meaning this is designed to be distributed to multiple systems; the same user cannot log into an IRC chat at the same time. In Figure 12 we can also see that the bot is getting “You are not channel operator” suggesting it does not have permission to run a command which is expected.

Here we can also see a potential author “Julian Murphy” and the origins of the malware which is part of a University of Portsmouth malware server.

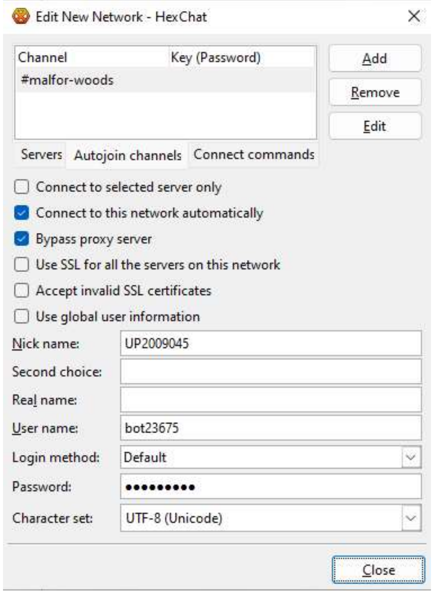
Analysis Conclusion

After analysing the executable provided, it is clearly designed to be distributed and function as a bot which can be controlled via IRC. After executing the malware, the program hides in the background logging into the server and channel awaiting a response from the command and control server. Once the IRC operator sends a command, for example “EXEC IEEXPLORER.exe” this will then conduct the command on each device infected with the malware.

Looking over the program, it seems the only resemblance to an author would be that displayed in the contact information “julian.murphy@port.ac.uk”.

Accessing the Server

This section will provide an overview of using the information gathered in previous steps to gain access to the server and channel. Using Hexchat, and client which can be used to interact with IRC servers and their channels.



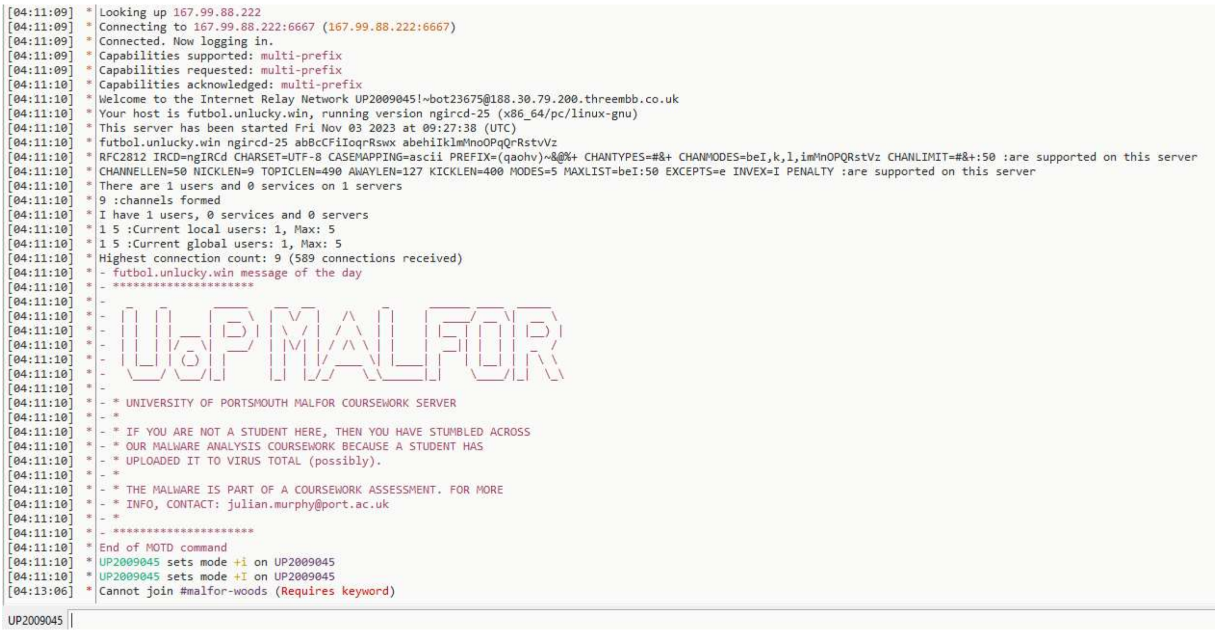
Channel	Key (Password)
#malfor-woods	

☐ Connect to selected server only
☑ Connect to this network automatically
☑ Bypass proxy server
☐ Use SSL for all the servers on this network
☐ Accept invalid SSL certificates
☐ Use global user information

Nick name: UP2009045
Second choice:
Real name:
User name: bot23675
Login method: Default
Password: •••••••••
Character set: UTF-8 (Unicode)

Figure 15 - Connecting to server via Hexchat using credentials found in Wireshark

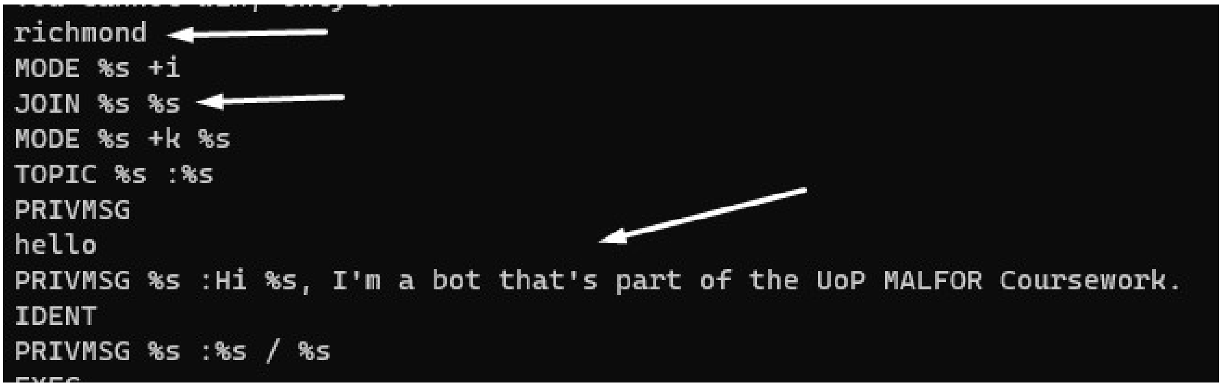
As shown in Figure 13, I installed Hexchat on the test machine and entered the username and password collected from the previous steps. I purposefully left the channel key empty here as I wanted to test if there were any difference between auto-joining versus joining manually.



```
[04:11:09] * Looking up 167.99.88.222
[04:11:09] * Connecting to 167.99.88.222:6667 (167.99.88.222:6667)
[04:11:09] * Connected. Now logging in.
[04:11:09] * Capabilities supported: multi-prefix
[04:11:09] * Capabilities requested: multi-prefix
[04:11:10] * Capabilities acknowledged: multi-prefix
[04:11:10] * Welcome to the Internet Relay Network UP2009045!~bot23675@188.30.79.200.threembb.co.uk
[04:11:10] * Your host is futbol.unlucky.win, running version ngircd-25 (x86_64/pc/linux-gnu)
[04:11:10] * This server has been started Fri Nov 03 2023 at 09:27:38 (UTC)
[04:11:10] * futbol.unlucky.win ngircd-25 abBcCFiIoqrRswx abehiIklmMnoOPqQrRstvVz
[04:11:10] * RFC2812 IRCD=ngIRCd CHARSET=UTF-8 CASEMAPPING=ascii PREFIX=(qaohv)~&@%+ CHANTYPES=#&+ CHANMODES=beI,k,l,imMnOPQRstVz CHANLIMIT=#&+:50 :are supported on this server
[04:11:10] * CHANNELLEN=50 NICKLEN=9 TOPICLEN=490 AWAYLEN=127 KICKLEN=400 MODES=5 MAXLIST=beI:50 EXCEPTS=e INVEX=I PENALTY :are supported on this server
[04:11:10] * There are 1 users and 0 services on 1 servers
[04:11:10] * 9 :channels formed
[04:11:10] * I have 1 users, 0 services and 0 servers
[04:11:10] * 1 5 :Current local users: 1, Max: 5
[04:11:10] * 1 5 :Current global users: 1, Max: 5
[04:11:10] * Highest connection count: 9 (589 connections received)
[04:11:10] * - futbol.unlucky.win message of the day
[04:11:10] * - ********************
[04:11:10] * -
[04:11:10] * -  _    _       _____    __  __            _        ______  ____   _____
[04:11:10] * - | |  | |     |  __ \  |  \/  |    /\    | |      |  ____|/ __ \ |  __ \
[04:11:10] * - | |  | | ___ | |__) | | \  / |   /  \   | |      | |__  | |  | || |__) |
[04:11:10] * - | |  | |/ _ \|  ___/  | |\/| |  / /\ \  | |      |  __| | |  | ||  _  /
[04:11:10] * - | |__| | (_) | |      | |  | | / ____ \ | |____  | |    | |__| || | \ \
[04:11:10] * -  \____/ \___/|_|      |_|  |_|/_/    \_\|______| |_|     \____/ |_|  \_\
[04:11:10] * -
[04:11:10] * - * UNIVERSITY OF PORTSMOUTH MALFOR COURSEWORK SERVER
[04:11:10] * - *
[04:11:10] * - * IF YOU ARE NOT A STUDENT HERE, THEN YOU HAVE STUMBLED ACROSS
[04:11:10] * - * OUR MALWARE ANALYSIS COURSEWORK BECAUSE A STUDENT HAS
[04:11:10] * - * UPLOADED IT TO VIRUS TOTAL (possibly).
[04:11:10] * - *
[04:11:10] * - * THE MALWARE IS PART OF A COURSEWORK ASSESSMENT. FOR MORE
[04:11:10] * - * INFO, CONTACT: julian.murphy@port.ac.uk
[04:11:10] * - *
[04:11:10] * - ********************
[04:11:10] * End of MOTD command
[04:11:10] * UP2009045 sets mode +i on UP2009045
[04:11:10] * UP2009045 sets mode +I on UP2009045
[04:13:06] * Cannot join #malfor-woods (Requires keyword)
UP2009045 |
```

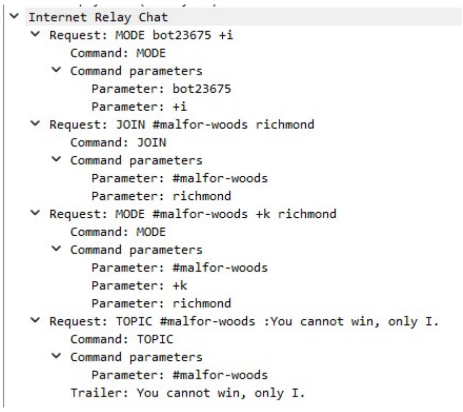
Figure 16 - Successful log in to server, channel requires separate password

After connecting, the server outputs the exact same MOTD and information, Figure 14, that can be seen in Figure 12.



```
You cannot win, only I.
richmond
MODE %s +i
JOIN %s %s
MODE %s +k %s
TOPIC %s :%s
PRIVMSG
hello
PRIVMSG %s :Hi %s, I'm a bot that's part of the UoP MALFOR Coursework.
IDENT
PRIVMSG %s :%s / %s
EXEC
```

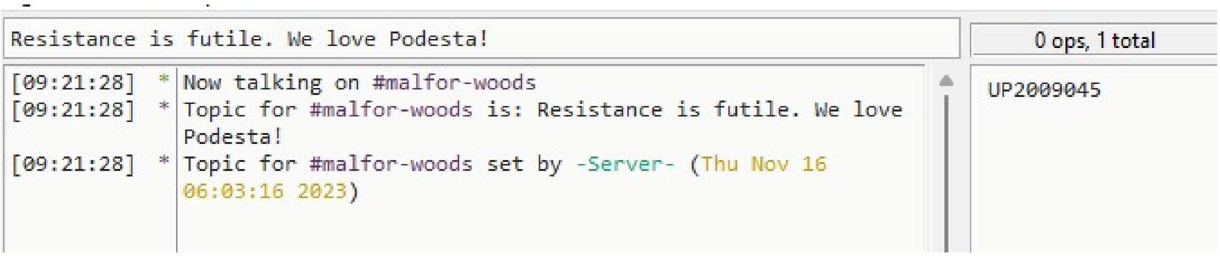
Figure 17 - Reviewing strings, a section for joining a channel can be found containing the password



```
v Internet Relay Chat
  v Request: MODE bot23675 +i
      Command: MODE
    v Command parameters
        Parameter: bot23675
        Parameter: +i
  v Request: JOIN #malfor-woods richmond
      Command: JOIN
    v Command parameters
        Parameter: #malfor-woods
        Parameter: richmond
  v Request: MODE #malfor-woods +k richmond
      Command: MODE
    v Command parameters
        Parameter: #malfor-woods
        Parameter: +k
        Parameter: richmond
  v Request: TOPIC #malfor-woods :You cannot win, only I.
      Command: TOPIC
    v Command parameters
        Parameter: #malfor-woods
      Trailer: You cannot win, only I.
```

Figure 18 - Wireshark frame displaying channel connection and password

Performing some further basic static and dynamic analysis, we can see that the password previously found in the TCP stream within Wireshark also seen in Figure 16, can be retrieved using strings, Figure 15.



```
Resistance is futile. We love Podesta!
[09:21:28]  * Now talking on #malfor-woods
[09:21:28]  * Topic for #malfor-woods is: Resistance is futile. We love
              Podesta!
[09:21:28]  * Topic for #malfor-woods set by -Server- (Thu Nov 16
              06:03:16 2023)
```

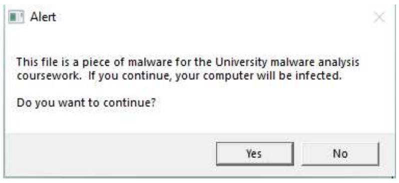
Figure 19 - Using the password 'richmond' provides access to #malfor-woods

With the password which can be found in Wireshark, Figures 14,16, during dynamic analysis and in strings, Figure 15, we were able to gain access to the botnet server channel “#malfor-woods”. From here we can review some commands to control the botnet which can be found looking at both Strings and IDA.

Debugging & Further Data Gathering

Using IDA Freeware we carried out debugging on the malware file, running through the executable step by step and analysing the file further to find methods of and how this could exactly be executed.

Anti-Debugging Technique

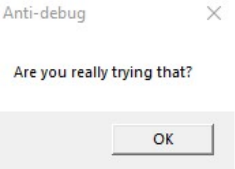


Alert

This file is a piece of malware for the University malware analysis coursework. If you continue, your computer will be infected.

Do you want to continue?

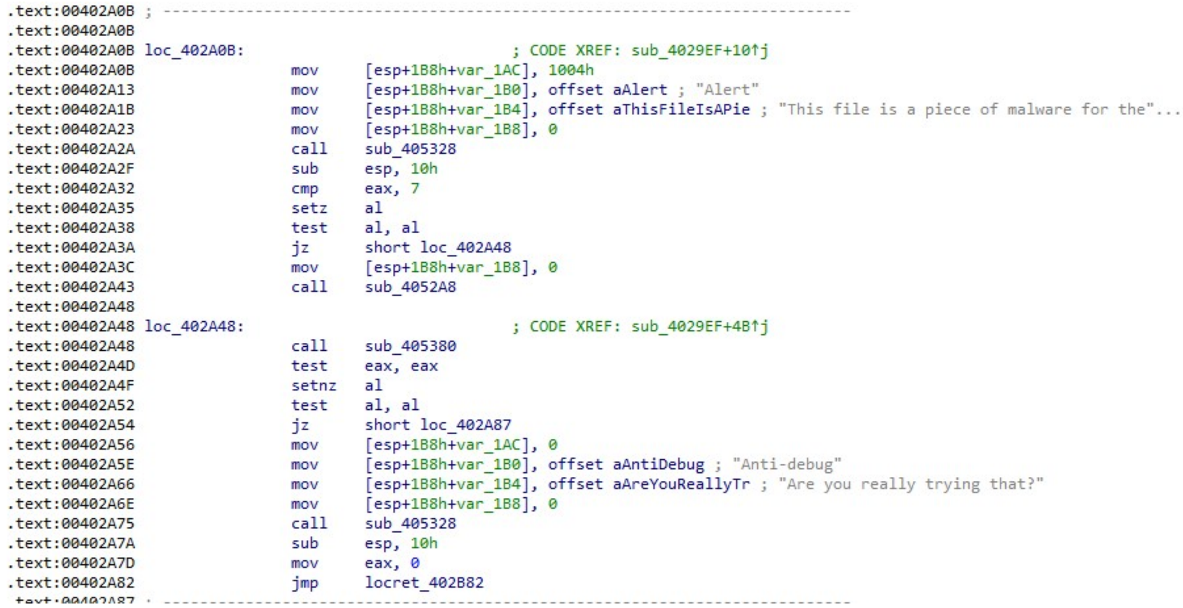
Figure 20 - First window generated by malware



Anti-debug

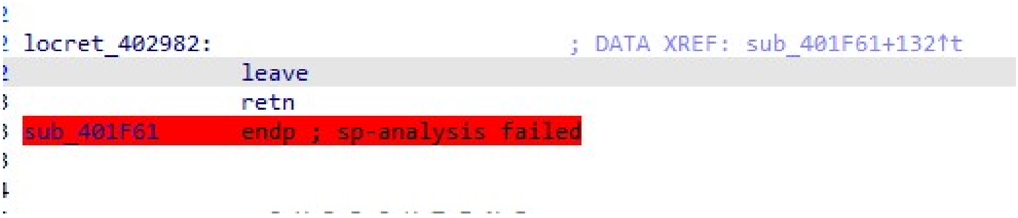
Are you really trying that?

Figure 21 - Ant-Debugging window generated by malware



```
.text:00402A0B ; ---------------------------------------------------------------------------
.text:00402A0B
.text:00402A0B loc_402A0B:                             ; CODE XREF: sub_4029EF+10↑j
.text:00402A0B                 mov     [esp+1B8h+var_1AC], 1004h
.text:00402A13                 mov     [esp+1B8h+var_1B0], offset aAlert ; "Alert"
.text:00402A1B                 mov     [esp+1B8h+var_1B4], offset aThisFileIsAPie ; "This file is a piece of malware for the"...
.text:00402A23                 mov     [esp+1B8h+var_1B8], 0
.text:00402A2A                 call    sub_405328
.text:00402A2F                 sub     esp, 10h
.text:00402A32                 cmp     eax, 7
.text:00402A35                 setz    al
.text:00402A38                 test    al, al
.text:00402A3A                 jz      short loc_402A48
.text:00402A3C                 mov     [esp+1B8h+var_1B8], 0
.text:00402A43                 call    sub_4052A8
.text:00402A48
.text:00402A48 loc_402A48:                             ; CODE XREF: sub_4029EF+4B↑j
.text:00402A48                 call    sub_405380
.text:00402A4D                 test    eax, eax
.text:00402A4F                 setnz   al
.text:00402A52                 test    al, al
.text:00402A54                 jz      short loc_402A87
.text:00402A56                 mov     [esp+1B8h+var_1AC], 0
.text:00402A5E                 mov     [esp+1B8h+var_1B0], offset aAntiDebug ; "Anti-debug"
.text:00402A66                 mov     [esp+1B8h+var_1B4], offset aAreYouReallyTr ; "Are you really trying that?"
.text:00402A6E                 mov     [esp+1B8h+var_1B8], 0
.text:00402A75                 call    sub_405328
.text:00402A7A                 sub     esp, 10h
.text:00402A7D                 mov     eax, 0
.text:00402A82                 jmp     locret_402B82
.text:00402A87 ; ---------------------------------------------------------------------------
```

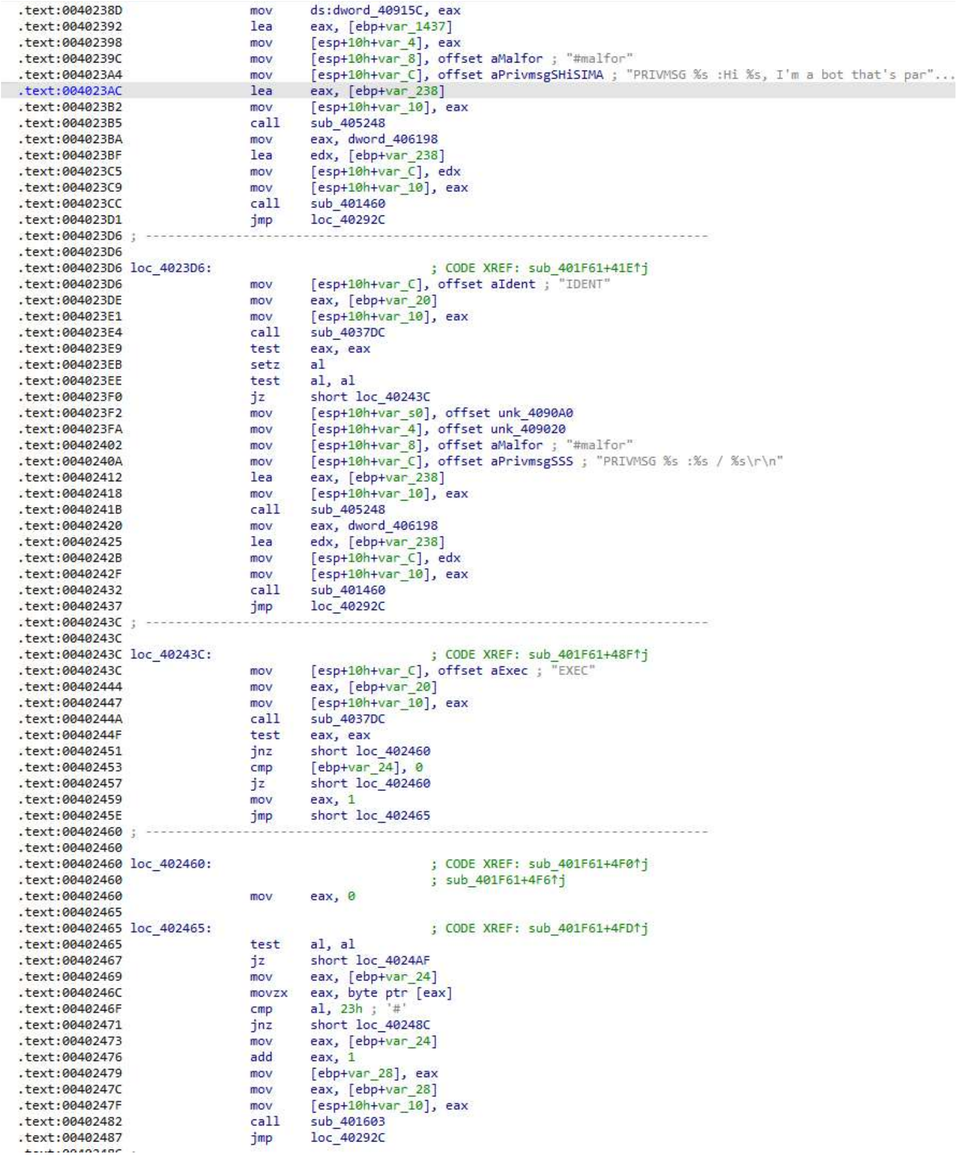
Figure 22 - Code block for first two forms



```
locret_402982:                          ; DATA XREF: sub_401F61+132↑t
                leave
                retn
sub_401F61      endp ; sp-analysis failed
```

Figure 23 - Code block for exiting debugging

C2 Commands



```
.text:0040238D                 mov     ds:dword_40915C, eax
.text:00402392                 lea     eax, [ebp+var_1437]
.text:00402398                 mov     [esp+10h+var_4], eax
.text:0040239C                 mov     [esp+10h+var_8], offset aMalfor ; "#malfor"
.text:004023A4                 mov     [esp+10h+var_C], offset aPrivmsgSHiSIMA ; "PRIVMSG %s :Hi %s, I'm a bot that's par"...
.text:004023AC                 lea     eax, [ebp+var_238]
.text:004023B2                 mov     [esp+10h+var_10], eax
.text:004023B5                 call    sub_405248
.text:004023BA                 mov     eax, dword_406198
.text:004023BF                 lea     edx, [ebp+var_238]
.text:004023C5                 mov     [esp+10h+var_C], edx
.text:004023C9                 mov     [esp+10h+var_10], eax
.text:004023CC                 call    sub_401460
.text:004023D1                 jmp     loc_40292C
.text:004023D6 ; ---------------------------------------------------------------------------
.text:004023D6
.text:004023D6 loc_4023D6:                             ; CODE XREF: sub_401F61+41E↑j
.text:004023D6                 mov     [esp+10h+var_C], offset aIdent ; "IDENT"
.text:004023DE                 mov     eax, [ebp+var_20]
.text:004023E1                 mov     [esp+10h+var_10], eax
.text:004023E4                 call    sub_4037DC
.text:004023E9                 test    eax, eax
.text:004023EB                 setz    al
.text:004023EE                 test    al, al
.text:004023F0                 jz      short loc_40243C
.text:004023F2                 mov     [esp+10h+var_s0], offset unk_4090A0
.text:004023FA                 mov     [esp+10h+var_4], offset unk_409020
.text:00402402                 mov     [esp+10h+var_8], offset aMalfor ; "#malfor"
.text:0040240A                 mov     [esp+10h+var_C], offset aPrivmsgSSS ; "PRIVMSG %s :%s / %s\r\n"
.text:00402412                 lea     eax, [ebp+var_238]
.text:00402418                 mov     [esp+10h+var_10], eax
.text:0040241B                 call    sub_405248
.text:00402420                 mov     eax, dword_406198
.text:00402425                 lea     edx, [ebp+var_238]
.text:0040242B                 mov     [esp+10h+var_C], edx
.text:0040242F                 mov     [esp+10h+var_10], eax
.text:00402432                 call    sub_401460
.text:00402437                 jmp     loc_40292C
.text:0040243C ; ---------------------------------------------------------------------------
.text:0040243C
.text:0040243C loc_40243C:                             ; CODE XREF: sub_401F61+48F↑j
.text:0040243C                 mov     [esp+10h+var_C], offset aExec ; "EXEC"
.text:00402444                 mov     eax, [ebp+var_20]
.text:00402447                 mov     [esp+10h+var_10], eax
.text:0040244A                 call    sub_4037DC
.text:0040244F                 test    eax, eax
.text:00402451                 jnz     short loc_402460
.text:00402453                 cmp     [ebp+var_24], 0
.text:00402457                 jz      short loc_402460
.text:00402459                 mov     eax, 1
.text:0040245E                 jmp     short loc_402465
.text:00402460 ; ---------------------------------------------------------------------------
.text:00402460
.text:00402460 loc_402460:                             ; CODE XREF: sub_401F61+4F0↑j
.text:00402460                                         ; sub_401F61+4F6↑j
.text:00402460                 mov     eax, 0
.text:00402465
.text:00402465 loc_402465:                             ; CODE XREF: sub_401F61+4FD↑j
.text:00402465                 test    al, al
.text:00402467                 jz      short loc_4024AF
.text:00402469                 mov     eax, [ebp+var_24]
.text:0040246C                 movzx   eax, byte ptr [eax]
.text:0040246F                 cmp     al, 23h ; '#'
.text:00402471                 jnz     short loc_40248C
.text:00402473                 mov     eax, [ebp+var_24]
.text:00402476                 add     eax, 1
.text:00402479                 mov     [ebp+var_28], eax
.text:0040247C                 mov     eax, [ebp+var_28]
.text:0040247F                 mov     [esp+10h+var_10], eax
.text:00402482                 call    sub_401603
.text:00402487                 jmp     loc_40292C
```

Figure 24 - Commands available

After beginning the debugging, the first window is generated by the malware, Figure 18, this contains a warning and asks for confirmation to continue. This is then followed by an Anti-debug window, Figure 19, to prevent the debugging process, after “OK” is selected debugging automatically exits.

These are shown further in depth via the code view within Figure 20. It is here that we can see the ‘jmp’ reference to “locret_402B82”. Figure 21 displays this return instruction and shows the purpose to end debugging.

Recommendations for malware remediation and future prevention

Remediation

Beginning with the remediation of the executable, any affected systems within Gazprom's internal infrastructure should be isolated and have their network connection terminated. This is to prevent any further infection via the botnet/re-infection.

Isolation

Once these systems have been isolated, the executable process should also be terminated, this will close the malware and its connection the command and control server, removing the ability for the owner to send commands.

Removal

Next the removal of the malicious files, i.e., malfor-cw-sample.exe should be removed from all systems, affected or unaffected.

Review

Finally, review currently affected systems ensuring the malware has been sufficiently removed and these can be released from isolation.

Future Prevention

As this data leak was caused by an insider of the estate for future prevention, we need to look at two areas, staff and hardware/software:

Update

Firstly, any operating systems and software should be kept up to date, scanning such as Tenable's Nessus can be implemented to automate these vulnerability assessments and find anything which is out of date or vulnerable and needs patching against new CVE's. Ensuring systems are up to date and patches against any newly found vulnerabilities will greatly reduce the risk of threat actors gaining access to and exfiltrating data.

Endpoint Security

XDR/SIEM Implementation

Secondly, Gazprom should also consider introducing if not already implemented an XDR solution or at the very least a SIEM which will produce alerts based on values found during collection of network logs then any system activity such as the executed commands will be detected and flagged as malicious as they will not have come from that user.

Firewall & Device Security

Gazprom should also review the effectiveness of their current endpoint security on their firewalls and user devices. Firewalls should be setup with sufficient rules to detect and prevent any malicious traffic by controlling what inbound/outbound traffic is denied or allowed throughout the network.

Creating a strong endpoint security infrastructure both through implementation of threat detection and good security practices on endpoints will greatly increase the likelihood of an attacker being alerted on if they get into the network and reduce the risk of attackers gaining access to core IP.

Staff training

As this was a rogue employee within the estate, it’s of utmost importance to Gazprom that they provide staff training to employees within the organization to highlight the key factors and devastating outcomes of breaking data security, not only for the organization, but the user themselves. Reinforcement of security policies, identifying suspicious activities and reporting concerns via security awareness training is important to ensure best security practices throughout the business (ADJACENT DIGITAL POLITICS LTD, 2023)

Contents

Checkpoint Research Team. (2015). *Intelligence Report: Equation Group - Check Point Blog*. https://blog.checkpoint.com/security/intelligence-report-equation-group/

Wireshark. (2020, August 11). *IRC*. https://wiki.wireshark.org/IRC.md

Table 1 – Discovered Items

Item	Significance
Username	botxxxxx
Channel Name	Malfor-woods
Password	Fancybear
Password	Richmond
IP Address	167.99.88.222
Port	6667
Malware Attack	Internet Relay Chat Botnet
Author	Julian Murphy
Origin	University of Portsmouth
Commands	‘EXEC’ ‘OPEN’ ‘PING’ ‘!’ ‘MSPAINT’ ‘SHUTDOWN’ ‘HELLO’ ‘IDENT’