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

FileCreationEvents

count

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.

```
| 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
```

```
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
```

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

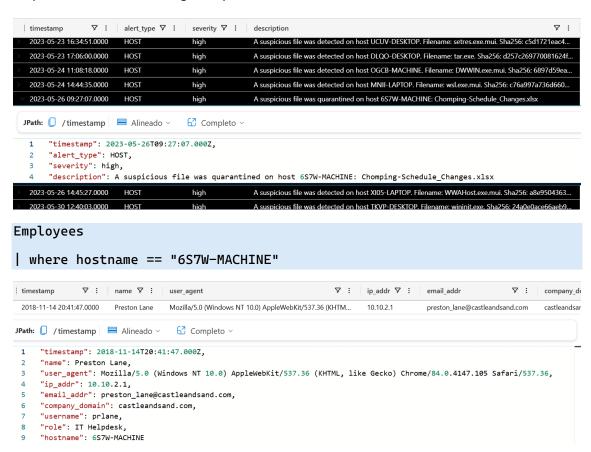
De los equipos afectados, 18 de ellos pertenecen a empleados dentro del departamento de IT. Vamos a revisar en cuántos dispositivos se ha registrado una alerta relacionada con la nota de ransomware que dejaron los atacantes.

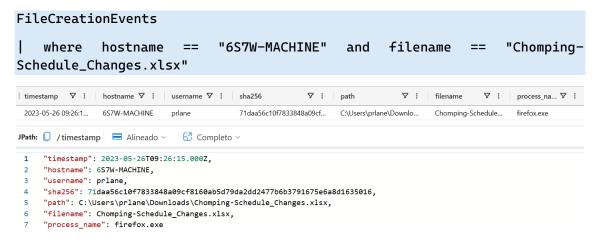
```
let impact_hosts = FileCreationEvents
| where filename == 'PAY_UP_OR_SWIM_WITH_THE_FISHES.txt'
| distinct hostname;
SecurityAlerts
| where description has_any (impact_hosts)
| count
>>> 652
```

```
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
```

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.





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 €
> 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 ϵ
> 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.:
> 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://shark fin.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://shark fin.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://shark fin.com/modules/public/published/Chomping-Sche	Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6
> 2023-05-27 10:44:42.0000	GET	10.10.4.231	http://shark fin.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://shark fin.com/modules/public/published/Chomping-Sche	Mozilla/5.0 (Windows NT 5.1; WOW64; rv:48.0) Ge

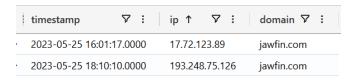
Los dos dominios relacionados con la descarga de estos ficheros son jawfin[.]com y sharkfin[.]com. Concretamente, el empleado del departamento de IT donde se encontró el fichero en cuarentena descargó el fichero desde una página con dominio jawfin[.]com, ya que en la tabla anterior podemos ver que la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

PassiveDns

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.

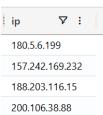
PassiveDns

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



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

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



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;
```

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.

157.242.169.232 Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gecko/201...

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...

Mozilla/5.0 (Android 1.6; Mobile; rv:24.0) Gecko/24....

Mozilla/5.0 (X11; Linux i686; rv:1.9.6.20) Gecko/201...

Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_5_6; r...

http://castleandsand.com/search?query=retail...

http://castleandsand.com/search?query=how...

https://castleandsand.com/search?query=beac..

GET 200.106.38.88
GET 17.72.123.89
GET

134.136.25.2

2023-05-20 07:22:50.0000

2023-05-22 07:08:56.0000

2023-05-22 07:43:20.0000

2023-05-22 18:14:38.0000 GET

```
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	ν 7	recipient	7	subject	
2023-05-25 16:33:09.00	000 legal.sand@	verizon.com urgent_	urgent@yandex.co	m preston_lane	@castleandsand.com	[EXTERNA	AL] FW: Take key beach reviews and re
2023-05-25 16:33:09.00	000 legal.sand@	verizon.com urgent_	urgent@yandex.co	m charles_lowe(@castleandsand.com	[EXTERNA	AL] FW: Take key beach reviews and re
2023-05-25 16:33:09.00	000 legal.sand@	verizon.com urgent_	urgent@yandex.co	m daniel_muse(@castleandsand.com	[EXTERNA	AL] FW: Take key beach reviews and re
2023-05-25 16:33:09.00	000 legal.sand@	verizon.com urgent_	urgent@yandex.co	m george_abne	y@castleandsand.com	[EXTERNA	AL] FW: Take key beach reviews and re
2023-05-25 16:33:09.00	000 legal.sand@	verizon.com urgent_	urgent@yandex.co	m louise_baltod	ano@castleandsand.co	m [EXTERN	AL] FW: Take key beach reviews and re
2023-05-25 16:33:09.00	000 legal.sand@	verizon.com urgent_	urgent@yandex.co	m james_wall@d	castleandsand.com	[EXTERNA	AL] FW: Take key beach reviews and re
2023-05-25 16:33:09.00	000 legal.sand@	verizon.com urgent_	urgent@yandex.co	m christy_holbro	ook@castleandsand.co	m [EXTERN	AL] FW: Take key beach reviews and re
2023-05-25 16:33:09.00	000 legal.sand@	verizon.com urgent_	urgent@yandex.co	m jeremy_davis	@castleandsand.com	[EXTERNA	AL] FW: Take key beach reviews and re
2023-05-26 19:58:05.00	000 legal.sand@	verizon.com castle@	hotmail.com	charlene_josh	ua@castleandsand.cor	n [EXTERN	AL] RE: LI accessories II great it sand re
2023-05-26 19:58:05.00	000 legal.sand@	verizon.com castle@	hotmail.com	patrick_dietz(@castleandsand.com	[EXTERNA	AL] RE: LI accessories II great it sand re
2023-05-26 19:58:05.00	000 legal.sand@	verizon.com castle@	hotmail.com	wallace_gum	@castleandsand.com	[EXTERNA	AL] RE: LI accessories II great it sand re
2023-05-26 19:58:05.00	000 legal.sand@	verizon.com castle@	hotmail.com	patricia_willia	ms@castleandsand.cor	n [EXTERN	AL] RE: LI accessories II 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.

```
| 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)
```

>>>6

count

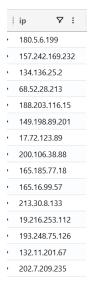
| distinct Result

Email

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
```



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
```

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"}
```

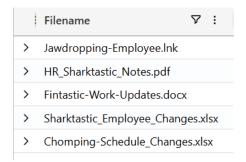
Además, podemos aprovechar la función parse_path para obtener el archivo de forma separada de la ruta incluida en la URL (documentación oficial <u>aquí</u>).

```
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
```



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")
```

timestamp	⊽ :	ра ∇ ᠄	parent_process_hash	♡ :	pr ↓ ▽ :	proce ∇ :	process_hash
2023-05-27 13:17:	25.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	b9f5112ba6025385a2b37a470aa
2023-05-28 11:22:	12.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	916a61b9e46a898d541fc1669d9
2023-05-28 13:20:	00.000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	35f866c698b2caa8834801bb38b
2023-05-29 15:19:	58.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	a4800b2f1d293afe8b8c24f48539
2023-05-30 15:15:	18.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	42488f2444cb9cfeead24c0adfbb
2023-05-31 13:10:	55.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	cda4b4d31c8c05b534559cbf76b
2023-06-01 12:03:	03.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	c173d433ce59420478245ef355f
2023-06-01 13:36:	57.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	6cfcb91e0154d5626a6ab160001
2023-06-04 11:19:	14.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	625b51595ba444fe253249eb192
2023-06-04 14:57:	07.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	4a3ec35c2b39bfbca71994be9a3
2023-06-05 11:29:	51.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	70fef9ef08893567e1d285cee207
2023-06-05 14:37:	47.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	5370c9658fa262534cbde09e761
2023-06-06 10:08:	50.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	1f0a9e2967aa95a76fc55dca02a0
2023-06-07 09:21:	15.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	56f2455b0b7e260c20bab1d12a6
2023-06-07 10:21:	17.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	53cbc7b7dc01b6b6db79ccba21
2023-06-07 10:54:	32.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	107b3217dbe46705b6a1d939b7
2023-06-07 17:00:	33.0000	services.exe	c3c259ae4640cded730676a6956ba	fea4f9bf20ed460a61c62c	winlogon.exe	winlogon.exe	bacef4ab8a3fa5c6059ce01b0d8e
			2 252 1612 1 1722576 6251				ar