RSAConference2019
Asia Pacific & Japan
Singapore | 16–18 July | Marina Bay Sands

BETTER.

SESSION ID: LAB-T05

Velociraptor: Digging deeper!

Dr Michael Cohen

Digital Paleontologist
Velocidex Enterprises

Nick Klein

Director, Velocidex Enterprises
Director, Klein & Co. Computer Forensics
SANS DFIR Certified Instructor

Who are we?

Dr Michael Cohen

- Experienced digital forensic software developer.
- Developer of foundation forensic tools including Volatility and Rekall.
- Former lead developer of Grr Rapid Response at Google Inc.

Nick Klein

- Director of Klein & Co. digital forensic and cyber response team.
- SANS DFIR Certified Instructor.

What will you need today?

A Windows computer or virtual machine, with admin access.

A copy of Velociraptor from our official release page:

https://github.com/Velocidex/velociraptor/releases

A hunting frame of mind.

What is Velociraptor?

Velociraptor is a unique DFIR tool, giving *you* power and flexibility through the Velociraptor Query Language (VQL)

VQL is used for everything:

- Collecting information from endpoints
- Controlling monitoring and response on endpoints
- Controlling and managing the Velociraptor server.

Velociraptor overview

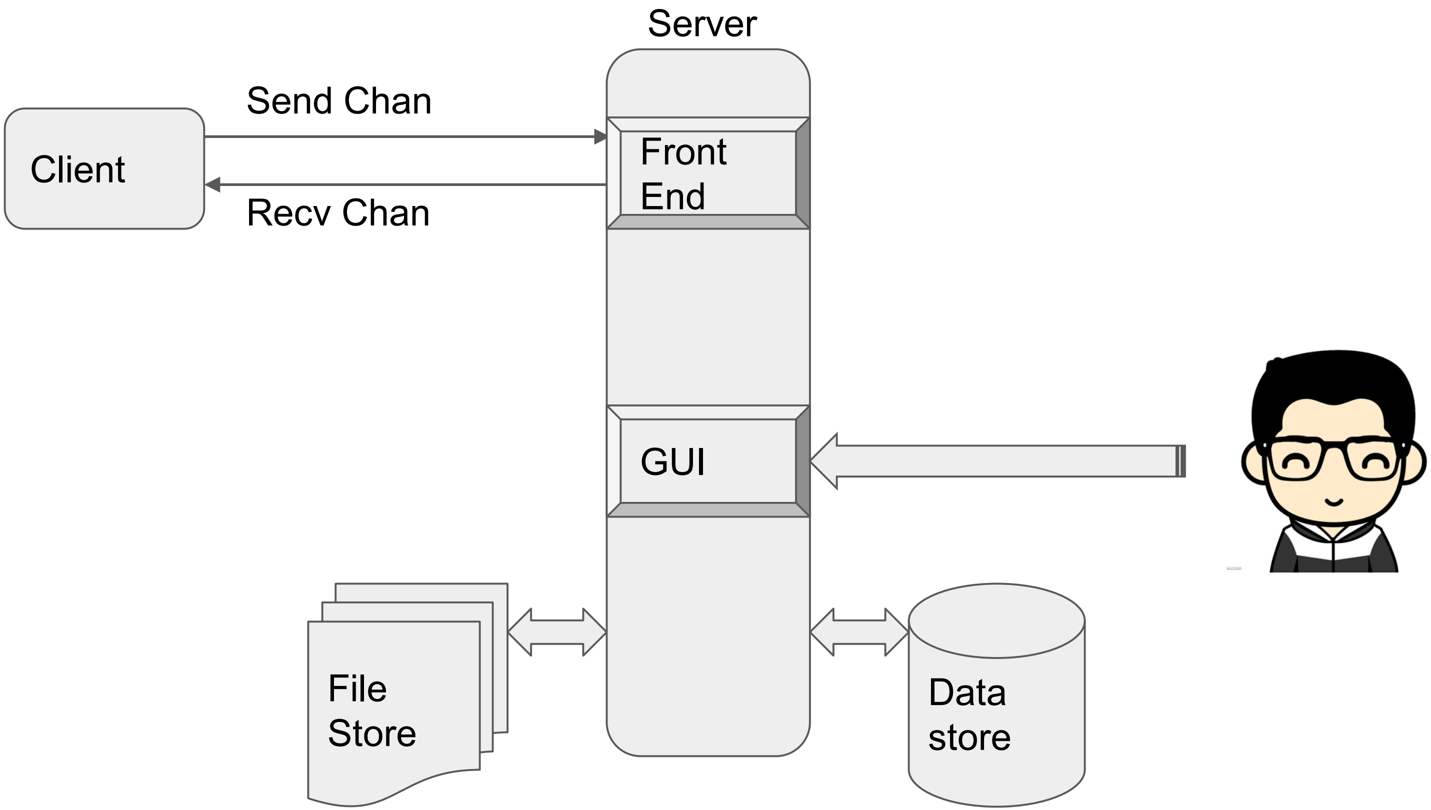
Everything uses the same binary - both clients and server.

- The server is controlled via the server configuration file.
- The client is controlled via the client configuration file.

In this lab, we run the server ***and*** client on the same machine.

In real cases, we typically deploy a Velociraptor server in the cloud.

Architecture overview



Installing Velociraptor

Download the Windows MSI from our releases page:

https://github.com/Velocidex/velociraptor/releases

On Windows, double-click the MSI to install.

Or run:

```
C:> msiexec /i velociraptor.msi
```

Note: You can try other OS versions, but today we'll use Windows.

Configuring Velociraptor

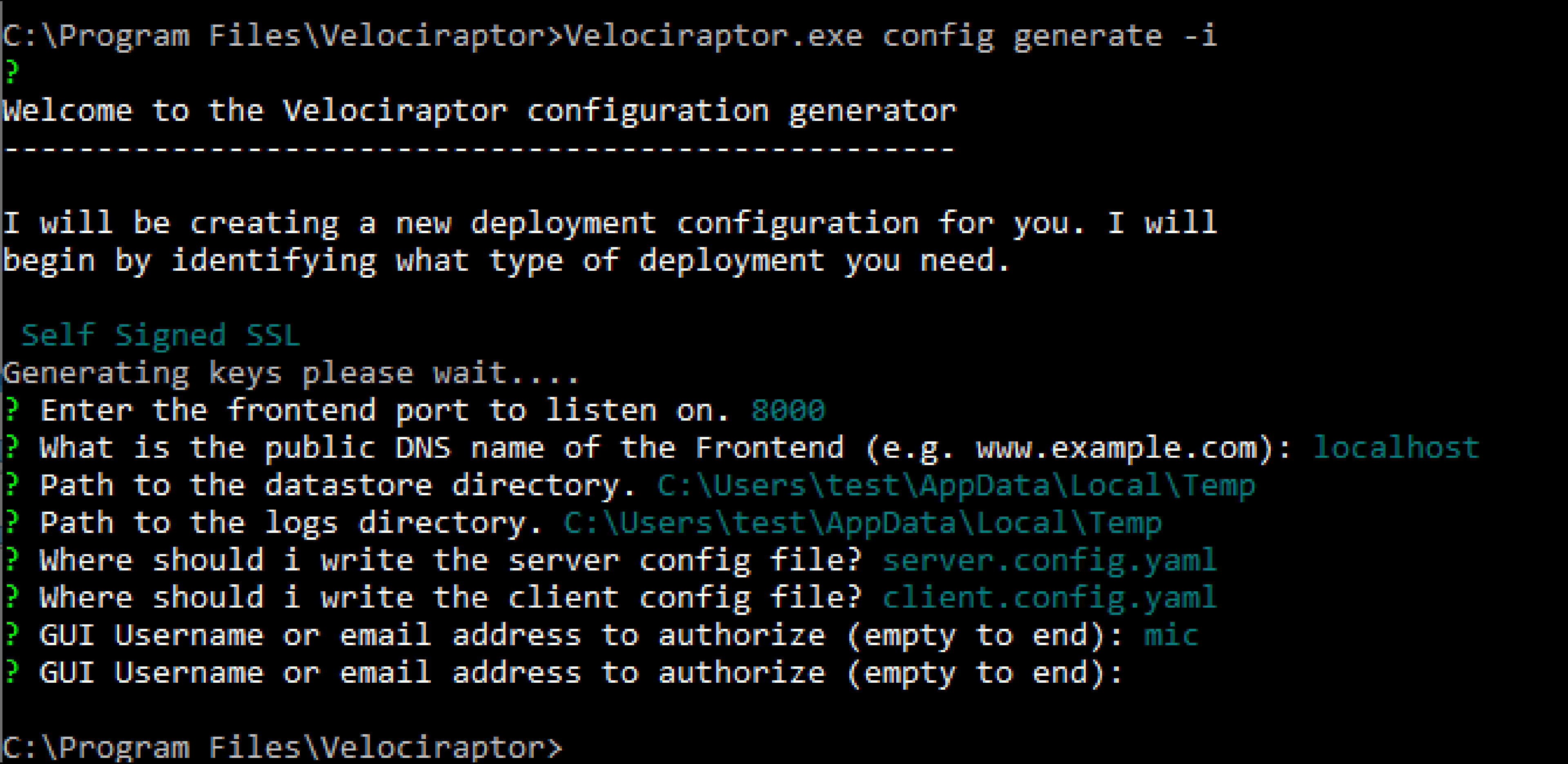
Everything is controlled by a pair of configuration files.

The configuration files contain key data, making them unique (and secure) to your deployment.

The server configuration file contains private keys - *make sure to secure it!*

Genering new configuration files is easy:

```
C:> cd “c:\Program Files\Velociraptor”
C:> Velociraptor.exe config generate -i
```



```
C:\Program Files\Velociraptor>Velociraptor.exe config generate -i
?
Welcome to the Velociraptor configuration generator
---------------------------------------------------


I will be creating a new deployment configuration for you. I will
begin by identifying what type of deployment you need.

 Self Signed SSL
Generating keys please wait....
? Enter the frontend port to listen on. 8000
? What is the public DNS name of the Frontend (e.g. www.example.com): localhost
? Path to the datastore directory. C:\Users\test\AppData\Local\Temp
? Path to the logs directory. C:\Users\test\AppData\Local\Temp
? Where should i write the server config file? server.config.yaml
? Where should i write the client config file? client.config.yaml
? GUI Username or email address to authorize (empty to end): mic
? GUI Username or email address to authorize (empty to end):

C:\Program Files\Velociraptor>
```

Starting the server

The same binary acts as a server or client depending on configuration options.

The previous step generated two files:

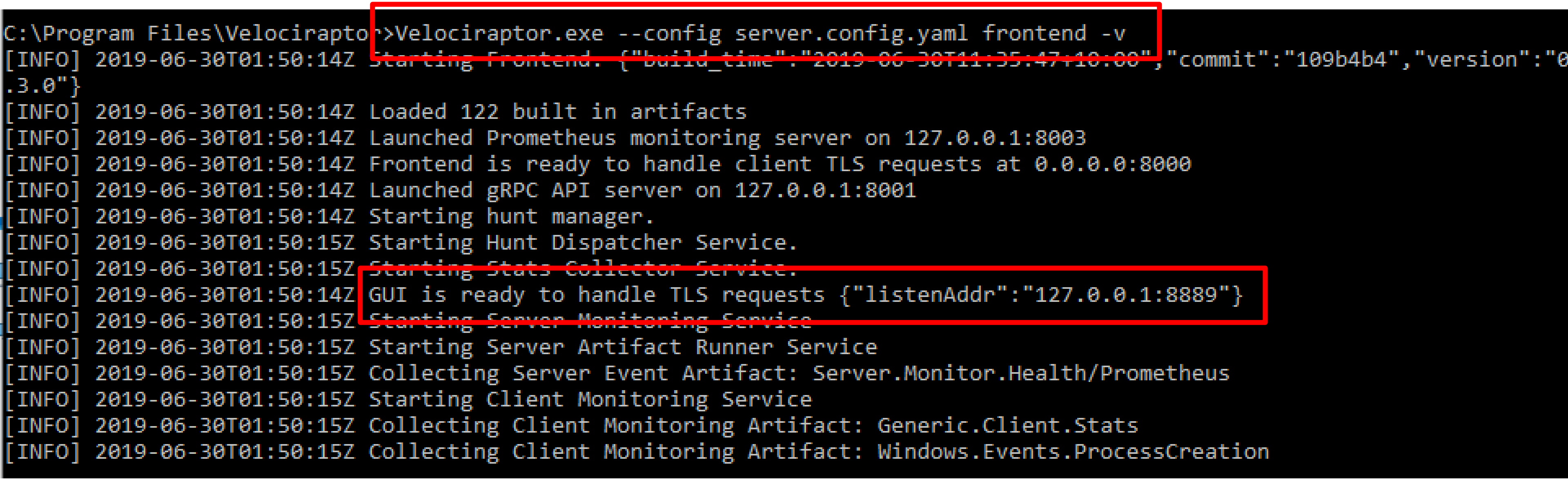
```
client.config.yaml
    server.config.yaml
```

Open two **Command Prompt** windows as administrator.

Start the Velociraptor server and frontend:

```
velociraptor.exe --config server.config.yaml frontend -v
```

Starting the server



```
C:\Program Files\Velociraptor>Velociraptor.exe --config server.config.yaml frontend -v
[INFO] 2019-06-30T01:50:14Z Starting Frontend: {"build_time":"2019-06-30T11:35:47+10:00","commit":"109b4b4","version":"0
.3.0"}
[INFO] 2019-06-30T01:50:14Z Loaded 122 built in artifacts
[INFO] 2019-06-30T01:50:14Z Launched Prometheus monitoring server on 127.0.0.1:8003
[INFO] 2019-06-30T01:50:14Z Frontend is ready to handle client TLS requests at 0.0.0.0:8000
[INFO] 2019-06-30T01:50:14Z Launched gRPC API server on 127.0.0.1:8001
[INFO] 2019-06-30T01:50:14Z Starting hunt manager.
[INFO] 2019-06-30T01:50:15Z Starting Hunt Dispatcher Service.
[INFO] 2019-06-30T01:50:15Z Starting Stats Collector Service.
[INFO] 2019-06-30T01:50:14Z GUI is ready to handle TLS requests {"listenAddr":"127.0.0.1:8889"}
[INFO] 2019-06-30T01:50:15Z Starting Server Monitoring Service
[INFO] 2019-06-30T01:50:15Z Starting Server Artifact Runner Service
[INFO] 2019-06-30T01:50:15Z Collecting Server Event Artifact: Server.Monitor.Health/Prometheus
[INFO] 2019-06-30T01:50:15Z Starting Client Monitoring Service
[INFO] 2019-06-30T01:50:15Z Collecting Client Monitoring Artifact: Generic.Client.Stats
[INFO] 2019-06-30T01:50:15Z Collecting Client Monitoring Artifact: Windows.Events.ProcessCreation
```

Test that the GUI works

Connect to the GUI address mentioned previously:

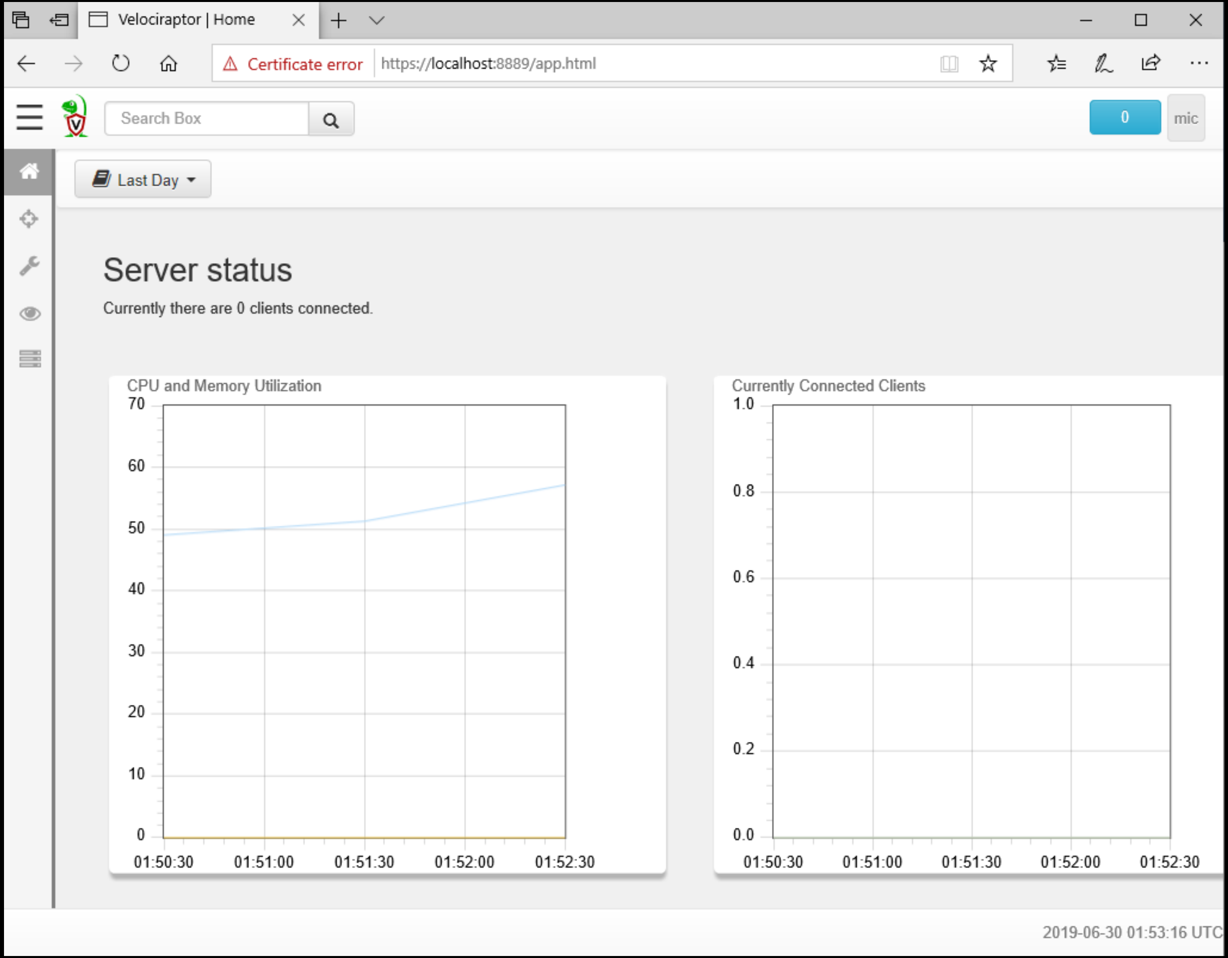
https://localhost:8889/

Note the certificate error - *this is OK*. It's because we chose self-signed SSL mode. You can click through the warning for now.

In real deployments we use proper SSL certificates.

The server is ready.

Now let’s configure some clients.



Server status

Currently there are 0 clients connected.

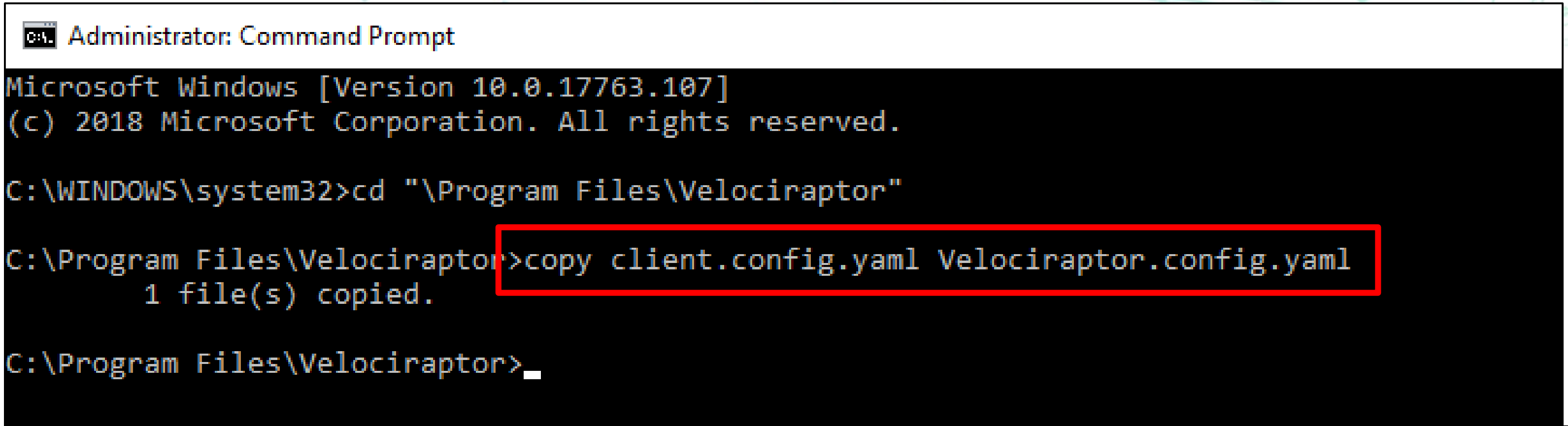
Starting a client

In Windows, installing the Velociraptor MSI installs a client service.

The service needs the client configuration file.

Simply move the client configuration file into plase (see next slide).

When deploying at scale, you can use **SCCM** or **Group Policy** to do this - today we simply use Windows Explorer or the shell.



Administrator: Command Prompt

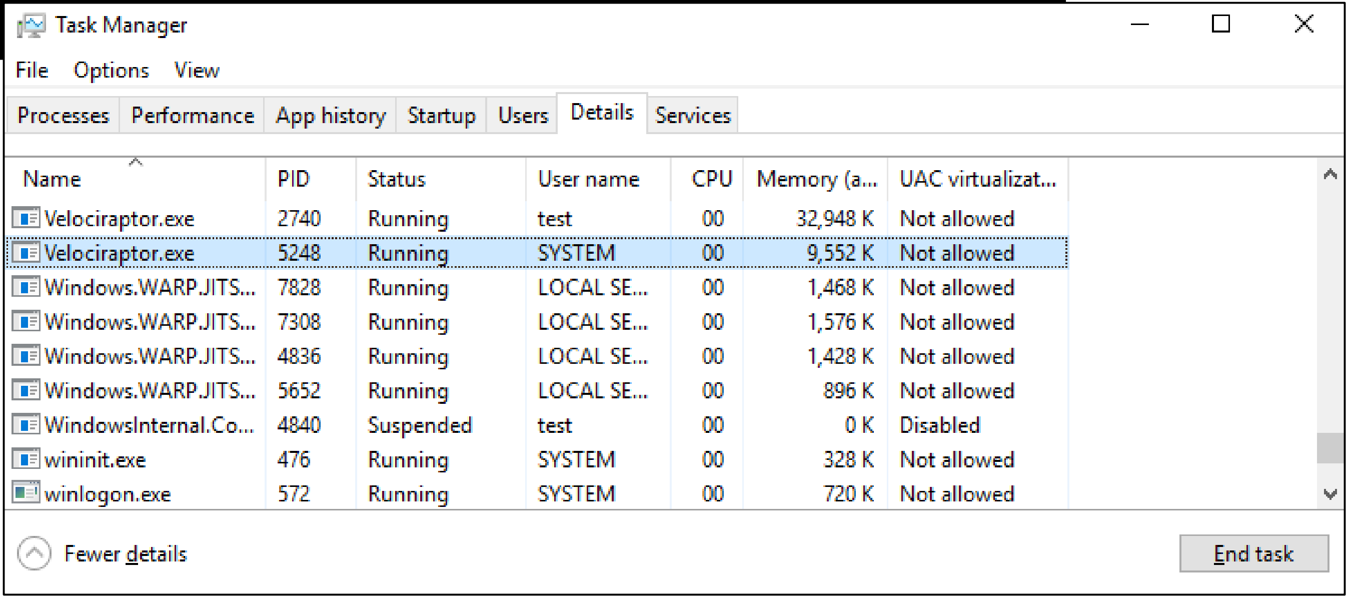
```
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd "\Program Files\Velociraptor"

C:\Program Files\Velociraptor>copy client.config.yaml Velociraptor.config.yaml
        1 file(s) copied.

C:\Program Files\Velociraptor>_
```

Copy the client configuration file, then start the Velociraptor service.



Name	PID	Status	User name	CPU	Memory (a...	UAC virtualizat...
Velociraptor.exe	2740	Running	test	00	32,948 K	Not allowed
Velociraptor.exe	5248	Running	SYSTEM	00	9,552 K	Not allowed
Windows.WARP.JITS...	7828	Running	LOCAL SE...	00	1,468 K	Not allowed
Windows.WARP.JITS...	7308	Running	LOCAL SE...	00	1,576 K	Not allowed
Windows.WARP.JITS...	4836	Running	LOCAL SE...	00	1,428 K	Not allowed
Windows.WARP.JITS...	5652	Running	LOCAL SE...	00	896 K	Not allowed
WindowsInternal.Co...	4840	Suspended	test	00	0 K	Disabled
wininit.exe	476	Running	SYSTEM	00	328 K	Not allowed
winlogon.exe	572	Running	SYSTEM	00	720 K	Not allowed

The Dashboard

The **Dashboard** shows the current state of the installation:

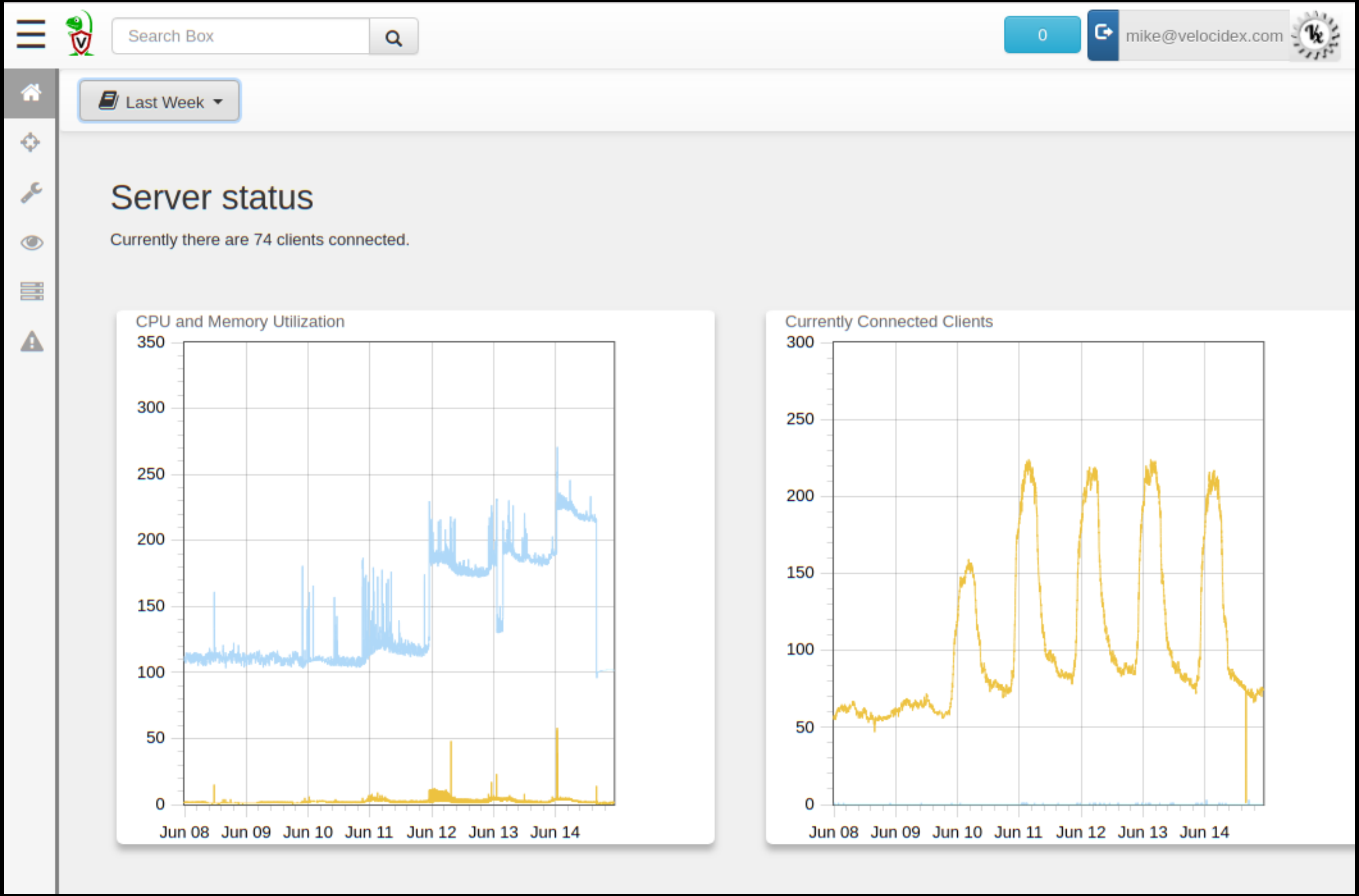
- How many clients are connected
- Current CPU load and memory footprint on the server.

When running hunts or intensive processing, memory and CPU requirements will increase but not too much.

You can customize the dashboard - it’s also just an artifact.

Clients have a persistent connection to the server.

They're ready to receive your commands.



Server status

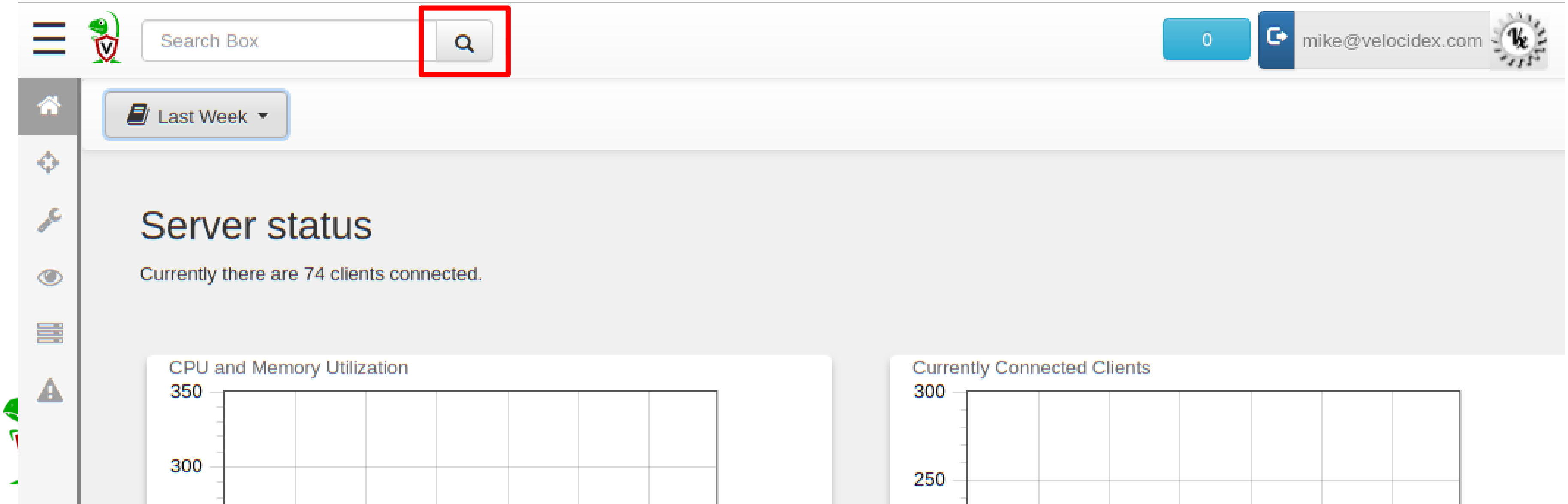
Currently there are 74 clients connected.

Interactive investigations on individual clients

Searching for a client

Sometimes we want to see information about a client.

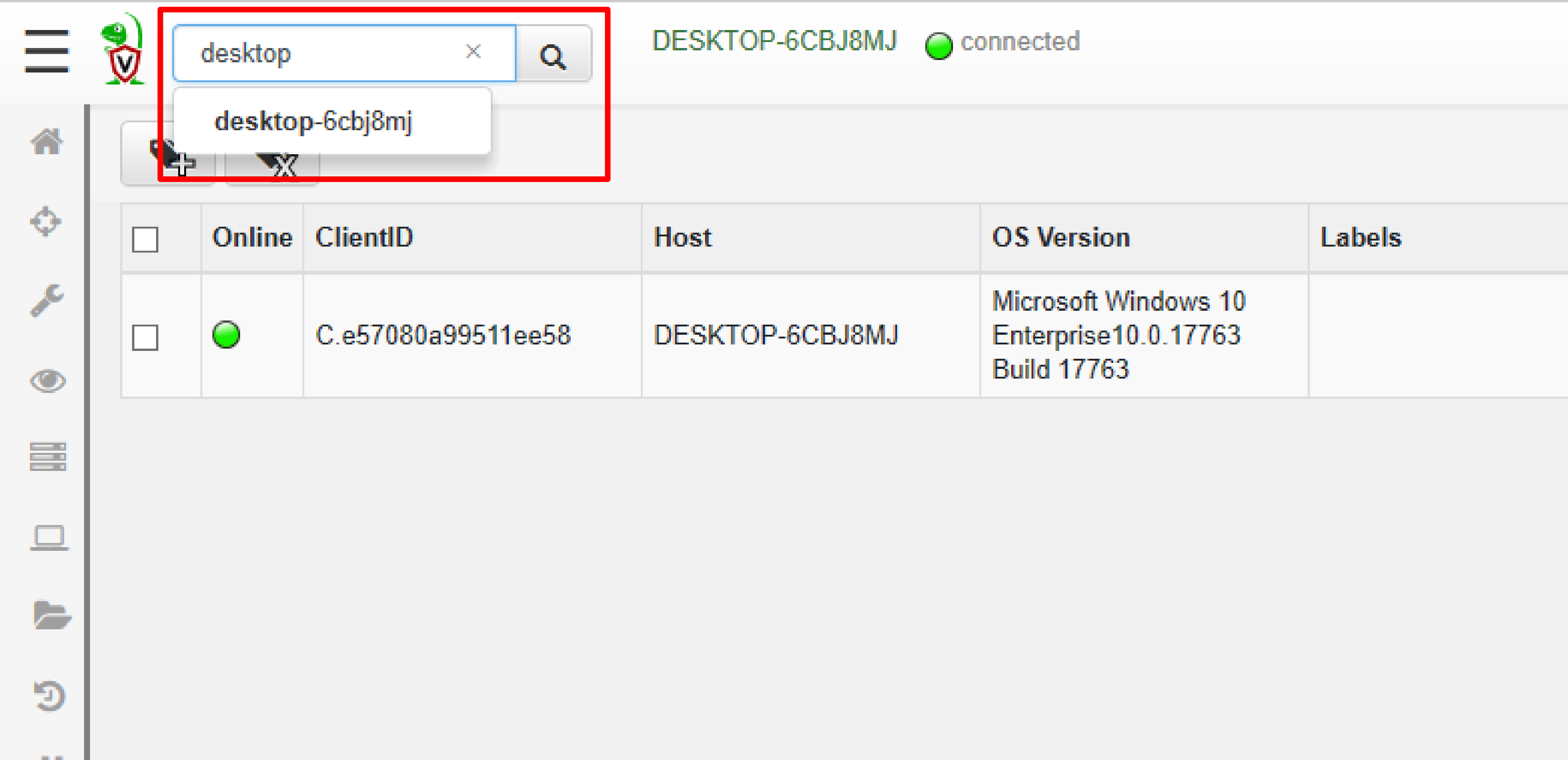
Press the **Search** icon to see all the clients



Server status

Currently there are 74 clients connected.

Or search for clients by hostname, label or client ID.



DESKTOP-6CBJ8MJ ● connected

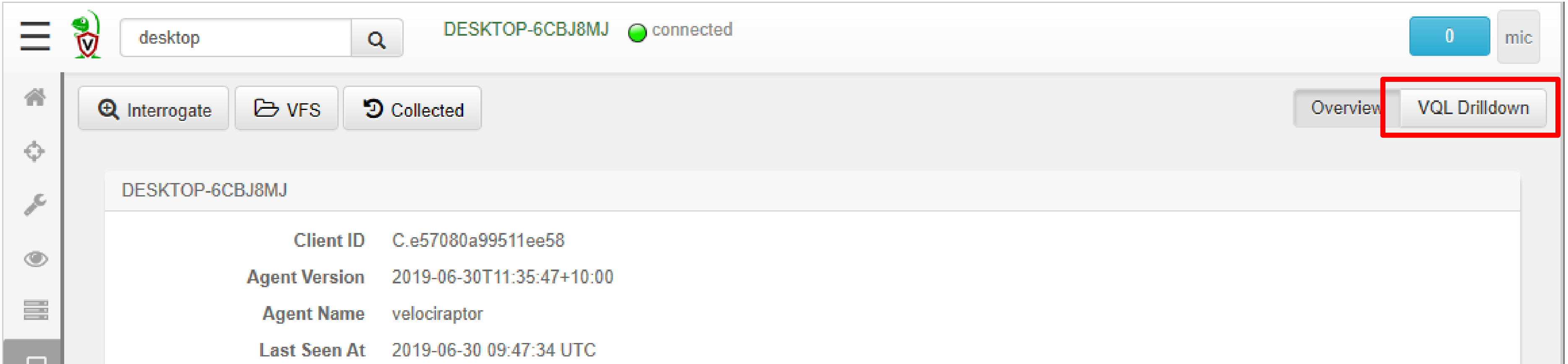
□	Online	ClientID	Host	OS Version	Labels
□	●	C.e57080a99511ee58	DESKTOP-6CBJ8MJ	Microsoft Windows 10 Enterprise10.0.17763 Build 17763	

Client overview

This provides some general information about a client.

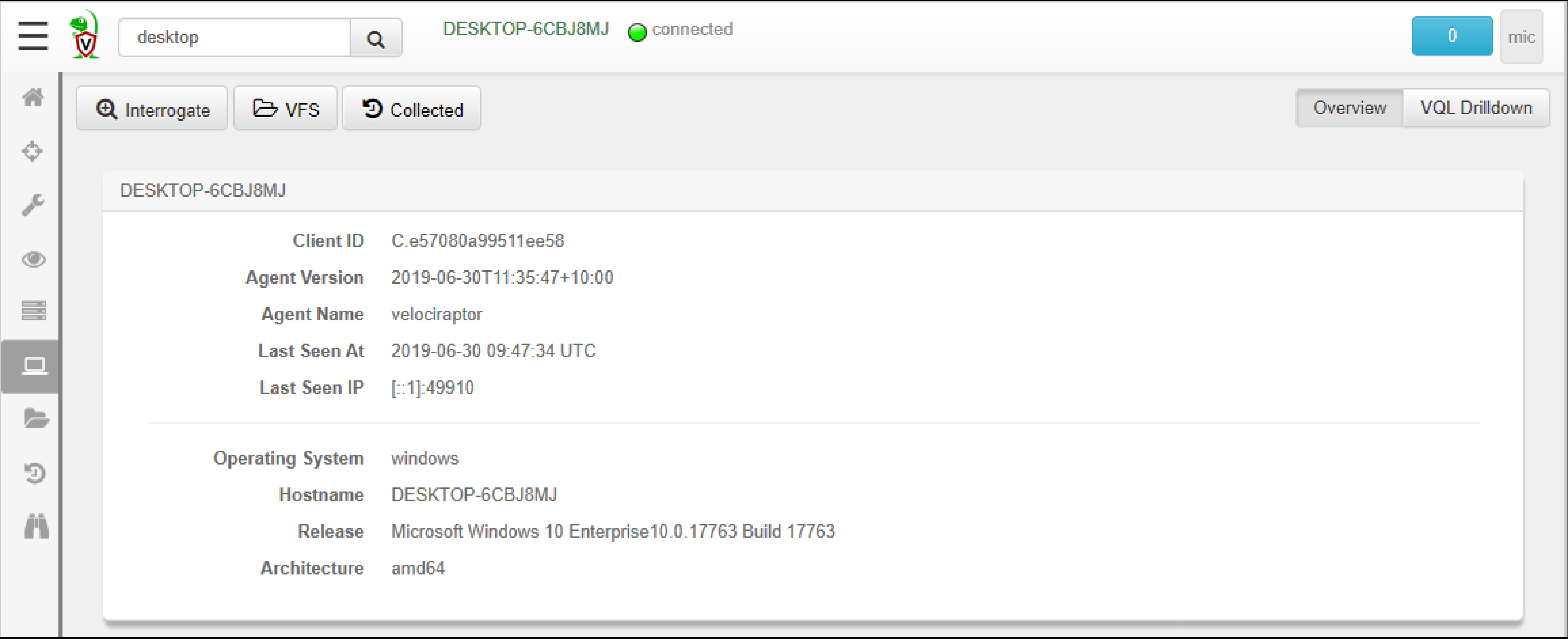
Click **VQL Drilldown** to see more detailed information.

You can customize the information collected and shown by editing the configuration file, to add extra VQL queries.



DESKTOP-6CBJ8MJ

Client ID	C.e57080a99511ee58
Agent Version	2019-06-30T11:35:47+10:00
Agent Name	velociraptor
Last Seen At	2019-06-30 09:47:34 UTC



DESKTOP-6CBJ8MJ

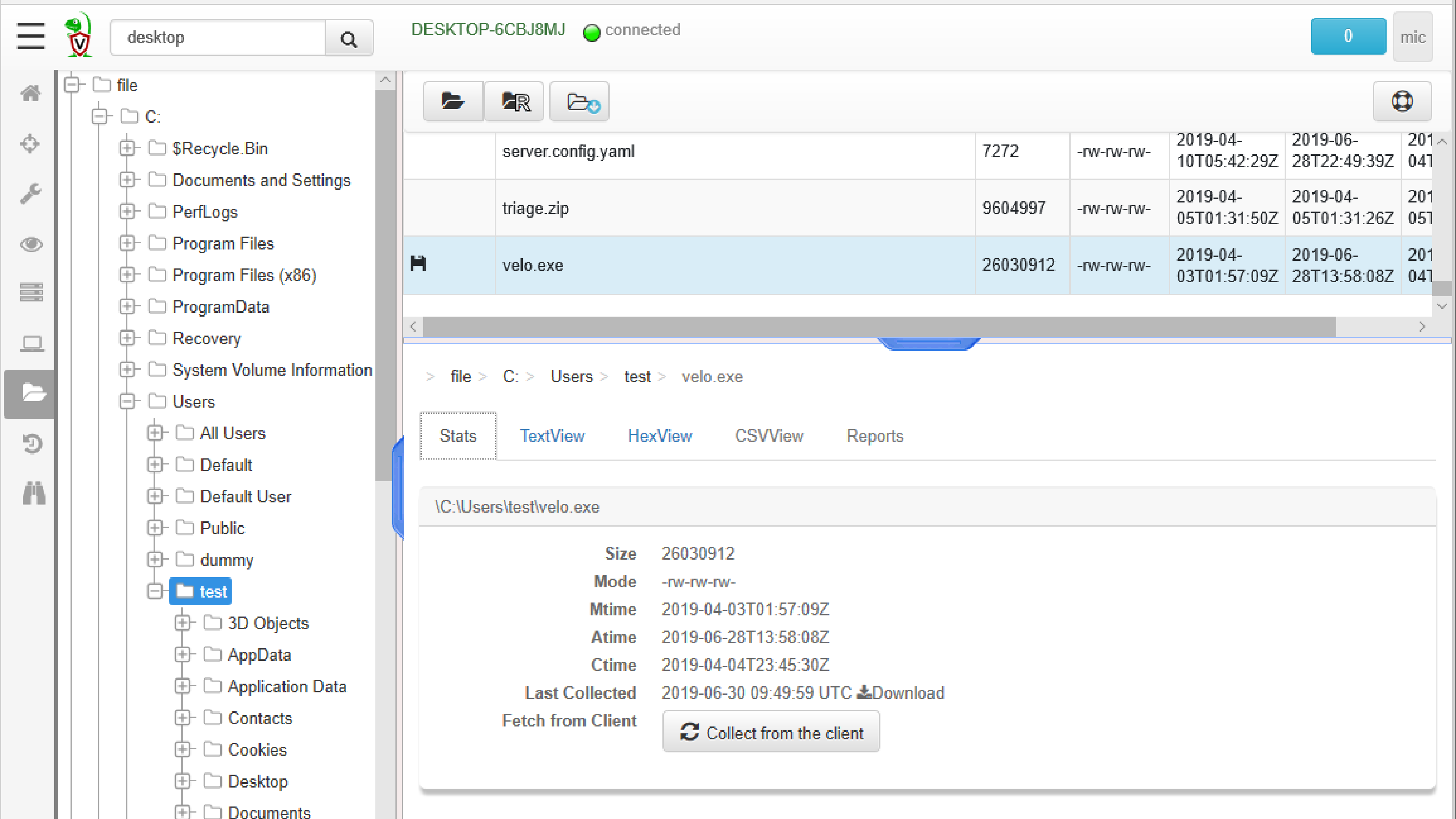
Client ID	C.e57080a99511ee58
Agent Version	2019-06-30T11:35:47+10:00
Agent Name	velociraptor
Last Seen At	2019-06-30 09:47:34 UTC
Last Seen IP	[::1]:49910
Operating System	windows
Hostname	DESKTOP-6CBJ8MJ
Release	Microsoft Windows 10 Enterprise10.0.17763 Build 17763
Architecture	amd64

The Virtual File System (VFS)

The VFS visualizes some server-side information we collect about the clients.

Top level corresponds to the type of information we collect:

- **File** - Access the file system using the filesystem API
- **NTFS** - Access the file system using raw NTFS parsing
- **Registry** - Access the Windows Registry using the Registry API
- **Artifacts** - A view of all artifacts collected from the client.



	server.config.yaml	7272	-rw-rw-rw-	2019-04-10T05:42:29Z	2019-06-28T22:49:39Z	201 04T
	triage.zip	9604997	-rw-rw-rw-	2019-04-05T01:31:50Z	2019-04-05T01:31:26Z	201 05T
💾	velo.exe	26030912	-rw-rw-rw-	2019-04-03T01:57:09Z	2019-06-28T13:58:08Z	201 04T

\C:\Users\test\velo.exe

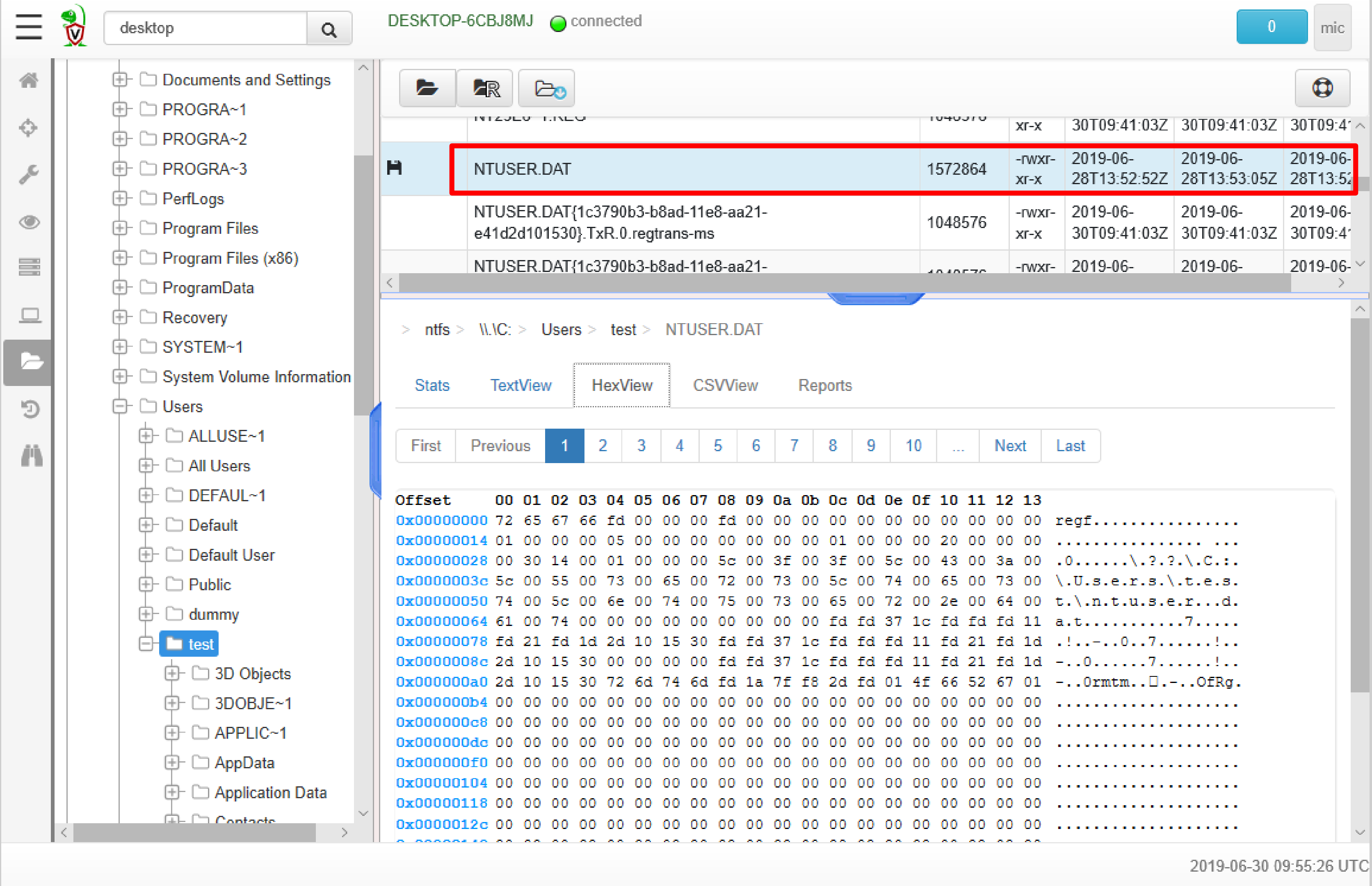
Size	26030912
Mode	-rw-rw-rw-
Mtime	2019-04-03T01:57:09Z
Atime	2019-06-28T13:58:08Z
Ctime	2019-04-04T23:45:30Z
Last Collected	2019-06-30 09:49:59 UTC Download
Fetch from Client	Collect from the client

Exercise: Browse the client file system using VFS

Task: Find your user NTUSER.DAT file and download it locally.

Hunting hints:

- NTUSER.DAT stores the Registry for your user account
- It’s locked when the user is logged in
- Therefore you need to fetch it using raw NTFS access
- Do you know where this file is located?



NTUSER.DAT	1572864	-rwxr-xr-x	2019-06-28T13:52:52Z	2019-06-28T13:53:05Z	2019-06-28T13:52
NTUSER.DAT{1c3790b3-b8ad-11e8-aa21-e41d2d101530}.TxR.0.regtrans-ms	1048576	-rwxr-xr-x	2019-06-30T09:41:03Z	2019-06-30T09:41:03Z	2019-06-30T09:4

```
Offset     00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
0x00000000 72 65 67 66 fd 00 00 00 fd 00 00 00 00 00 00 00 00 00 00 00  regf................
0x00000014 01 00 00 00 05 00 00 00 00 00 00 00 01 00 00 00 20 00 00 00  ................ ...
0x00000028 00 30 14 00 01 00 00 00 5c 00 3f 00 3f 00 5c 00 43 00 3a 00  .0......\.?.?.\.C.:.
0x0000003c 5c 00 55 00 73 00 65 00 72 00 73 00 5c 00 74 00 65 00 73 00  \.U.s.e.r.s.\.t.e.s.
0x00000050 74 00 5c 00 6e 00 74 00 75 00 73 00 65 00 72 00 2e 00 64 00  t.\.n.t.u.s.e.r...d.
0x00000064 61 00 74 00 00 00 00 00 00 00 00 00 fd fd 37 1c fd fd fd 11  a.t...........7.....
0x00000078 fd 21 fd 1d 2d 10 15 30 fd fd 37 1c fd fd fd 11 fd 21 fd 1d  .!..-..0..7......!..
0x0000008c 2d 10 15 30 00 00 00 00 fd fd 37 1c fd fd fd 11 fd 21 fd 1d  -..0......7......!..
0x000000a0 2d 10 15 30 72 6d 74 6d fd 1a 7f f8 2d fd 01 4f 66 52 67 01  -..0rmtm..□.-..OfRg.
0x000000b4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
0x000000c8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
0x000000dc 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
0x000000f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
0x00000104 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
0x00000118 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
0x0000012c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
```

2019-06-30 09:55:26 UTC

Use Velociraptor artifacts to automate everything

Use Velociraptor artifacts to automate everything

We can collect information about *many* things in DFIR cases:

- Registry keys, files, WMI queries, sqlite databases …

But we really just want to answer specific questions:

- What program did the attacker run?
- What files were downloaded?
- What DNS lookups occured?
- Did a particular file exist on a client?

Velociraptor uses expert knowledge to find the evidence

A key objective of Velociraptor is encapsulating DFIR knowledge into the tool

- We have high level questions to answer
- We know where to look for evidence of user / system activities

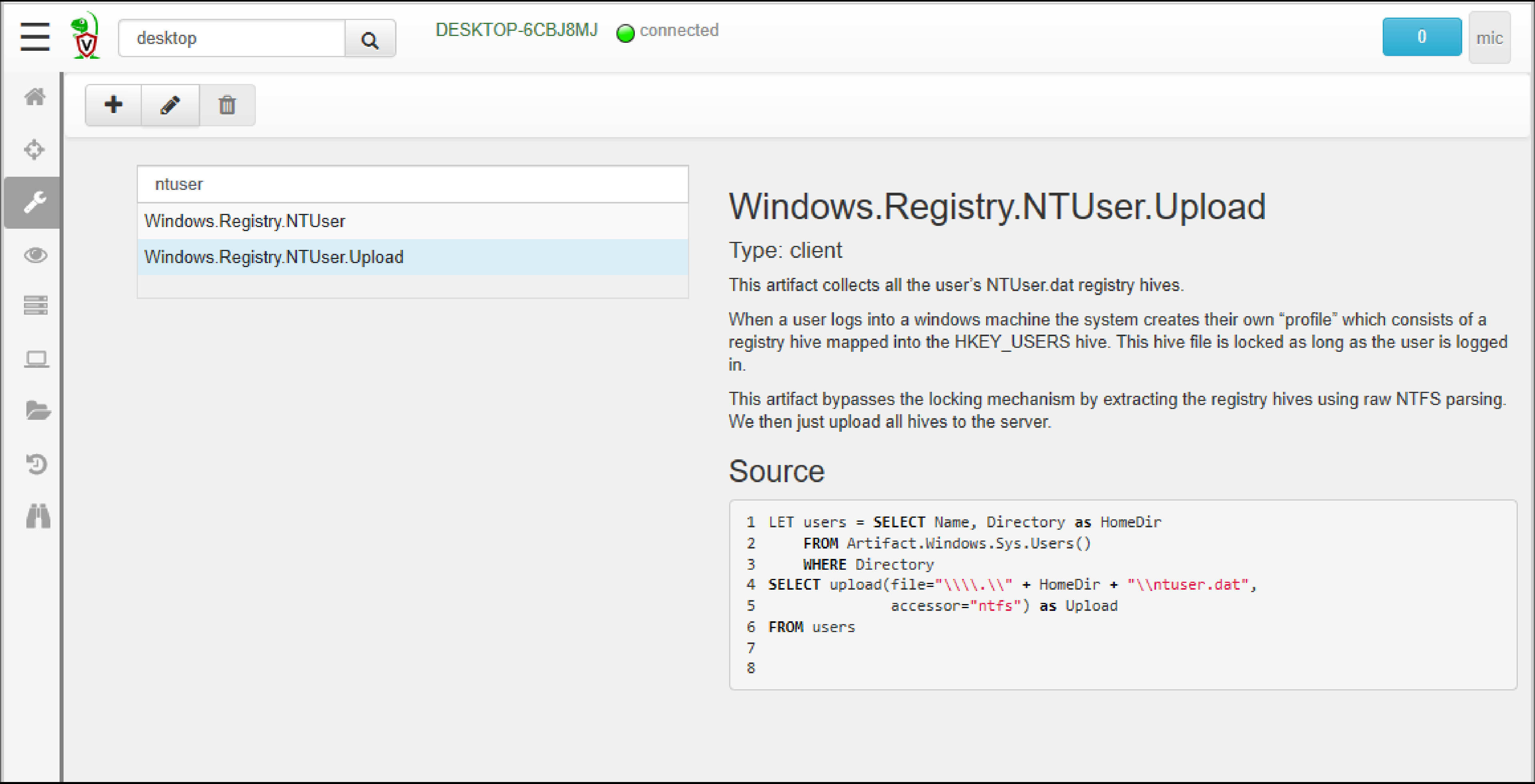
We build artifacts to collect and analyze the evidence in order to answer our investigative questions.

Velociraptor's unique feature - user specified artifacts

An artifact is a YAML file …

- (therefore user-readable, shareable and editable)
- … that answers a question …
- … by collecting data from the endpoint …
- … and reporting on this data in a human readable way.

Artifacts encode expert knowledge into human reusable components.



Windows.Registry.NTUser.Upload

Type: client

This artifact collects all the user's NTUser.dat registry hives.

When a user logs into a windows machine the system creates their own "profile" which consists of a registry hive mapped into the HKEY_USERS hive. This hive file is locked as long as the user is logged in.

This artifact bypasses the locking mechanism by extracting the registry hives using raw NTFS parsing. We then just upload all hives to the server.

Source

```
LET users = SELECT Name, Directory as HomeDir
    FROM Artifact.Windows.Sys.Users()
    WHERE Directory
SELECT upload(file="\\\\.\\" + HomeDir + "\\ntuser.dat",
              accessor="ntfs") as Upload
FROM users


```

Exercise: Collect all user NTUSER.DAT files

Previously we downloaded one NTUSER.DAT - let’s get them all.

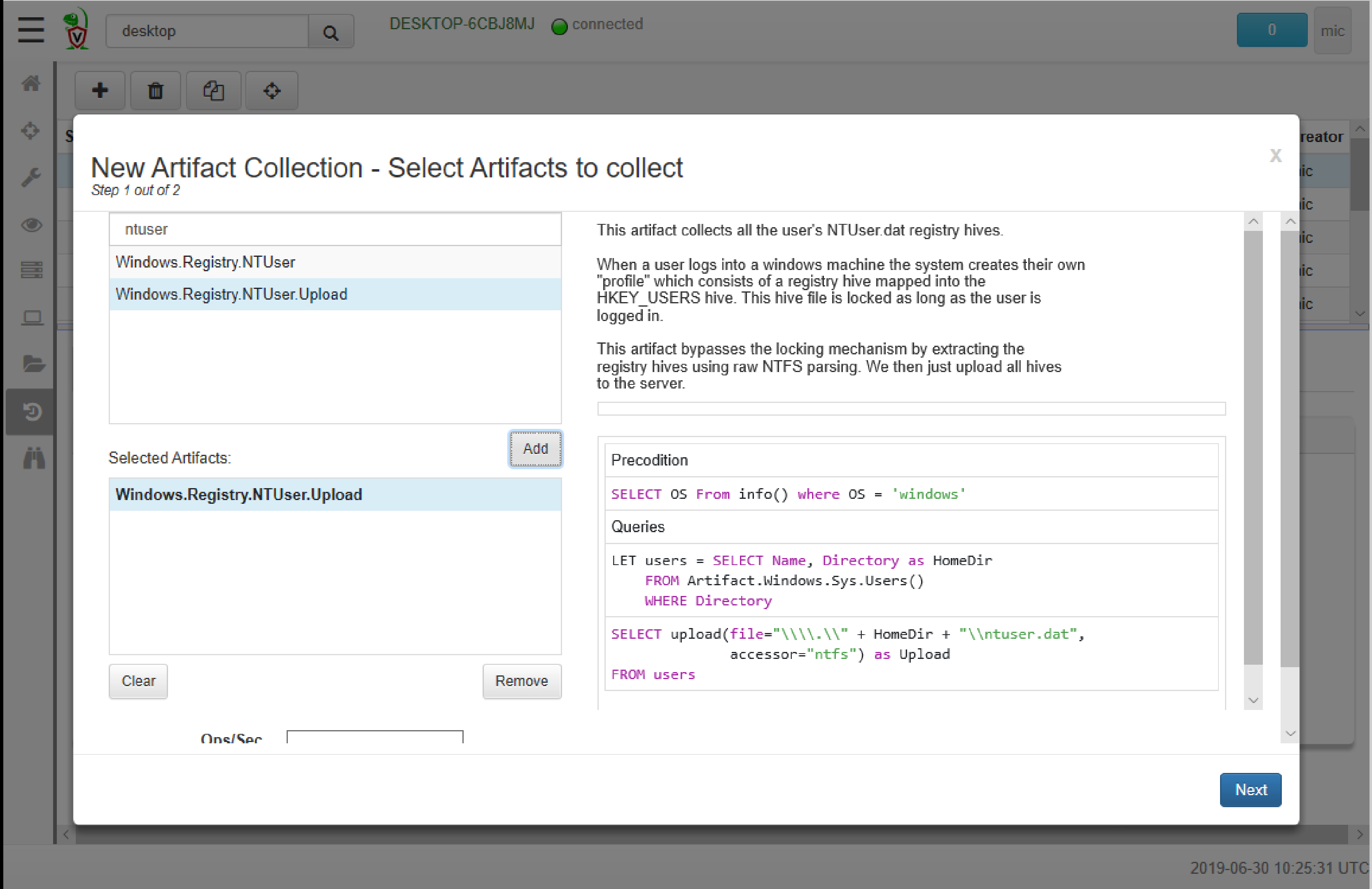
Select **Collected Artifacts** to view all artifacts previously collected.

Click **Collect More Artifacts** to open the **New Artifact Wizard**.

Search for an artifact that fetches NTUSER.DAT files.

Click **Add** to add the artifact to the list for collection.

Click **Next** to start the collection.



New Artifact Collection - Select Artifacts to collect

Step 1 out of 2

ntuser

Windows.Registry.NTUser

Windows.Registry.NTUser.Upload

Selected Artifacts:

Windows.Registry.NTUser.Upload

This artifact collects all the user's NTUser.dat registry hives.

When a user logs into a windows machine the system creates their own "profile" which consists of a registry hive mapped into the HKEY_USERS hive. This hive file is locked as long as the user is logged in.

This artifact bypasses the locking mechanism by extracting the registry hives using raw NTFS parsing. We then just upload all hives to the server.

Precondition

```
SELECT OS From info() where OS = 'windows'
```

Queries

```
LET users = SELECT Name, Directory as HomeDir
    FROM Artifact.Windows.Sys.Users()
    WHERE Directory
```

```
SELECT upload(file="\\\\.\\" + HomeDir + "\\ntuser.dat",
              accessor="ntfs") as Upload
FROM users
```

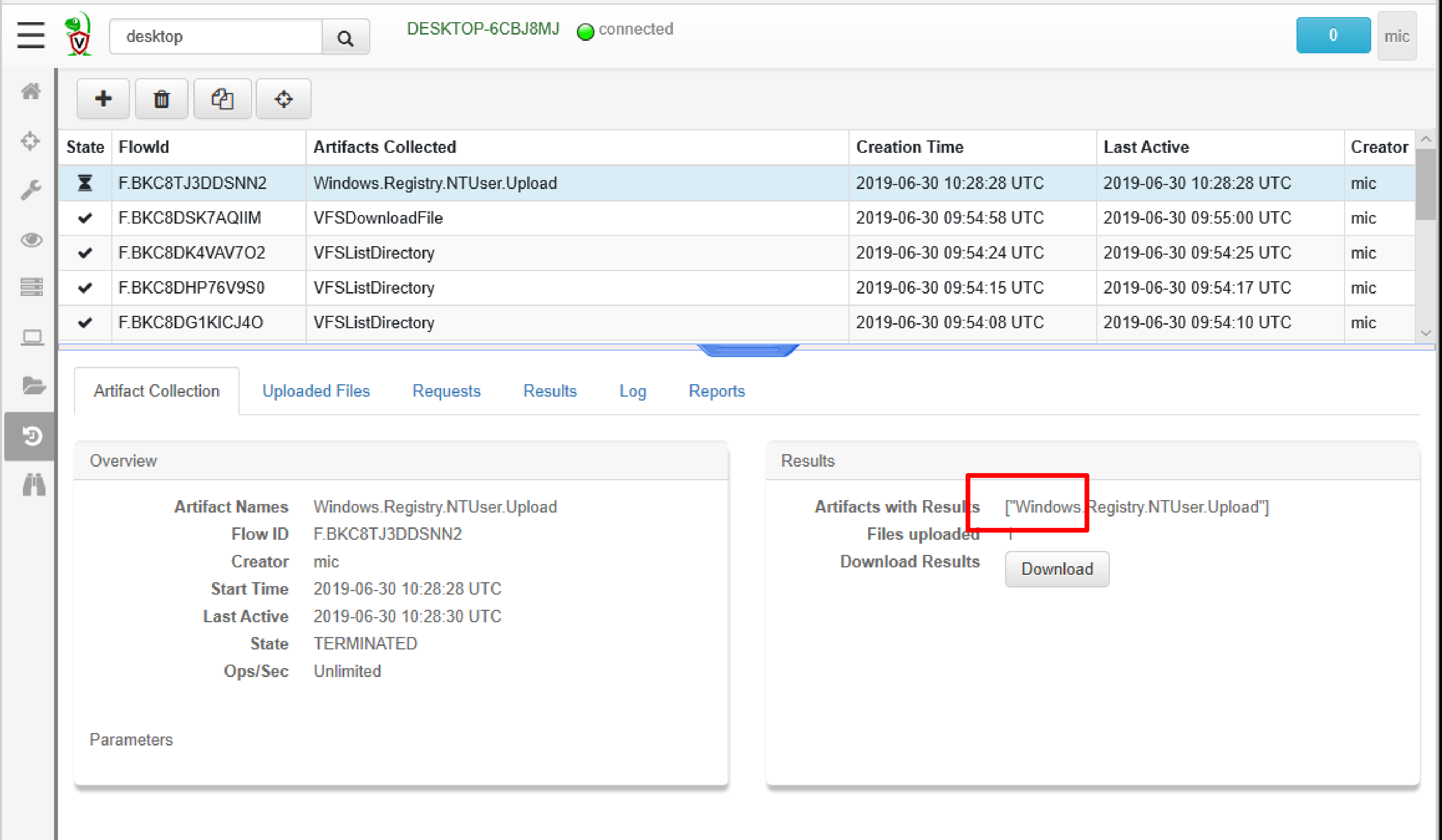
2019-06-30 10:25:31 UTC

Get the collected data

One file will be downloaded for every user on the client.

Click **Download** to download the results of this artifact collection through your web browser (see next slide).

The result is a ZIP file with the collected files (NTUSER.DAT) and a CSV file of the collection results.



State	FlowId	Artifacts Collected	Creation Time	Last Active	Creator
⌛	F.BKC8TJ3DDSNN2	Windows.Registry.NTUser.Upload	2019-06-30 10:28:28 UTC	2019-06-30 10:28:28 UTC	mic
✓	F.BKC8DSK7AQIIM	VFSDownloadFile	2019-06-30 09:54:58 UTC	2019-06-30 09:55:00 UTC	mic
✓	F.BKC8DK4VAV7O2	VFSListDirectory	2019-06-30 09:54:24 UTC	2019-06-30 09:54:25 UTC	mic
✓	F.BKC8DHP76V9S0	VFSListDirectory	2019-06-30 09:54:15 UTC	2019-06-30 09:54:17 UTC	mic
✓	F.BKC8DG1KICJ4O	VFSListDirectory	2019-06-30 09:54:08 UTC	2019-06-30 09:54:10 UTC	mic

Overview

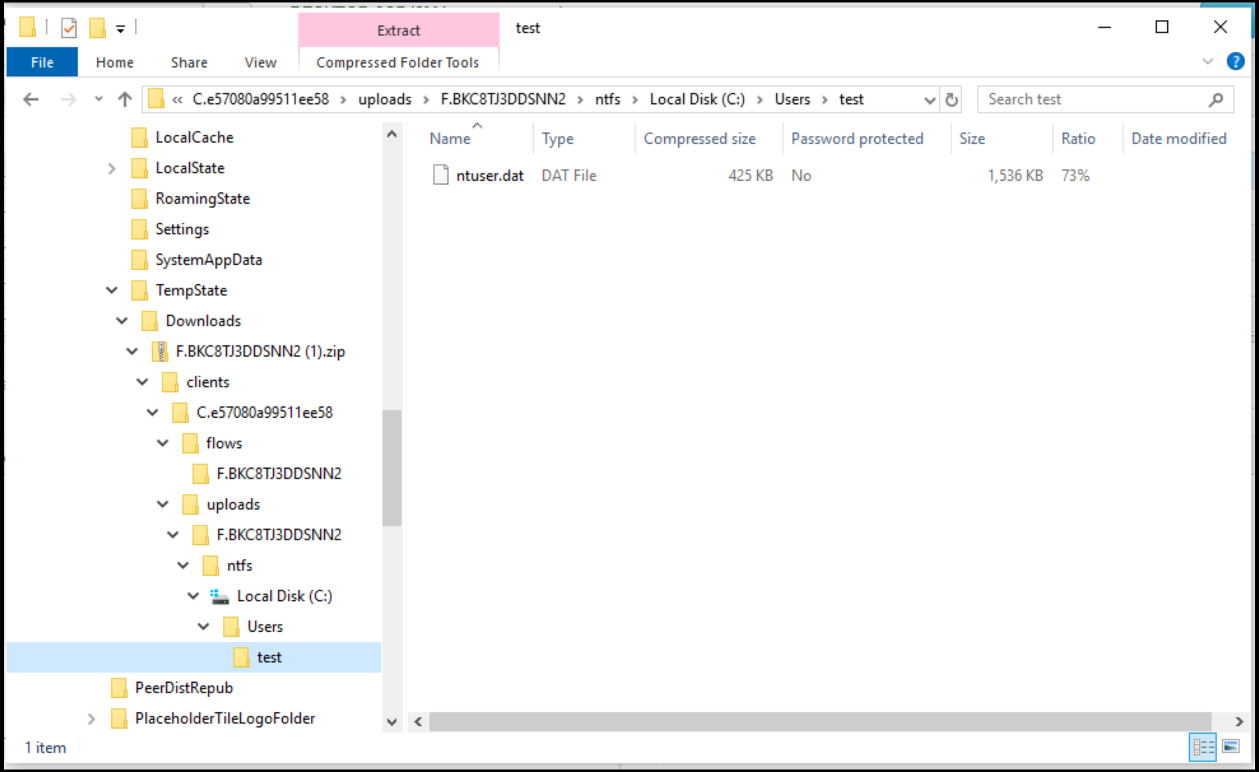
Artifact Names	Windows.Registry.NTUser.Upload
Flow ID	F.BKC8TJ3DDSNN2
Creator	mic
Start Time	2019-06-30 10:28:28 UTC
Last Active	2019-06-30 10:28:30 UTC
State	TERMINATED
Ops/Sec	Unlimited

Parameters

Results

Artifacts with Results	["Windows.Registry.NTUser.Upload"]
Files uploaded	1
Download Results	Download

The ZIP file contains a directory structure for each client mirroring the original directory structure on the client.



« C.e57080a99511ee58 › uploads › F.BKC8TJ3DDSNN2 › ntfs › Local Disk (C:) › Users › test

- LocalCache
- LocalState
- RoamingState
- Settings
- SystemAppData
- TempState
  - Downloads
    - F.BKC8TJ3DDSNN2 (1).zip
      - clients
        - C.e57080a99511ee58
          - flows
            - F.BKC8TJ3DDSNN2
          - uploads
            - F.BKC8TJ3DDSNN2
              - ntfs
                - Local Disk (C:)
                  - Users
                    - test
- PeerDistRepub
- PlaceholderTileLogoFolder

Name	Type	Compressed size	Password protected	Size	Ratio	Date modified
ntuser.dat	DAT File	425 KB	No	1,536 KB	73%	

Hunting across the whole network

Hunting is the collection of artifacts across the network

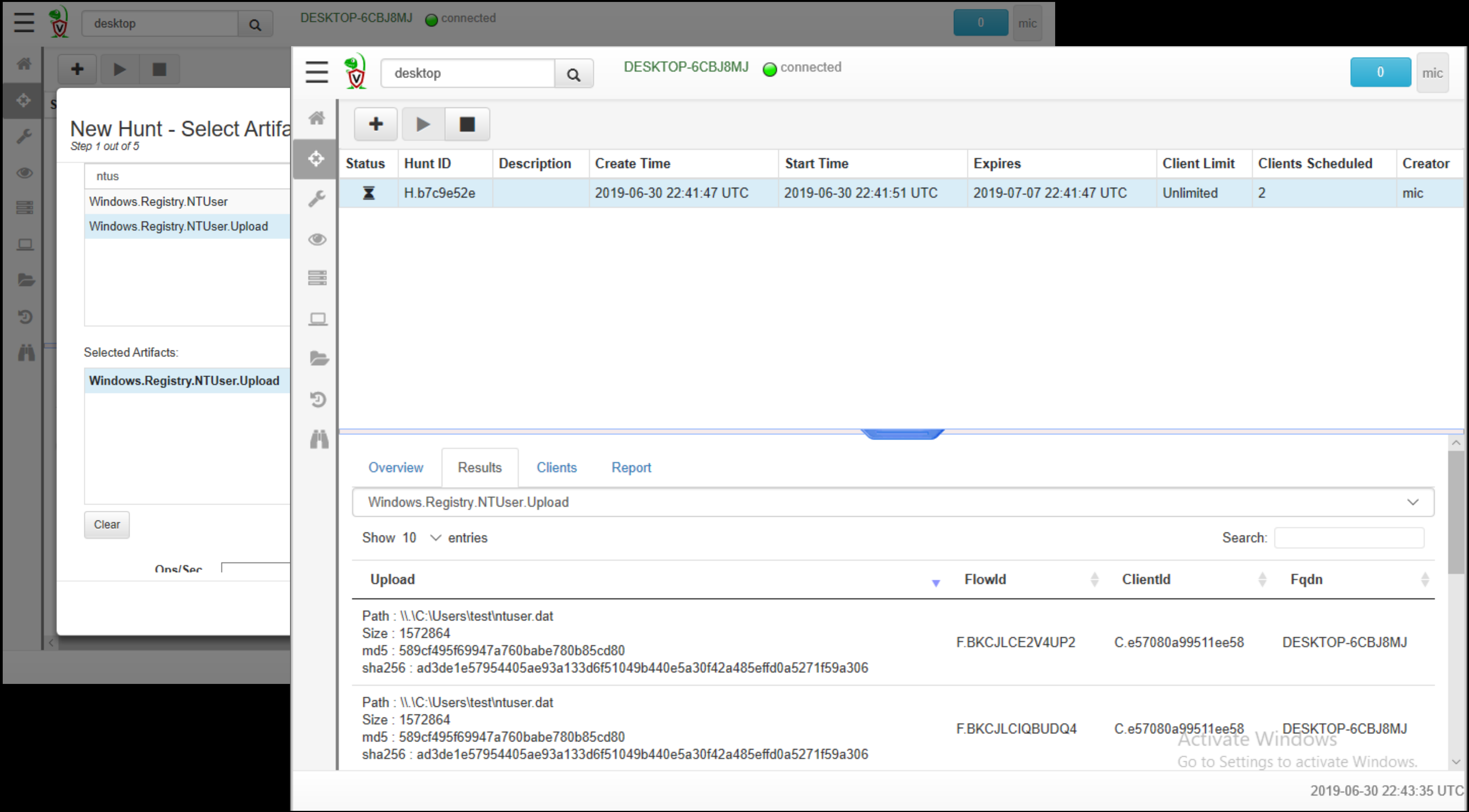
Any artifact that can be collected on a single computer, can be hunted across the network.

A hunt can cover a group of clients, or the whole network.

A hunt will continue running until it expires, or is stopped.

As new machines appear, they automatically join in the hunt.

Downloading the hunt results generates a ZIP file with all the uploaded files (in this exercise NTUSER.DAT files).



New Hunt - Select Artifa

Step 1 out of 5

ntus

Windows.Registry.NTUser

Windows.Registry.NTUser.Upload

Selected Artifacts:

Windows.Registry.NTUser.Upload

Status	Hunt ID	Description	Create Time	Start Time	Expires	Client Limit	Clients Scheduled	Creator
⧗	H.b7c9e52e		2019-06-30 22:41:47 UTC	2019-06-30 22:41:51 UTC	2019-07-07 22:41:47 UTC	Unlimited	2	mic

Windows.Registry.NTUser.Upload

Show 10 entries

Search:

Upload	FlowId	ClientId	Fqdn
Path : \\.\C:\Users\test\ntuser.dat Size : 1572864 md5 : 589cf495f69947a760babe780b85cd80 sha256 : ad3de1e57954405ae93a133d6f51049b440e5a30f42a485effd0a5271f59a306	F.BKCJLCE2V4UP2	C.e57080a99511ee58	DESKTOP-6CBJ8MJ
Path : \\.\C:\Users\test\ntuser.dat Size : 1572864 md5 : 589cf495f69947a760babe780b85cd80 sha256 : ad3de1e57954405ae93a133d6f51049b440e5a30f42a485effd0a5271f59a306	F.BKCJLCIQBUDQ4	C.e57080a99511ee58	DESKTOP-6CBJ8MJ

2019-06-30 22:43:35 UTC

Surgical collection of evidence

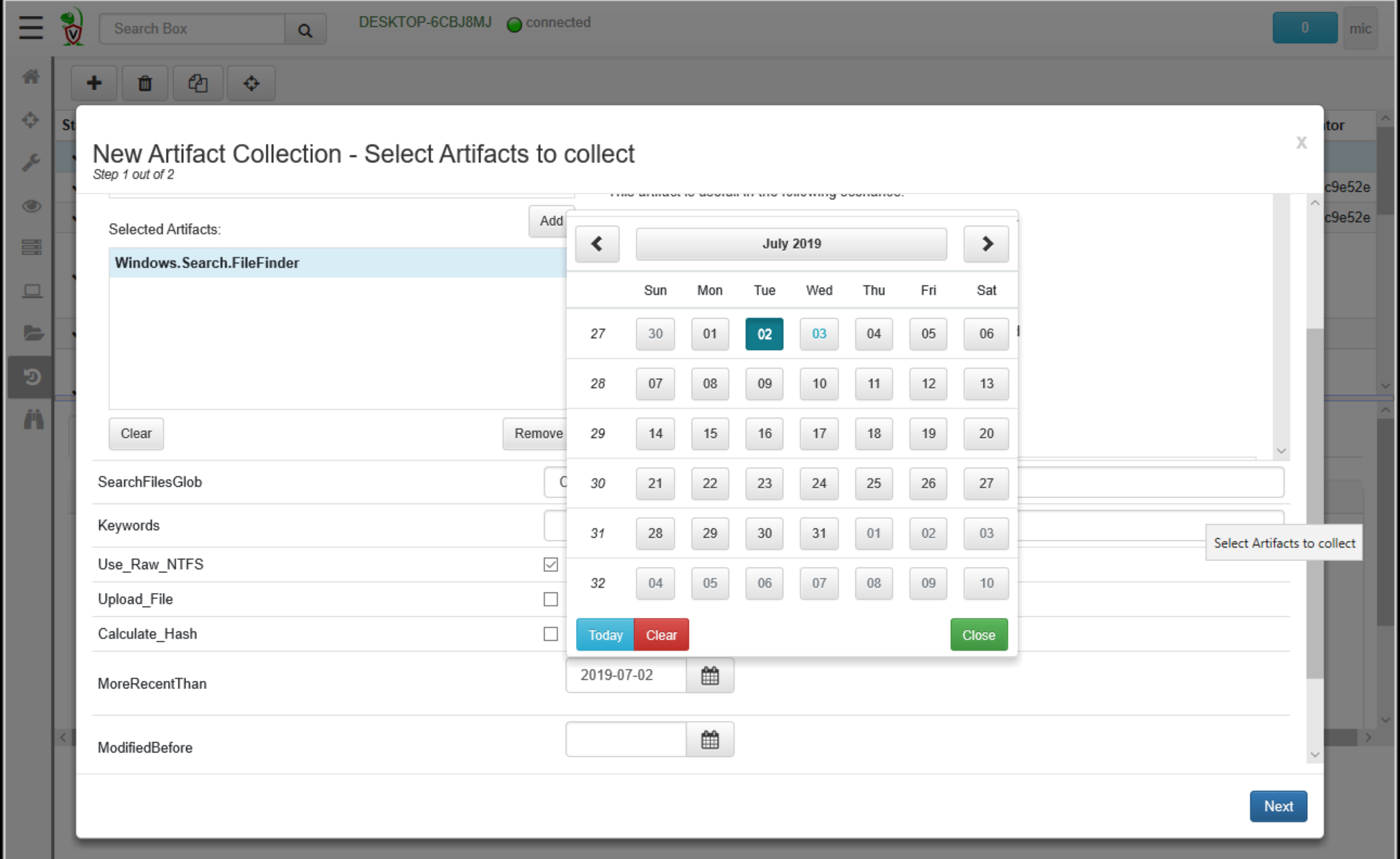
Finding files

Searching for files is a fundamental capability.

Velociraptor provides a powerful **File Finder** artifact for this.

- Use wildcards to ‘glob’ over directories
- Use Yara to search the contents of files for keywords
- Filter by modified or created dates
- Upload matching files to the server, for further analysis.

The **Windows.Search.FileFinder** is a great start for many custom artifacts - just copy/paste and pre-populate with the right defaults.



New Artifact Collection - Select Artifacts to collect

Step 1 out of 2

This artifact is useful in the following scenarios:

Selected Artifacts:

Windows.Search.FileFinder

July 2019

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	30	01	02	03	04	05	06
28	07	08	09	10	11	12	13
29	14	15	16	17	18	19	20
30	21	22	23	24	25	26	27
31	28	29	30	31	01	02	03
32	04	05	06	07	08	09	10

SearchFilesGlob	C
Keywords	
Use_Raw_NTFS	☑
Upload_File	□
Calculate_Hash	□
MoreRecentThan	2019-07-02
ModifiedBefore	

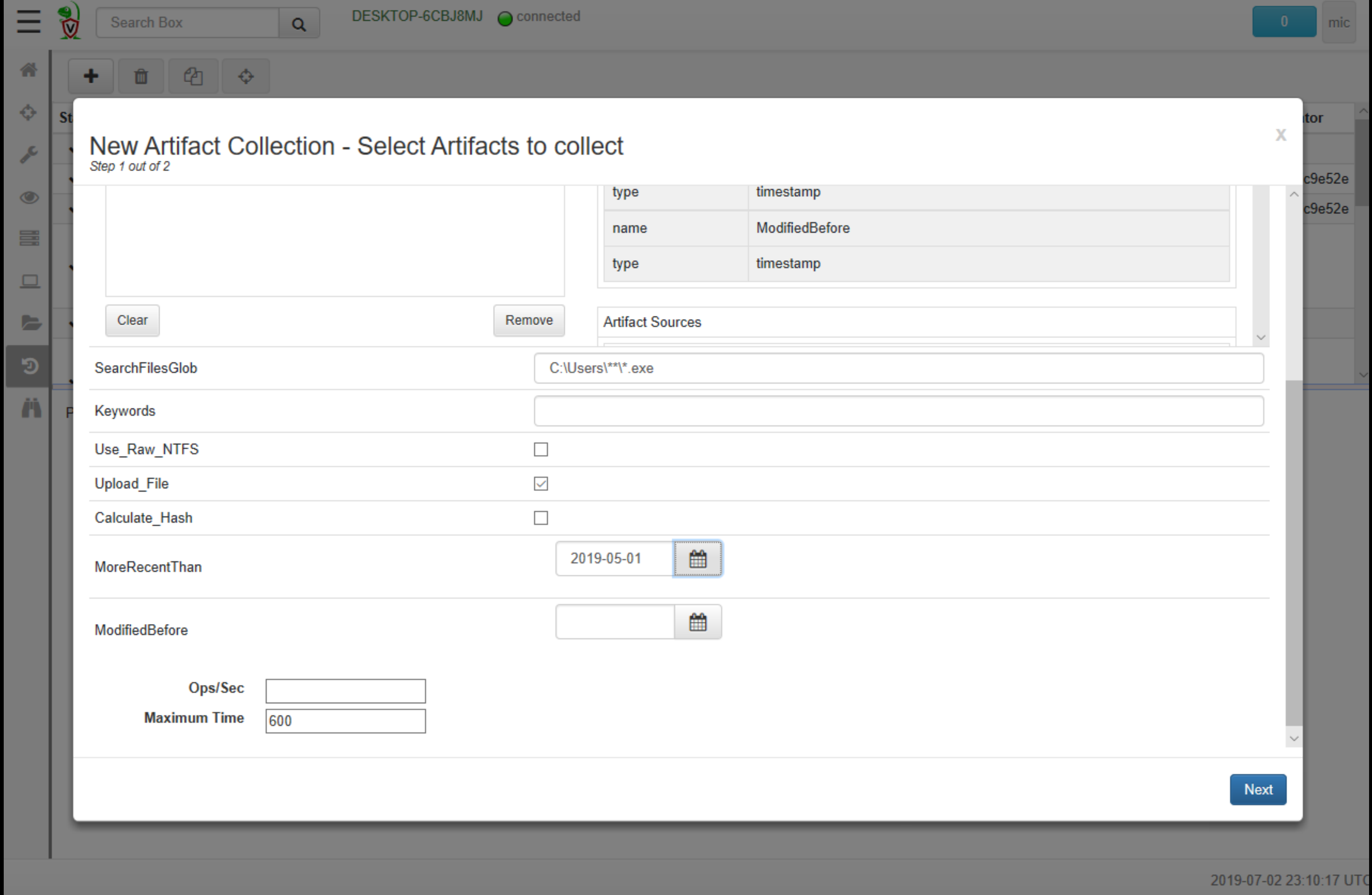
Exercise: File collections

Tasks:

- Collect all exe’s created in a home directory in the last day
- Collect all text files containing a keyword.

Hunting hints:

- Create a text file containing the keyword *“secret”*
- Search for it as before.



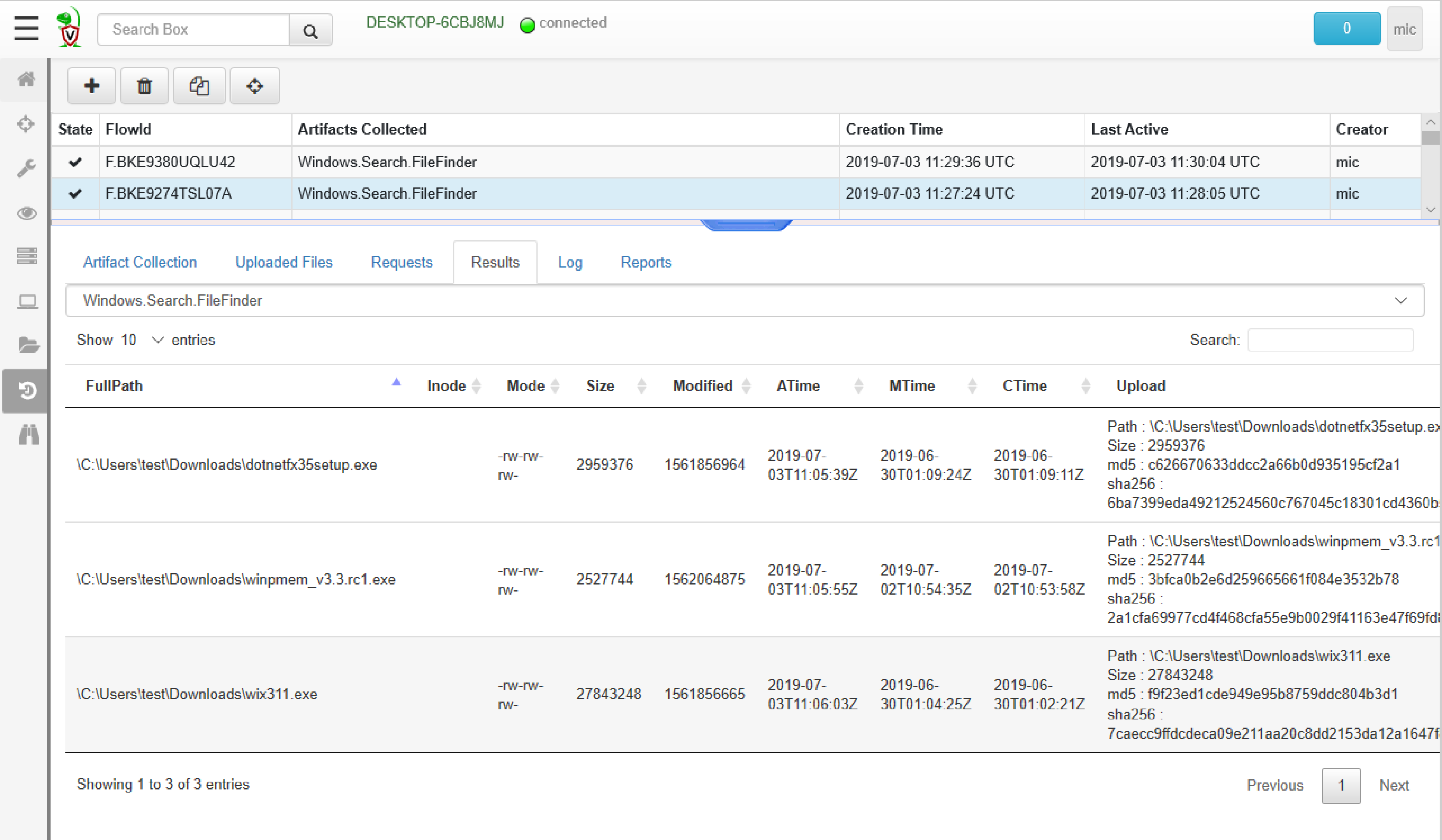
New Artifact Collection - Select Artifacts to collect

Step 1 out of 2

type	timestamp
name	ModifiedBefore
type	timestamp

SearchFilesGlob	C:\Users\***.exe
Keywords	
Use_Raw_NTFS	□
Upload_File	☑
Calculate_Hash	□
MoreRecentThan	2019-05-01
ModifiedBefore	

Ops/Sec	
Maximum Time	600



State	FlowId	Artifacts Collected	Creation Time	Last Active	Creator
✓	F.BKE9380UQLU42	Windows.Search.FileFinder	2019-07-03 11:29:36 UTC	2019-07-03 11:30:04 UTC	mic
✓	F.BKE9274TSL07A	Windows.Search.FileFinder	2019-07-03 11:27:24 UTC	2019-07-03 11:28:05 UTC	mic

FullPath	Inode	Mode	Size	Modified	ATime	MTime	CTime	Upload
\C:\Users\test\Downloads\dotnetfx35setup.exe		-rw-rw-rw-	2959376	1561856964	2019-07-03T11:05:39Z	2019-06-30T01:09:24Z	2019-06-30T01:09:11Z	Path : \C:\Users\test\Downloads\dotnetfx35setup.ex Size : 2959376 md5 : c626670633ddcc2a66b0d935195cf2a1 sha256 : 6ba7399eda49212524560c767045c18301cd4360b
\C:\Users\test\Downloads\winpmem_v3.3.rc1.exe		-rw-rw-rw-	2527744	1562064875	2019-07-03T11:05:55Z	2019-07-02T10:54:35Z	2019-07-02T10:53:58Z	Path : \C:\Users\test\Downloads\winpmem_v3.3.rc1 Size : 2527744 md5 : 3bfca0b2e6d259665661f084e3532b78 sha256 : 2a1cfa69977cd4f468cfa55e9b0029f41163e47f69fd
\C:\Users\test\Downloads\wix311.exe		-rw-rw-rw-	27843248	1561856665	2019-07-03T11:06:03Z	2019-06-30T01:04:25Z	2019-06-30T01:02:21Z	Path : \C:\Users\test\Downloads\wix311.exe Size : 27843248 md5 : f9f23ed1cde949e95b8759ddc804b3d1 sha256 : 7caecc9ffdcdeca09e211aa20c8dd2153da12a1647f

Showing 1 to 3 of 3 entries

Exercise: File collections - Microsoft Word docs

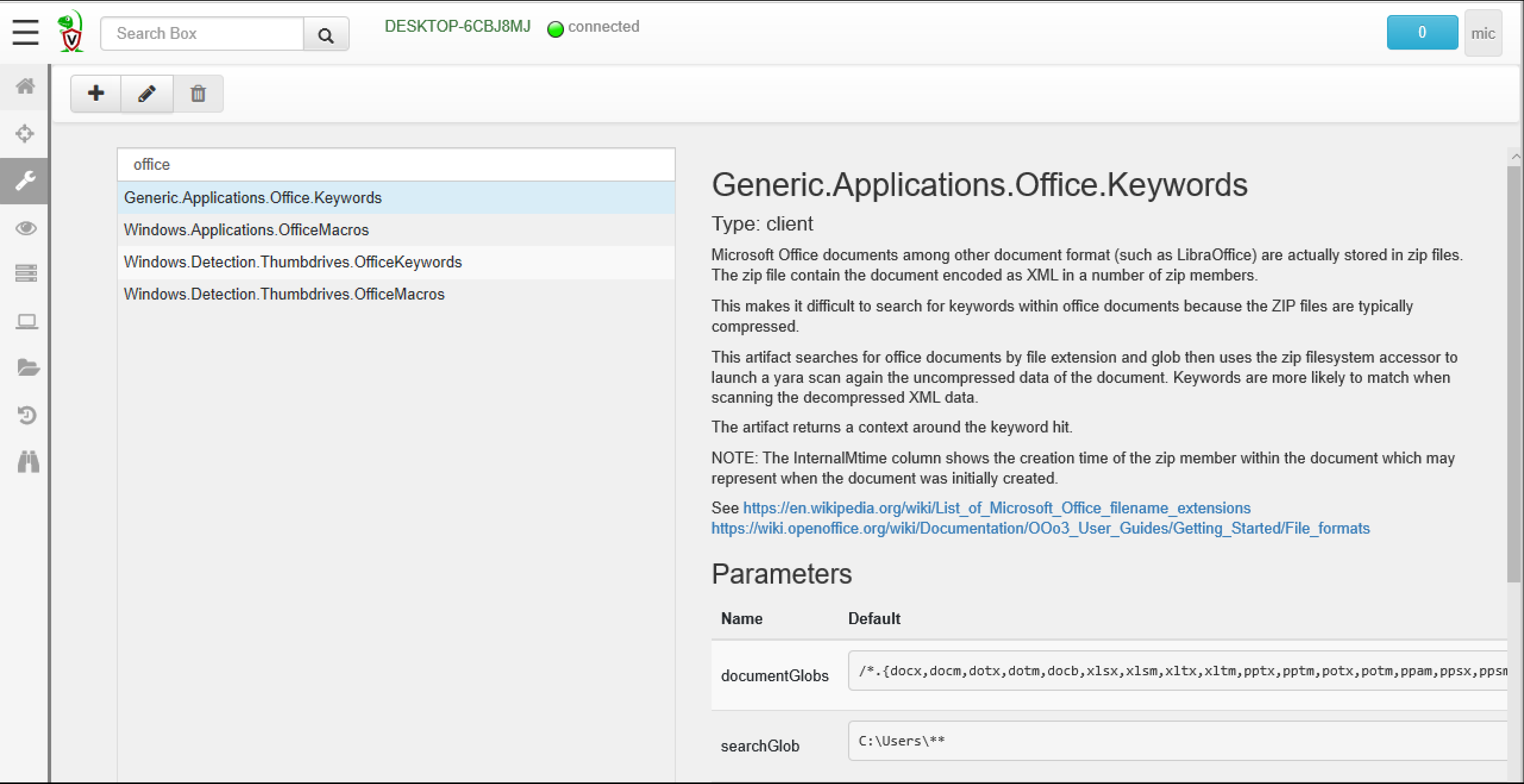
Task: Collect Microsoft Word documents containing a keyword.

Hunting hints:

- Create a Word document containing the word *"secret"*
- Search for it as before - does it work?
  - (it won't work because Word documents are compressed)

What can we do?

- *We have an artifact for that ...*



office

Generic.Applications.Office.Keywords
Windows.Applications.OfficeMacros
Windows.Detection.Thumbdrives.OfficeKeywords
Windows.Detection.Thumbdrives.OfficeMacros

Generic.Applications.Office.Keywords

Type: client

Microsoft Office documents among other document format (such as LibraOffice) are actually stored in zip files. The zip file contain the document encoded as XML in a number of zip members.

This makes it difficult to search for keywords within office documents because the ZIP files are typically compressed.

This artifact searches for office documents by file extension and glob then uses the zip filesystem accessor to launch a yara scan again the uncompressed data of the document. Keywords are more likely to match when scanning the decompressed XML data.

The artifact returns a context around the keyword hit.

NOTE: The InternalMtime column shows the creation time of the zip member within the document which may represent when the document was initially created.

See https://en.wikipedia.org/wiki/List_of_Microsoft_Office_filename_extensions https://wiki.openoffice.org/wiki/Documentation/OOo3_User_Guides/Getting_Started/File_formats

Parameters

Name	Default
documentGlobs	`/*.{docx,docm,dotx,dotm,docb,xlsx,xlsm,xltx,xltm,pptx,pptm,potx,potm,ppam,ppsx,ppsm`
searchGlob	`C:\Users\**`

Scenario: Chrome extensions

Chrome extensions can be very dangerous.

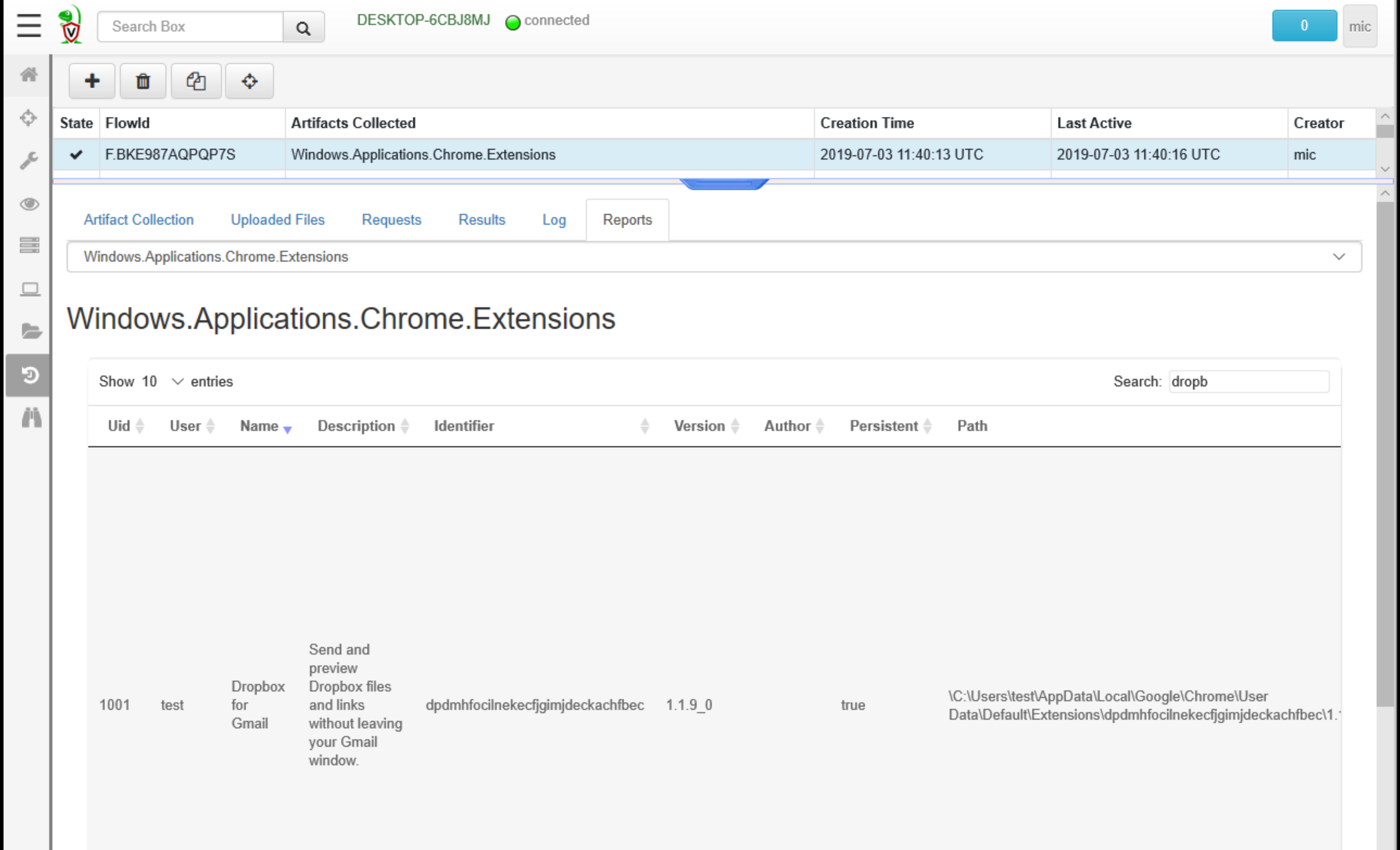
They could access all website data including cookies and logon creds.

They can create XSS opportunities for complete compromise.

Exfil is difficult to spot, since all communications occur over SSL.

Many Chrome extensions have been found to be malicious or vulnerable.

So what Chrome extensions do your users have installed?



State	FlowId	Artifacts Collected	Creation Time	Last Active	Creator
✓	F.BKE987AQPQP7S	Windows.Applications.Chrome.Extensions	2019-07-03 11:40:13 UTC	2019-07-03 11:40:16 UTC	mic

Windows.Applications.Chrome.Extensions

Show 10 entries

Search: dropb

Uid	User	Name	Description	Identifier	Version	Author	Persistent	Path
1001	test	Dropbox for Gmail	Send and preview Dropbox files and links without leaving your Gmail window.	dpdmhfocilnekecfjgimjdeckachfbec	1.1.9_0		true	\C:\Users\test\AppData\Local\Google\Chrome\User Data\Default\Extensions\dpdmhfocilnekecfjgimjdeckachfbec\1.

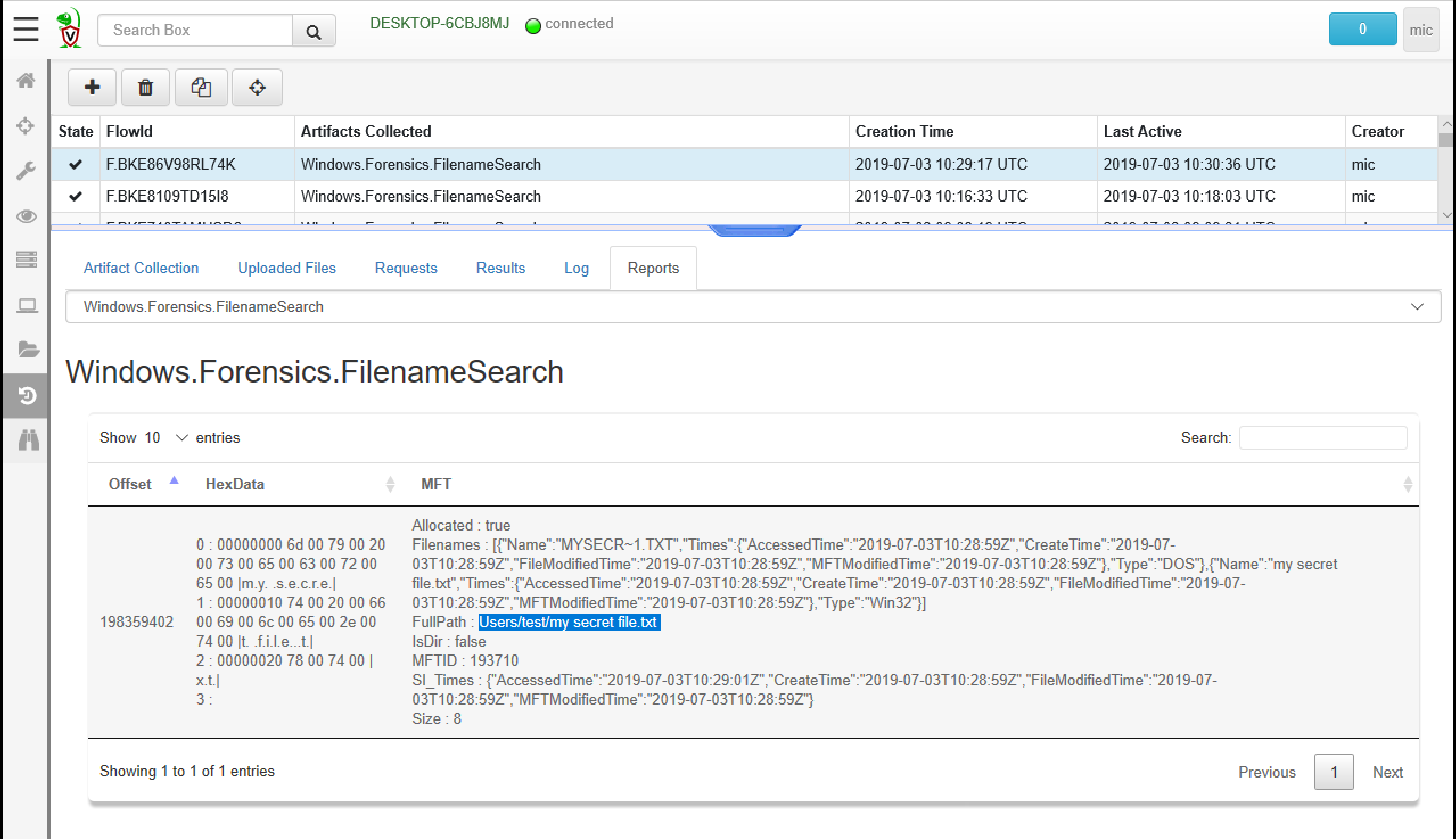
Exercise: IP theft

We’ve just been advised that our confidential data has been found on the dark web.

Task: We need to know which machines had this file in the past.

Hunting hints:

- Create a new file called **my secret file.txt** on your client
- Scan your MFT for the unique string
- This may work even if the file is deleted.



State	FlowId	Artifacts Collected	Creation Time	Last Active	Creator
✓	F.BKE86V98RL74K	Windows.Forensics.FilenameSearch	2019-07-03 10:29:17 UTC	2019-07-03 10:30:36 UTC	mic
✓	F.BKE8109TD15I8	Windows.Forensics.FilenameSearch	2019-07-03 10:16:33 UTC	2019-07-03 10:18:03 UTC	mic

Windows.Forensics.FilenameSearch

Windows.Forensics.FilenameSearch

Show 10 entries Search:

Offset	HexData	MFT
198359402	0 : 00000000 6d 00 79 00 20 00 73 00 65 00 63 00 72 00 65 00 |m.y. .s.e.c.r.e.| 1 : 00000010 74 00 20 00 66 00 69 00 6c 00 65 00 2e 00 74 00 |t. .f.i.l.e...t.| 2 : 00000020 78 00 74 00 | x.t.| 3 :	Allocated : true Filenames : [{"Name":"MYSECR~1.TXT","Times":{"AccessedTime":"2019-07-03T10:28:59Z","CreateTime":"2019-07-03T10:28:59Z","FileModifiedTime":"2019-07-03T10:28:59Z","MFTModifiedTime":"2019-07-03T10:28:59Z"},"Type":"DOS"},{"Name":"my secret file.txt","Times":{"AccessedTime":"2019-07-03T10:28:59Z","CreateTime":"2019-07-03T10:28:59Z","FileModifiedTime":"2019-07-03T10:28:59Z","MFTModifiedTime":"2019-07-03T10:28:59Z"},"Type":"Win32"}] FullPath : Users/test/my secret file.txt IsDir : false MFTID : 193710 SI_Times : {"AccessedTime":"2019-07-03T10:29:01Z","CreateTime":"2019-07-03T10:28:59Z","FileModifiedTime":"2019-07-03T10:28:59Z","MFTModifiedTime":"2019-07-03T10:28:59Z"} Size : 8

Showing 1 to 1 of 1 entries

Previous 1 Next

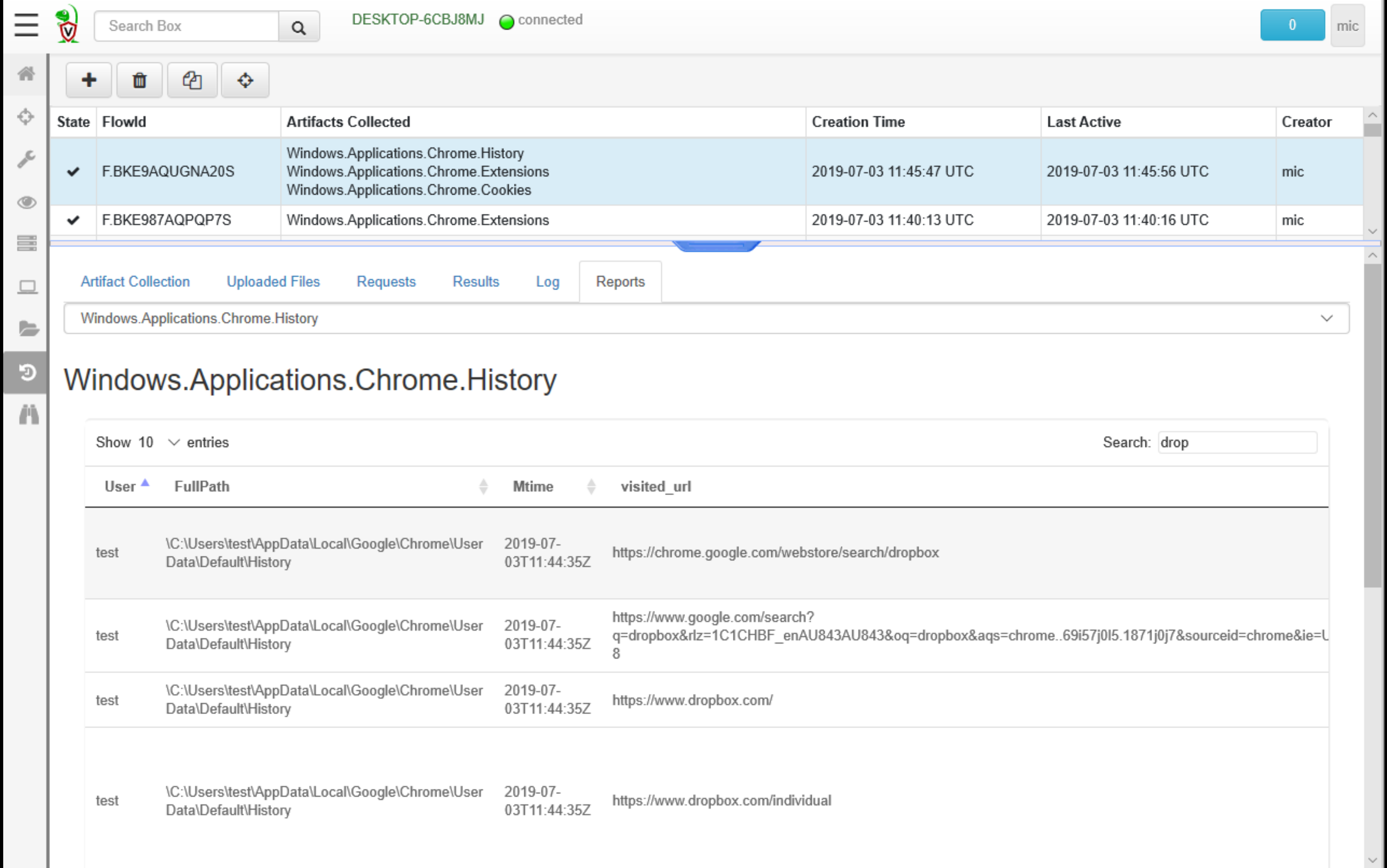
Scenario: Hunt down “shadow IT”

Dropbox is one common “shadow IT” threat.

It can be accessed through a web browser or an installed program.

Exercise:

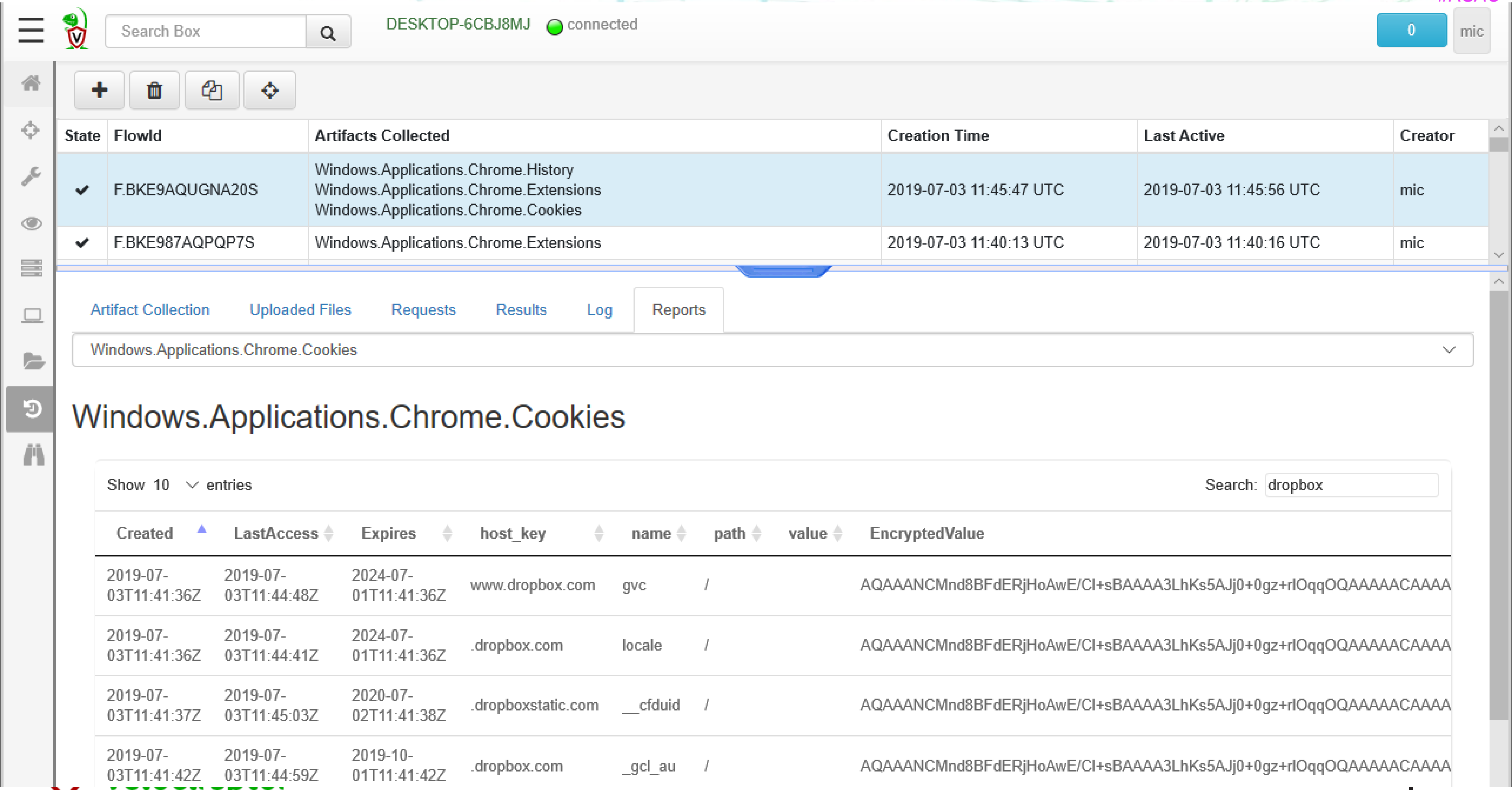
- Which of your users have Dropbox accounts?
- When did they access Dropbox through their web browsers?
- What confidential documents are shared through Dropbox?
- Let’s search web browsing history for accesses to Dropbox.



State	FlowId	Artifacts Collected	Creation Time	Last Active	Creator
✓	F.BKE9AQUGNA20S	Windows.Applications.Chrome.History Windows.Applications.Chrome.Extensions Windows.Applications.Chrome.Cookies	2019-07-03 11:45:47 UTC	2019-07-03 11:45:56 UTC	mic
✓	F.BKE987AQPQP7S	Windows.Applications.Chrome.Extensions	2019-07-03 11:40:13 UTC	2019-07-03 11:40:16 UTC	mic

Windows.Applications.Chrome.History

User	FullPath	Mtime	visited_url
test	\C:\Users\test\AppData\Local\Google\Chrome\User Data\Default\History	2019-07-03T11:44:35Z	https://chrome.google.com/webstore/search/dropbox
test	\C:\Users\test\AppData\Local\Google\Chrome\User Data\Default\History	2019-07-03T11:44:35Z	https://www.google.com/search?q=dropbox&rlz=1C1CHBF_enAU843AU843&oq=dropbox&aqs=chrome..69i57j0l5.1871j0j7&sourceid=chrome&ie=U 8
test	\C:\Users\test\AppData\Local\Google\Chrome\User Data\Default\History	2019-07-03T11:44:35Z	https://www.dropbox.com/
test	\C:\Users\test\AppData\Local\Google\Chrome\User Data\Default\History	2019-07-03T11:44:35Z	https://www.dropbox.com/individual



State	FlowId	Artifacts Collected	Creation Time	Last Active	Creator
✓	F.BKE9AQUGNA20S	Windows.Applications.Chrome.History Windows.Applications.Chrome.Extensions Windows.Applications.Chrome.Cookies	2019-07-03 11:45:47 UTC	2019-07-03 11:45:56 UTC	mic
✓	F.BKE987AQPQP7S	Windows.Applications.Chrome.Extensions	2019-07-03 11:40:13 UTC	2019-07-03 11:40:16 UTC	mic

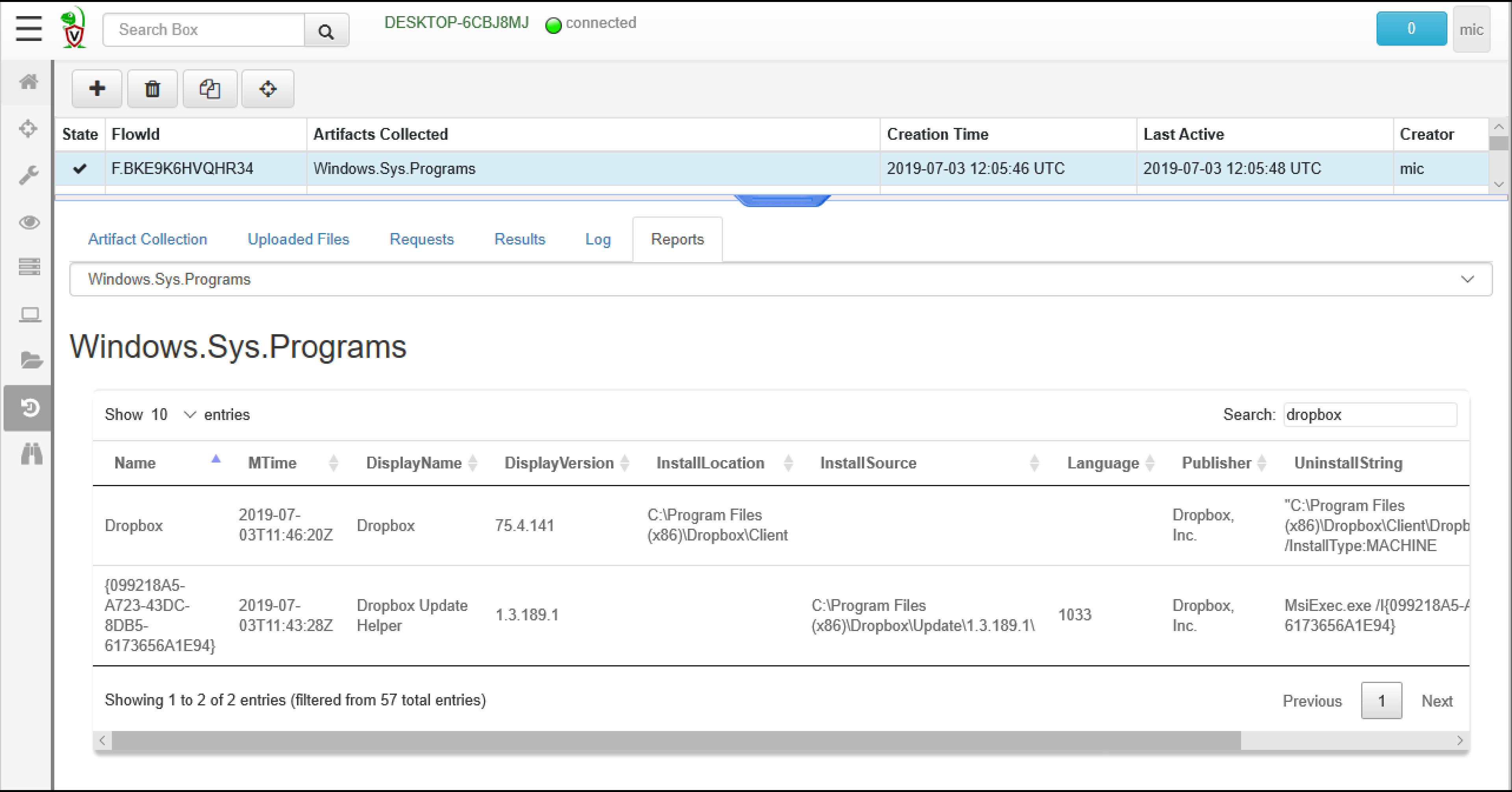
Windows.Applications.Chrome.Cookies

Windows.Applications.Chrome.Cookies

Show 10 entries

Search: dropbox

Created	LastAccess	Expires	host_key	name	path	value	EncryptedValue
2019-07-03T11:41:36Z	2019-07-03T11:44:48Z	2024-07-01T11:41:36Z	www.dropbox.com	gvc	/		AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAA3LhKs5AJj0+0gz+rlOqqOQAAAAACAAAA
2019-07-03T11:41:36Z	2019-07-03T11:44:41Z	2024-07-01T11:41:36Z	.dropbox.com	locale	/		AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAA3LhKs5AJj0+0gz+rlOqqOQAAAAACAAAA
2019-07-03T11:41:37Z	2019-07-03T11:45:03Z	2020-07-02T11:41:38Z	.dropboxstatic.com	__cfduid	/		AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAA3LhKs5AJj0+0gz+rlOqqOQAAAAACAAAA
2019-07-03T11:41:42Z	2019-07-03T11:44:59Z	2019-10-01T11:41:42Z	.dropbox.com	_gcl_au	/		AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAA3LhKs5AJj0+0gz+rlOqqOQAAAAACAAAA



State	FlowId	Artifacts Collected	Creation Time	Last Active	Creator
✓	F.BKE9K6HVQHR34	Windows.Sys.Programs	2019-07-03 12:05:46 UTC	2019-07-03 12:05:48 UTC	mic

Windows.Sys.Programs

Show 10 entries

Search: dropbox

Name	MTime	DisplayName	DisplayVersion	InstallLocation	InstallSource	Language	Publisher	UninstallString
Dropbox	2019-07-03T11:46:20Z	Dropbox	75.4.141	C:\Program Files (x86)\Dropbox\Client			Dropbox, Inc.	"C:\Program Files (x86)\Dropbox\Client\Dropb /InstallType:MACHINE
{099218A5-A723-43DC-8DB5-6173656A1E94}	2019-07-03T11:43:28Z	Dropbox Update Helper	1.3.189.1		C:\Program Files (x86)\Dropbox\Update\1.3.189.1\	1033	Dropbox, Inc.	MsiExec.exe /I{099218A5-A 6173656A1E94}

Showing 1 to 2 of 2 entries (filtered from 57 total entries)

Scenario: Use of Microsoft SysInternal tools

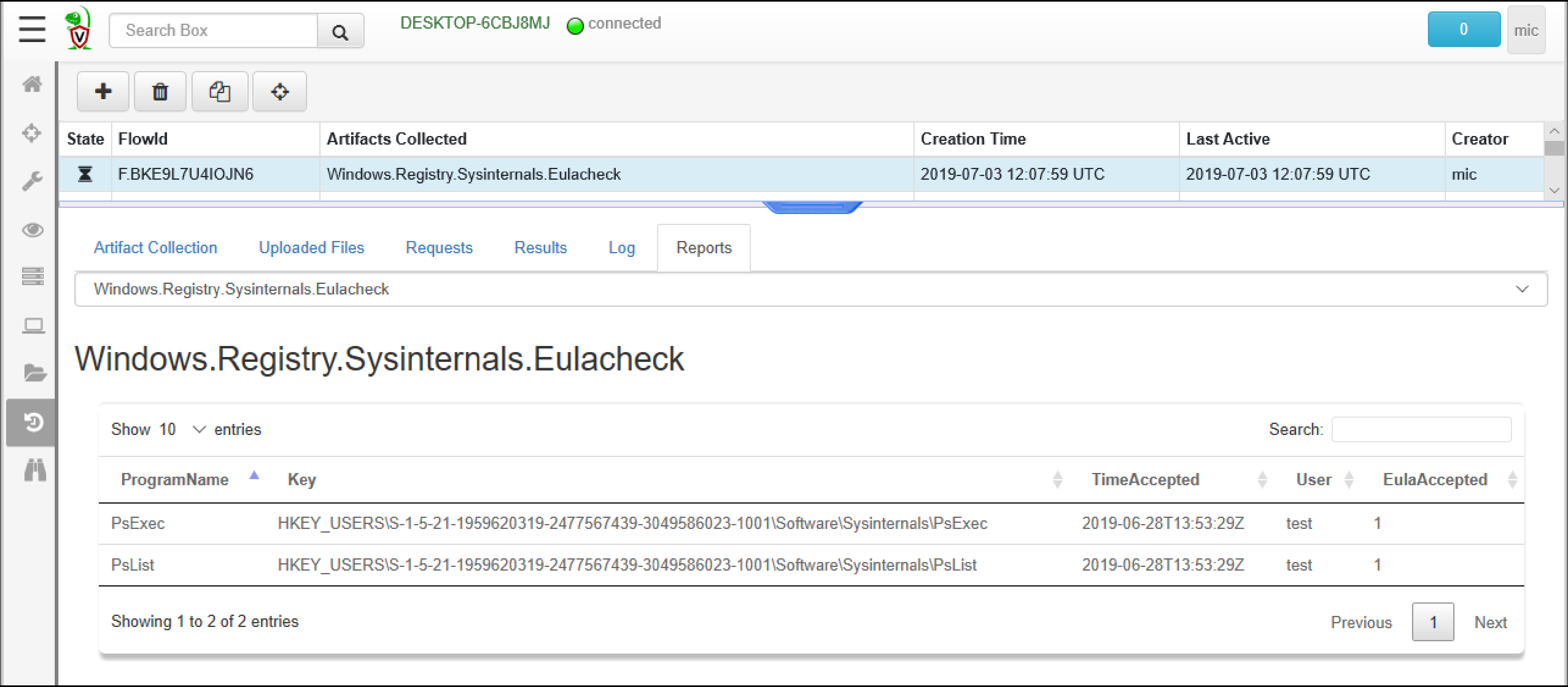
SysInternal tools are powerful system administration tools which are also used by attackers “living off the land”.

Did any SysInternal tools ever run on your endpoint?

For non-administrator accounts, this is very suspicious.

- Hint: Sysinternals tools require the user accepting a EULA, which leaves an interesting forensic artifact - a Registry key showing the user accepted the EULA.

We have an artifact for that too!



State	FlowId	Artifacts Collected	Creation Time	Last Active	Creator
⧗	F.BKE9L7U4IOJN6	Windows.Registry.Sysinternals.Eulacheck	2019-07-03 12:07:59 UTC	2019-07-03 12:07:59 UTC	mic

Windows.Registry.Sysinternals.Eulacheck

Windows.Registry.Sysinternals.Eulacheck

Show 10 entries

Search:

ProgramName	Key	TimeAccepted	User	EulaAccepted
PsExec	HKEY_USERS\S-1-5-21-1959620319-2477567439-3049586023-1001\Software\Sysinternals\PsExec	2019-06-28T13:53:29Z	test	1
PsList	HKEY_USERS\S-1-5-21-1959620319-2477567439-3049586023-1001\Software\Sysinternals\PsList	2019-06-28T13:53:29Z	test	1

Showing 1 to 2 of 2 entries

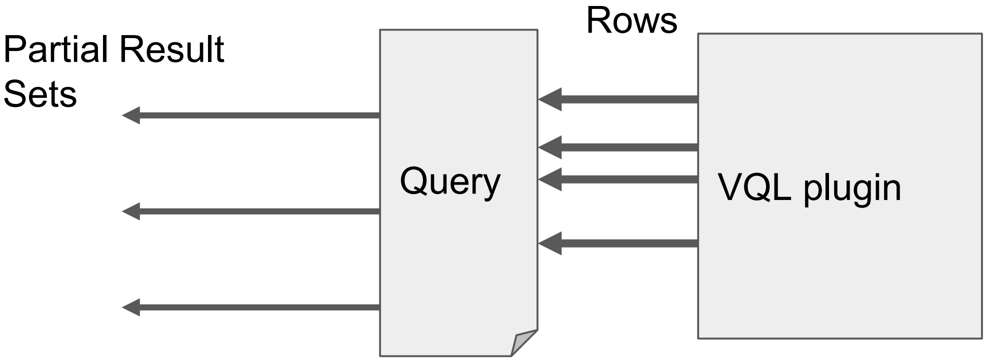
Event artifacts and endpoint monitoring

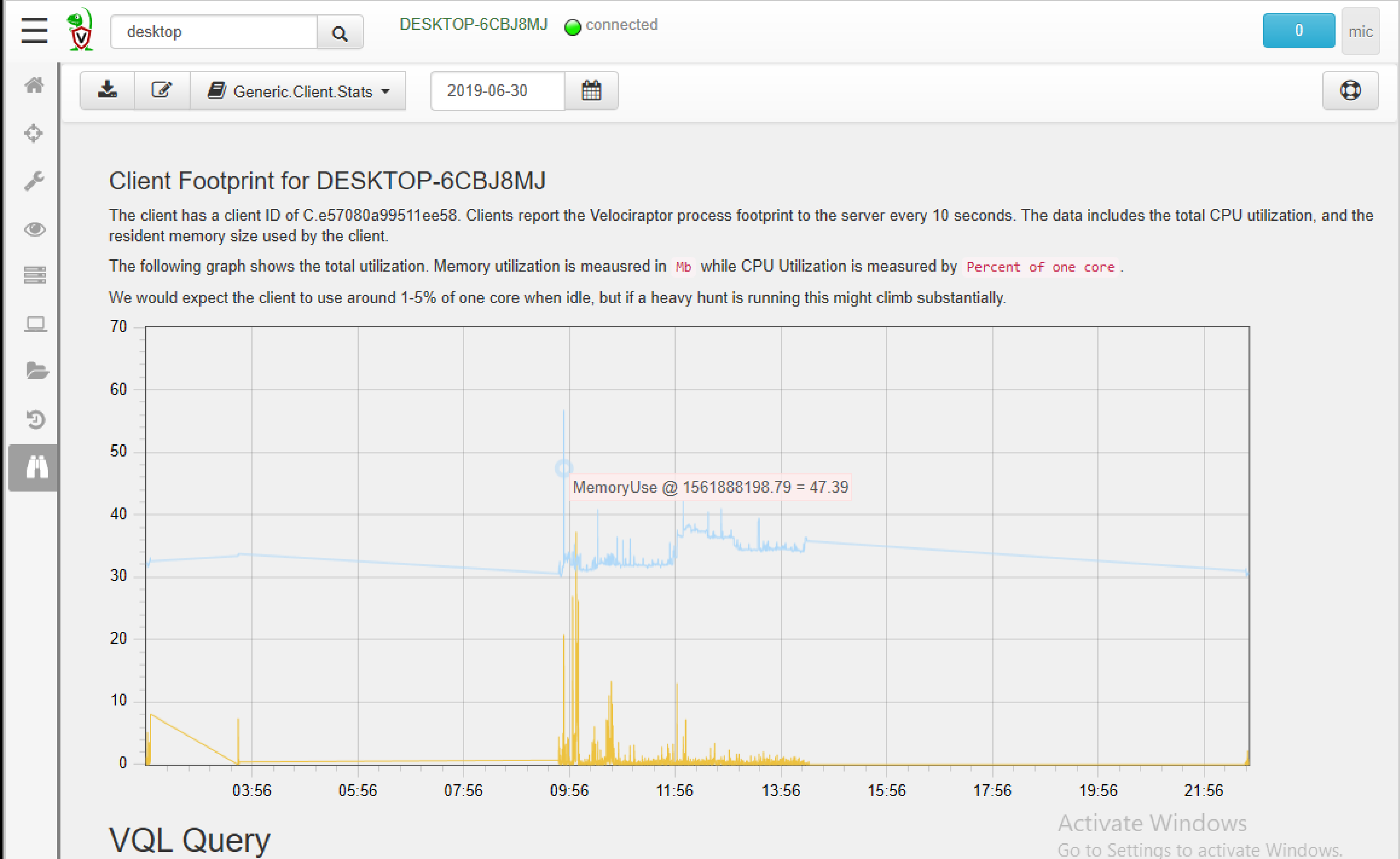
What are event artifacts?

Event artifacts are never-ending VQL queries that watch for events on clients and stream those events to the server.

Example:

Generic.Client.Stats





Client Footprint for DESKTOP-6CBJ8MJ

The client has a client ID of C.e57080a99511ee58. Clients report the Velociraptor process footprint to the server every 10 seconds. The data includes the total CPU utilization, and the resident memory size used by the client.

The following graph shows the total utilization. Memory utilization is meausred in `Mb` while CPU Utilization is measured by `Percent of one core`.

We would expect the client to use around 1-5% of one core when idle, but if a heavy hunt is running this might climb substantially.

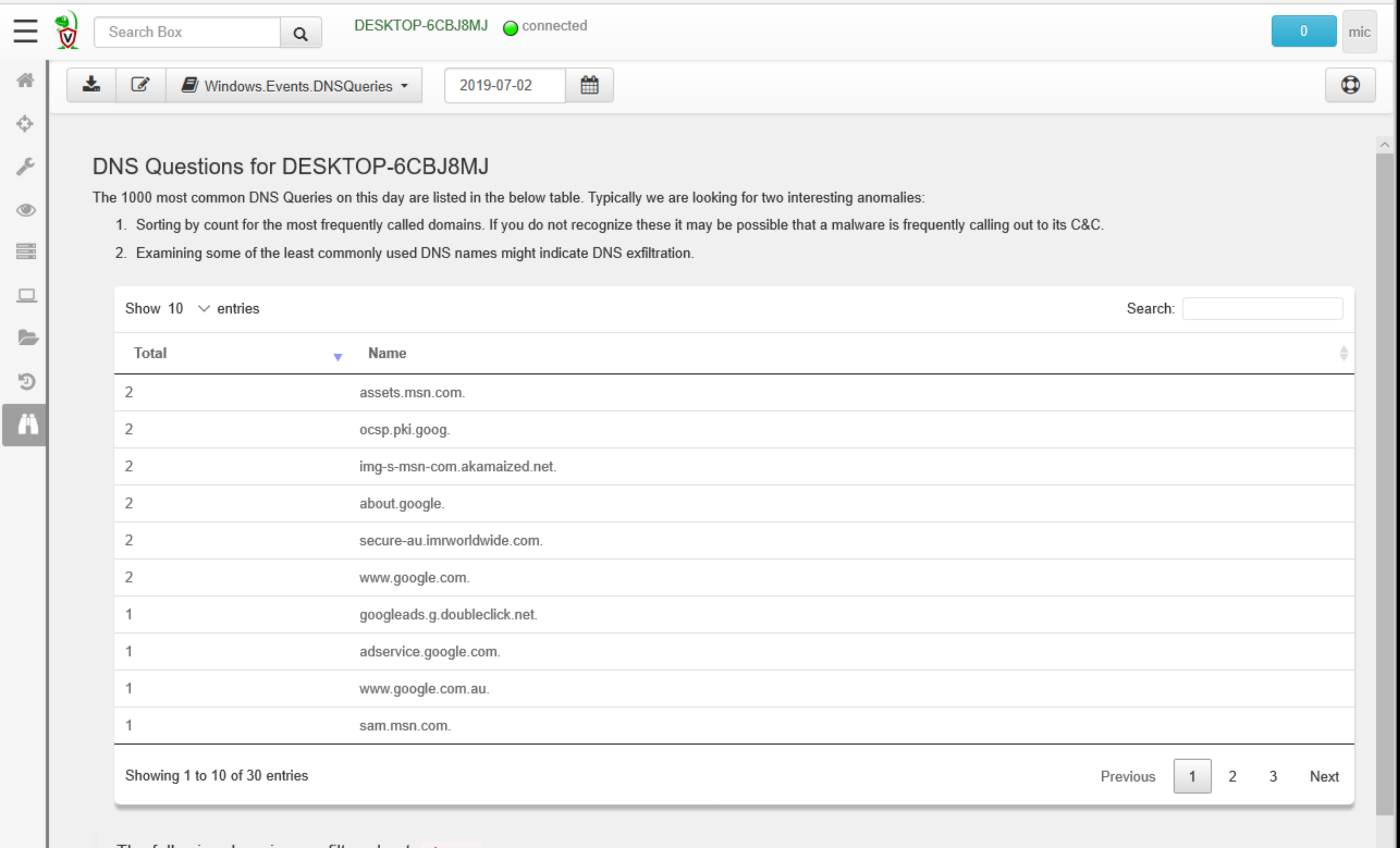
VQL Query

Scenario: Monitor all DNS lookups

DNS lookups are an *excellent* network signal.

They can reveal C2 activity and help scope the extent of compromise across a network by showing all clients attempting to connect to known-bad domains.

We can store all DNS lookups from clients, then search this data when threat intel reveals C2 and other suspicious DNS names.



DNS Questions for DESKTOP-6CBJ8MJ

The 1000 most common DNS Queries on this day are listed in the below table. Typically we are looking for two interesting anomalies:

1. Sorting by count for the most frequently called domains. If you do not recognize these it may be possible that a malware is frequently calling out to its C&C.
2. Examining some of the least commonly used DNS names might indicate DNS exfiltration.

Show 10 entries Search:

Total	Name
2	assets.msn.com.
2	ocsp.pki.goog.
2	img-s-msn-com.akamaized.net.
2	about.google.
2	secure-au.imrworldwide.com.
2	www.google.com.
1	googleads.g.doubleclick.net.
1	adservice.google.com.
1	www.google.com.au.
1	sam.msn.com.

Showing 1 to 10 of 30 entries

Scenario: Monitor endpoint for USB drive insertion

USB drives are a constant threat:

- They can introduce malware
- They’re commonly used to exfiltrate confidential documents.

We want an artifact that watches every client for USB drives being inserted, then sends us a listing of all files copied to them.

This has long been a limitation of Windows forensic artifacts!

Automating response with server event artifacts

Post-process client events

Server event artifacts are similar to the client event artifacts, except they run on the server.

The server listens for events and responds to them.

The events may originate with the clients **or** post-process any other activity on the server.

Exercise: Decode encoded PowerShell commands

PowerShell can accept a base64 encoded command line, often used by attackers to pass commands and script blocks.

These are easy to decode individually, but harder at scale.

Velociraptor can decode these automatically.

Test this by running the following encoded PowerShell:

```
powershell -encodedCommand
ZABpAHIAIAAiAGMAOgBcAHAAcgBvAGcAcgBhAG0AIABmAGkAbABlAHMAI
gAgAA==
```

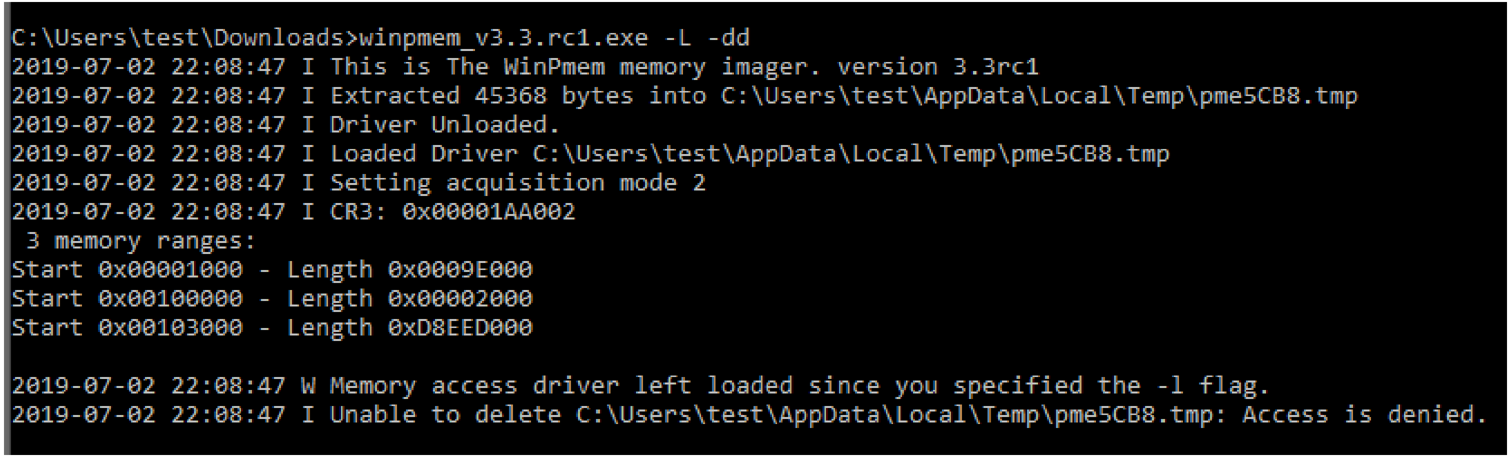
Exercise: Alert if a new service is installed

Installation of new services could indicate attacker activities.

Example: **winpmem** is a tool used to obtain memory images.

It installs a kernel driver and a service called **pmem**.

Velociraptor can easily send an email alert if this is detected.



```
C:\Users\test\Downloads>winpmem_v3.3.rc1.exe -L -dd
2019-07-02 22:08:47 I This is The WinPmem memory imager. version 3.3rc1
2019-07-02 22:08:47 I Extracted 45368 bytes into C:\Users\test\AppData\Local\Temp\pme5CB8.tmp
2019-07-02 22:08:47 I Driver Unloaded.
2019-07-02 22:08:47 I Loaded Driver C:\Users\test\AppData\Local\Temp\pme5CB8.tmp
2019-07-02 22:08:47 I Setting acquisition mode 2
2019-07-02 22:08:47 I CR3: 0x00001AA002
 3 memory ranges:
Start 0x00001000 - Length 0x0009E000
Start 0x00100000 - Length 0x00002000
Start 0x00103000 - Length 0xD8EED000

2019-07-02 22:08:47 W Memory access driver left loaded since you specified the -l flag.
2019-07-02 22:08:47 I Unable to delete C:\Users\test\AppData\Local\Temp\pme5CB8.tmp: Access is denied.
```

Velociraptor

Customizing artifacts

Customizing artifacts

Artifacts simply contain VQL statements.

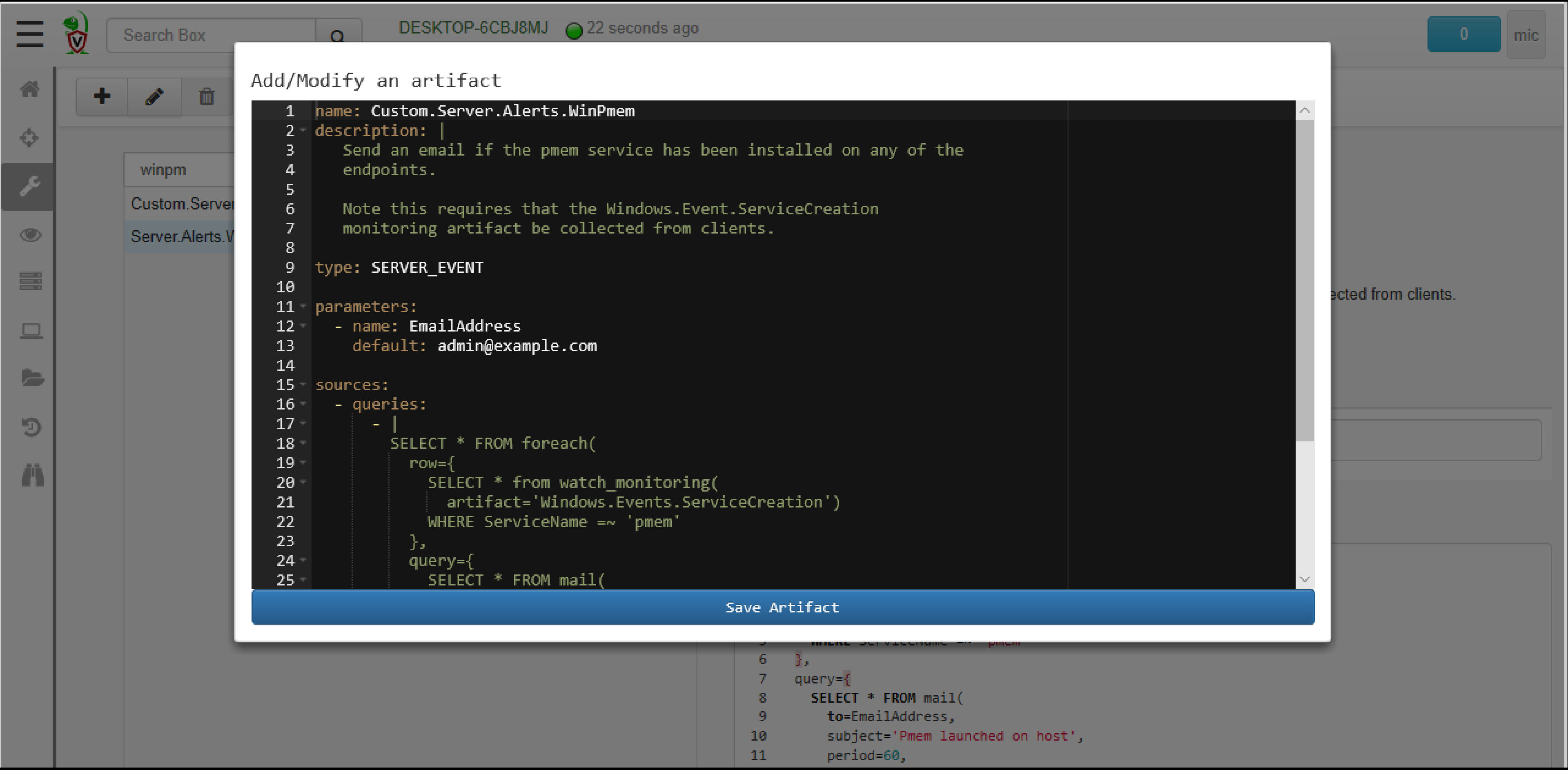
It’s easy to modify existing artifacts to your needs.

As you learn VQL, you can easily write your own.

Custom artifacts start with the **Custom.** Prefix.

You can use official or custom artifacts interchangeably.

You can also contribute your artifacts to the Velociraptor project.



```
name: Custom.Server.Alerts.WinPmem
description: |
   Send an email if the pmem service has been installed on any of the
   endpoints.

   Note this requires that the Windows.Event.ServiceCreation
   monitoring artifact be collected from clients.

type: SERVER_EVENT

parameters:
  - name: EmailAddress
    default: admin@example.com

sources:
  - queries:
      - |
        SELECT * FROM foreach(
          row={
            SELECT * from watch_monitoring(
              artifact='Windows.Events.ServiceCreation')
            WHERE ServiceName =~ 'pmem'
          },
          query={
            SELECT * FROM mail(
```

```
},
query={
  SELECT * FROM mail(
    to=EmailAddress,
    subject='Pmem launched on host',
    period=60,
```

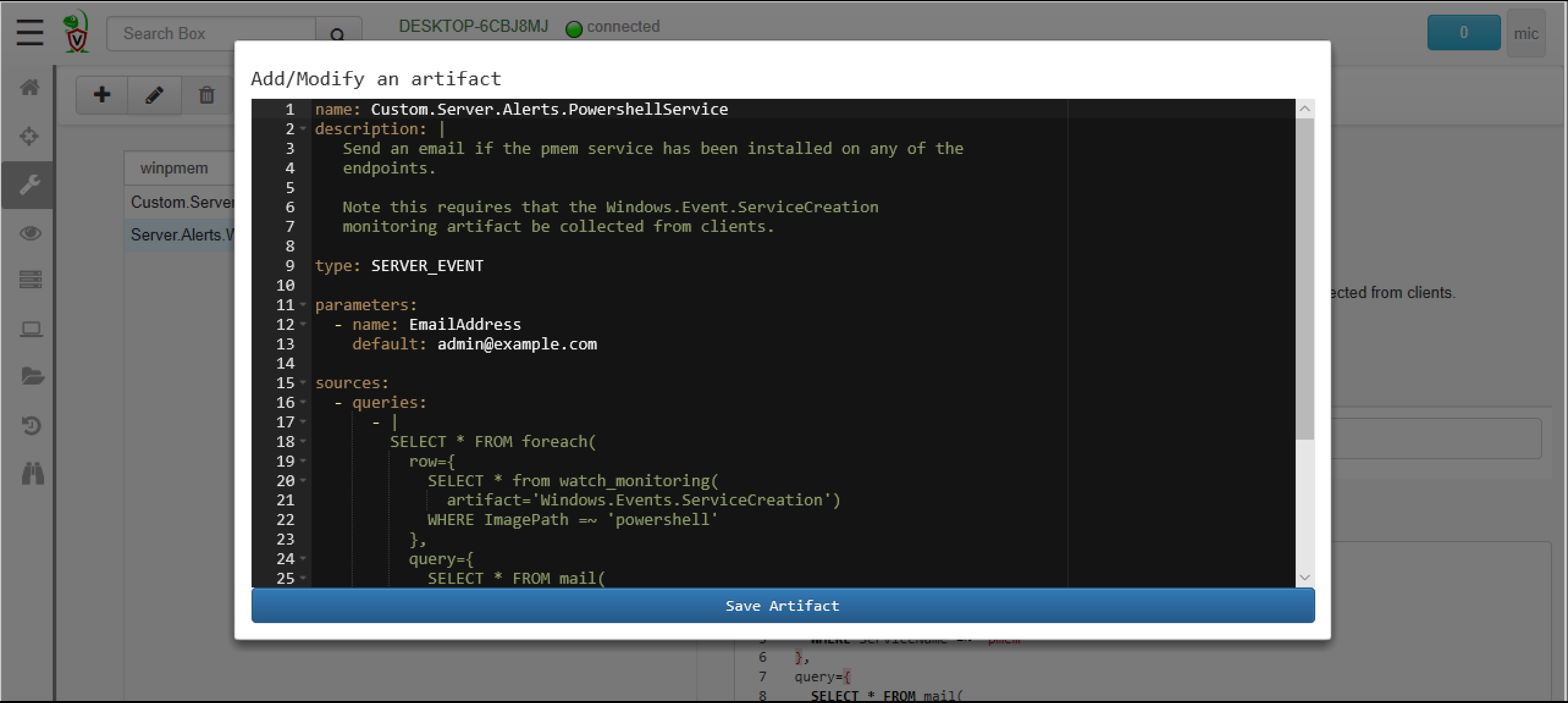
Scenario: Detecting lateral movement

Imagine a new service spawning PowerShell:

```
C:> sc create FakeDriver binpath="cmd.exe /Q /c
powershell.exe -nop -c dir"

C:>sc start FakeDriver
```

We can monitor clients for service creation events and alert when a service is installed using PowerShell.



Add/Modify an artifact

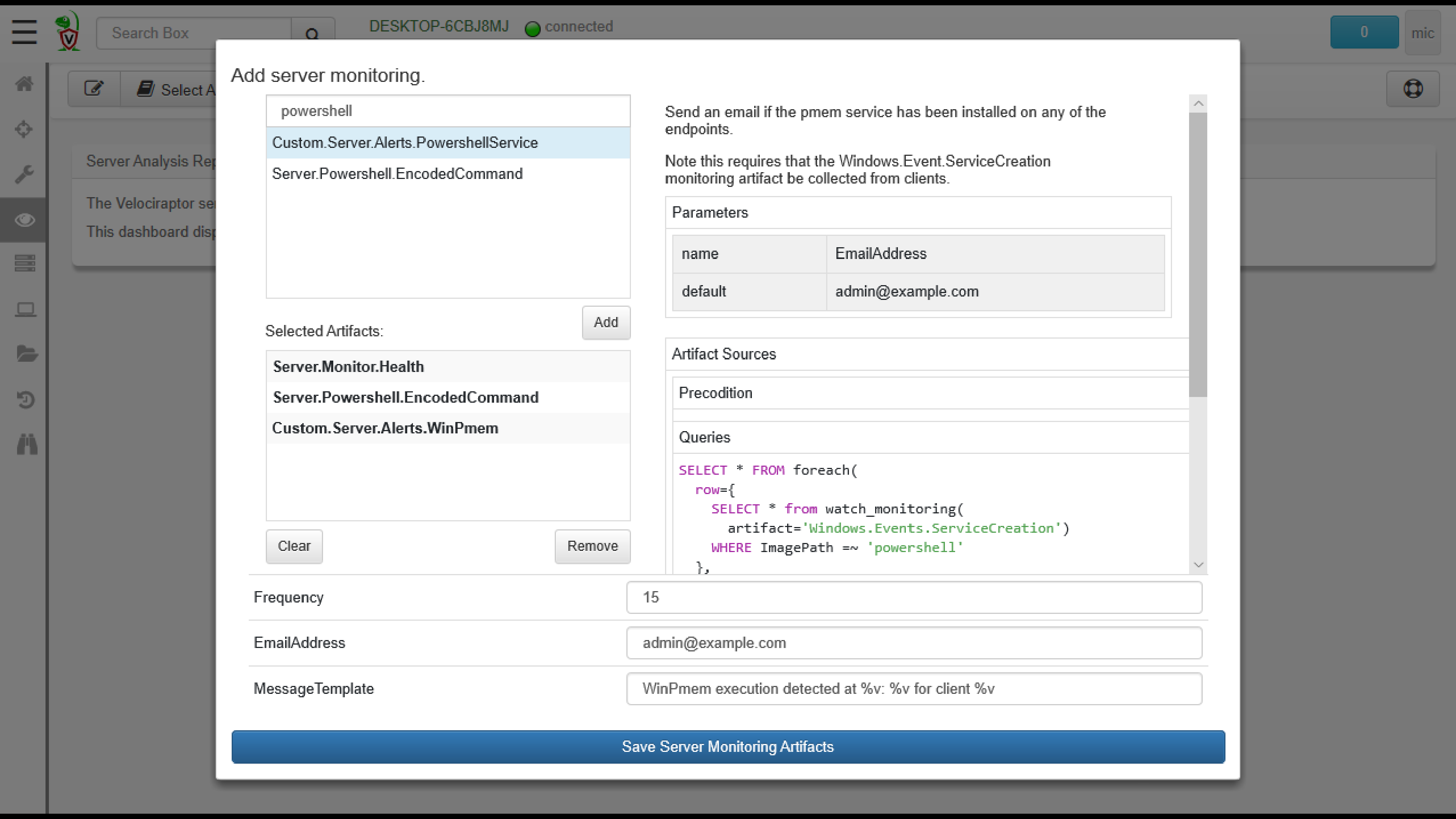
```
name: Custom.Server.Alerts.PowershellService
description: |
   Send an email if the pmem service has been installed on any of the
   endpoints.

   Note this requires that the Windows.Event.ServiceCreation
   monitoring artifact be collected from clients.

type: SERVER_EVENT

parameters:
  - name: EmailAddress
    default: admin@example.com

sources:
  - queries:
      - |
        SELECT * FROM foreach(
          row={
            SELECT * from watch_monitoring(
              artifact='Windows.Events.ServiceCreation')
            WHERE ImagePath =~ 'powershell'
          },
          query={
            SELECT * FROM mail(
```



The Velociraptor se

This dashboard disp

Add server monitoring.

powershell

- Custom.Server.Alerts.PowershellService
- Server.Powershell.EncodedCommand

Selected Artifacts:

- **Server.Monitor.Health**
- **Server.Powershell.EncodedCommand**
- **Custom.Server.Alerts.WinPmem**

Send an email if the pmem service has been installed on any of the endpoints.

Note this requires that the Windows.Event.ServiceCreation monitoring artifact be collected from clients.

Parameters

name	EmailAddress
default	admin@example.com

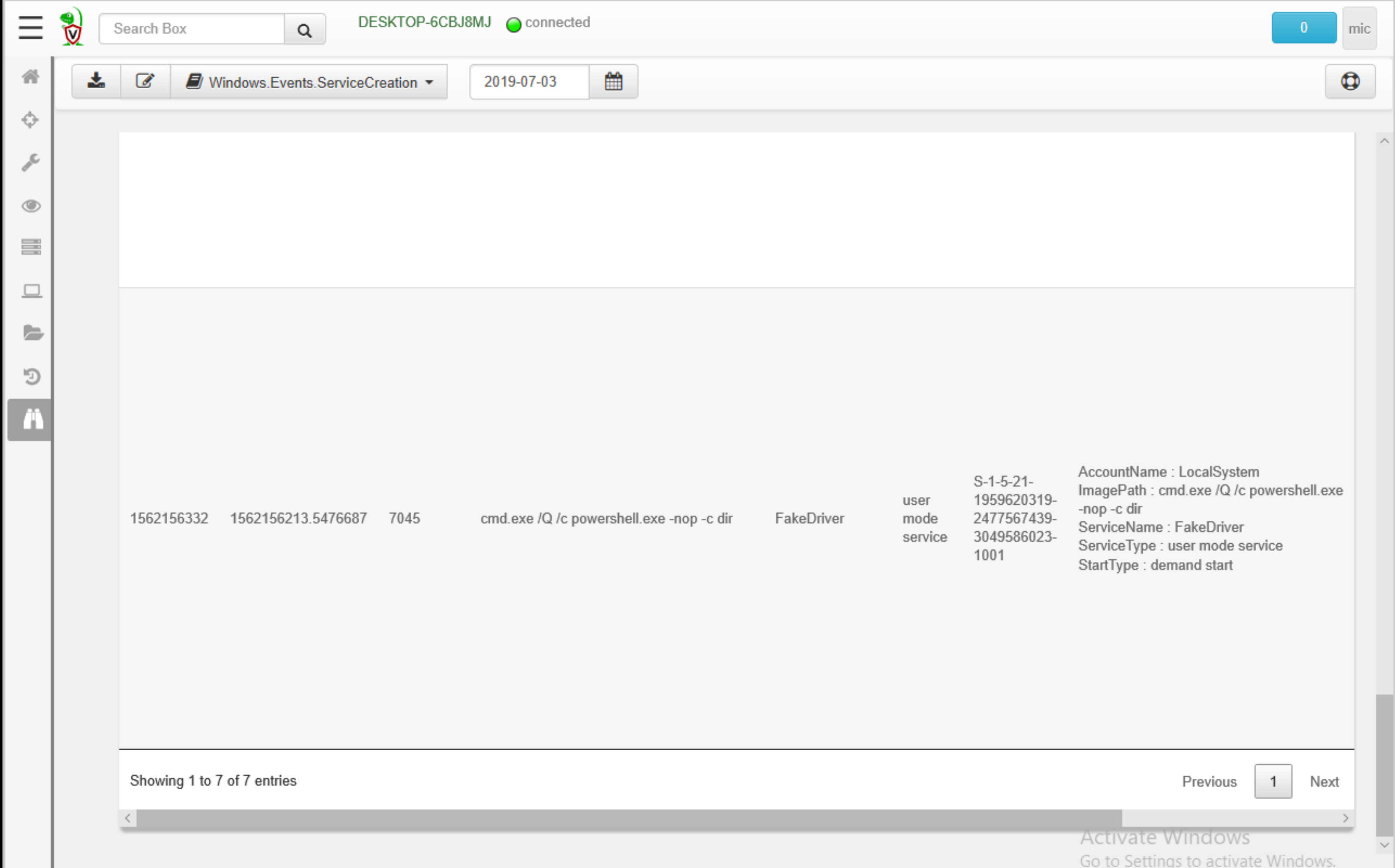
Artifact Sources

Precodition

Queries

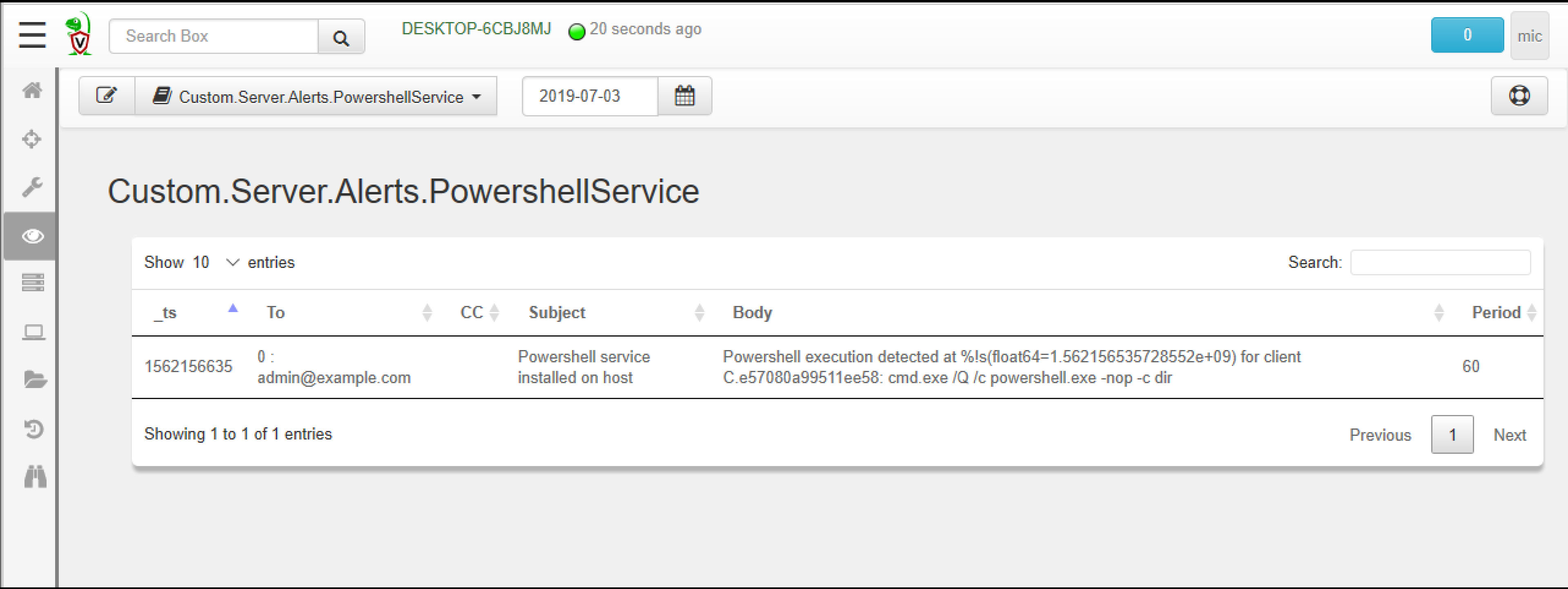
```
SELECT * FROM foreach(
  row={
    SELECT * from watch_monitoring(
      artifact='Windows.Events.ServiceCreation')
    WHERE ImagePath =~ 'powershell'
  },
```

Frequency	15
EmailAddress	admin@example.com
MessageTemplate	WinPmem execution detected at %v: %v for client %v



1562156332	1562156213.5476687	7045	cmd.exe /Q /c powershell.exe -nop -c dir	FakeDriver	user mode service	S-1-5-21-1959620319-2477567439-3049586023-1001	AccountName : LocalSystem ImagePath : cmd.exe /Q /c powershell.exe -nop -c dir ServiceName : FakeDriver ServiceType : user mode service StartType : demand start

Showing 1 to 7 of 7 entries



Custom.Server.Alerts.PowershellService

Show 10 entries

Search:

_ts	To	CC	Subject	Body	Period
1562156635	0 : admin@example.com		Powershell service installed on host	Powershell execution detected at %!s(float64=1.562156535728552e+09) for client C.e57080a99511ee58: cmd.exe /Q /c powershell.exe -nop -c dir	60

Showing 1 to 1 of 1 entries

Apply what you learned

Ultimately we are trying to answer **Questions** about our endpoints.

Now think of how to answer **Questions** using your endpoint monitoring tool of choice - *think outside the box.*

Share your method with others so they can easily apply your work.

Start hunting today

- Download Velociraptor from www.velocidex.com or GitHub
- Review the **Quick Start** documentation.
- Setup a Velociraptor server and deploy some test clients.
- Start by hunting for some pre-built artefacts.
- Then customise some hunts to your own requirements.
- Contribute back with your feedback and ideas.

RSAConference2019
Asia Pacific & Japan

Thank you.

www.velocidex.com