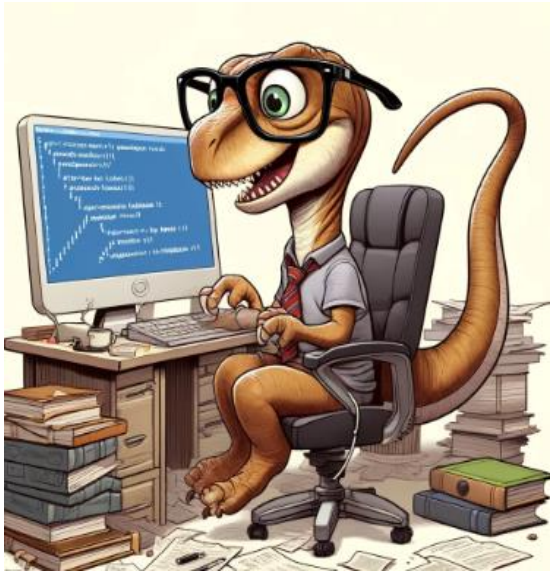
(WIP)Velociraptor(WIP)



First the below link is the official documentation to velociraptor

https://docs.velociraptor.app/

The next link is the velociraptor github repository

https://github.com/Velocidex/velociraptor

This link is where you actually download the velociraptor binary

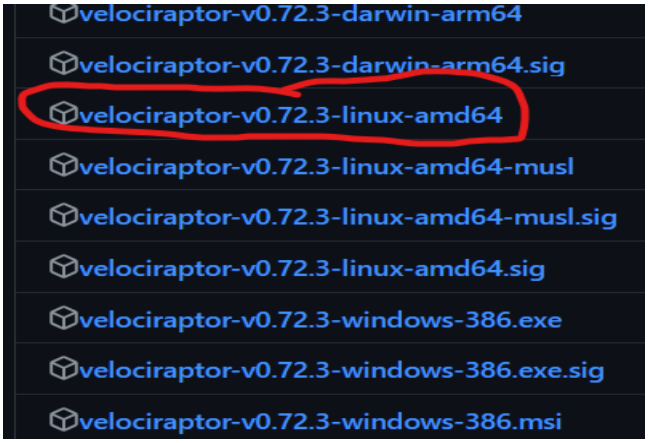
https://github.com/Velocidex/velociraptor/releases/tag/v0.72

What is Velociraptor - Velociraptor is a unique, advanced open-source endpoint monitoring, digital forensic and cyber response platform.

How to get it- it's free and open source

Download the corresponding version of velociraptor here (depending on your OS version, We will try Linux in this scenario as the server) . At the time of making this, 72.3 is the most recent version..

https://github.com/Velocidex/velociraptor/releases/tag/v0.72



At a high level, your Velociraptor deployment will consist of 3 tasks: setting up a server, deploying clients, and granting user access to the console.

Velociraptor is distributed as a **Single Binary**, which can act as a server, client or a number of utility programs depending on command line flags. Velociraptor does not use an external datastore - all data is stored within the server's filesystem in regular files and directories

3 types of deployment

1. Self-Signed SSL - recommended for on-premises environments
2. Cloud Deployment - recommended for easy deployments
3. Instant Velociraptor - recommended if you want to install Velociraptor as a self-contained client and server on your local machine for testing purposes

Lets try to configure the server. This simulates a sort of "jump box" on mission. Where we need to grab forensic evidence on a remote host (client)

- Download the binary

https://github.com/Velocidex/velociraptor/releases/tag/v0.72

- For Linux, make sure the file can execute: execute the below command

```
chmod +x velociraptor-v0.72.3-linux-amd64
```

- Lets make a directory for future velociraptor output

```
mkdir ~/Desktop/velociraptor
```

Once downloaded, navigate to the location of the download and run the executable with the

following command:

```
./velociraptor-v0.72.3-linux-amd64 config generate -i
```

- This will prompt you with questions about the server's configuration. Answer them accordingly Choose the OS you'll be using (Linux)
- Choose where to store data. Lets place it in our made directory. Sometime permission issues may arise when outputting in the /opt directory

```
Path to the datastore directory. (/opt/velociraptor) ~/Desktop/velociraptor
```

- Pick Self Signed SSL

```
> Self Signed SSL
  Automatically provision certificates with Lets Encrypt
  Authenticate users with SSO
```

- Next it will ask the DNS name. We won't have one, we need to use the ip of our host server. This is the IP that clients will reach out to.

```
What is the public DNS name of the Master Frontend (e.g. www.example.com): [? for help] (localhost) 192.168.1.x
```

- Choose the port the server will listen on, default is 8000. We'll leave it as is.

```
? Enter the frontend port to listen on. (8000)
```

- Next it asked what port the GUI will listen on. We'll leave it as is, 8889

```
Enter the port for the GUI to listen on. (8889)
```

- Answer "N" for the next two questions, then none

```
Websocket is a bidirectional low latency communication protocol supported by
most modern proxies and load balancers. This method is more efficient and
portable than plain HTTP. Be sure to test this in your environment.
 No
? Would you like to use the registry to store the writeback files? (Experimental) No
? Which DynDns provider do you use? none
```

- Create a Username and password

```
GUI Username or email address to authorize (empty to end): Admin
Password ********
```

- Press enter for key generation
- Enter our directory for output logs

```
? Path to the logs directory. (~/Desktop/velociraptor/logs) ~/Desktop/velociraptor
```

- Enter Y and enter twice to write a server/client config file

```
? Do you want to restrict VQL functionality on the server?

This is useful for a shared server where users are not fully trusted.
It removes potentially dangerous plugins like execve(), filesystem access etc.

NOTE: This is an experimental feature only useful in limited situations. If you
do not know you need it select N here!
 Yes
? Where should I write the server config file? server.config.yaml
? Where should I write the client config file? client.config.yaml
```

We are done with configuring the server! If you look now two .yaml files will be in the directory. One for server and one for client.

```
patrick@patrick-virtual-machine:~/Desktop$ ls
client.config.yaml  server.config.yaml  velociraptor-v0.72.3-linux-amd64
patrick@patrick-virtual-machine:~/Desktop$
```

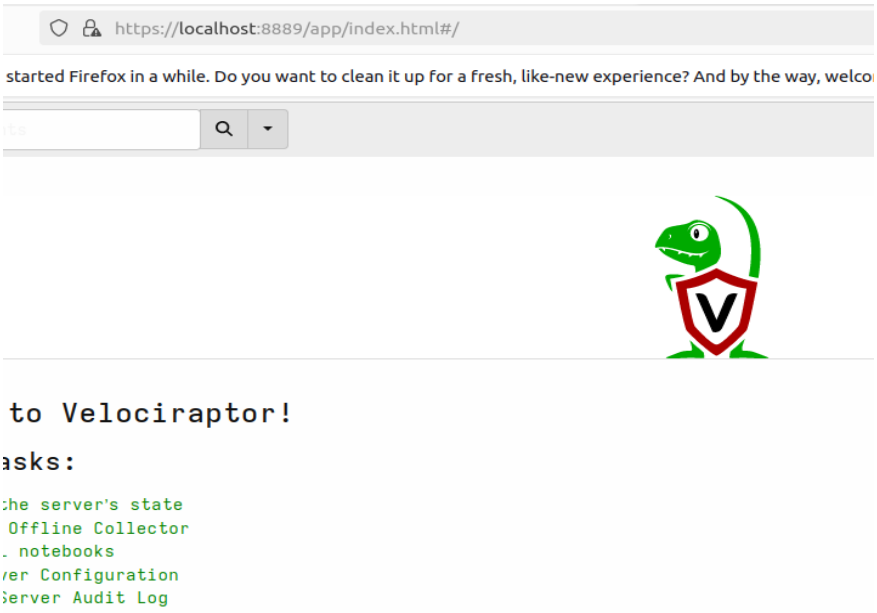
Now we will start the sever! Run the below command

```
./velociraptor-v0.72.3-linux-amd64 --config server.config.yaml frontend -v
```

Now the server is running. Remember, it is listening on port 8000 (tcp). We will need to allow this through the firewall. On mission this will most likely be through pfsense. In this scenario I'll just edit the firewall config on my host with the below command.

```
sudo ufw allow 8000/tcp
```

Lets check if the GUI is up. Should be at https://localhost:8889 if you recall



Looks like the server is running. Remember the GUI is only available on our host for management. We

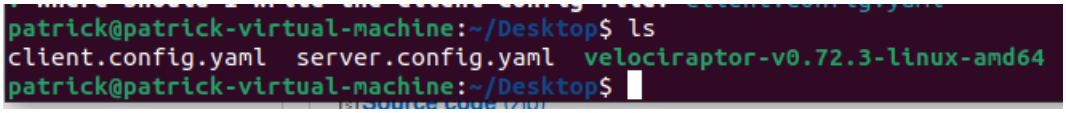
didn't open 8889 through the firewall.

Use the link below to learn basic navigation of the Admin GUI

https://docs.velociraptor.app/docs/gui

Now lets add a client to gather forensic evidence!

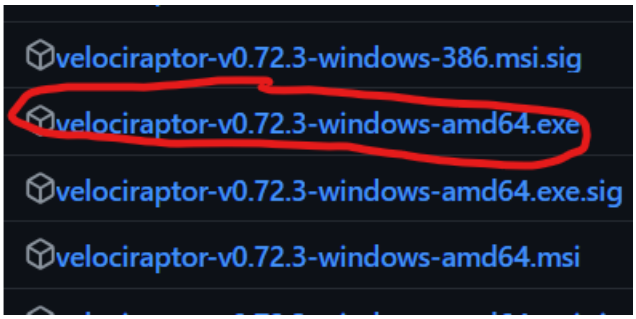
Remember that client configuration file? We're going to need that



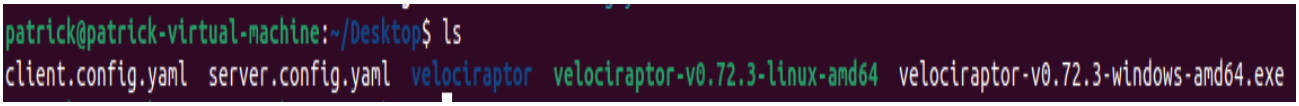
First, what is the OS of the client??? Lets do a windows host

We need to download velociraptor for windows.

https://github.com/Velocidex/velociraptor/releases/tag/v0.72

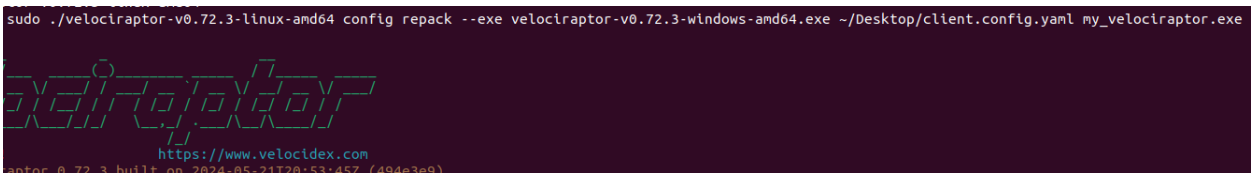


Save it to the directory that has your Linux Velociraptor



Now we can use Velociraptor to create our own easy binary to send to a Windows host, that will contain the server config file. Run the command below.

Sudo ./velociraptor-v0.72.3-linux-amd config repack –exe velociraptor-v0.72.3-windows-amd64.exe ~/Desktop/client.config.yaml my_velociraptor.exe



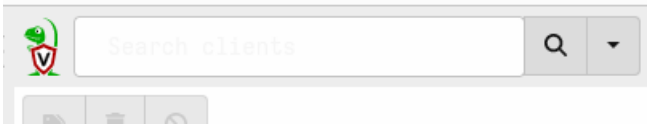
We are basically combining the windows executable with our client config yaml to make things easier. We're naming this new executable my_velociraptor.exe

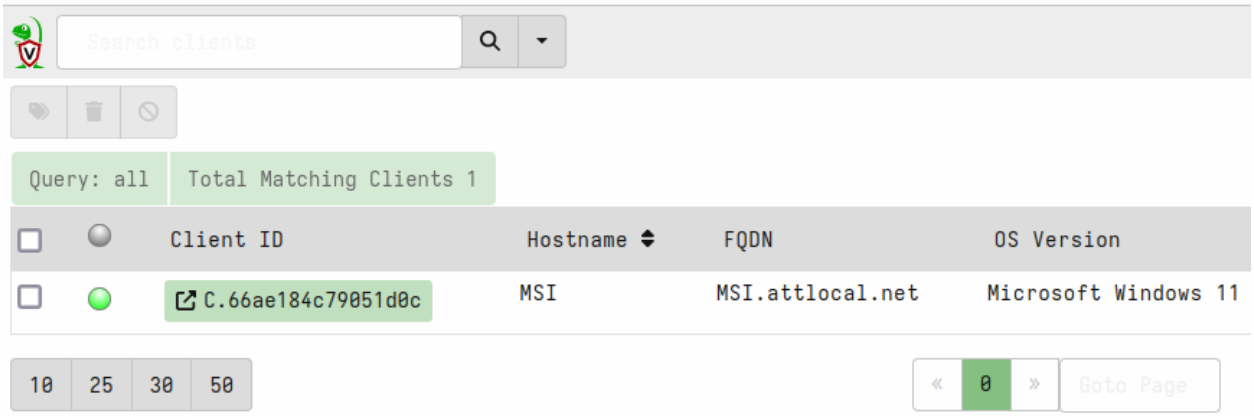
Now if we run my_velociraptor.exe on a host that's traffic isn't blocked to our serve, it should theoretically report itself. Now we have to transfer my_velociraptor.exe to our test client. Since this is a walkthrough on Velociraptor, I'm not going to go into file transfer (many different ways to do this). I'll go ahead a run it on the client.

Run the below command on the Windows client

```
.\my_velociraptor.exe service install
```

Now lets head back over to the admin GUI and check if the client is reporting. Click the magnifying glass on the upper left corner to reveal clients.

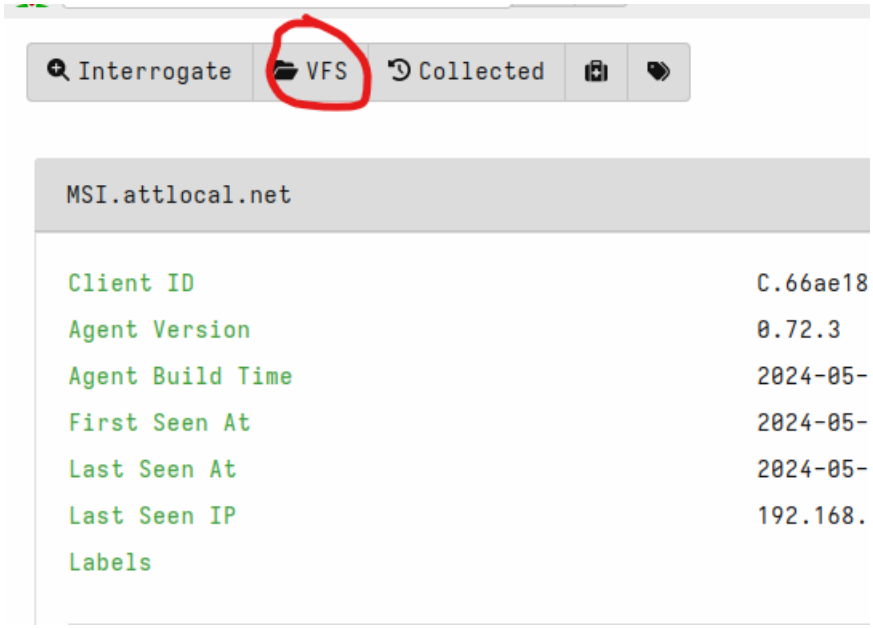




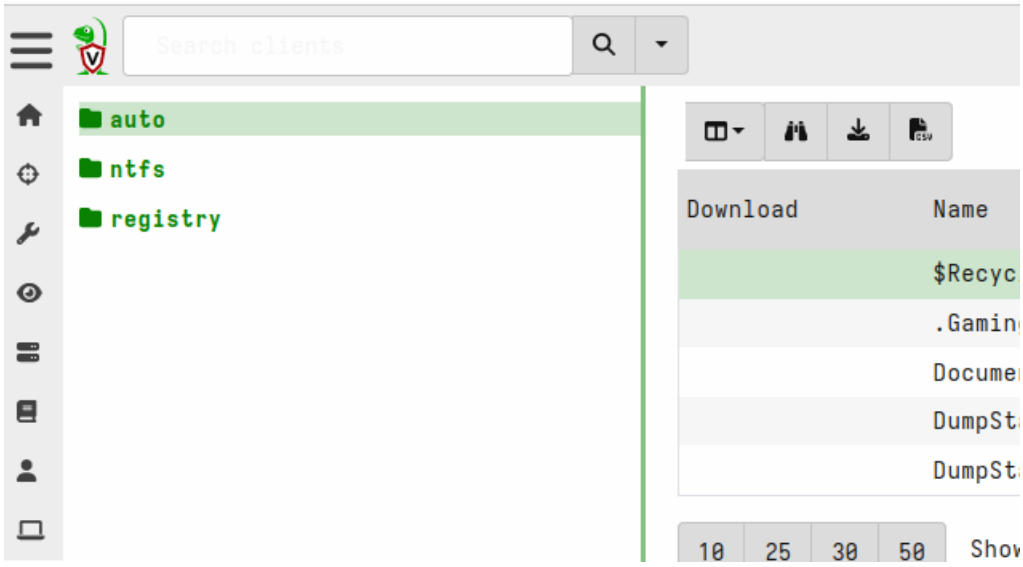
And it is! We have successfully started a velociraptor server and now have a client we can do forensics on!

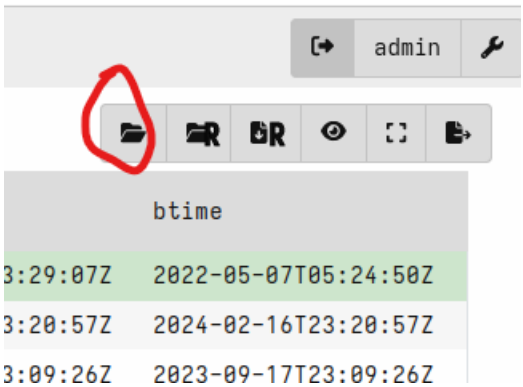
Forensic Investigation

- Click on the client ID and select VFS, this is the Virtual File system

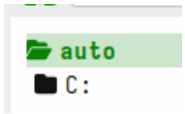


Navigating the VFS is a little strange. To navigate the filesystem in VFS, click on “auto” and then in the upper right corner click the refresh folder



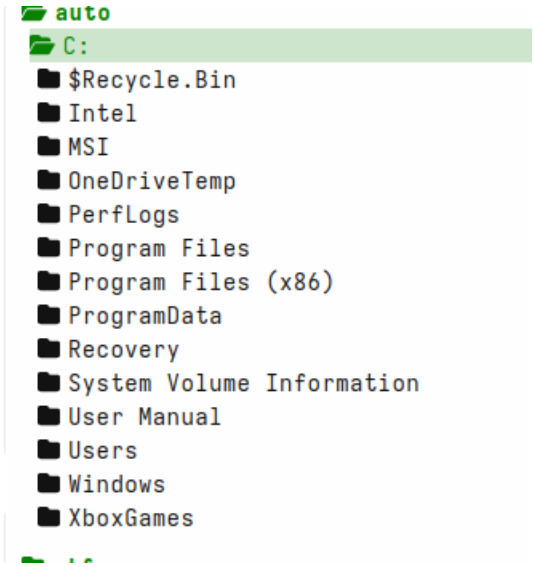


Now a C:\ should appear under auto (or whatever the drive name is).



Now you must click on the C: and hit the refresh folder again and keep doing this to navigate the file system.

Keep going until you get to the artifact you want to grab



Lets extract a potential IOC on the Desktop (Grab_this_file.txt)

Download	Name	Size	Mode	mtime
	GCFA	0	drwxrwxrwx	2023-11-24T20:16:10Z
	Grab_this_file.txt	0	-rw-rw-rw-	2024-05-24T19:13:35Z
	Virtual_Machines	0	drwxrwxrwx	2024-05-24T19:02:28Z

Select the file you want to grab, on the bottom of the screen you'll see a "collect from the client" button. Click that

After, it will hash the file and give you a download option

That's just a sample of what it can do

Automated Artifact retrieval

VQL – velociraptor's own query language. It closely resembles SQL

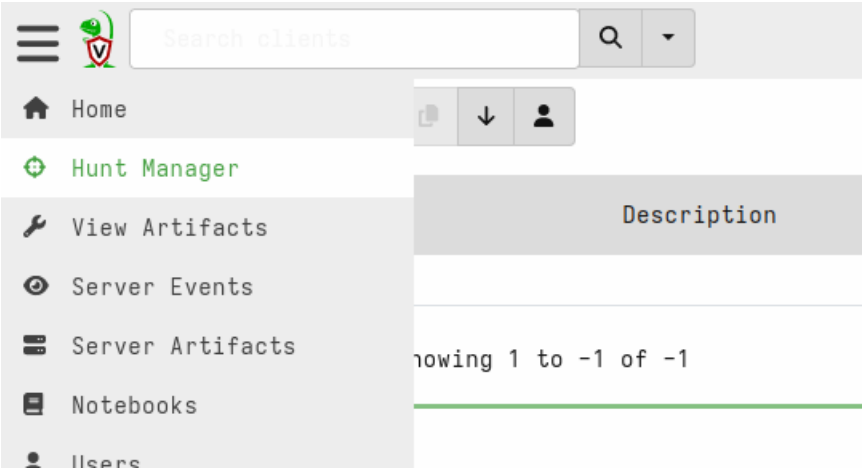
Within the admin GUI there are already pre-written VQL queries to retrieve certain windows aritifacts

You can also write your own queries (the link below has the basics) :

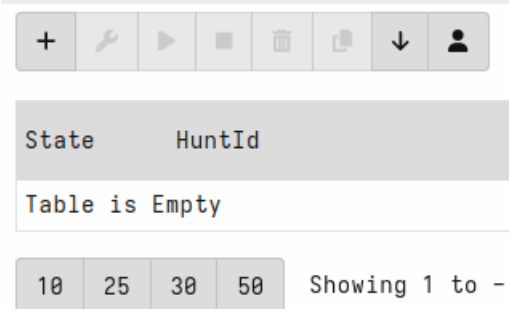
https://docs.velociraptor.app/docs/vql/

Lets see how to collect specific artifacts.

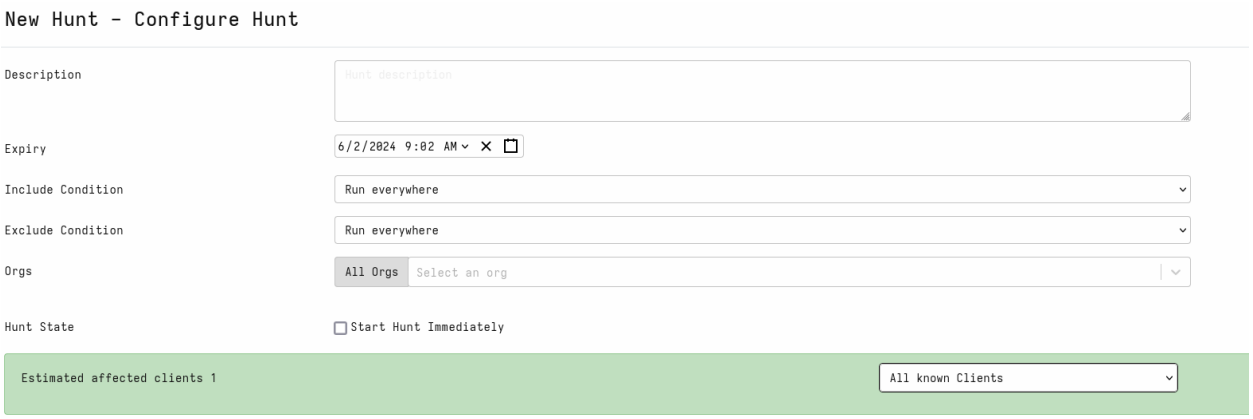
In the admin GUI click the hamburger icon in the upper left corner to reveal the drop down and select "hunt manger"



Click the "+" icon to start a new hunt

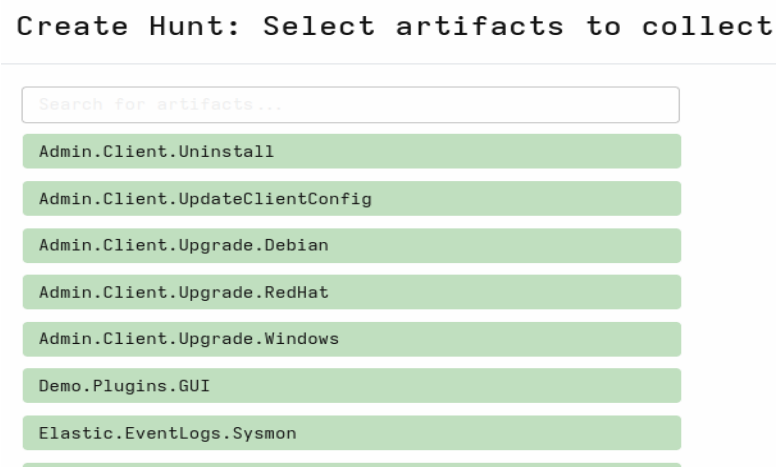


You will see a page to fill out basic information



Fill in the required information and at the bottom select “select Artifacts”

A list of premade VQL queries will appear. You can search for a select what you want to grab.

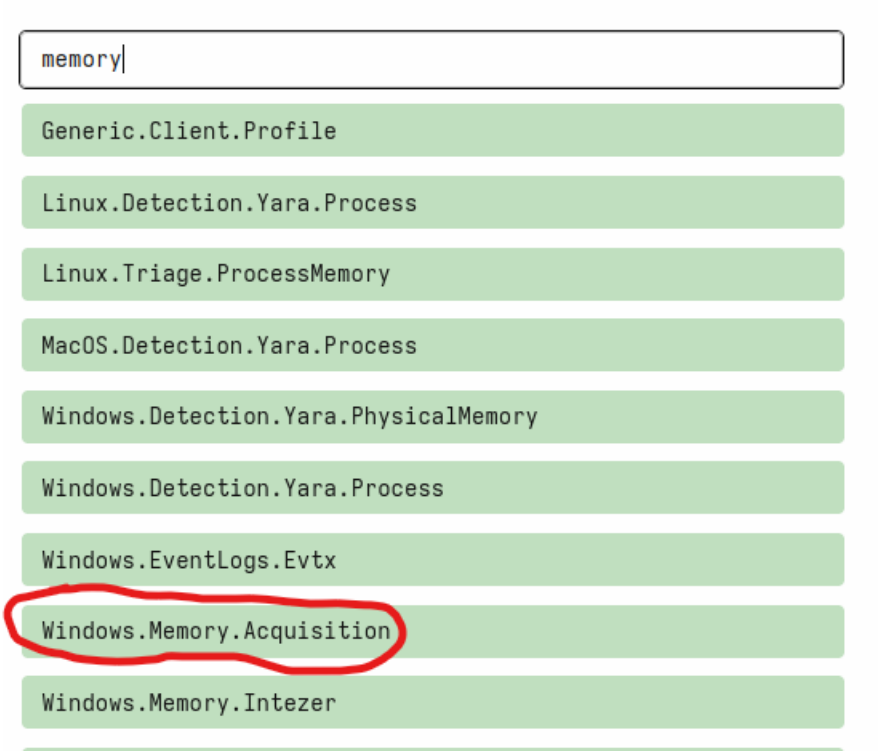


Lets try to use a premade VQL query

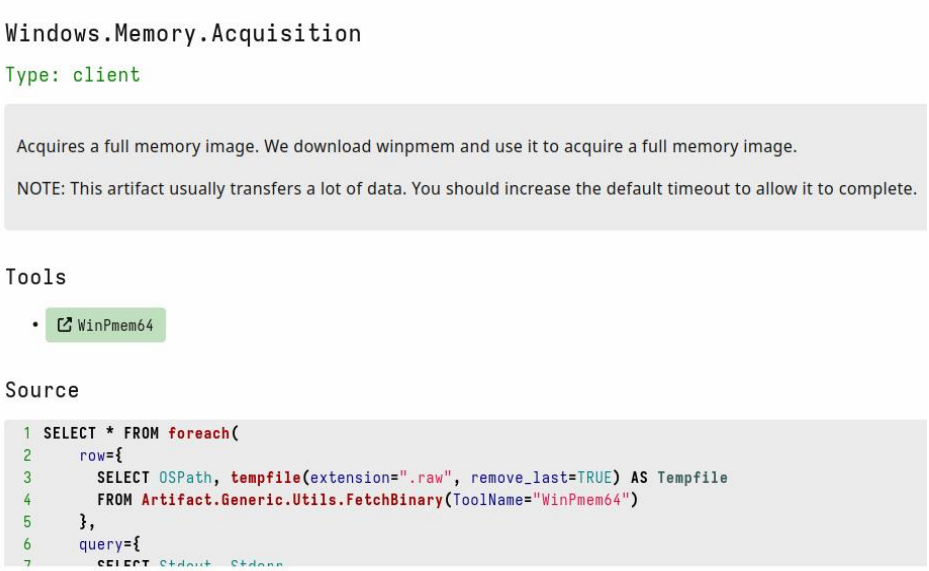
Memory acquisition

We will attempt to grab a memory dump from our client

Go to the premade VQL queries and search for memory

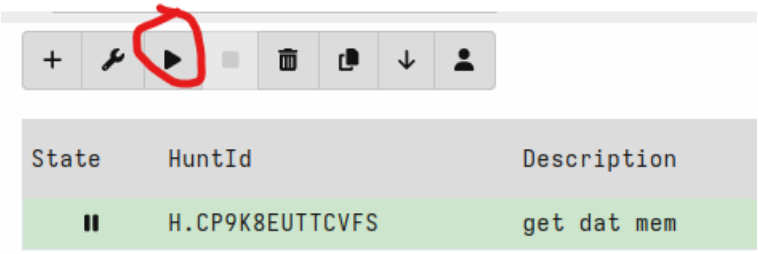


Select “Windows.Memory.Acquisition”. It will show you the contents of the VQL with a description of what id does.



Review the rest of the tabs, when ready select “Launch”. A cool feature is you can specify resources, lets say if it’s an important machine that you want to lower risks of DOSing the machine

After you select launch a new hunt will appear with a hunt id. Click on it and then click the play button to run the hunt



You will see a status under the state column