# Threat Hunting: Velociraptor for Endpoint Monitoring (Part 2)

September 26, 2020 By Raj Chandel

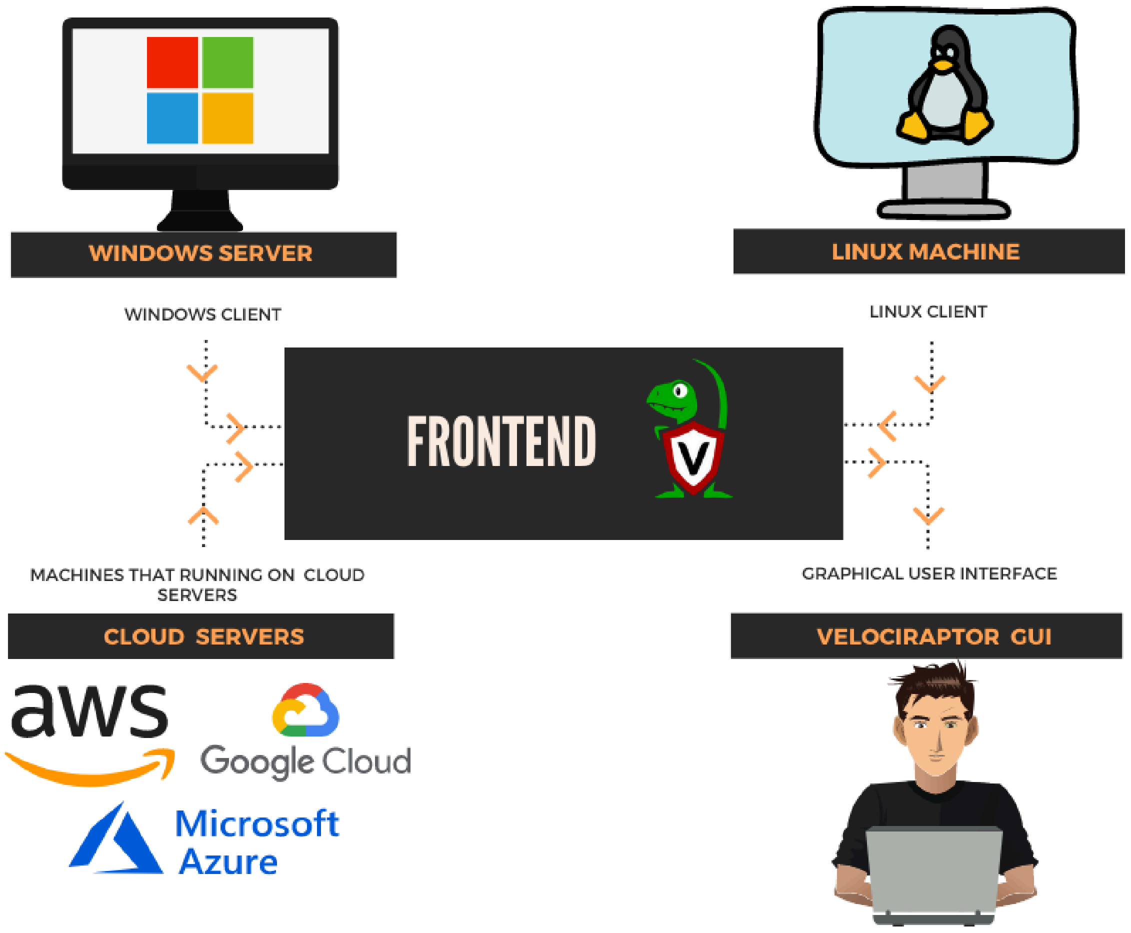
In our previous article, we have covered with Velociraptor master server setup with a brief demonstration of Velociraptor installation, GUI interface set up with some of the forensics Artifacts

If you didn't read that then don't worry you can visit that article from **here**.

## Threat Hunting: Velociraptor for Endpoint Monitoring

Once done with a complete server setup we need to focus on "how to Add Hosts or clients of our network environment" for Quick incident Response, forensics, Malware Analysis, and Threat Hunting. In this Blog, we are going to focus our attention only on those machines who shows potential sign of compromises

Now we see how to add a client to the Velociraptor server for further investigations.



## Table of Contents

- Prerequisites

- Agent or Client Environment
- Agent installation
  - For Linux Systems
  - For Windows server or windows 10
- Configure Agent to send Data to Velociraptor Server
- Forensics investigation / Threat Hunting

## Prerequisites

To configure **Velociraptor Agent** on your client-server, there are some prerequisites required for installation or pen-testing.

- Windows, Linux systems, or cloud servers with admin access.
- Velociraptor Agents
- Attacker: Kali Linux

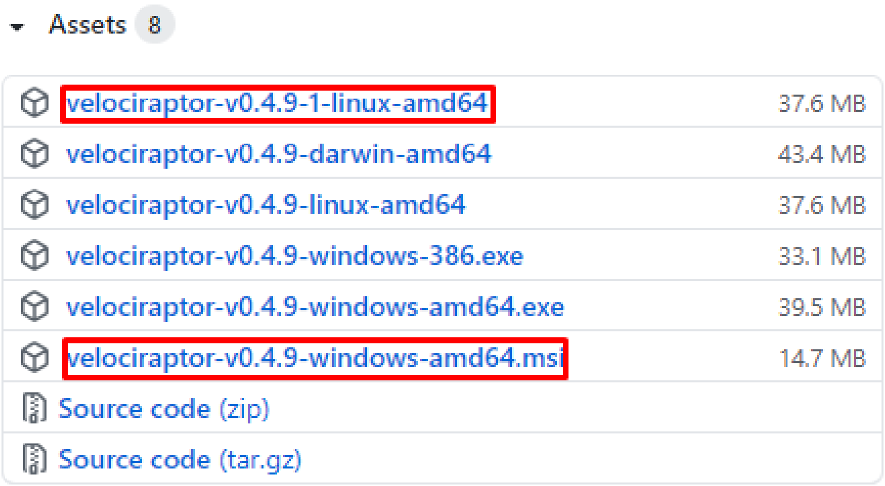
## Agent or Client Environment.

In this article, we will target to install Velociraptor Agents on a Windows server and Linux environments. You can download Velociraptor Agents by following the below link.

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

Choose your installation package

- Go to the official GitHub page of Velociraptor by following the above Link
- Select and install Velociraptor Agents as per your client system



Assets 8

| | |
|---|---|
| velociraptor-v0.4.9-1-linux-amd64 | 37.6 MB |
| velociraptor-v0.4.9-darwin-amd64 | 43.4 MB |
| velociraptor-v0.4.9-linux-amd64 | 37.6 MB |
| velociraptor-v0.4.9-windows-386.exe | 33.1 MB |
| velociraptor-v0.4.9-windows-amd64.exe | 39.5 MB |
| velociraptor-v0.4.9-windows-amd64.msi | 14.7 MB |
| Source code (zip) | |
| Source code (tar.gz) | |

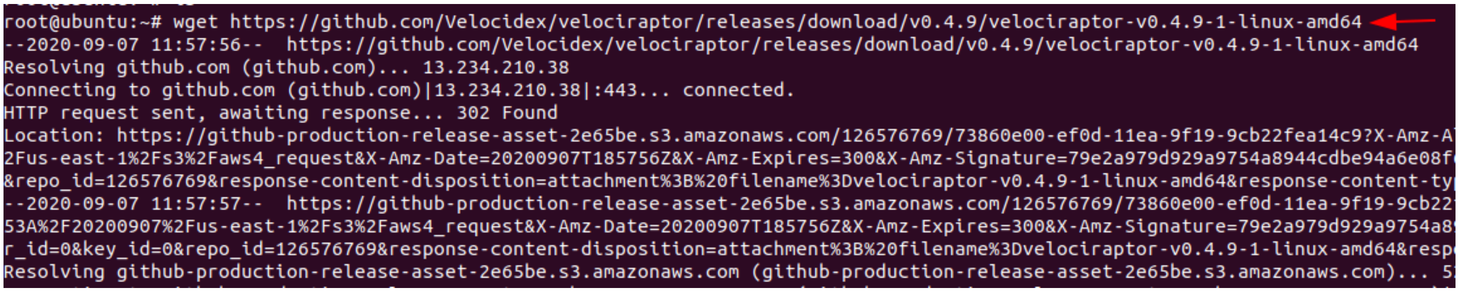
## Agent installation

### For Linux Systems !!

To install Velociraptor Agent into your Linux systems, follow the steps as described below: Visit to the official GitHub page of Velociraptor locate and select Velociraptor-Linux-amd64 Package

I prefer to download this package via terminal with wget. To download Agent issue the following command into the terminal.

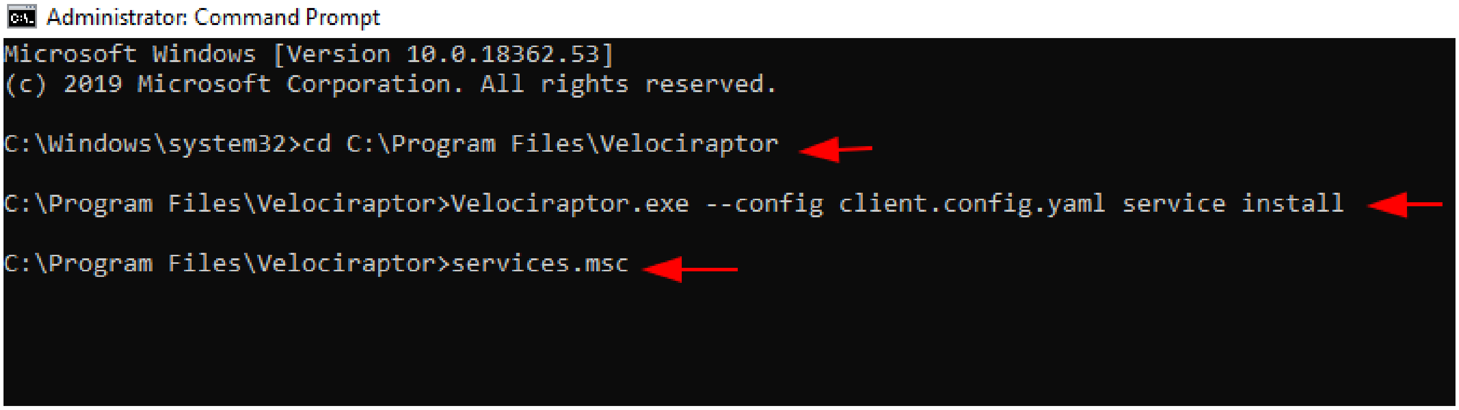
```
wget https://github.com/Velocidex/velociraptor/releases/download/v0.4.9/velocirapt
```



```
root@ubuntu:~# wget https://github.com/Velocidex/velociraptor/releases/download/v0.4.9/velociraptor-v0.4.9-1-linux-amd64
--2020-09-07 11:57:56--  https://github.com/Velocidex/velociraptor/releases/download/v0.4.9/velociraptor-v0.4.9-1-linux-amd64
Resolving github.com (github.com)... 13.234.210.38
Connecting to github.com (github.com)|13.234.210.38|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/126576769/73860e00-ef0d-11ea-9f19-9cb22fea14c9?X-Amz-A
2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200907T185756Z&X-Amz-Expires=300&X-Amz-Signature=79e2a979d929a9754a8944cdbe94a6e08f
&repo_id=126576769&response-content-disposition=attachment%3B%20filename%3Dvelociraptor-v0.4.9-1-linux-amd64&response-content-ty
--2020-09-07 11:57:57--  https://github-production-release-asset-2e65be.s3.amazonaws.com/126576769/73860e00-ef0d-11ea-9f19-9cb22
53A%2F20200907%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200907T185756Z&X-Amz-Expires=300&X-Amz-Signature=79e2a979d929a9754a8
r_id=0&key_id=0&repo_id=126576769&response-content-disposition=attachment%3B%20filename%3Dvelociraptor-v0.4.9-1-linux-amd64&resp
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 5
```

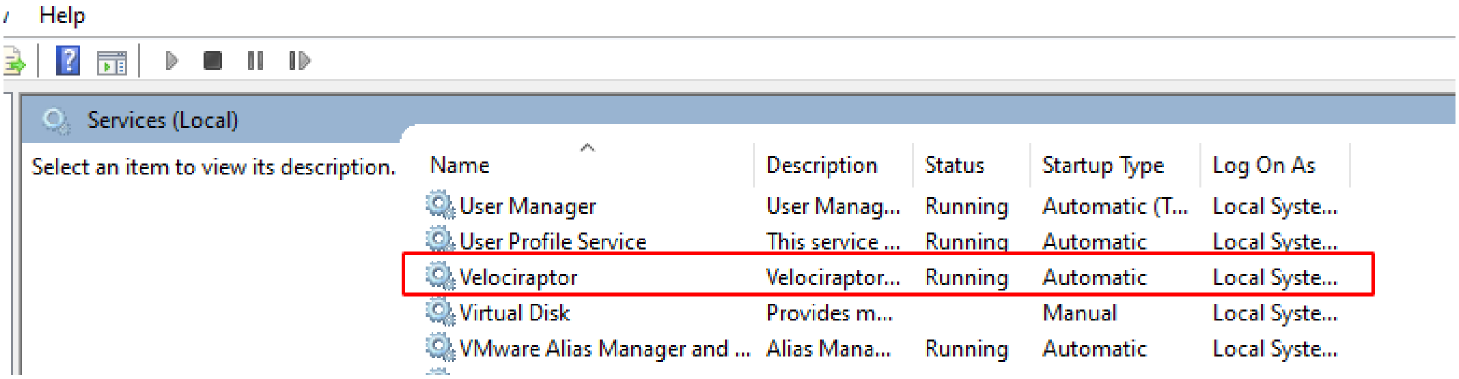
After downloading it, return to your Velociraptor Master Server and issue the following command to install a client service into the server so that it becomes active to accept connections from the client.

```
cd C:\Program Files\Velociraptor
Velociraptor.exe --config server.config.yaml service install
services.msc
```



```
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\Velociraptor

C:\Program Files\Velociraptor>Velociraptor.exe --config client.config.yaml service install

C:\Program Files\Velociraptor>services.msc
```

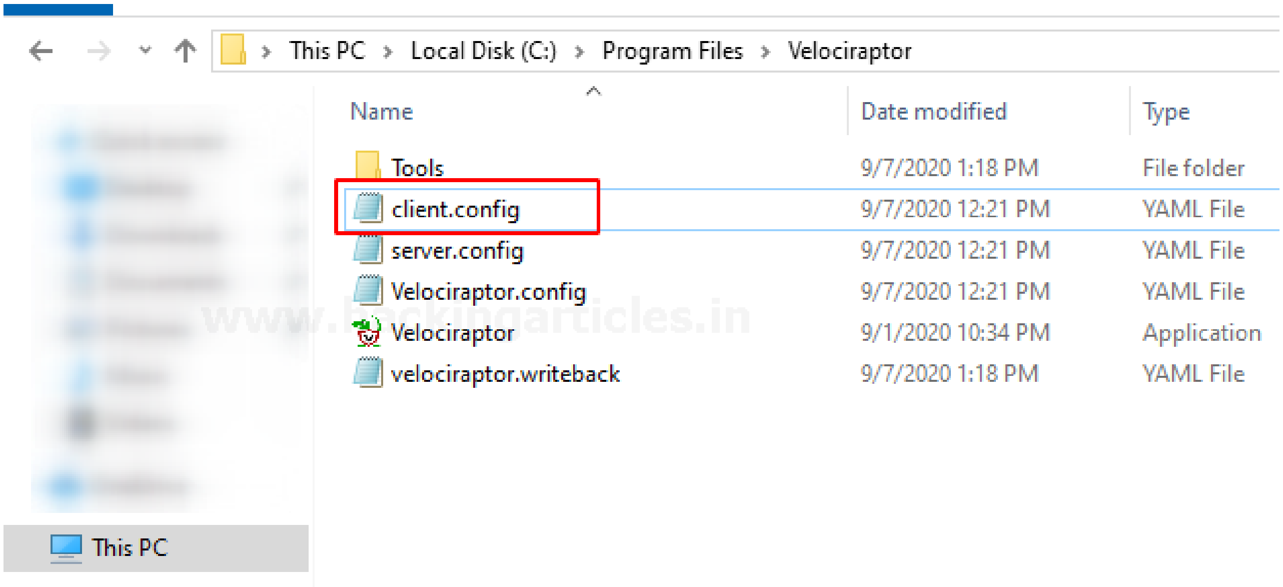
Also, you can verify whether the service is running or not by issuing command **services.msc** it will open a prompt on your screen as shown below:



| Name | Description | Status | Startup Type | Log On As |
|---|---|---|---|---|
| User Manager | User Manag... | Running | Automatic (T... | Local Syste... |
| User Profile Service | This service ... | Running | Automatic | Local Syste... |
| Velociraptor | Velociraptor... | Running | Automatic | Local Syste... |
| Virtual Disk | Provides m... | | Manual | Local Syste... |
| VMware Alias Manager and ... | Alias Mana... | Running | Automatic | Local Syste... |

Nice! As we can see service is enabled or running.

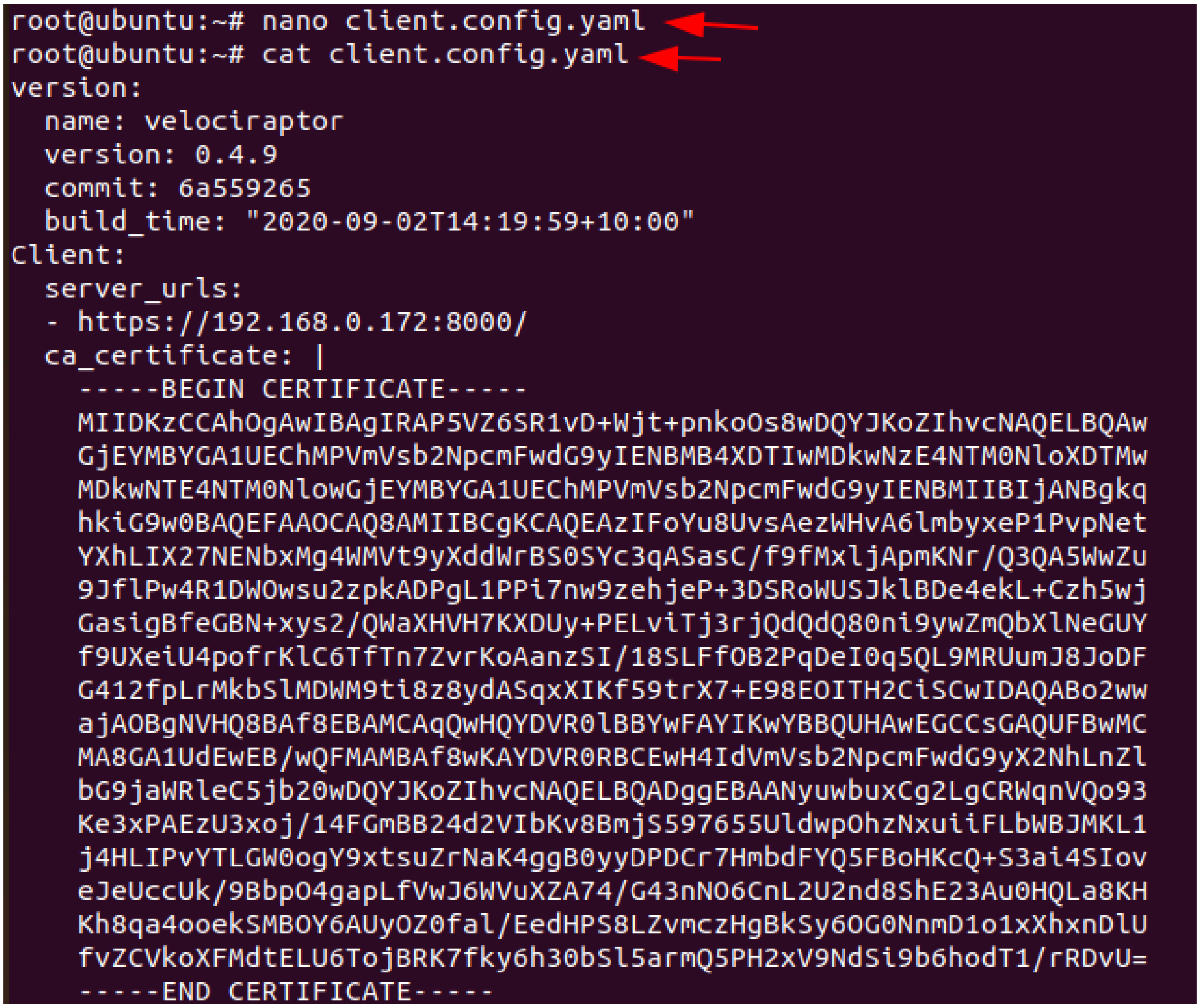
Next, come to the Directory where the Velociraptor server installed and copy the configuration of the client.config



| Name | Date modified | Type |
|---|---|---|
| Tools | 9/7/2020 1:18 PM | File folder |
| client.config | 9/7/2020 12:21 PM | YAML File |
| server.config | 9/7/2020 12:21 PM | YAML File |
| Velociraptor.config | 9/7/2020 12:21 PM | YAML File |
| Velociraptor | 9/1/2020 10:34 PM | Application |
| velociraptor.writeback | 9/7/2020 1:18 PM | YAML File |

## Configure Agent to send data to Velociraptor server

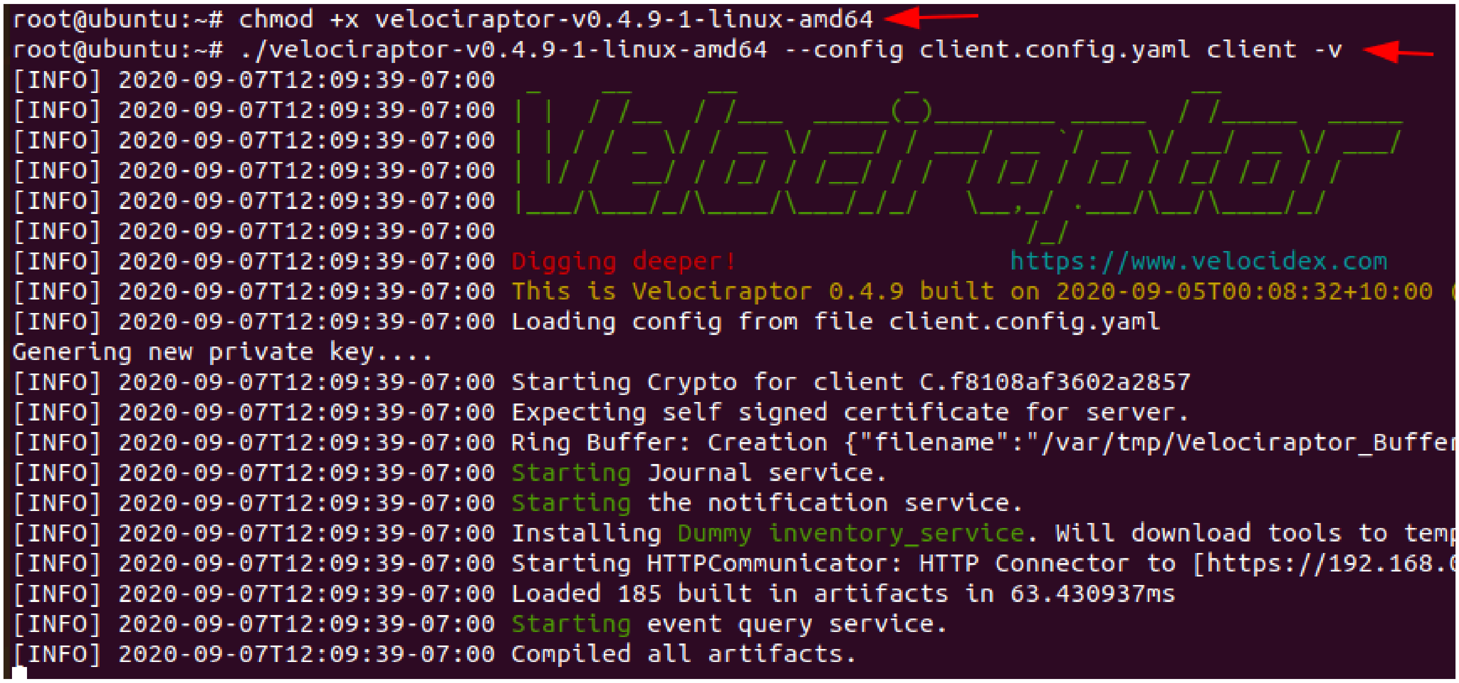
Return to Linux machine and create a client.config.yaml file and paste the configuration of the client.config file which we have copied above inside a **client.config.yaml**.



```
root@ubuntu:~# nano client.config.yaml
root@ubuntu:~# cat client.config.yaml
version:
  name: velociraptor
  version: 0.4.9
  commit: 6a559265
  build_time: "2020-09-02T14:19:59+10:00"
Client:
  server_urls:
  - https://192.168.0.172:8000/
  ca_certificate: |
    -----BEGIN CERTIFICATE-----
    MIIDKzCCAhOgAwIBAgIRAP5VZ6SR1vD+Wjt+pnkoOs8wDQYJKoZIhvcNAQELBQAw
    GjEYMBYGA1UEChMPVmVsb2NpcmFwdG9yIENBMB4XDTIwMDkwNzE4NTM0NloXDTMw
    MDkwNTE4NTM0NlowGjEYMBYGA1UEChMPVmVsb2NpcmFwdG9yIENBMIIBIjANBgkq
    hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzIFoYu8UvsAezWHvA6lmbyxeP1PvpNet
    YXhLIX27NENbxMg4WMVt9yXddWrBS0SYc3qASasC/f9fMxljApmKNr/Q3QA5WwZu
    9JflPw4R1DWOwsu2zpkADPgL1PPi7nw9zehjeP+3DSRoWUSJklBDe4ekL+Czh5wj
    GasigBfeGBN+xys2/QWaXHVH7KXDUy+PELviTj3rjQdQdQ80ni9ywZmQbXlNeGUY
    f9UXeiU4pofrKlC6TfTn7ZvrKoAanzSI/18SLFfOB2PqDeI0q5QL9MRUumJ8JoDF
    G412fpLrMkbSlMDWM9ti8z8ydASqxXIKf59trX7+E98EOITH2CiSCwIDAQABo2ww
    ajAOBgNVHQ8BAf8EBAMCAqQwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMC
    MA8GA1UdEwEB/wQFMAMBAf8wKAYDVR0RBCEwH4IdVmVsb2NpcmFwdG9yX2NhLnZl
    bG9jaWRleC5jb20wDQYJKoZIhvcNAQELBQADggEBAANyuwbuxCg2LgCRWqnVQo93
    Ke3xPAEzU3xoj/14FGmBB24d2VIbKv8BmjS597655UldwpOhzNxuiiFLbWBJMKL1
    j4HLIPvYTLGW0ogY9xtsuZrNaK4ggB0yyDPDCr7HmbdFYQ5FBoHKcQ+S3ai4SIov
    eJeUccUk/9BbpO4gapLfVwJ6WVuXZA74/G43nNO6CnL2U2nd8ShE23Au0HQLa8KH
    Kh8qa4ooekSMBOY6AUyOZ0fal/EedHPS8LZvmczHgBkSy6OG0NnmD1o1xXhxnDlU
    fvZCVkoXFMdtELU6TojBRK7fky6h30bSl5armQ5PH2xV9NdSi9b6hodT1/rRDvU=
    -----END CERTIFICATE-----
```

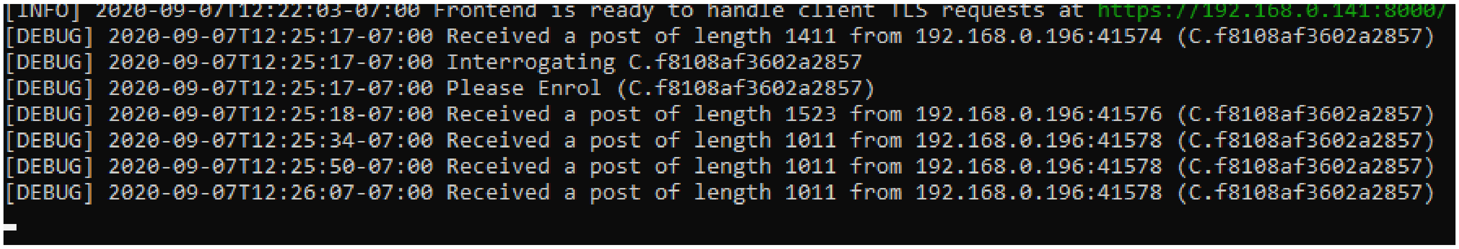
This client configuration file contains a CA certificate that is used for authentication between the client's machine to the Velociraptor Master server.  After that change permission of the Downloaded Velociraptor Agent to make it executable and then deploy the client to Velociraptor by executing the following command:

```
chmod +x velociraptor -v0.4.9-1-Linux-amd64
./velociraptor-v0.4.9-1-Linux-amd64 –config client.config.yaml client -v
```



```
root@ubuntu:~# chmod +x velociraptor-v0.4.9-1-linux-amd64
root@ubuntu:~# ./velociraptor-v0.4.9-1-linux-amd64 --config client.config.yaml client -v
[INFO] 2020-09-07T12:09:39-07:00  _    __     __           _                    __
[INFO] 2020-09-07T12:09:39-07:00 | |  / /__  / /___  _____(_)________ _____  / /_____  _____
[INFO] 2020-09-07T12:09:39-07:00 | | / / _ \/ / __ \/ ___/ / ___/ __ `/ __ \/ __/ __ \/ ___/
[INFO] 2020-09-07T12:09:39-07:00 | |/ /  __/ / /_/ / /__/ / /  / /_/ / /_/ / /_/ /_/ / /
[INFO] 2020-09-07T12:09:39-07:00 |___/\___/_/\____/\___/_/_/   \__,_/ .___/\__/\____/_/
[INFO] 2020-09-07T12:09:39-07:00                                  /_/
[INFO] 2020-09-07T12:09:39-07:00 Digging deeper!                  https://www.velocidex.com
[INFO] 2020-09-07T12:09:39-07:00 This is Velociraptor 0.4.9 built on 2020-09-05T00:08:32+10:00 (
[INFO] 2020-09-07T12:09:39-07:00 Loading config from file client.config.yaml
Genering new private key....
[INFO] 2020-09-07T12:09:39-07:00 Starting Crypto for client C.f8108af3602a2857
[INFO] 2020-09-07T12:09:39-07:00 Expecting self signed certificate for server.
[INFO] 2020-09-07T12:09:39-07:00 Ring Buffer: Creation {"filename":"/var/tmp/Velociraptor_Buffer
[INFO] 2020-09-07T12:09:39-07:00 Starting Journal service.
[INFO] 2020-09-07T12:09:39-07:00 Starting the notification service.
[INFO] 2020-09-07T12:09:39-07:00 Installing Dummy inventory_service. Will download tools to temp
[INFO] 2020-09-07T12:09:39-07:00 Starting HTTPCommunicator: HTTP Connector to [https://192.168.(
[INFO] 2020-09-07T12:09:39-07:00 Loaded 185 built in artifacts in 63.430937ms
[INFO] 2020-09-07T12:09:39-07:00 Starting event query service.
[INFO] 2020-09-07T12:09:39-07:00 Compiled all artifacts.
```

Hmm:) !! As you can see service is started sending logs to the Velociraptor server. You can ensure the integration of the client (Ubuntu) machine with the server inside the Velociraptor Master Server which will generate logs for the client connectivity as shown in the image.

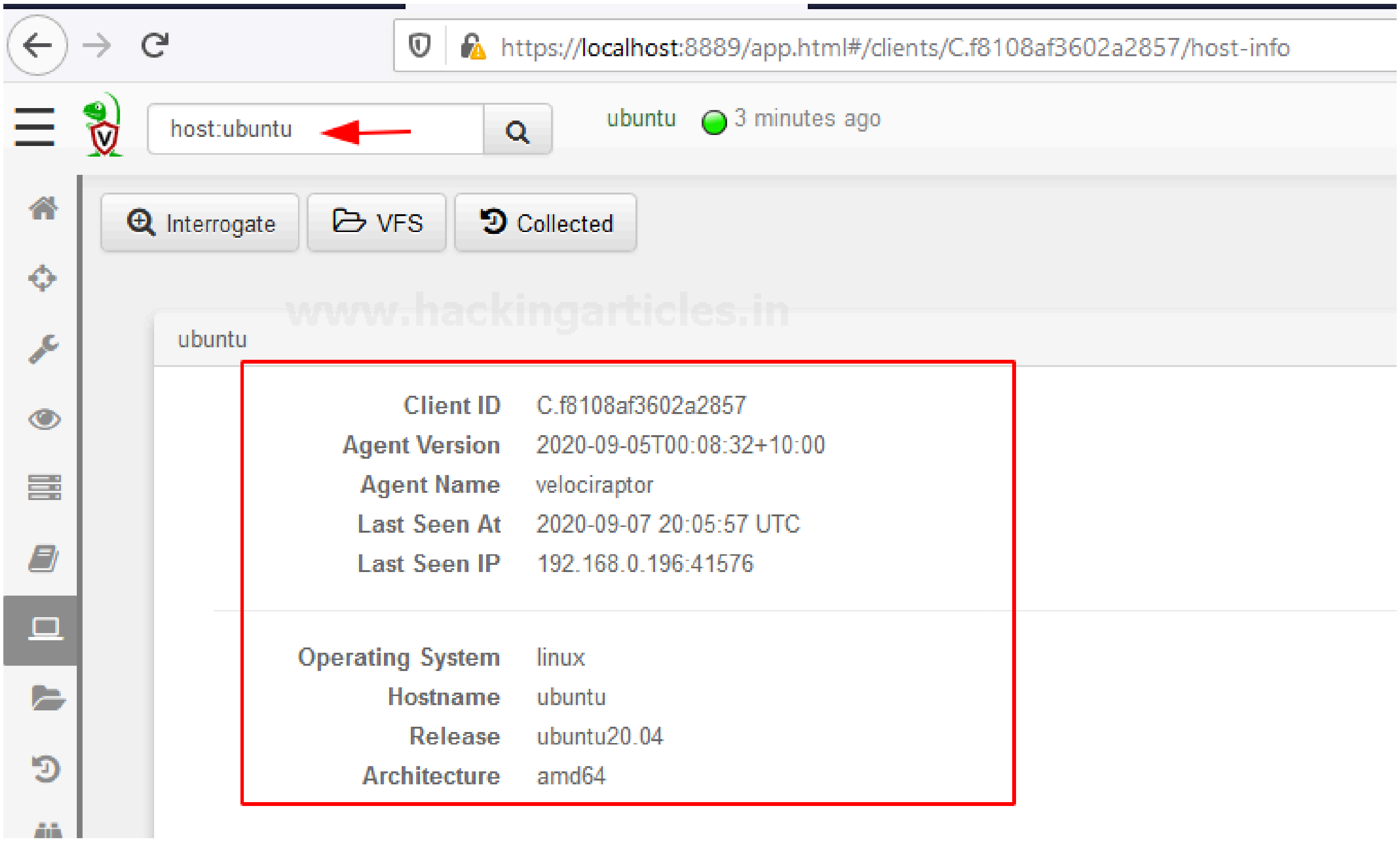


```
[INFO] 2020-09-07T12:22:03-07:00 Frontend is ready to handle client TLS requests at https://192.168.0.141:8000/
[DEBUG] 2020-09-07T12:25:17-07:00 Received a post of length 1411 from 192.168.0.196:41574 (C.f8108af3602a2857)
[DEBUG] 2020-09-07T12:25:17-07:00 Interrogating C.f8108af3602a2857
[DEBUG] 2020-09-07T12:25:17-07:00 Please Enrol (C.f8108af3602a2857)
[DEBUG] 2020-09-07T12:25:18-07:00 Received a post of length 1523 from 192.168.0.196:41576 (C.f8108af3602a2857)
[DEBUG] 2020-09-07T12:25:34-07:00 Received a post of length 1011 from 192.168.0.196:41578 (C.f8108af3602a2857)
[DEBUG] 2020-09-07T12:25:50-07:00 Received a post of length 1011 from 192.168.0.196:41578 (C.f8108af3602a2857)
[DEBUG] 2020-09-07T12:26:07-07:00 Received a post of length 1011 from 192.168.0.196:41578 (C.f8108af3602a2857)
```

Let's navigate to **http://localhost:8889** to access the GUI interface and verify whether the client is reflected on the interface or not by simply running a query in the search bar

```
host:ubuntu
```

where Ubuntu is my client's system name


https://localhost:8889/app.html#/clients/C.f8108af3602a2857/host-info
host:ubuntu
ubuntu
3 minutes ago
Interrogate
VFS
Collected
ubuntu

| | |
|---|---|
| Client ID | C.f8108af3602a2857 |
| Agent Version | 2020-09-05T00:08:32+10:00 |
| Agent Name | velociraptor |
| Last Seen At | 2020-09-07 20:05:57 UTC |
| Last Seen IP | 192.168.0.196:41576 |
| Operating System | linux |
| Hostname | ubuntu |
| Release | ubuntu20.04 |
| Architecture | amd64 |

Ok ? !! you have successfully added the Linux system as a client

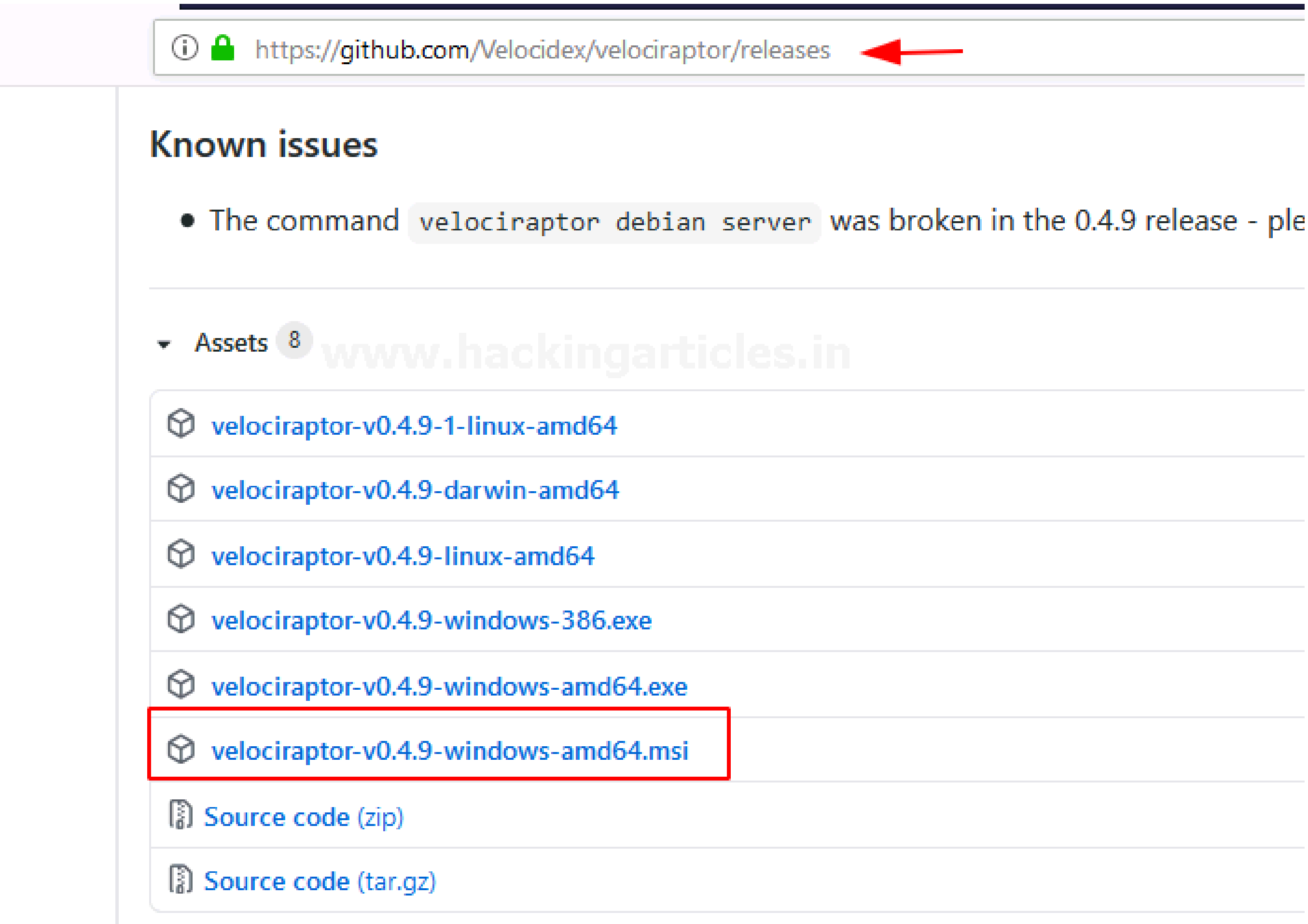
## For Windows Systems !!

As described above you can download Velociraptor Agent for your windows system by official GitHub page of a velociraptor

In my case, I will target to install Velociraptor agent in Windows server 2016.

Let’s begin the installation !!

Download package **velociraptor-v0.4.9-windows-amd6464.msi**, It will download a ZIP file into Your downloads open it install into the system.



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

## Known issues

- The command `velociraptor debian server` was broken in the 0.4.9 release - ple

Assets 8

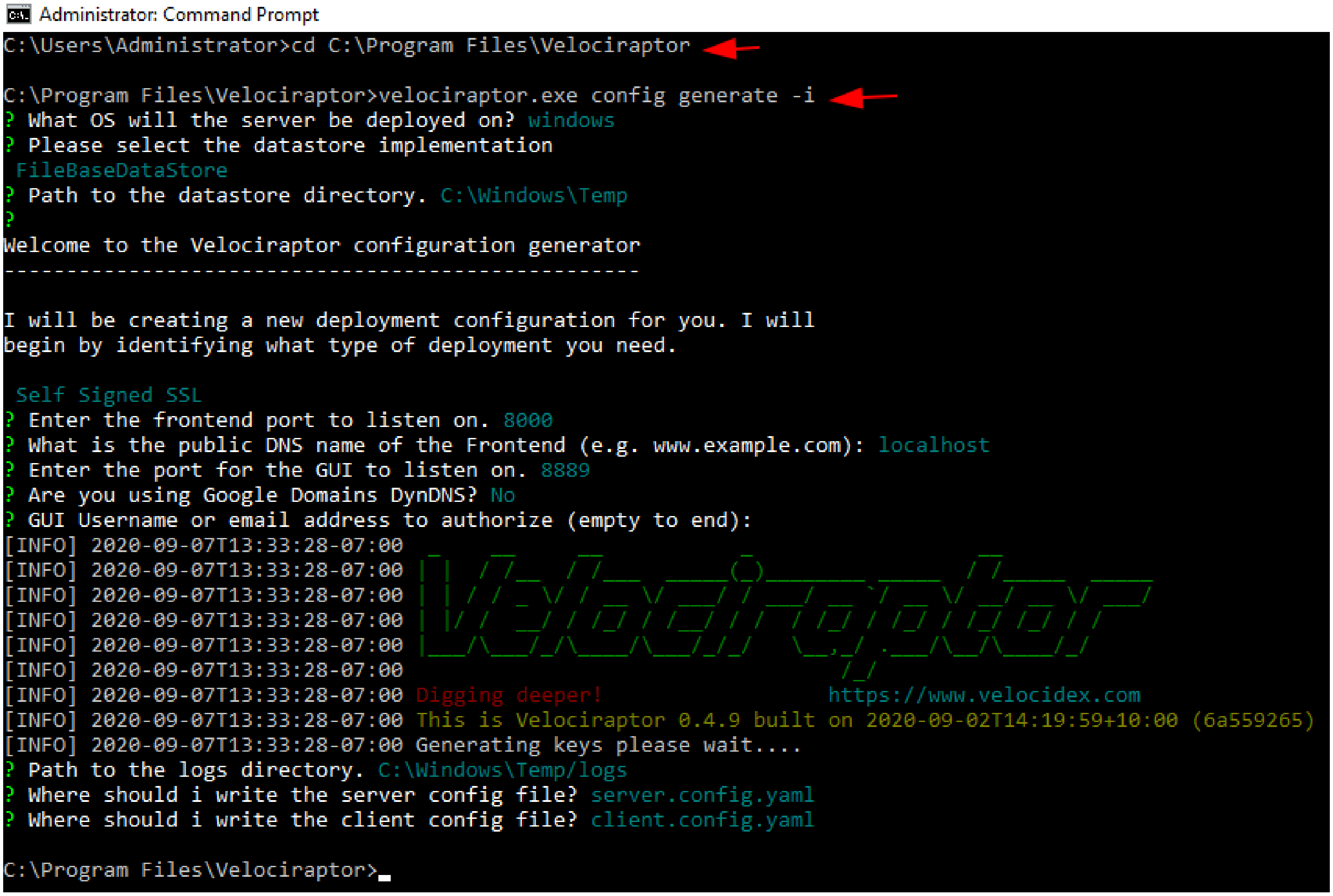
- velociraptor-v0.4.9-1-linux-amd64
- velociraptor-v0.4.9-darwin-amd64
- velociraptor-v0.4.9-linux-amd64
- velociraptor-v0.4.9-windows-386.exe
- velociraptor-v0.4.9-windows-amd64.exe
- velociraptor-v0.4.9-windows-amd64.msi
- Source code (zip)
- Source code (tar.gz)

Configure Agent to send data to Velociraptor server Open the command prompt with administrator privilege and navigate to velociraptor folder.

```
cd C:\Program Files\Velociraptor
```

So now what we need to do is to generate the configuration. To generate the configuration execute the following command.

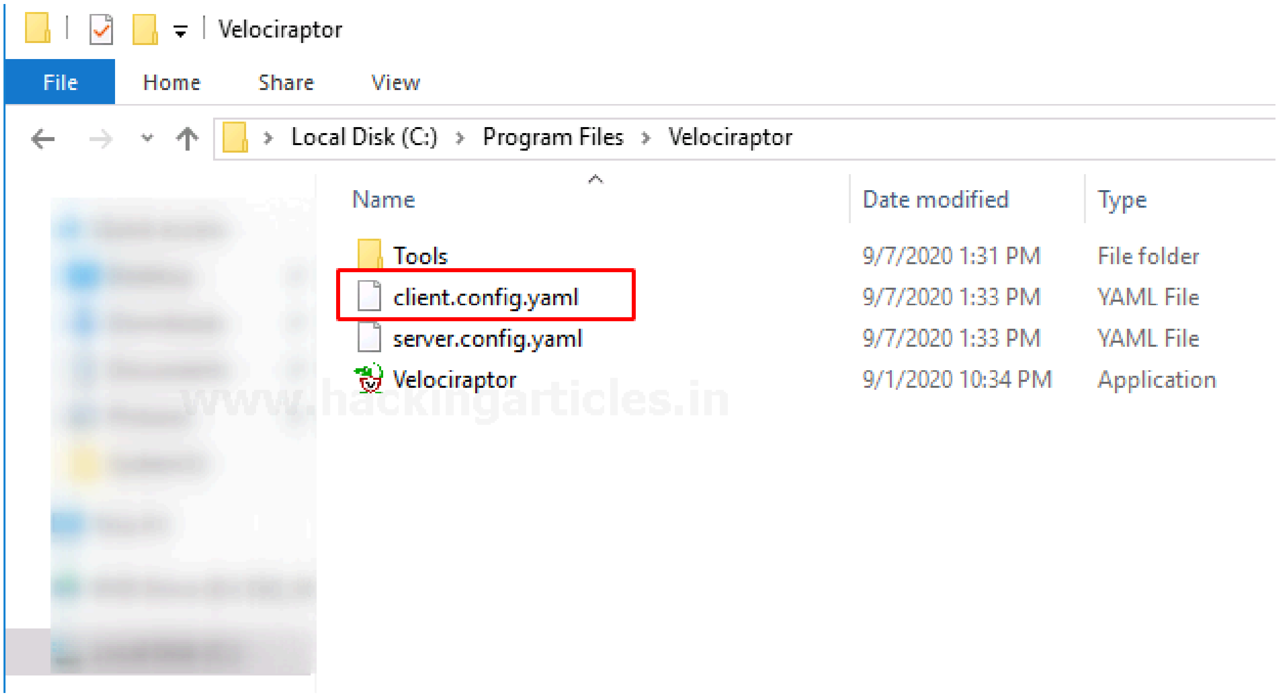
```
velociraptor.exe config generate -i
```



```
C:\Users\Administrator>cd C:\Program Files\Velociraptor

C:\Program Files\Velociraptor>velociraptor.exe config generate -i
? What OS will the server be deployed on? windows
? Please select the datastore implementation
 FileBaseDataStore
? Path to the datastore directory. C:\Windows\Temp
?
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
? Enter the frontend port to listen on. 8000
? What is the public DNS name of the Frontend (e.g. www.example.com): localhost
? Enter the port for the GUI to listen on. 8889
? Are you using Google Domains DynDNS? No
? GUI Username or email address to authorize (empty to end):
[INFO] 2020-09-07T13:33:28-07:00  _    __     __           _                  __
[INFO] 2020-09-07T13:33:28-07:00 | |  / /__  / /___  _____(_)________ _____  / /_____  _____
[INFO] 2020-09-07T13:33:28-07:00 | | / / _ \/ / __ \/ ___/ / ___/ __ `/ __ \/ __/ __ \/ ___/
[INFO] 2020-09-07T13:33:28-07:00 | |/ /  __/ / /_/ / /__/ / /  / /_/ / /_/ / /_/ /_/ / /
[INFO] 2020-09-07T13:33:28-07:00 |___/\___/_/\____/\___/_/_/   \__,_/ .___/\__/\____/_/
[INFO] 2020-09-07T13:33:28-07:00                                   /_/
[INFO] 2020-09-07T13:33:28-07:00 Digging deeper!                  https://www.velocidex.com
[INFO] 2020-09-07T13:33:28-07:00 This is Velociraptor 0.4.9 built on 2020-09-02T14:19:59+10:00 (6a559265)
[INFO] 2020-09-07T13:33:28-07:00 Generating keys please wait....
? Path to the logs directory. C:\Windows\Temp/logs
? Where should i write the server config file? server.config.yaml
? Where should i write the client config file? client.config.yaml

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

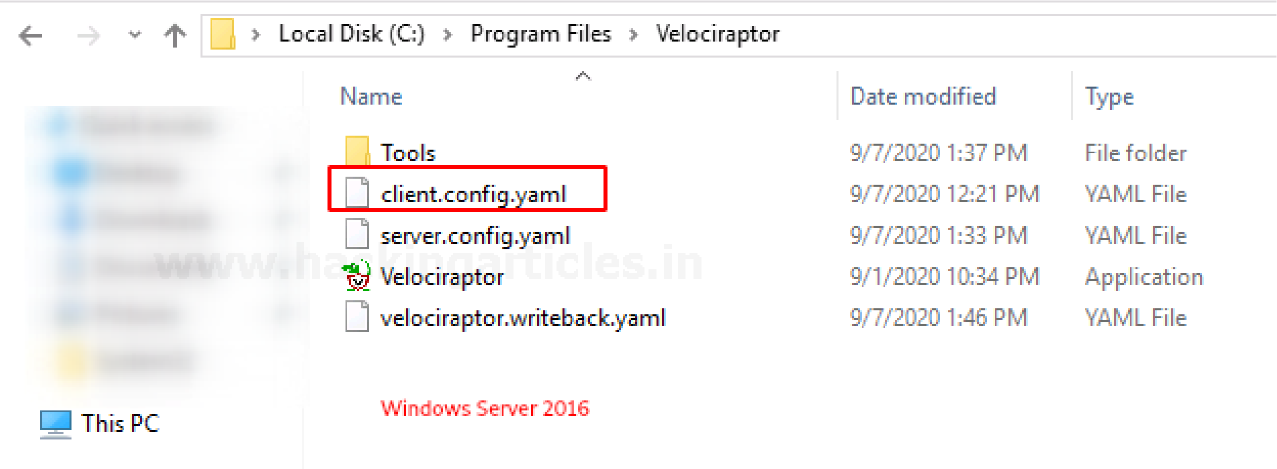
Hmm great !! as we can see the agent is installed successfully. Now, since we have this part done

Return to the Velociraptor master server and go to the directory where it is installed and what we need to do is to copy the client.config.yaml file.



| Name | Date modified | Type |
|---|---|---|
| Tools | 9/7/2020 1:31 PM | File folder |
| client.config.yaml | 9/7/2020 1:33 PM | YAML File |
| server.config.yaml | 9/7/2020 1:33 PM | YAML File |
| Velociraptor | 9/1/2020 10:34 PM | Application |

Then come back to the windows machine open the directory where Agent is installed and replace the client.config,yaml by simply pasting the file into that directory



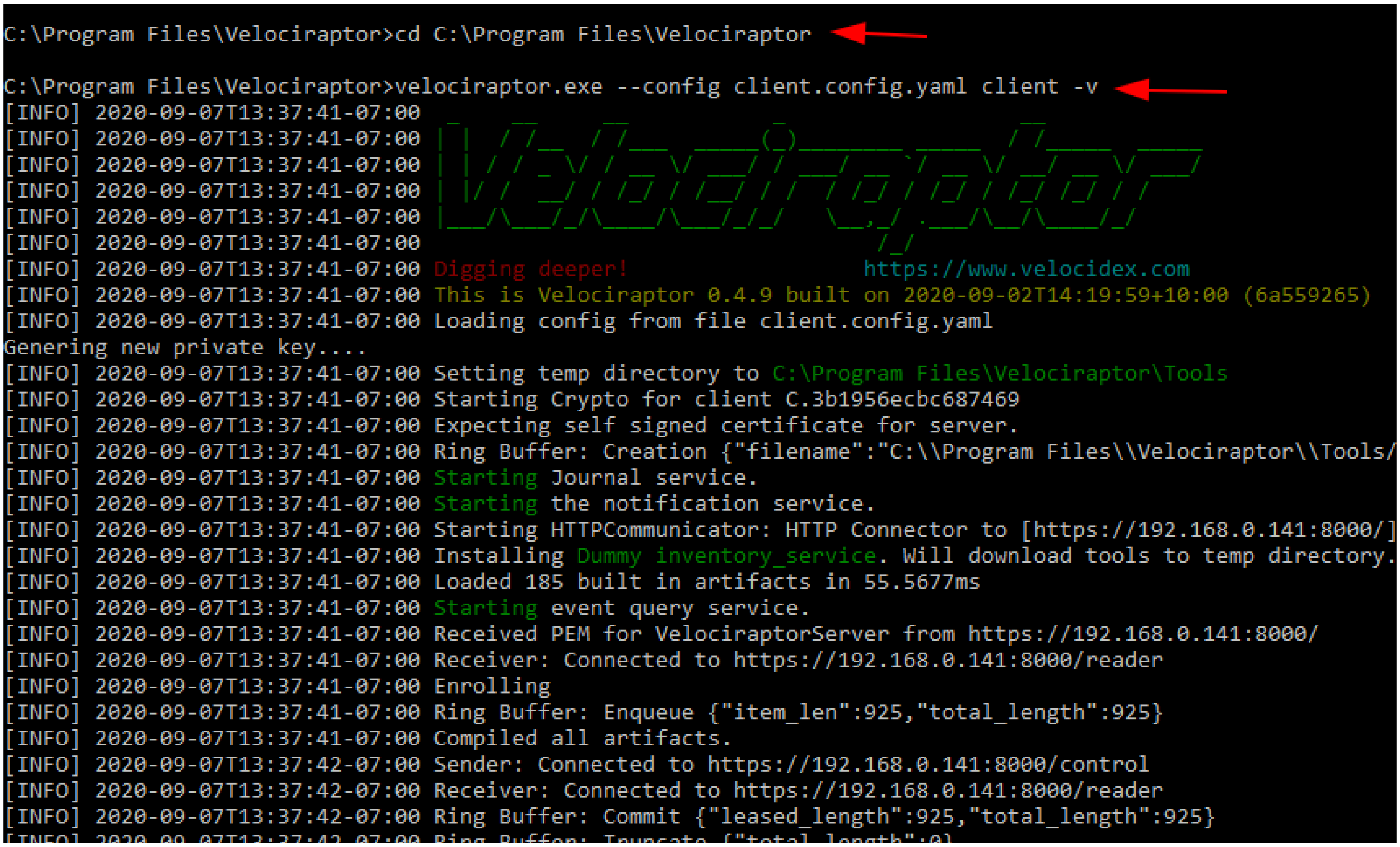
Local Disk (C:) › Program Files › Velociraptor

| Name | Date modified | Type |
|---|---|---|
| Tools | 9/7/2020 1:37 PM | File folder |
| client.config.yaml | 9/7/2020 12:21 PM | YAML File |
| server.config.yaml | 9/7/2020 1:33 PM | YAML File |
| Velociraptor | 9/1/2020 10:34 PM | Application |
| velociraptor.writeback.yaml | 9/7/2020 1:46 PM | YAML File |

Windows Server 2016

Come back to CMD prompt and deploy your client to the Velociraptor server by issuing the following command

```
Velociraptor.exe --config client.config.yaml client -v
```



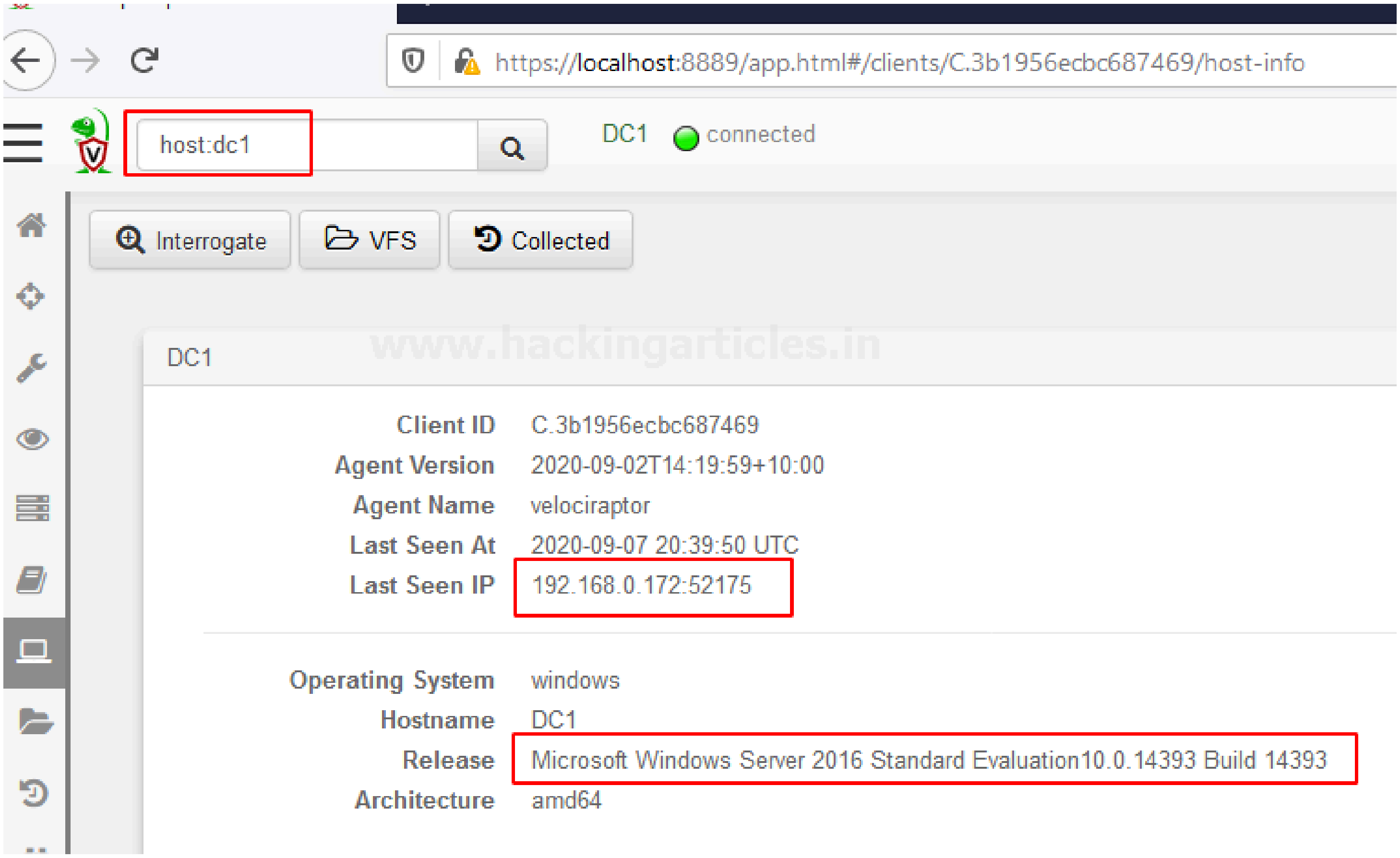
```
C:\Program Files\Velociraptor>cd C:\Program Files\Velociraptor

C:\Program Files\Velociraptor>velociraptor.exe --config client.config.yaml client -v
[INFO] 2020-09-07T13:37:41-07:00
[INFO] 2020-09-07T13:37:41-07:00
[INFO] 2020-09-07T13:37:41-07:00
[INFO] 2020-09-07T13:37:41-07:00
[INFO] 2020-09-07T13:37:41-07:00
[INFO] 2020-09-07T13:37:41-07:00
[INFO] 2020-09-07T13:37:41-07:00 Digging deeper!                  https://www.velocidex.com
[INFO] 2020-09-07T13:37:41-07:00 This is Velociraptor 0.4.9 built on 2020-09-02T14:19:59+10:00 (6a559265)
[INFO] 2020-09-07T13:37:41-07:00 Loading config from file client.config.yaml
Genering new private key....
[INFO] 2020-09-07T13:37:41-07:00 Setting temp directory to C:\Program Files\Velociraptor\Tools
[INFO] 2020-09-07T13:37:41-07:00 Starting Crypto for client C.3b1956ecbc687469
[INFO] 2020-09-07T13:37:41-07:00 Expecting self signed certificate for server.
[INFO] 2020-09-07T13:37:41-07:00 Ring Buffer: Creation {"filename":"C:\\Program Files\\Velociraptor\\Tools/
[INFO] 2020-09-07T13:37:41-07:00 Starting Journal service.
[INFO] 2020-09-07T13:37:41-07:00 Starting the notification service.
[INFO] 2020-09-07T13:37:41-07:00 Starting HTTPCommunicator: HTTP Connector to [https://192.168.0.141:8000/]
[INFO] 2020-09-07T13:37:41-07:00 Installing Dummy inventory_service. Will download tools to temp directory.
[INFO] 2020-09-07T13:37:41-07:00 Loaded 185 built in artifacts in 55.5677ms
[INFO] 2020-09-07T13:37:41-07:00 Starting event query service.
[INFO] 2020-09-07T13:37:41-07:00 Received PEM for VelociraptorServer from https://192.168.0.141:8000/
[INFO] 2020-09-07T13:37:41-07:00 Receiver: Connected to https://192.168.0.141:8000/reader
[INFO] 2020-09-07T13:37:41-07:00 Enrolling
[INFO] 2020-09-07T13:37:41-07:00 Ring Buffer: Enqueue {"item_len":925,"total_length":925}
[INFO] 2020-09-07T13:37:41-07:00 Compiled all artifacts.
[INFO] 2020-09-07T13:37:42-07:00 Sender: Connected to https://192.168.0.141:8000/control
[INFO] 2020-09-07T13:37:42-07:00 Receiver: Connected to https://192.168.0.141:8000/reader
[INFO] 2020-09-07T13:37:42-07:00 Ring Buffer: Commit {"leased_length":925,"total_length":925}
[INFO] 2020-09-07T13:37:42-07:00 Ring Buffer: Truncate {"total_length":0}
```

Nice ? !! You can ensure the integration of the client (Windows) machine with the server inside the Velociraptor Master Server which will generate logs for the client connectivity as shown in the image.

Come back to the Velociraptor server and verify, whether the client is reflected on the GUI interface or not by simply running a query in the search box

```
host:dc1
```

where dc:1 is my client's system name Hmm ? !! you have successfully added the Windows system as a client. Now, We have successfully added both Machines that will be monitored by Velociraptor server.


https://localhost:8889/app.html#/clients/C.3b1956ecbc687469/host-info
host:dc1
DC1
connected
Interrogate
VFS
Collected
DC1
www.hackingarticles.in

Client ID C.3b1956ecbc687469
Agent Version 2020-09-02T14:19:59+10:00
Agent Name velociraptor
Last Seen At 2020-09-07 20:39:50 UTC
Last Seen IP 192.168.0.172:52175

Operating System windows
Hostname DC1
Release Microsoft Windows Server 2016 Standard Evaluation10.0.14393 Build 14393
Architecture amd64

## Forensic Investigation / Threat Hunting

Let's begin some forensics investigation or Threat Hunting

Now if you go back to the homepage you could be able to see your host by searching in the filter box

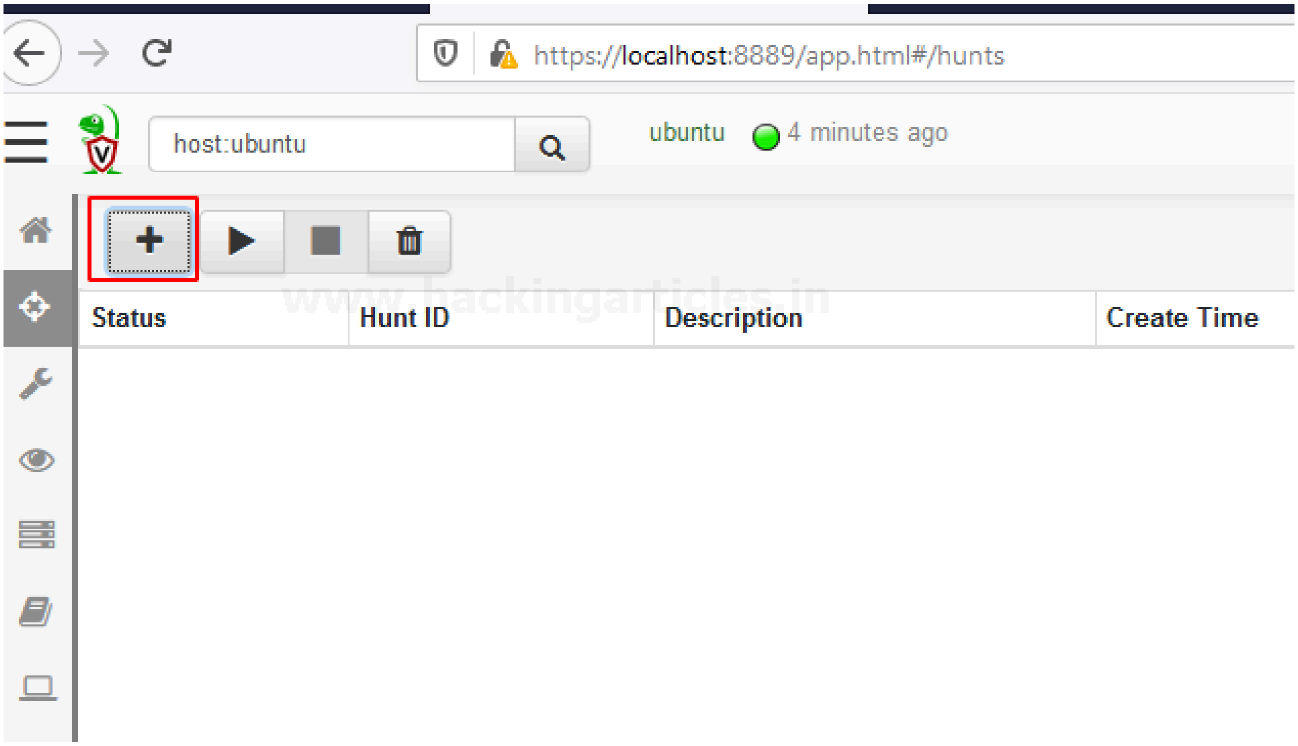
As we have 2 clients connected to velociraptor

Let's start an investigation with Machine-1 (Ubuntu) !!

So now we have Hunt Manager you can easily find it on your Dashboard

Hunt manager allows you to hunt for the specific events that happened to your client and also you can view specific artifacts and server events.

we need to create a hunt with specific artifacts to do this move your cursor to the “+” button and select it as shown below.

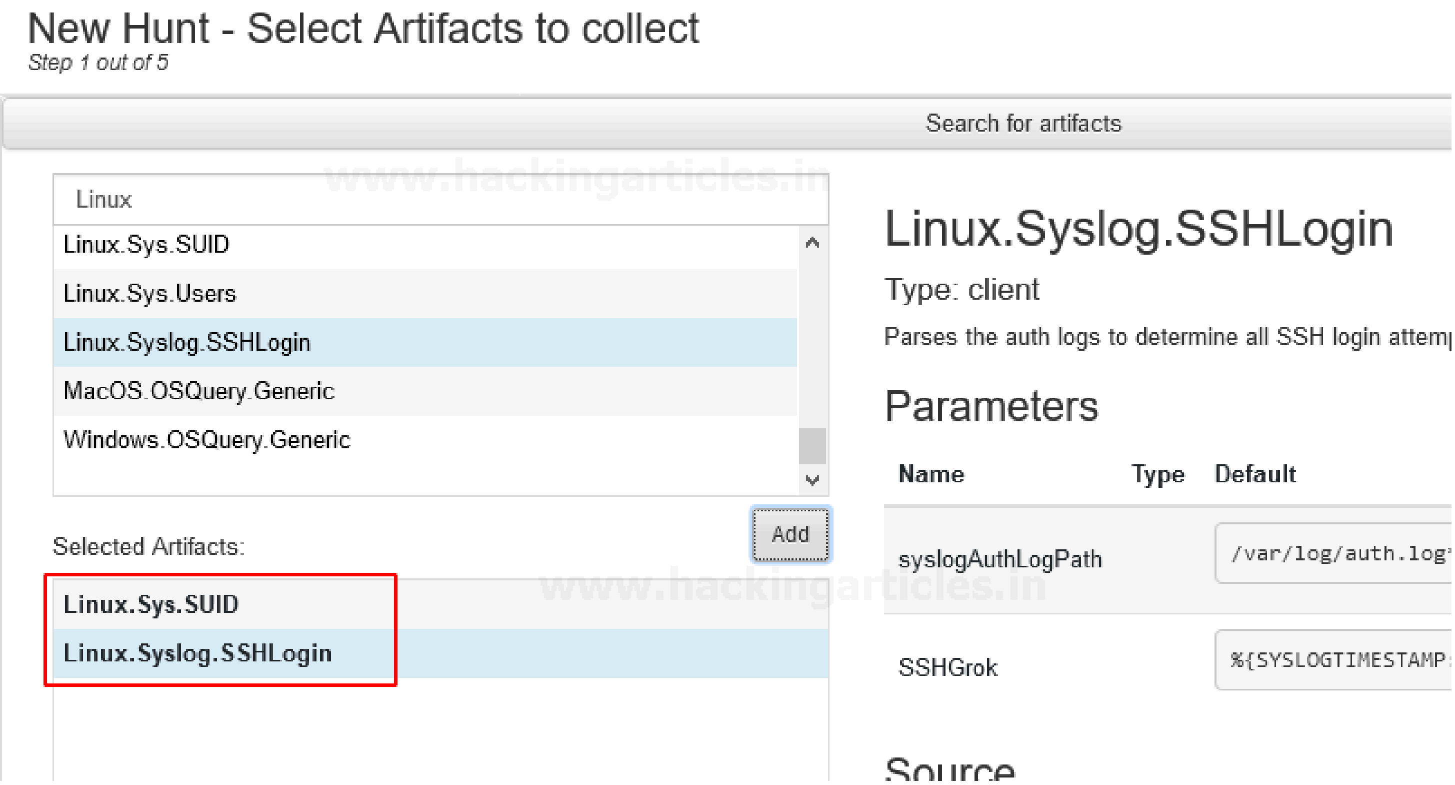


To create a new hunt in the search window start typing Linux then select the artifacts that you want to hunt and add then select **“Next”**,

Some prebuilt Artifacts can be used for forensics of Linux systems Available on Velociraptor as listed below

```
    Linux.Applications.Chrome.Extensions
    Linux.Applications.Chrome.Extensions.Upload
    Linux.Applications.Docker.Info
    Linux.Applications.Docker.Version
    Linux.Debian.AptSources
    Linux.Debian.Packages
    Linux.Mounts
    Linux.OSQuery.Generic
    Linux.Proc.Arp
    Linux.Proc.Modules
    Linux.Search.FileFinder
    Linux.Ssh.AuthorizedKeys
    Linux.Ssh.KnownHosts
    Linux.Ssh.PrivateKeys
    Linux.Sys.ACPITables
    Linux.Sys.BashShell
    Linux.Sys.CPUTime
    Linux.Sys.Crontab
    Linux.Sys.LastUserLogin
    Linux.Sys.Maps
    Linux.Sys.Pslist
    Linux.Sys.SUID
    Linux.Sys.Users
    Linux.Syslog.SSHLogin
```

In my case, I’m selecting Linux.Sys.SUID, Linux.Syslog.SSHLogin you can select as much you want.



New Hunt - Select Artifacts to collect
Step 1 out of 5

Search for artifacts

Linux

- Linux.Sys.SUID
- Linux.Sys.Users
- Linux.Syslog.SSHLogin
- MacOS.OSQuery.Generic
- Windows.OSQuery.Generic

Add

Selected Artifacts:

- Linux.Sys.SUID
- Linux.Syslog.SSHLogin

Linux.Syslog.SSHLogin

Type: client

Parses the auth logs to determine all SSH login attem

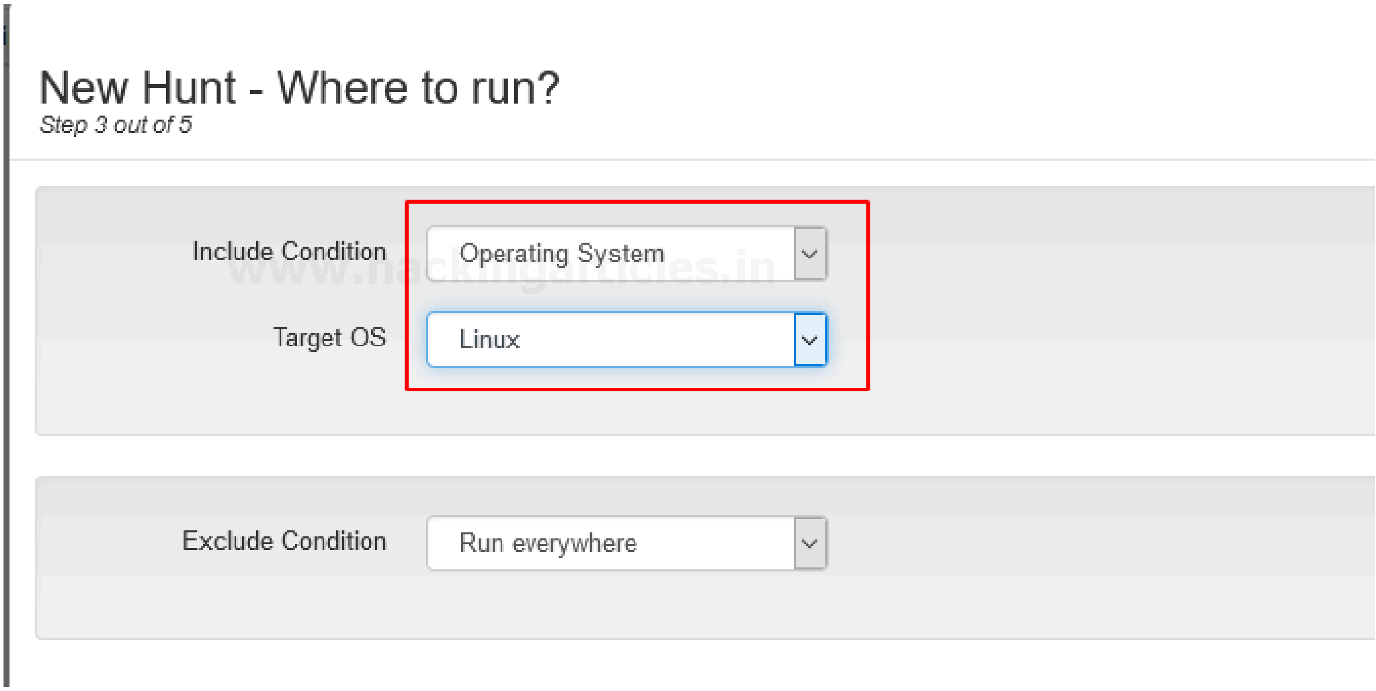
Parameters

| Name | Type | Default |
|---|---|---|
| syslogAuthLogPath | | /var/log/auth.log* |
| SSHGrok | | %{SYSLOGTIMESTAMP: |

Source

After selecting next, it will redirect to next prompt where you need to give Hunt Description and then select **"Next"**

Hunt conditions should be in "**operating system"** select it in the drop-down menu of Include Condition then select Target OS **"Linux"** and then hit **"Next"**



New Hunt - Where to run?
Step 3 out of 5

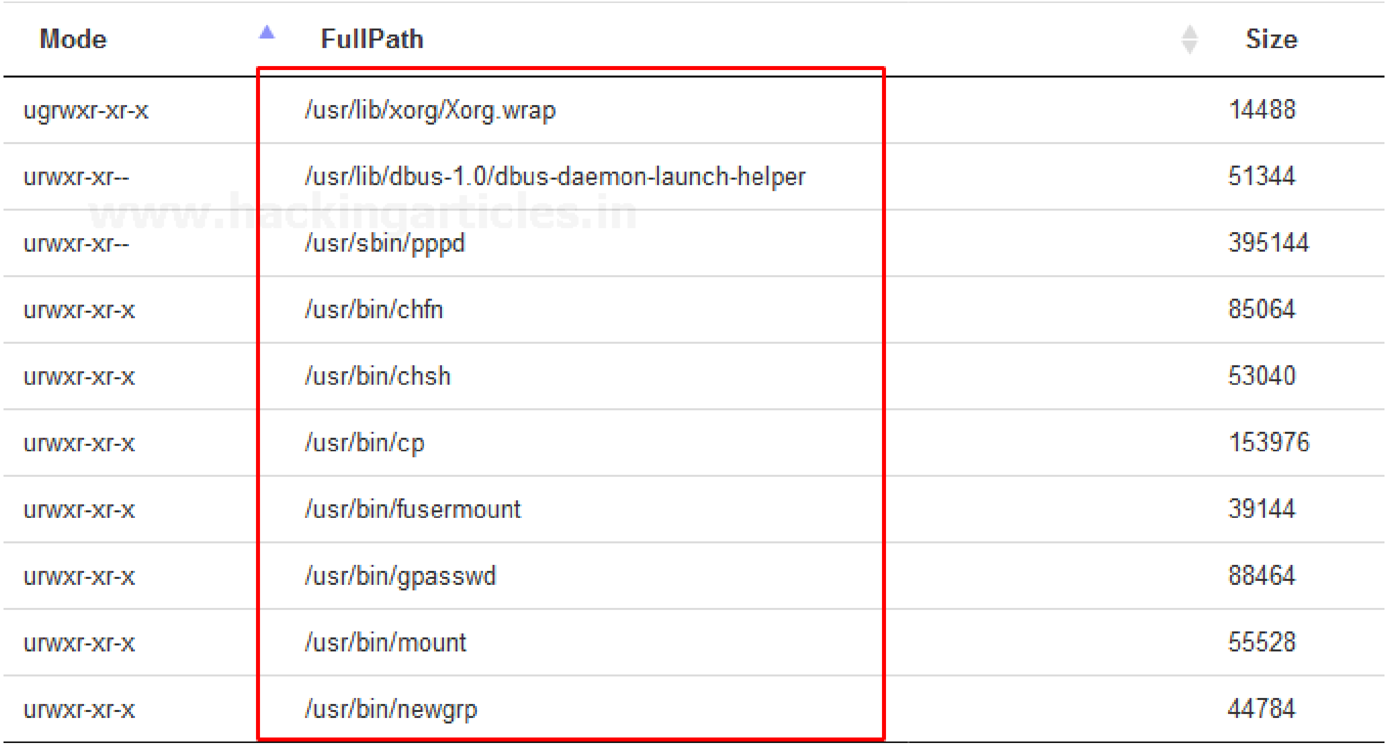
Include Condition: Operating System

Target OS: Linux

Exclude Condition: Run everywhere

At the next screen, you have your hunt Description or Artifact review, now a select the option **"Create Hunt"**

Now we have created a new Hunt Named Linux Hunt it reflects on our Hunts panel And We would like to run this hunt by pressing the play button to see what's next in the result…



| Mode | FullPath | Size |
|---|---|---|
| ugrwxr-xr-x | /usr/lib/xorg/Xorg.wrap | 14488 |
| urwxr-xr-- | /usr/lib/dbus-1.0/dbus-daemon-launch-helper | 51344 |
| urwxr-xr-- | /usr/sbin/pppd | 395144 |
| urwxr-xr-x | /usr/bin/chfn | 85064 |
| urwxr-xr-x | /usr/bin/chsh | 53040 |
| urwxr-xr-x | /usr/bin/cp | 153976 |
| urwxr-xr-x | /usr/bin/fusermount | 39144 |
| urwxr-xr-x | /usr/bin/gpasswd | 88464 |
| urwxr-xr-x | /usr/bin/mount | 55528 |
| urwxr-xr-x | /usr/bin/newgrp | 44784 |

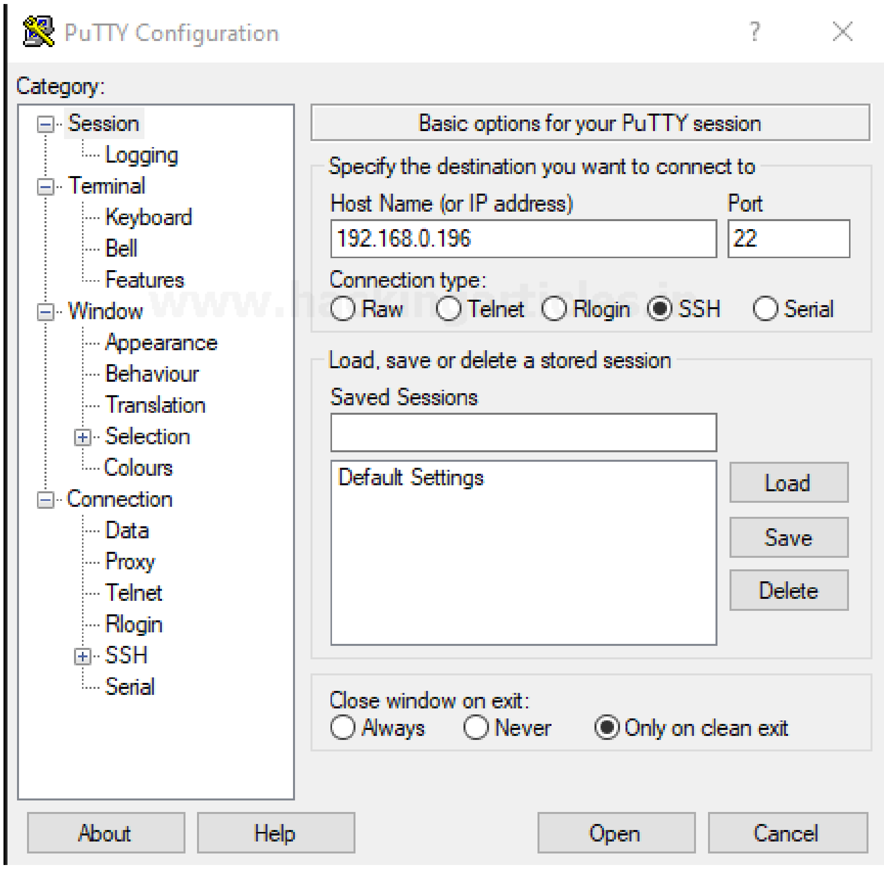
Wow ? !! As we can see here is the list of Linux system SUID

Wait this is not enough… Let's Dig it more Deeper

Let's take SSH of Linux client from Putty and perform a Brute-force attack from Attacker machine Kali Linux

Exited? let's do it ? !!

open Putty and enter the IP and port no. of the client and open the session



Specify the destination you want to connect to

Host Name (or IP address): 192.168.0.196    Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

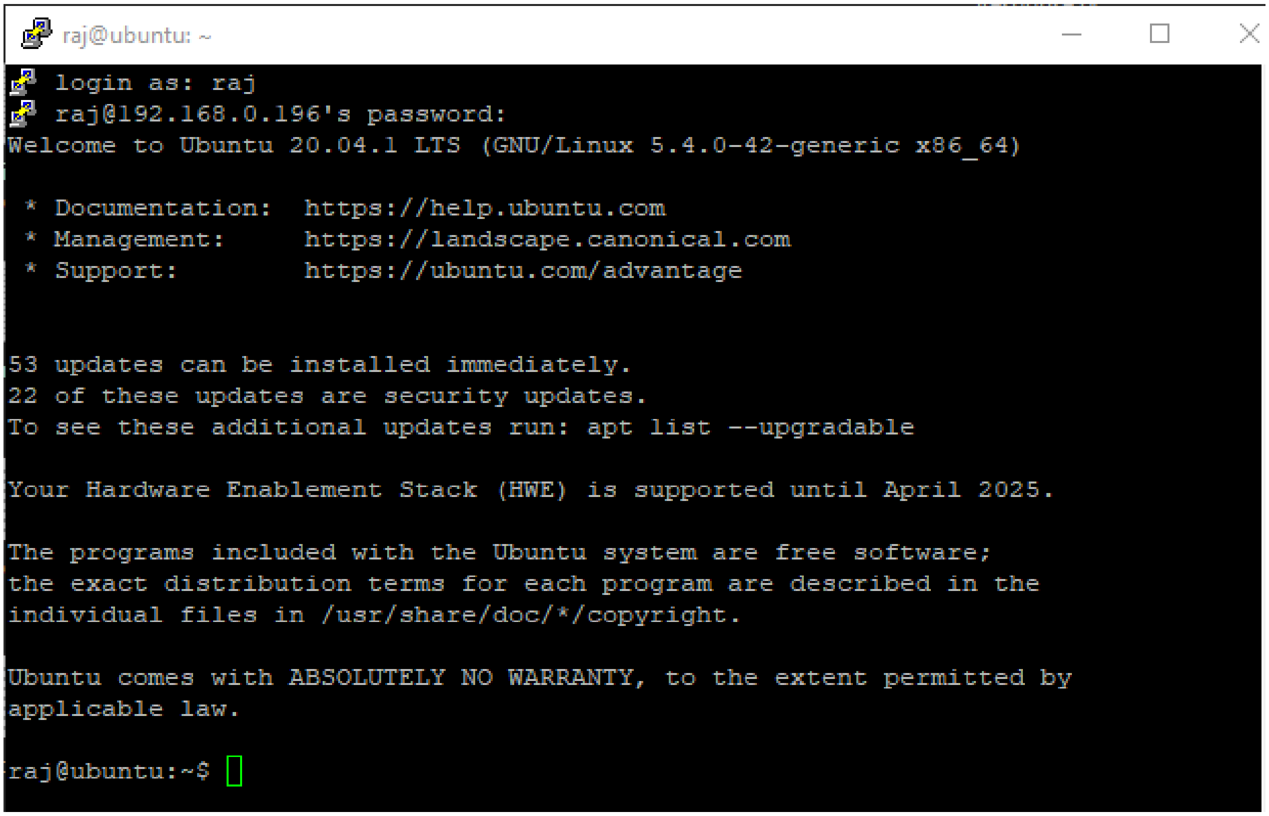
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

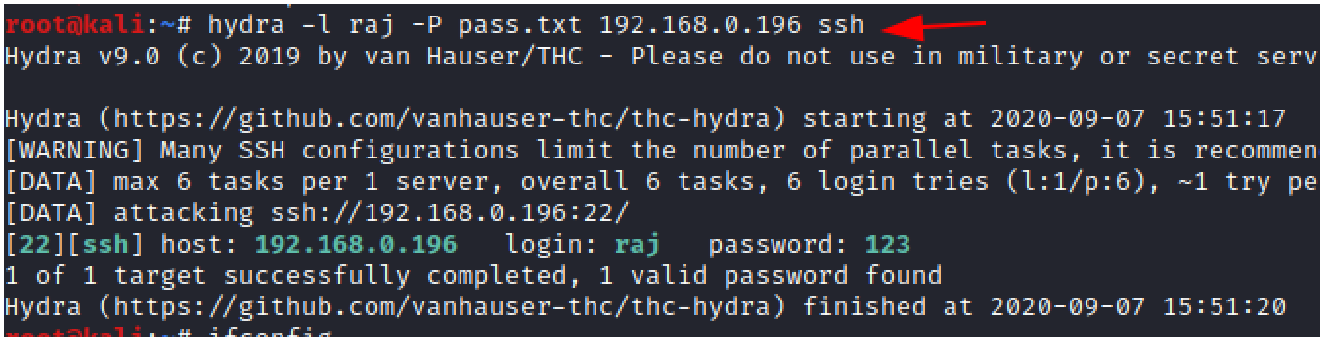
After the opening of the SSH shell login to the Client machine



```
login as: raj
raj@192.168.0.196's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


53 updates can be installed immediately.
22 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

raj@ubuntu:~$
```

Nice !! we have successfully logged in to the client machine  Let’s perform a Brute-force attack to check is Velociraptor able to detect the attack or not. Fire up the Attacker machine Kali Linux and run the following command

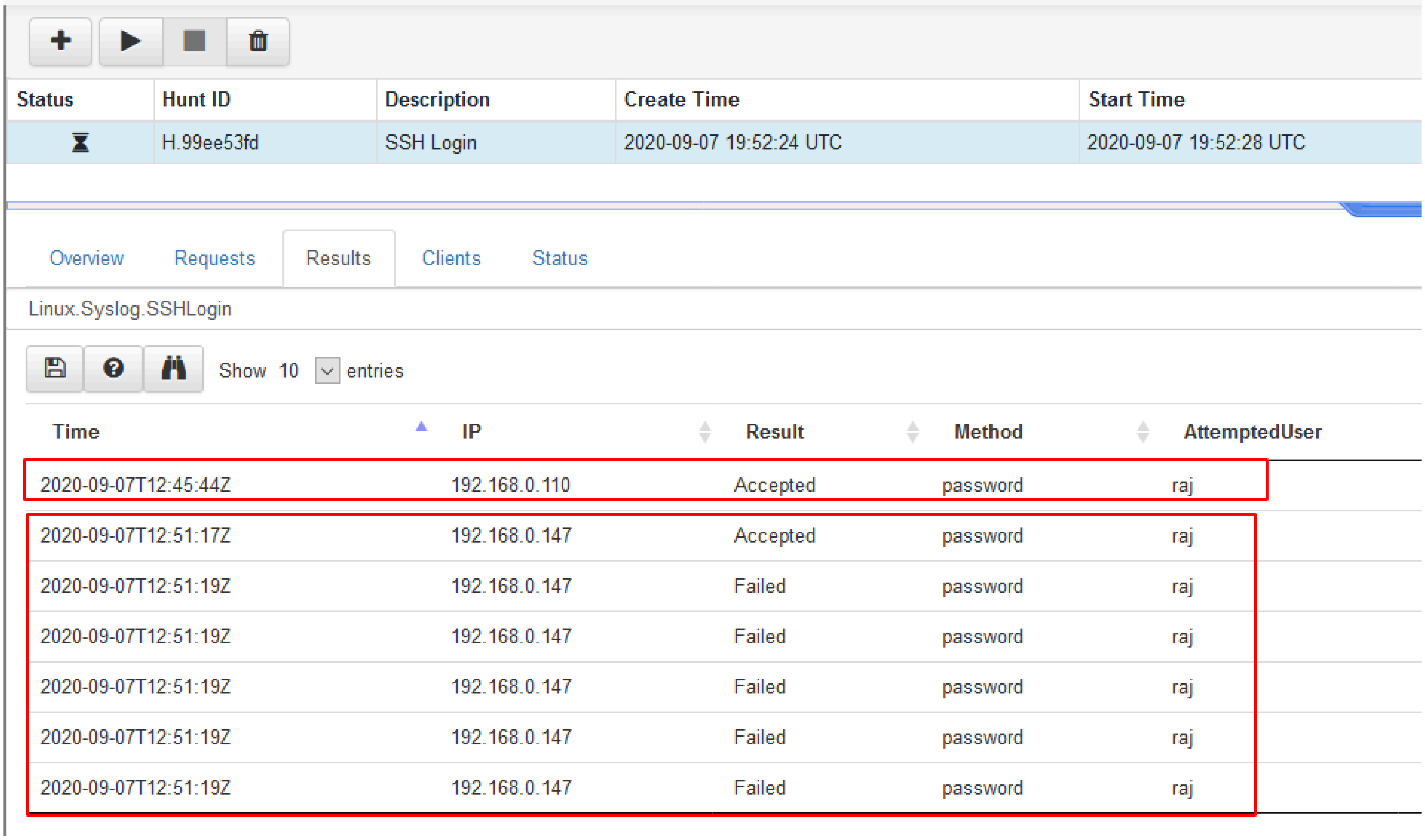
```
hydra -l raj -P pass.txt 192.168.0.196 ssh
```



```
root@kali:~# hydra -l raj -P pass.txt 192.168.0.196 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret serv

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-09-07 15:51:17
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommen
[DATA] max 6 tasks per 1 server, overall 6 tasks, 6 login tries (l:1/p:6), ~1 try pe
[DATA] attacking ssh://192.168.0.196:22/
[22][ssh] host: 192.168.0.196   login: raj   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-09-07 15:51:20
root@kali:~# ifconfig
```

Let’s check what happened to the GUI interface of Velociraptor.

Hold tight !!

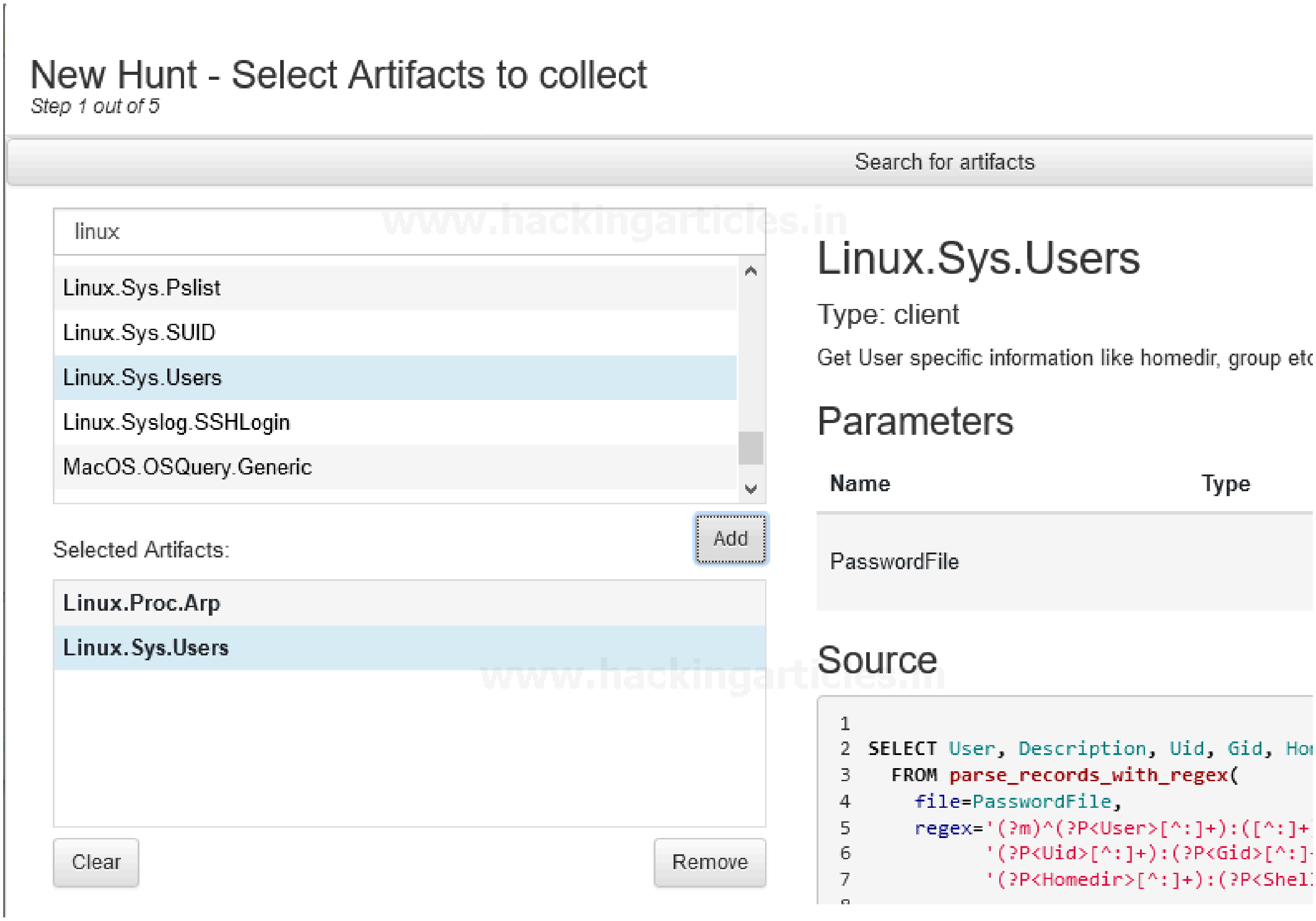


| Status | Hunt ID | Description | Create Time | Start Time |
|---|---|---|---|---|
| ⧗ | H.99ee53fd | SSH Login | 2020-09-07 19:52:24 UTC | 2020-09-07 19:52:28 UTC |

Linux.Syslog.SSHLogin

| Time | IP | Result | Method | AttemptedUser |
|---|---|---|---|---|
| 2020-09-07T12:45:44Z | 192.168.0.110 | Accepted | password | raj |
| 2020-09-07T12:51:17Z | 192.168.0.147 | Accepted | password | raj |
| 2020-09-07T12:51:19Z | 192.168.0.147 | Failed | password | raj |
| 2020-09-07T12:51:19Z | 192.168.0.147 | Failed | password | raj |
| 2020-09-07T12:51:19Z | 192.168.0.147 | Failed | password | raj |
| 2020-09-07T12:51:19Z | 192.168.0.147 | Failed | password | raj |
| 2020-09-07T12:51:19Z | 192.168.0.147 | Failed | password | raj |

wow !! As we can see it detects and shows 2 successful logins of different machines and 5 failed login attempts just because of Brute force Attack. Let’s check some more artifacts that show the Arp requests and Linux system users.



## New Hunt - Select Artifacts to collect

*Step 1 out of 5*

Search for artifacts

linux

- Linux.Sys.Pslist
- Linux.Sys.SUID
- Linux.Sys.Users
- Linux.Syslog.SSHLogin
- MacOS.OSQuery.Generic

Selected Artifacts:

- Linux.Proc.Arp
- Linux.Sys.Users

### Linux.Sys.Users

Type: client

Get User specific information like homedir, group etc

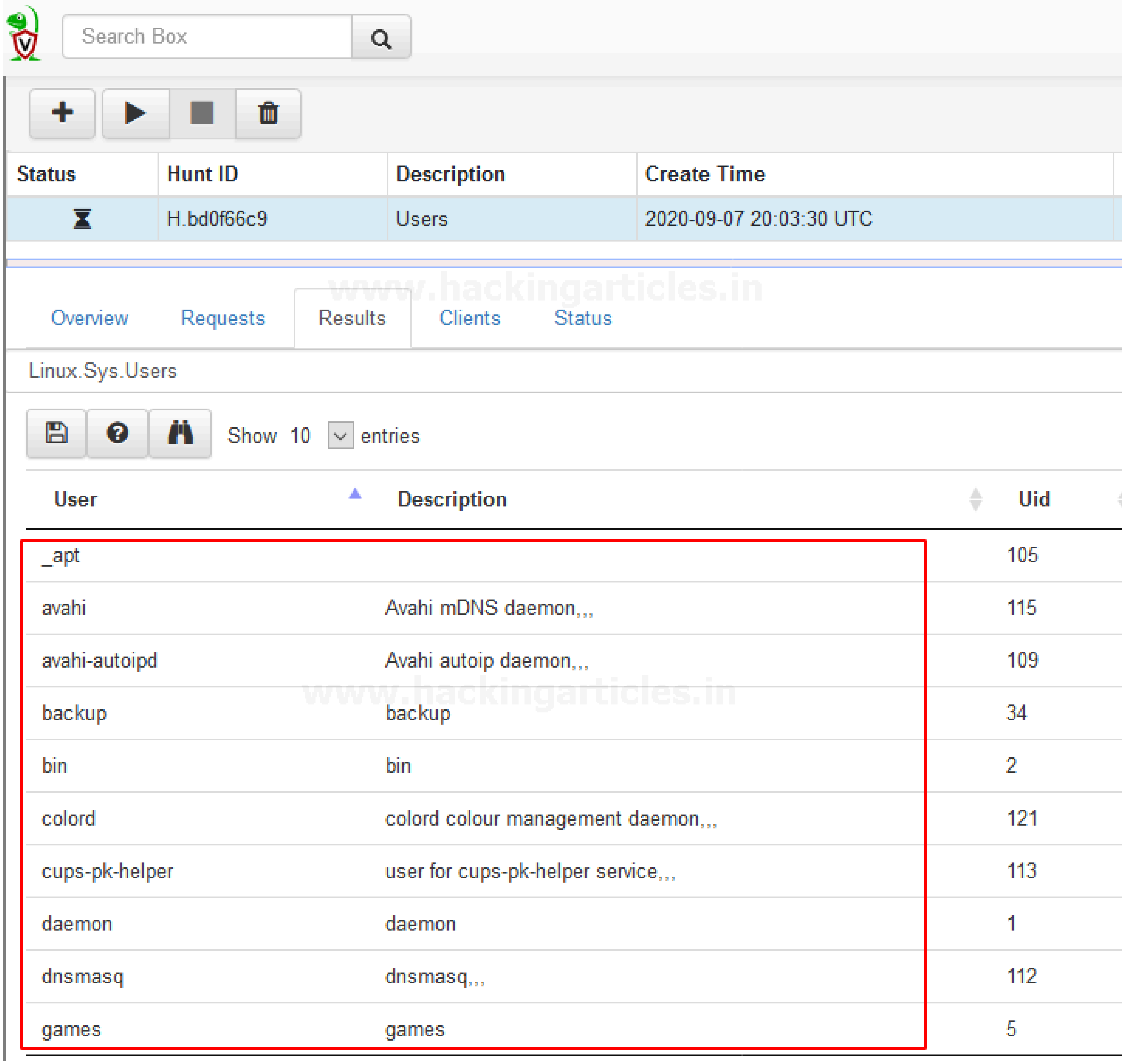
### Parameters

| Name | Type |
|---|---|
| PasswordFile | |

### Source

```

SELECT User, Description, Uid, Gid, Ho
  FROM parse_records_with_regex(
    file=PasswordFile,
    regex='(?m)^(?P<User>[^:]+):([^:]+
          '(?P<Uid>[^:]+):(?P<Gid>[^:]
          '(?P<Homedir>[^:]+):(?P<Shel
```

After creating the Hunt go to the result section and check what happens there…



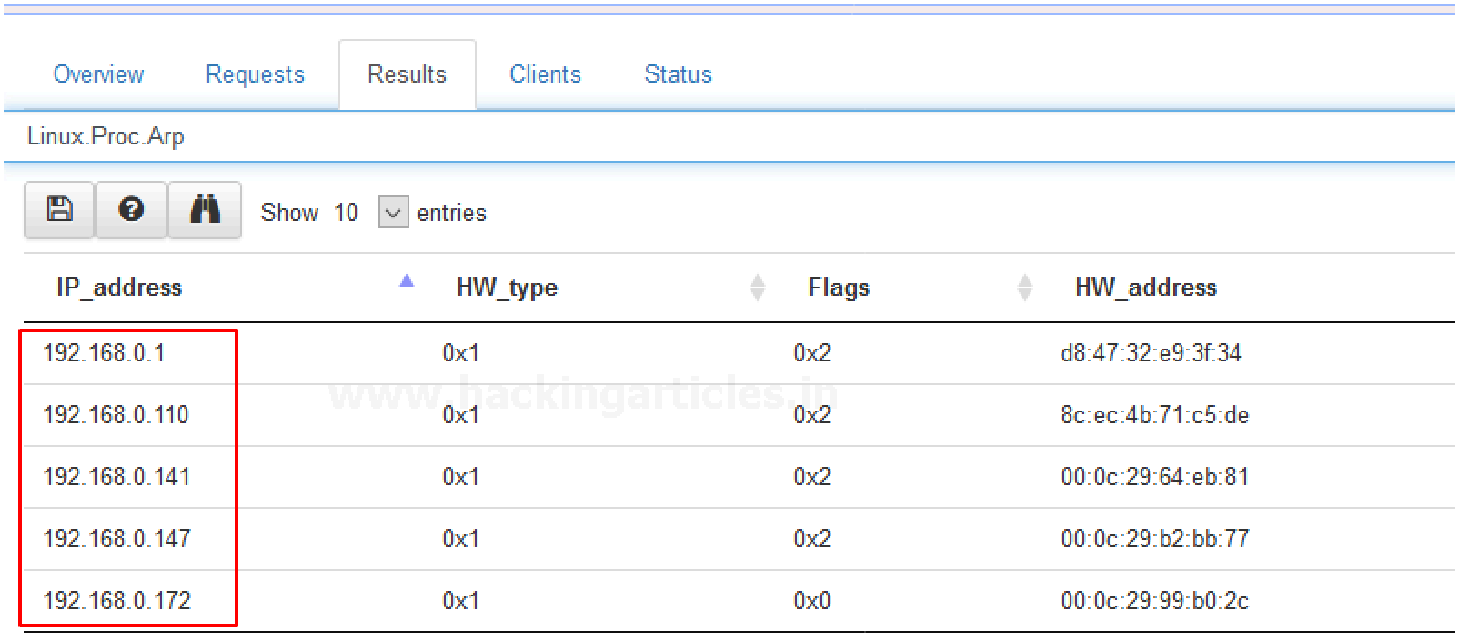
| Status | Hunt ID | Description | Create Time |
|---|---|---|---|
| ⧗ | H.bd0f66c9 | Users | 2020-09-07 20:03:30 UTC |

Linux.Sys.Users

| User | Description | Uid |
|---|---|---|
| _apt | | 105 |
| avahi | Avahi mDNS daemon,,, | 115 |
| avahi-autoipd | Avahi autoip daemon,,, | 109 |
| backup | backup | 34 |
| bin | bin | 2 |
| colord | colord colour management daemon,,, | 121 |
| cups-pk-helper | user for cups-pk-helper service,,, | 113 |
| daemon | daemon | 1 |
| dnsmasq | dnsmasq,,, | 112 |
| games | games | 5 |

As we can see it shows All Linux system users with their “UID” and a small description of the role of users.

Let’s check the “ARP” requests on the client



| IP_address | HW_type | Flags | HW_address |
|---|---|---|---|
| 192.168.0.1 | 0x1 | 0x2 | d8:47:32:e9:3f:34 |
| 192.168.0.110 | 0x1 | 0x2 | 8c:ec:4b:71:c5:de |
| 192.168.0.141 | 0x1 | 0x2 | 00:0c:29:64:eb:81 |
| 192.168.0.147 | 0x1 | 0x2 | 00:0c:29:b2:bb:77 |
| 192.168.0.172 | 0x1 | 0x0 | 00:0c:29:99:b0:2c |

Wow ? !! it contains quite enough useful information.

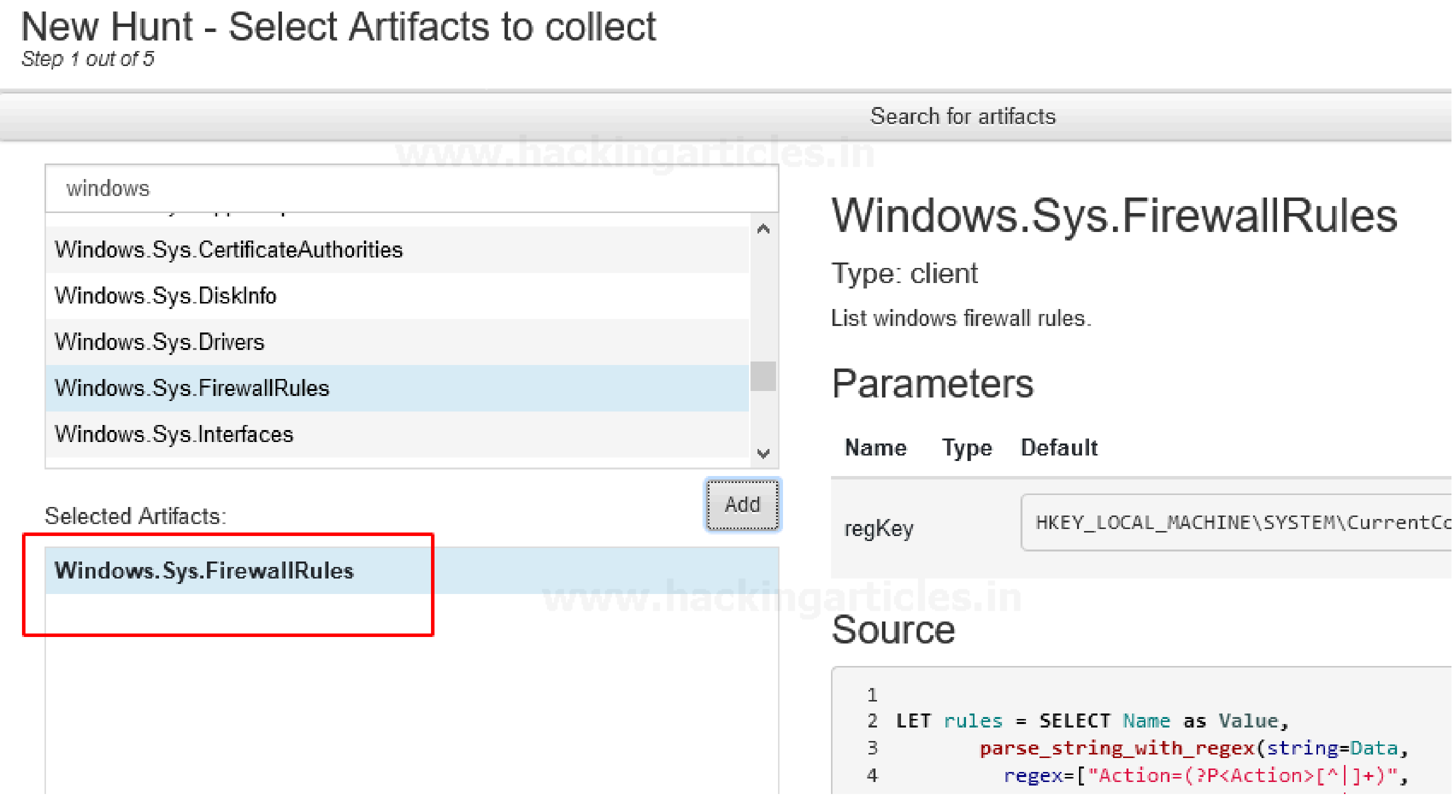
Based on these artifacts you can investigate the scene or your client by creating Hunt as per your requirements also you can create your artifacts if you have good knowledge of VQL.

Let’s investigate our Windows client !! ?

Form Dashboard set the host to windows or whatever the client’s computer name.

Then create a Hunt

I’m going to use Artifact “**Windows.Sys.FirewallRules**”



New Hunt - Select Artifacts to collect
Step 1 out of 5

Search for artifacts

windows

- Windows.Sys.CertificateAuthorities
- Windows.Sys.DiskInfo
- Windows.Sys.Drivers
- Windows.Sys.FirewallRules
- Windows.Sys.Interfaces

Selected Artifacts:

**Windows.Sys.FirewallRules**

Windows.Sys.FirewallRules

Type: client

List windows firewall rules.

Parameters

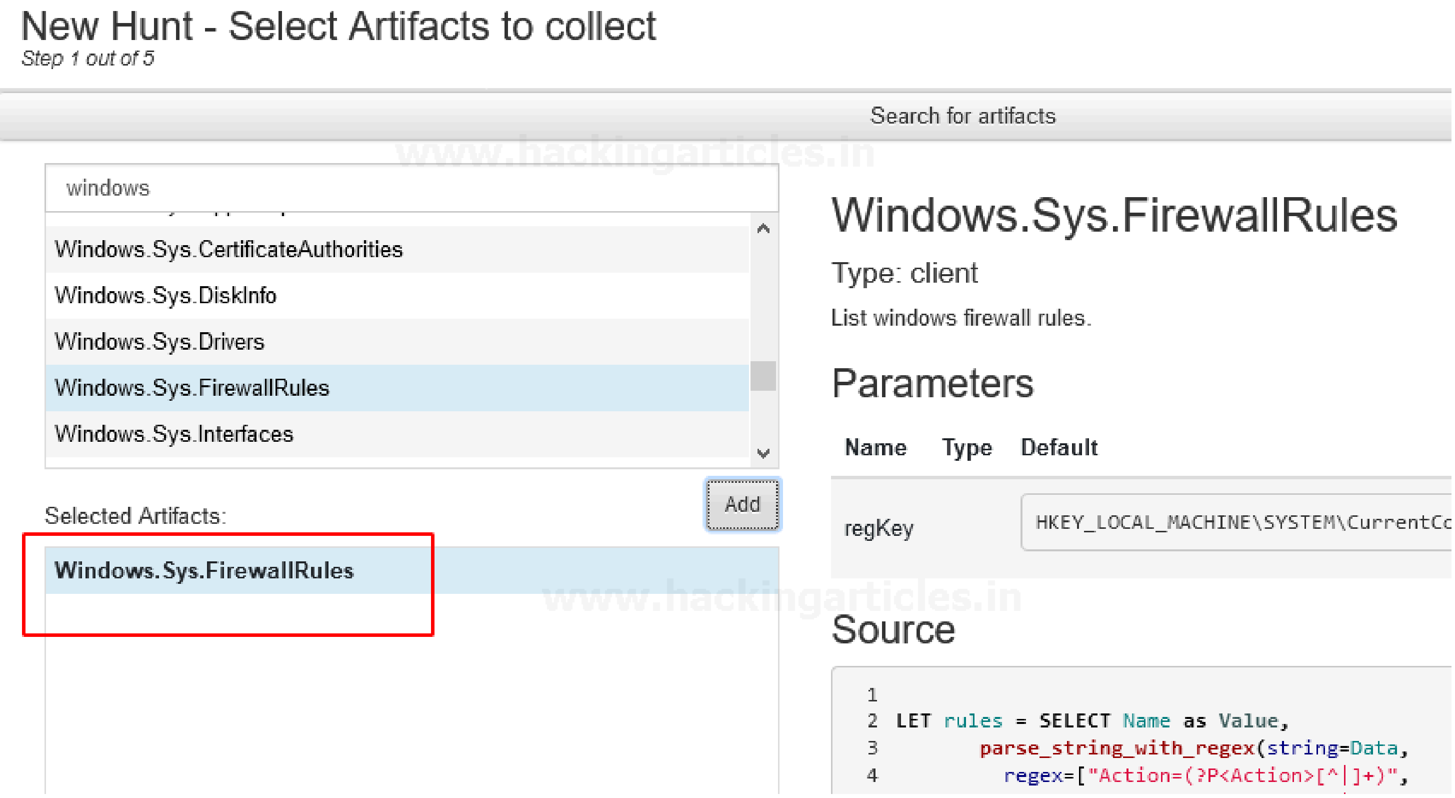
| Name | Type | Default |
|---|---|---|
| regKey | | HKEY_LOCAL_MACHINE\SYSTEM\CurrentCo |

Source

```

LET rules = SELECT Name as Value,
      parse_string_with_regex(string=Data,
        regex=["Action=(?P<Action>[^|]+)",
```

After selecting next it redirects you to next prompt when you need to Hunt Description and then select **“Next”**

Hunt conditions should be in “**operating system”** select it in the drop-down menu of Include Condition then select Target OS **“Windows”** and then hit **“Next”**



New Hunt - Select Artifacts to collect
Step 1 out of 5

Search for artifacts

windows

- Windows.Sys.CertificateAuthorities
- Windows.Sys.DiskInfo
- Windows.Sys.Drivers
- Windows.Sys.FirewallRules
- Windows.Sys.Interfaces

Selected Artifacts:

**Windows.Sys.FirewallRules**

Windows.Sys.FirewallRules

Type: client

List windows firewall rules.

Parameters

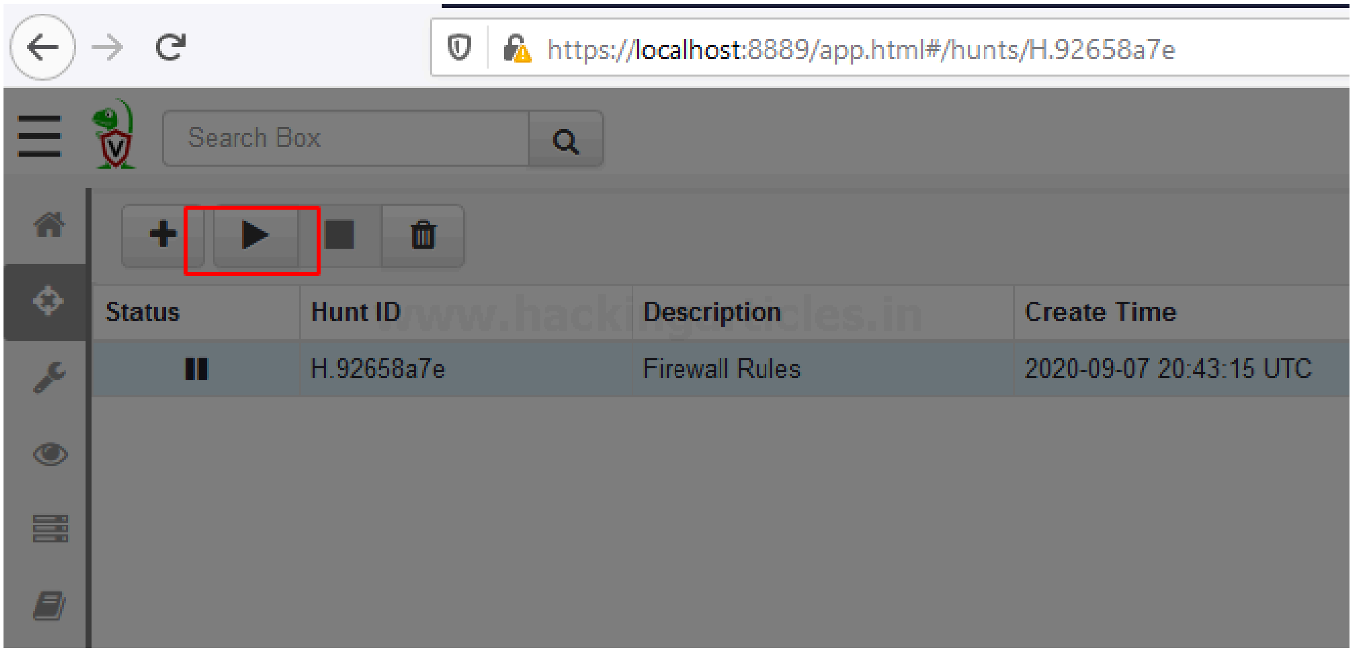
| Name | Type | Default |
|---|---|---|
| regKey | | HKEY_LOCAL_MACHINE\SYSTEM\CurrentCo |

Source

```

LET rules = SELECT Name as Value,
      parse_string_with_regex(string=Data,
        regex=["Action=(?P<Action>[^|]+)",
```

Now we have created a new Hunt Named Windows Hunt it reflects your Hunts panel And We would like to run this hunt by pressing the play button to see what's next in the result…

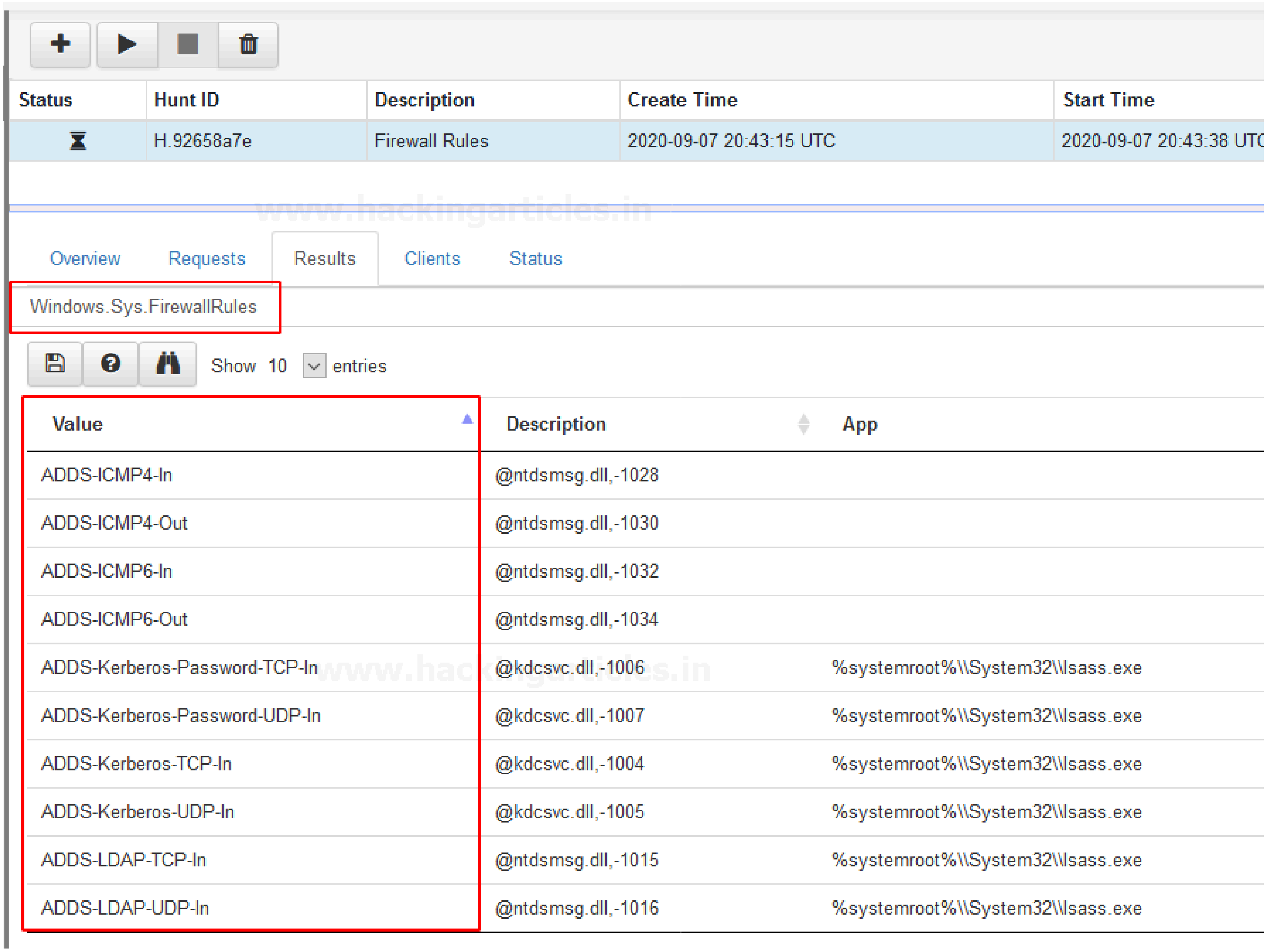
Let's check the result. Hold tight !!



https://localhost:8889/app.html#/hunts/H.92658a7e

| Status | Hunt ID | Description | Create Time |
|---|---|---|---|
| ❚❚ | H.92658a7e | Firewall Rules | 2020-09-07 20:43:15 UTC |

Let's check the result. Hold tight !!



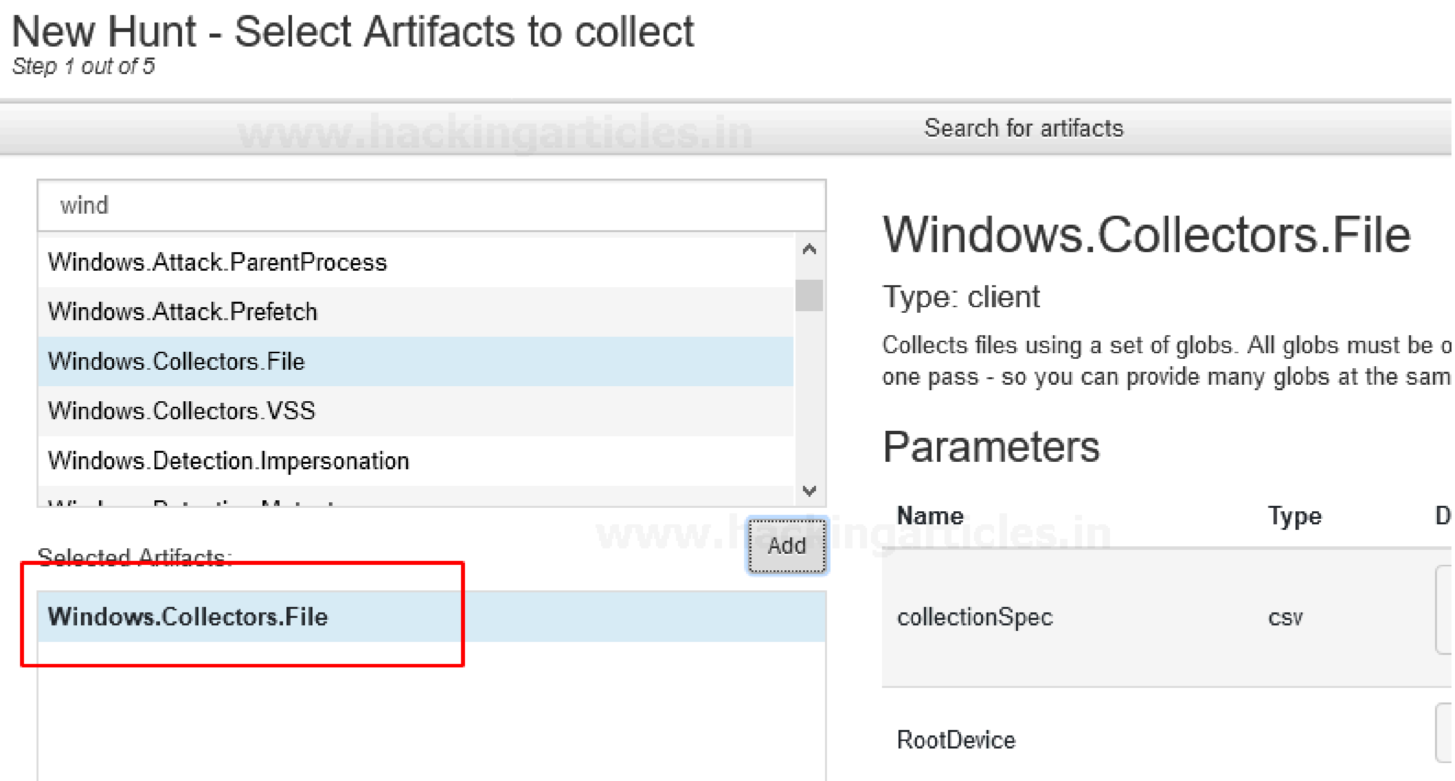
| Status | Hunt ID | Description | Create Time | Start Time |
|---|---|---|---|---|
| ⧗ | H.92658a7e | Firewall Rules | 2020-09-07 20:43:15 UTC | 2020-09-07 20:43:38 UTC |

| Value | Description | App |
|---|---|---|
| ADDS-ICMP4-In | @ntdsmsg.dll,-1028 | |
| ADDS-ICMP4-Out | @ntdsmsg.dll,-1030 | |
| ADDS-ICMP6-In | @ntdsmsg.dll,-1032 | |
| ADDS-ICMP6-Out | @ntdsmsg.dll,-1034 | |
| ADDS-Kerberos-Password-TCP-In | @kdcsvc.dll,-1006 | %systemroot%\\System32\\lsass.exe |
| ADDS-Kerberos-Password-UDP-In | @kdcsvc.dll,-1007 | %systemroot%\\System32\\lsass.exe |
| ADDS-Kerberos-TCP-In | @kdcsvc.dll,-1004 | %systemroot%\\System32\\lsass.exe |
| ADDS-Kerberos-UDP-In | @kdcsvc.dll,-1005 | %systemroot%\\System32\\lsass.exe |
| ADDS-LDAP-TCP-In | @ntdsmsg.dll,-1015 | %systemroot%\\System32\\lsass.exe |
| ADDS-LDAP-UDP-In | @ntdsmsg.dll,-1016 | %systemroot%\\System32\\lsass.exe |

Nice !! Here is the list of implemented Firewall Rule on the Client’s machine.

Let’s check out some more artifacts to dig it deeper.

Create a new hunt and add many artifacts as you want. Here I’m going to use “Windows.Collectors.File”



New Hunt - Select Artifacts to collect
Step 1 out of 5

Search for artifacts

wind

- Windows.Attack.ParentProcess
- Windows.Attack.Prefetch
- Windows.Collectors.File
- Windows.Collectors.VSS
- Windows.Detection.Impersonation

Add

Selected Artifacts:

Windows.Collectors.File

Windows.Collectors.File

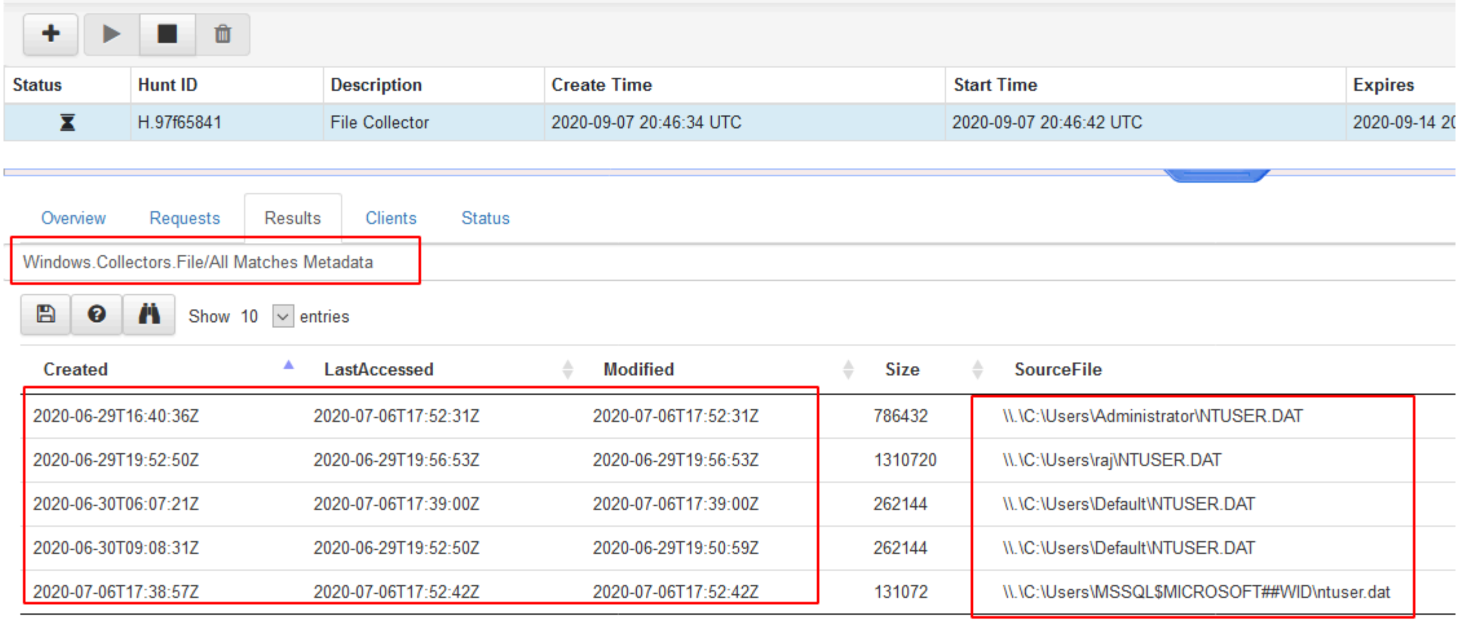
Type: client

Collects files using a set of globs. All globs must be o
one pass - so you can provide many globs at the sam

Parameters

| Name | Type | D |
|---|---|---|
| collectionSpec | csv | |
| RootDevice | | |

Let’s check what comes in result…..



| Status | Hunt ID | Description | Create Time | Start Time | Expires |
|---|---|---|---|---|---|
| ⧗ | H.97f65841 | File Collector | 2020-09-07 20:46:34 UTC | 2020-09-07 20:46:42 UTC | 2020-09-14 2( |

Windows.Collectors.File/All Matches Metadata

Show 10 entries

| Created | LastAccessed | Modified | Size | SourceFile |
|---|---|---|---|---|
| 2020-06-29T16:40:36Z | 2020-07-06T17:52:31Z | 2020-07-06T17:52:31Z | 786432 | \\.\C:\Users\Administrator\NTUSER.DAT |
| 2020-06-29T19:52:50Z | 2020-06-29T19:56:53Z | 2020-06-29T19:56:53Z | 1310720 | \\.\C:\Users\raj\NTUSER.DAT |
| 2020-06-30T06:07:21Z | 2020-07-06T17:39:00Z | 2020-07-06T17:39:00Z | 262144 | \\.\C:\Users\Default\NTUSER.DAT |
| 2020-06-30T09:08:31Z | 2020-06-29T19:52:50Z | 2020-06-29T19:50:59Z | 262144 | \\.\C:\Users\Default\NTUSER.DAT |
| 2020-07-06T17:38:57Z | 2020-07-06T17:52:42Z | 2020-07-06T17:52:42Z | 131072 | \\.\C:\Users\MSSQL$MICROSOFT##WID\ntuser.dat |

Wow!! As we can see it listed the All matches Metadata of windows.collectors

Similarly, you can Dig it much Deeper by adding as many artifacts as you need

Hang tight this is not enough!

More will be discussed in part3.