# Lab 03 – Network Acquisition

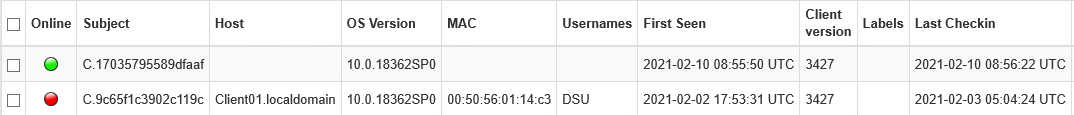
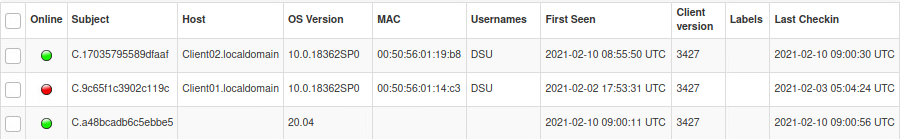
Our task in this lab is to demonstrate proficiency in capturing evidence using an open source incident response framework.

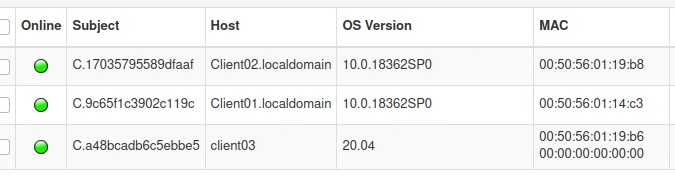
We will be using the vApp in the IA Lab labeled <username>\_CSC388\_Zwach\_GRR. You will find several VM’s there to simulate an environment where we can capture client machine data using GRR. Additionally, the sample images from the text are included on the *ClientForensics* VM; they're not required for this lab, but you may use them.

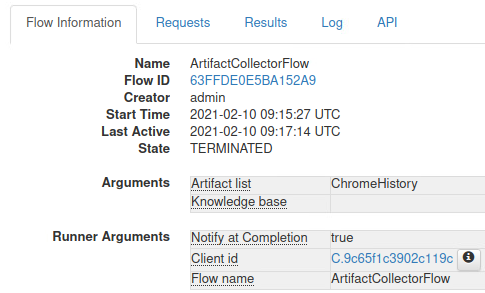
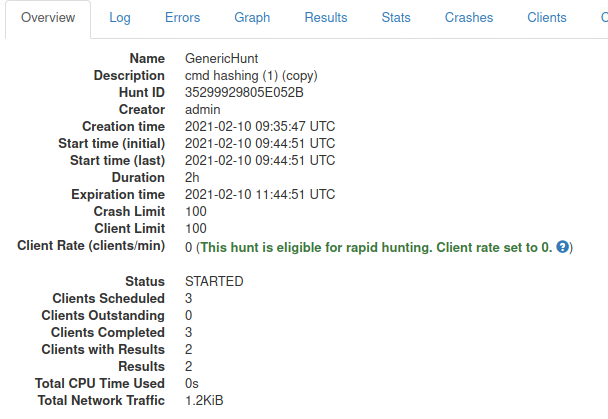
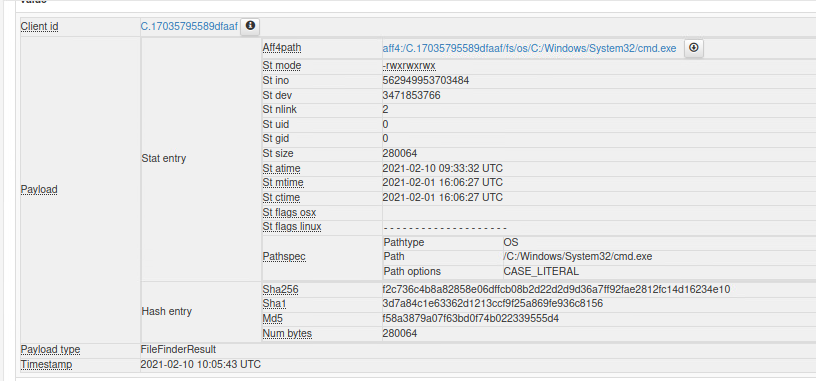
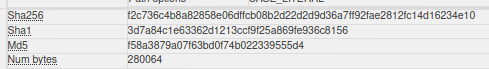
It will take a few minutes for GRR to startup so start the vApp, grab a coffee or water, and return in 5 minutes or so. If you have a client that hasn't checked in for a while, open that VM console and interact with the user interface a bit.

GRR Username: **admin**GRR Password: **password**GRR URL: <http://grrserver:8000> OR <http://192.168.1.5:8000>

## Lab Steps

1. On the *Client02* machine:
   1. Login with the DSU account (**Password1!**)
   2. Install the GRR client for Windows.
   3. Provide a screenshot of the client checking in
2. On the *Client03* machine:
   1. Login with the DSU account (**Password1!**)
   2. Install the GRR client for Debian/Ubuntu (.deb)
   3. Provide a screenshot of the client checking i

I went in to restart the watcher service on client 1 at this point

1. Using the *GRRServer* system:
   1. Navigate to the GRR URL and review your clients
   2. Run a flow on *Client01* to get web history from Chrome (Start New Flows->Collectors->ArtifactCollectorFlow->Chrome History)
   3. Provide a screenshot showing the flow completed
2. Again, from the *GRRServer* system:
   1. run a hunt to get the hash of cmd.exe on the Windows systems (Set the duration to 1h; it will take a while)
      1. Hint: %%environ\_windir%%\System32\cmd.exe
   2. Provide a screenshot showing the hunt completed and the results (hashes)

## Writing Prompts

1. Based on this brief experience with GRR, what are some pros and cons that you've encountered?

Pros:   
 - pretty intuitive interface  
 - free  
 - lots of prepackaged search forms/terms  
 - client installation is cake

Cons:  
 - remote login over http is it possible to set up https?  
 - client01 was crashed initially, which may be a flag for reliability concern. On the upside, we can identify which clients may be down/failing by their connectivity status.

1. Would you feel comfortable using GRR in an environment you control or have input on? Why or why not?

I like the tool and the features that it offers, however, since I didn’t configure - and I’m unfamiliar with - the server side of things, I am hesitant to say one way or the other. I would have some accountability concerns. First I would like to be able to create different accounts to access the server, and log use of the system, ideally hooking into active directory. Second I would post it up using HTTPS, instead of http if possible. With those two conditions I would be pretty satisfied with GRR.

## Notes

* OSForensics and the included evidence files on the desktop of the *ClientForensics* VM are **not** required for this lab.
* In a production deployment, you likely would manage the usage of the network forensics tool and its functions, but not its implementation. We practice installing clients here in case you are the person to install a network forensics client.
* Other tools may be more featureful in their collection abilities, but GRR is free which makes it greatly accessible for this course.
* To list all GRR clients, login and then click in the search box and press enter.