

GitHub - MythicAgents/Apollo: A .NET Framework 4.0 Windows Agent

By its-a-feature

Archived: 2026-04-05 21:41:25 UTC



Apollo is a Windows agent written in C# using the 4.0 .NET Framework designed to be used in SpecterOps training offerings.

Installation

To install Apollo, you'll need Mythic installed on a remote computer. You can find installation instructions for Mythic at the [Mythic project page](#).

From the Mythic install directory, use the following command to install Apollo as the **root** user:

```
./mythic-cli install github https://github.com/MythicAgents/Apollo.git
```

From the Mythic install directory, use the following command to install Apollo as a **non-root** user:

```
sudo -E ./mythic-cli install github https://github.com/MythicAgents/Apollo.git
```

Once installed, restart Mythic to build a new agent.

Notable Features

- P2P Communication
- Credential Tracking and Manipulation
- Unmanged PE, .NET Assembly, and PowerShell Script Execution
- User Exploitation Suite
- SOCKSv5 Support

- Advanced HTTPX Profile with Malleable Configuration Support
- Message Transform Support (Base64, NetBIOS, XOR, etc.)
- Domain Rotation and Proxy Support

Commands Manual Quick Reference

Command	Syntax	Description
assembly_inject	<code>assembly_inject -PID [pid] -Assembly [assembly] -Arguments [args]</code>	Execute .NET assembly in remote process.
blockdlls	<code>blockdlls -EnableBlock [false]</code>	Block non-Microsoft signed DLLs from loading into post-ex jobs.
cat	<code>cat -Path [file]</code>	Retrieve the output of a file.
cd	<code>cd -Path [dir]</code>	Change working directory.
cp	<code>cp -Path [source] -Destination [destination]</code>	Copy a file from path to destination.
dcsync	<code>dcsync -Domain contoso.local [-User username -DC dc.ip]</code>	DCSync one or more user credentials
download	<code>download -Path [path] [-Host [hostname]]</code>	Download a file off the target system.
execute_assembly	<code>execute_assembly -Assembly [assembly.exe] -Arguments [args]</code>	Execute a .NET assembly registered with <code>register_file</code>
execute_coff	<code>execute_coff -Coff [object.x64.o] -Function [go] -Timeout [30] [-Arguments [args]]</code>	Execute a object file (BOF) that's been registered with <code>register_file</code>
execute_pe	<code>execute_pe -PE [binary.exe] -Arguments [args]</code>	Execute a statically compiled executable that's been registered with <code>register_file</code>
exit	<code>exit</code>	Task agent to exit.
get_injection_techniques	<code>get_injection_techniques</code>	Show currently registered injection techniques as well as the current technique.

Command	Syntax	Description
getprivs	getprivs	Enable as many privileges as possible for the current access token.
ifconfig	ifconfig	Get Network Adapters and Interfaces
inject	inject	Inject a new payload into a remote process.
inline_assembly	inline_assembly -Assembly [Assembly.exe] -Arguments [Additional Args]	Execute a .NET assembly in the currently executing process that's been registered with register_file
jobkill	jobkill [jid]	Kill a running job in the agent.
jobs	jobs	List all running jobs.
keylog_inject	keylog_inject -PID [pid]	Inject a keylogger into a remote process.
kill	kill -PID [pid]	Attempt to kill the process specified by [pid] .
link	link	Link to a P2P agent via SMB or TCP. Modal popup only.
load	load command1 command2 ...	Load new commands into the agent.
ls	ls [-Path [path]]	List files and folders in [path] . Defaults to current working directory.
make_token	make_token	Impersonate a user using plaintext credentials. Modal popup.
mimikatz	mimikatz -Command [args]	Execute Mimikatz with the specified arguments.
mkdir	mkdir -Path [dir]	Create a directory.
mv	mv -Path [source] -Destination [destination]	Move a file from source to destination. Modal popup.

Command	Syntax	Description
net_dclist	<code>net_dclist [domain.local]</code>	List all domain controllers for the current or specified domain.
net_localgroup_member	<code>net_localgroup_member -Group [groupname] [-Computer [computername]]</code>	Retrieve membership information from a specified group on a given computer.
net_localgroup	<code>net_localgroup [computer]</code>	Retrieve local groups known by a computer. Default to localhost.
net_shares	<code>net_shares [-Computer [computer]]</code>	Show shares of a remote PC.
netstat	<code>netstat [-Tcp -Udp -Established - Listen]</code>	Get TCP and UDP connections
powerpick	<code>powerpick -Command [command]</code>	Executes PowerShell in a sacrificial process.
powershell	<code>powershell -Command [command]</code>	Executes PowerShell in your currently running process.
powershell_import	<code>powershell_import</code>	Register a new .ps1 file to be used in other PowerShell jobs
ppid	<code>ppid -PID [pid_integer]</code>	Set the PPID of sacrificial jobs to the specified PID.
printspoofer	<code>printspoofer -Command [command]</code>	Execute a command in SYSTEM integrity so long as you have SeImpersonate privileges.
ps	<code>ps</code>	List process information.
psinject	<code>psinject -PID [pid] -Command [command]</code>	Executes PowerShell in the process specified by <code>[pid]</code> . Note: Currently stdout is not captured of child processes if not explicitly captured into a variable or via inline execution (such as <code>\$(whoami)</code>).
pth	<code>pth -Domain [domain] -User [username] -NTLM [ntlm_hash] [-AES128 [aes128_key] -AES256 [aes256_key] -Run [program.exe]]</code>	Use <code>mimikatz</code> 's pth module to spawn a process with alternate credentials.

Command	Syntax	Description
pwd	<code>pwd</code>	Print working directory.
reg_query	<code>reg_query -Hive [HKCU:\ HKU:\ HKLM:\ HKCR:\] [-Key [keyname]]</code>	Query all subkeys of the specified registry path. Needs to be of the format <code>HKCU:\</code> , <code>HKLM:\</code> , or <code>HKCR:\</code> .
reg_write_value	<code>reg_write_value -Hive [HKCU:\ HKU:\ HKLM:\ HKCR:\] -Key [keyname] [-Name [value_name] -Value [value_value]]</code>	Write specified values to the registry keys.
register_assembly	<code>register_assembly</code>	Register a .NET assembly with the agent to be used in .NET post-exploitation activities
register_file	<code>register_file</code>	Register a file to the agent's file cache. Used to store assemblies, executables, and PowerShell scripts.
rev2self	<code>rev2self</code>	Revert the access token to the original access token.
rm	<code>rm -Path [path] [-Host [hostname] -File [filename]]</code>	Remove a file specified by <code>[path]</code> . Alternatively, if <code>-File</code> is provided, <code>-Path</code> will be used as the directory, and <code>-File</code> will be the filename.
run	<code>run -Executable [binary.exe] -Arguments [args]</code>	Runs the binary specified by <code>[binary.exe]</code> with passed arguments (if any).
sc	<code>sc [-Query -Start -Stop -Create -Delete] [-Computer [computername] -DisplayName [display_name] -ServiceName [servicename] -BinPath [binpath]]</code>	.NET implementation of the Service Control Manager.
screenshot_inject	<code>screenshot_inject -PID [pid] [-Interval [int] -Count [int]]</code>	Get a screenshot of the desktop session associated with <code>PID</code> every <code>Interval</code> seconds for <code>Count</code> screenshots.

Command	Syntax	Description
screenshot	screenshot	Get a screenshot of the current screen.
set_injection_technique	set_injection_technique [technique]	Set the injection technique used in post-ex jobs that require injection.
shell	shell [command]	Run a shell command which will translate to a process being spawned with command line: cmd.exe /S /c [command]
shinject	shinject	Inject given shellcode into a specified pid. Modal popup only.
sleep	sleep [seconds]	Set the callback interval of the agent in seconds.
socks	socks -Port [port]	Standup the socks server to proxy network traffic, routable via Mythic on [port] .
spawn	spawn	Spawn a new callback in the postex process specified by spawn*_* .
spawnto_x64	spawnto_x64 -Application [path] - Arguments [args]	Sets the process used in jobs requiring sacrificial processes to the specified [path] with arguments [args] .
spawnto_x86	spawnto_x86 -Application [path] - Arguments [args]	Sets the process used in jobs requiring sacrificial processes to the specified [path] with arguments [args] .
steal_token	steal_token [pid]	Attempts to steal the process's primary token specified by [pid] and apply it to our own session.
unlink	unlink	Unlink a callback linked to via the link command. Modal popup only.

Command	Syntax	Description
upload	<code>upload</code>	Upload a file to a remote path on the machine. Modal popup only.
whoami	<code>whoami</code>	Report access token for local and remote operations.

Supported C2 Profiles

[HTTP Profile](#)

The HTTP profile calls back to the Mythic server over the basic, non-dynamic profile. When selecting options to be stamped into Apollo at compile time, all options are respected with the exception of those parameters relating to GET requests.

[HTTPX Profile](#)

Advanced HTTP profile with malleable configuration support and message transforms. Provides significantly more flexibility and OPSEC benefits compared to the basic HTTP profile, making it ideal for red team operations.

[SMB Profile](#)

Establish communications over SMB named pipes. By default, the named pipe name will be a randomly generated GUID.

[TCP Profile](#)

Establish communications over a specified network socket. Note: If unelevated, the user may receive a prompt to allow communications from the binary to occur over the network.

[WebSocket Profile](#)

Establish communications over WebSocket connections for real-time bidirectional communication.

SOCKSv5 Support

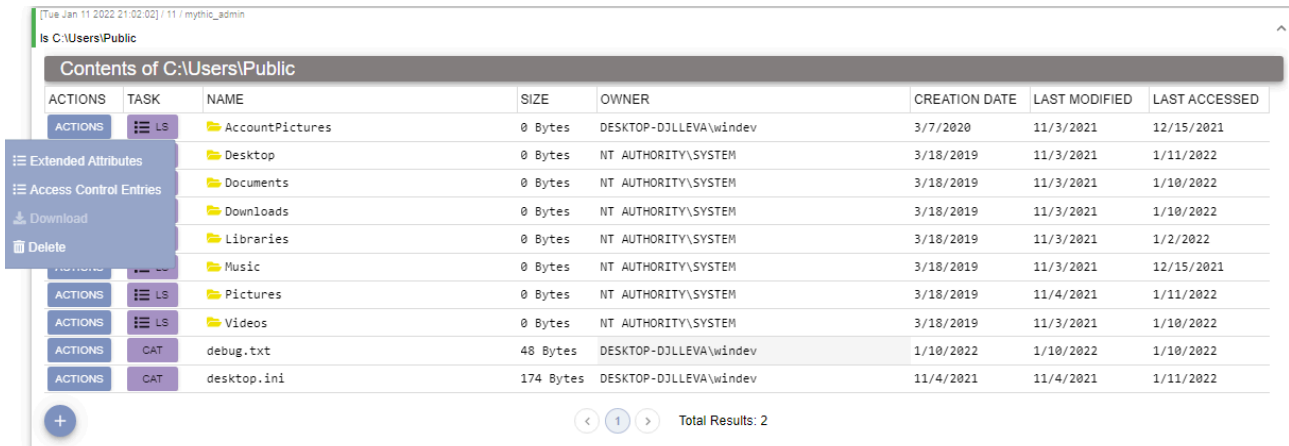
Apollo can route SOCKS traffic regardless of what other commands are compiled in. To start the socks server, issue `socks -Port [port]`. This starts a SOCKS server on the Mythic server which is `proxychains4` compatible. To stop the SOCKS proxy, navigate to the SOCKS page in the Mythic UI and terminate it.

Quality of Life Improvements

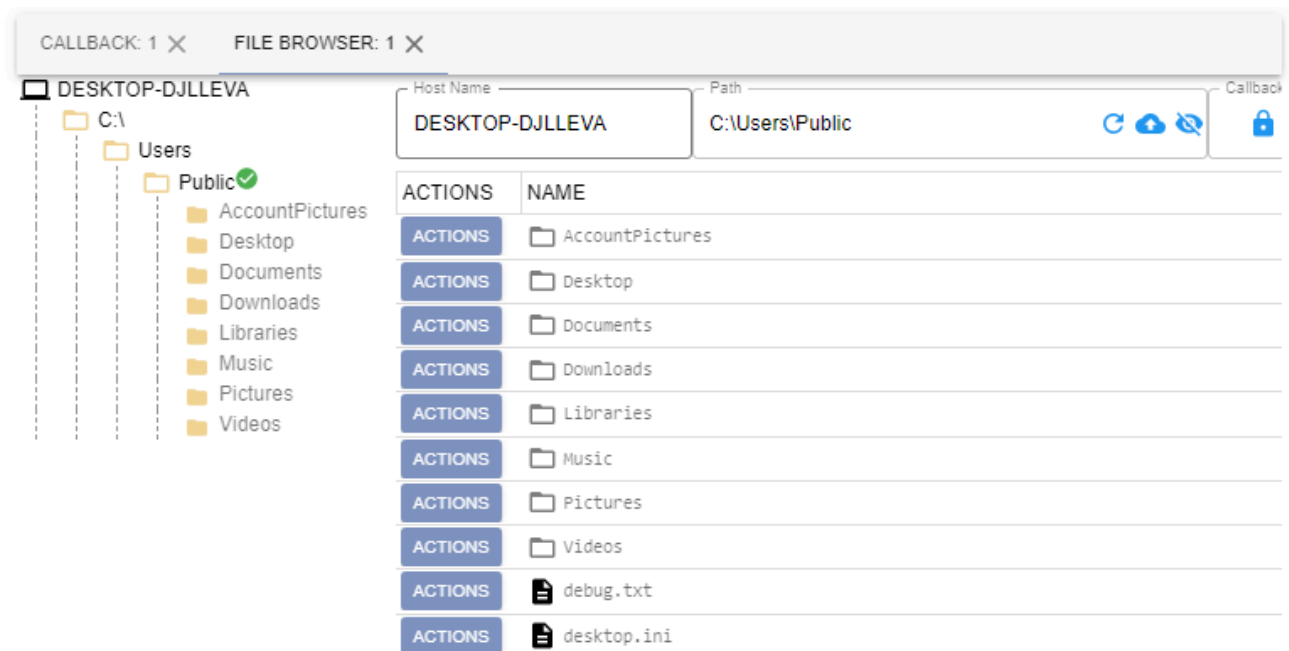
File Triage

The `ls` command reports back a wealth of information and allows operators to easily copy file paths and examine permissions of files, in addition to being able to sort and filter files. Clicking the icon under the ACLs column will show all the permissions of a file. Additionally, this hooks into Mythic's native file browser.

This shows typical `ls` output:



Interfaces with Mythic's filebrowser and caches data server-side:



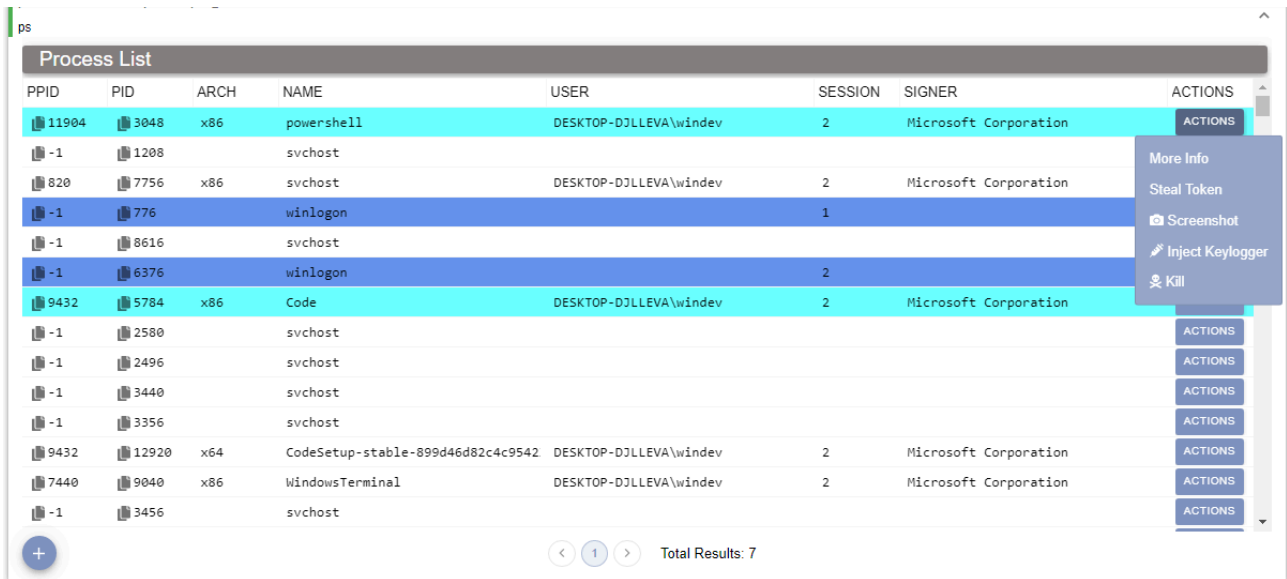
Process Listings

When issuing `ps`, additional details are retrieved such as:

- Company name of the process executable
- Description of the process executable
- Full path of the process
- Integrity level of the process
- Desktop session
- Process command line arguments

This process listing also interfaces with Mythic's builtin process browser, which allows you to see process trees more easily.

Lastly, the associated browser script will do row highlighting based on the process's name (in a one-to-one port of [this script](#))



Portable Executable, Assembly, and PowerShell Script Caching

Apollo can cache files for expeditious task execution. In general, control flow follows the `register_file` command followed by whatever command you wish to execute (`execute_assembly` , `powerpick` , `execute_pe` , etc.). These files are cached client side via DPAPI encrypted AES256 blobs, preventing their signatures being exposed outside of task execution.

Dynamic Injection Techniques

The agent can change what code injection technique is in use by post-exploitation jobs that require injection through a suite of injection commands. Currently, injection techniques that are supported:

- CreateRemoteThread
- QueueUserAPC (Early Bird)

[Tue Jan 11 2022 22:17:57] / 13 / mythic_admin

get_injection_techniques

Loaded Injection Techniques

SET	NAME
SET	QueueUserAPC
SET	CreateRemoteThread

+

[Tue Jan 11 2022 22:18:04] / 14 / mythic_admin

set_injection_technique QueueUserAPC

```
1 Set injection technique to QueueUserAPC
```

+

Job Tracking

Agent jobs are tracked by job ID, by command, and by the arguments passed to the command so that you know what job correlates to what tasking.

[Wed Jan 12 2022 18:25:03] / 28 / mythic_admin

jobs

Running Jobs

KILL	OPERATOR	COMMAND	ARGUMENTS
KILL	mythic_admin	powershell	-Command start-sleep 100

+ < 1

[Wed Jan 12 2022 18:25:09] / 29 / mythic_admin

jobkill 20fd0916-57b6-4eb7-a6d0-c89fdb5f9cd1

```
1 Killed 20fd0916-57b6-4eb7-a6d0-c89fdb5f9cd1
```

+ < 1

Artifact Tracking

Commands that manipulate the disk, create new logons, or spawn new processes will document those changes in the Artifact Reporting page as shown below.

CALLBACKS	TASKS	FILES	CREDENTIALS	KEYLOGS	ARTIFACTS	TOKENS	SOCKS
Search...						Search Artifacts by Artifact	
Type	Command	Task	Callback	Host	Artifact		
FileDelete	rm	12	1	DESKTOP-DJLLEVA	Deleted C:\Users\windev\Downloads\Rubeus.exe.config		
FileWrite	upload	11	1	DESKTOP-DJLLEVA	Wrote 161 bytes to C:\Users\windev\Downloads\Rubeus.exe.config		
ProcessKill	execute_pe	9	1	DESKTOP-DJLLEVA	Killed PID 10916		
ProcessInject	execute_pe	9	1	DESKTOP-DJLLEVA	Injected into PID 10916 using CreateRemoteThread		
ProcessCreate	execute_pe	9	1	DESKTOP-DJLLEVA	Started C:\Windows\System32\rundll32.exe (PID: 10916)		

And more!

There's a number of niceities that come with pairing an agent to Mythic - too many to list in one README. Install the agent and see for yourself!

Special Thanks

A big thanks goes to those who have contributed to the project in both major and minor ways.

- Cody Thomas, [@its_a_feature](#)
- Calvin Hedler, [@001SPARTaN](#)
- Lee Christensen, [@tifkin](#)
- Brandon Forbes, [@reznok](#)
- Thiago Mayllart, [@thiagomayllart](#)
- Matt Hand, [@matterpreter](#)
- Hope Walker, [@IceMoonHSV](#)
- Jack Ullrich, [@winternl_t](#)
- Elad Shamir, [@elad_shamir](#)
- Ben Turner [@benpturner](#)
- Ian Wallace [@strawp](#)
- m0rv4i [@m0rv4i](#)
- Harley Lebeau [@r3dQu1nn](#)
- Antonio Quina [@st3r30byt3](#)
- Sean Pierce [@secure_sean](#)
- Evan McBroom, [@EvanMcBroom](#)
- Matt Ehrnschwender, [@M_alphaaa](#)

Source: <https://github.com/MythicAgents/Apollo>