A Change of Mythic Proportions

By Cody Thomas

Published: 2020-08-13 · Archived: 2026-04-05 18:06:00 UTC

In early 2018, I looked into macOS tradecraft for red teaming and developed a new proof of concept agent in JavaScript for Automation (JXA). I looked through open source Command and Control (C2) frameworks to see if there was one I could easily integrate my agent into, but didn't find one to meet my needs. Many frameworks at the time weren't modular in agent support, and I was tired of learning a new C2 interface for every agent. So, I designed and released a new C2 framework in July 2018 called `Apfell`. The goal was a unified, web front-end with a bunch of quality-of-life improvements that supported multiple agents. Due to the nature of my work at the time, many of the initial agents were macOS or Linux based.

Fast-forward two years, and Apfell has really grown. Multiple people have contributed agents (Josiah Massari, Christopher Ross, Dwight Hohnstein) across Windows (atlas), macOS (apfell and poseidon), Linux (poseidon), and Chrome (leviathan), and many people at SpecterOps have spent countless hours finding bugs as we used Apfell on operations. Unfortunately, due to the logo and the name from years ago, many people think Apfell is a macOS only framework. So, I'm re-branding the framework to something broader in hopes that it breaks that stereotype.

I want every agent to feel unique, every contributor to feel like they can really leave their mark, and every developer to have as much freedom as possible. Every agent and C2 communications profile should be decoupled as much as possible, and every agent should be free to hook into as many or as few features of the framework as they want. To this end, the Apfell C2 framework is re-branded as `Mythic`:

Mythic C2 Logo

What does this mean as a user of Apfell?

I debated for a while how to handle the re-brand. I want to minimize breaking changes as much as possible, then I saw this from GitHub regarding repository name changes:

> In addition to redirecting web traffic, all `git clone`, `git fetch`, or `git push` operations targeting the previous location will continue to function as if made on the new location.

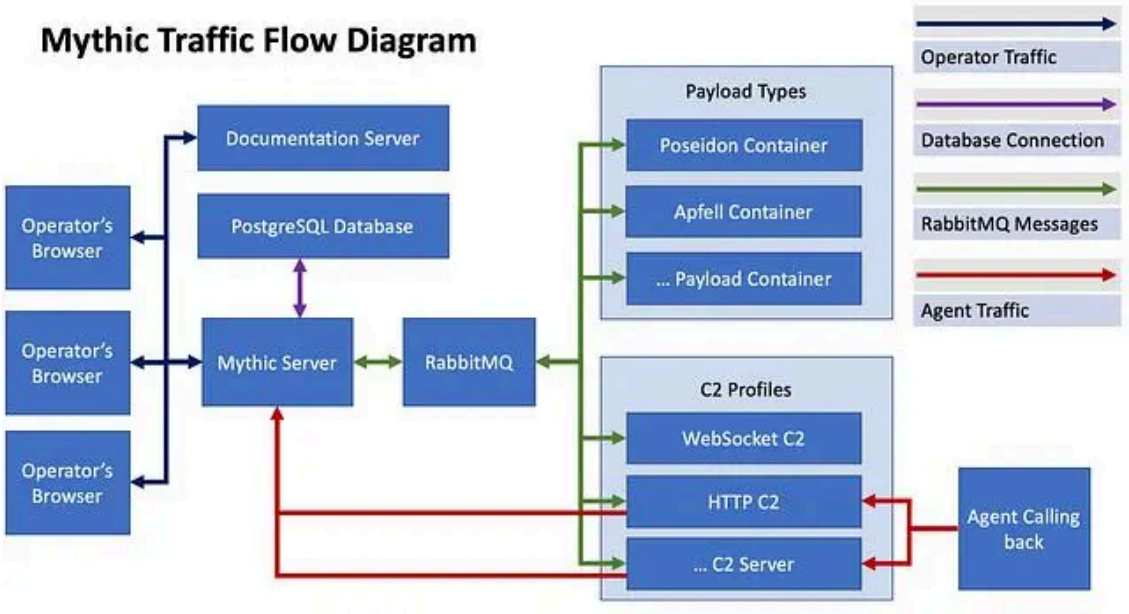
So, to make a smooth transition, the GitHub repository name will simply change from Apfell to Mythic. Code wise, this is simply treated as version 2.0.

All documentation on `https://docs.apfell.net` will redirect to the new url `https://docs.mythic-c2.net`. Lastly, the `apfell-jxa` payload that leverages JavaScript for Automation (JXA) is renamed to just the `apfell` payload.

What is Mythic?

Mythic is a multiplayer, command and control platform for red teaming operations across macOS, Windows, and Linux. One of the Mythic project's main goals is to provide quality of life improvements to operators, improve maintainability of agents, enable customizations, and provide more robust data analytic capabilities to operations.

Fundamentally, Mythic uses a web-based front end and Docker containers for the back-end; however, Mythic can leverage remote VMs or physical computers as compatible “containers” for agents and C2 profiles. This feature is particularly useful as certain agents might need to build on specific operating systems or might have highly complex build toolchains that make a linux Docker container unrealistic to leverage. The image below shows a high level diagram of how traffic flows between all the different Mythic components:



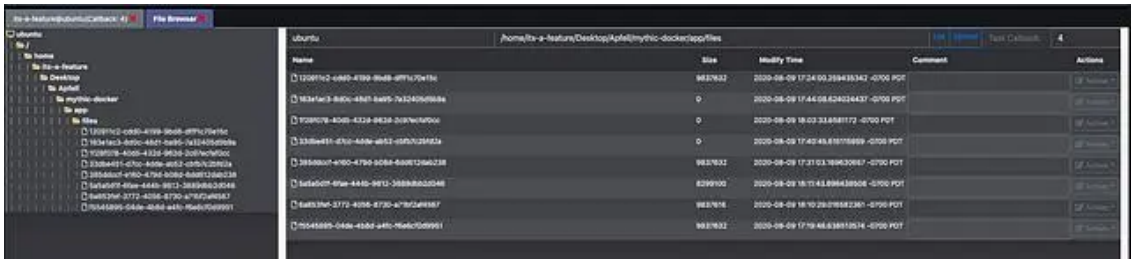
From an assessment perspective, Mythic provides fine grained control over what operators can do, allows scripting of the agent output, tracks tasking via MITRE ATT&CK, allows searching across an operation, and provides a comment mechanism for tasking and files. All of this provides quality of life improvements for assessment leads, deconflictions, and multi-operator operations.

Any new features with this release?

This isn’t simply an announcement of the name change, there are a few new features added to Mythic: file browser, SOCKS support, spectator view, and more documentation.

File Browser:

Cobalt Strike really made waves in the red teaming world with the addition of a file browser. This feature alone saves operators a lot of time and helps aggregate browsing data. Unfortunately, that implementation has some limitations. So, Mythic has its own implementation of a file browser with some notable additions. The data is shared across all operators, persistent across reboots, allows for comments, collects detailed permissions data, indicates if the file has been downloaded or not, and shows the download history for a file (in case you download a file multiple times).



SOCKS5 Support:

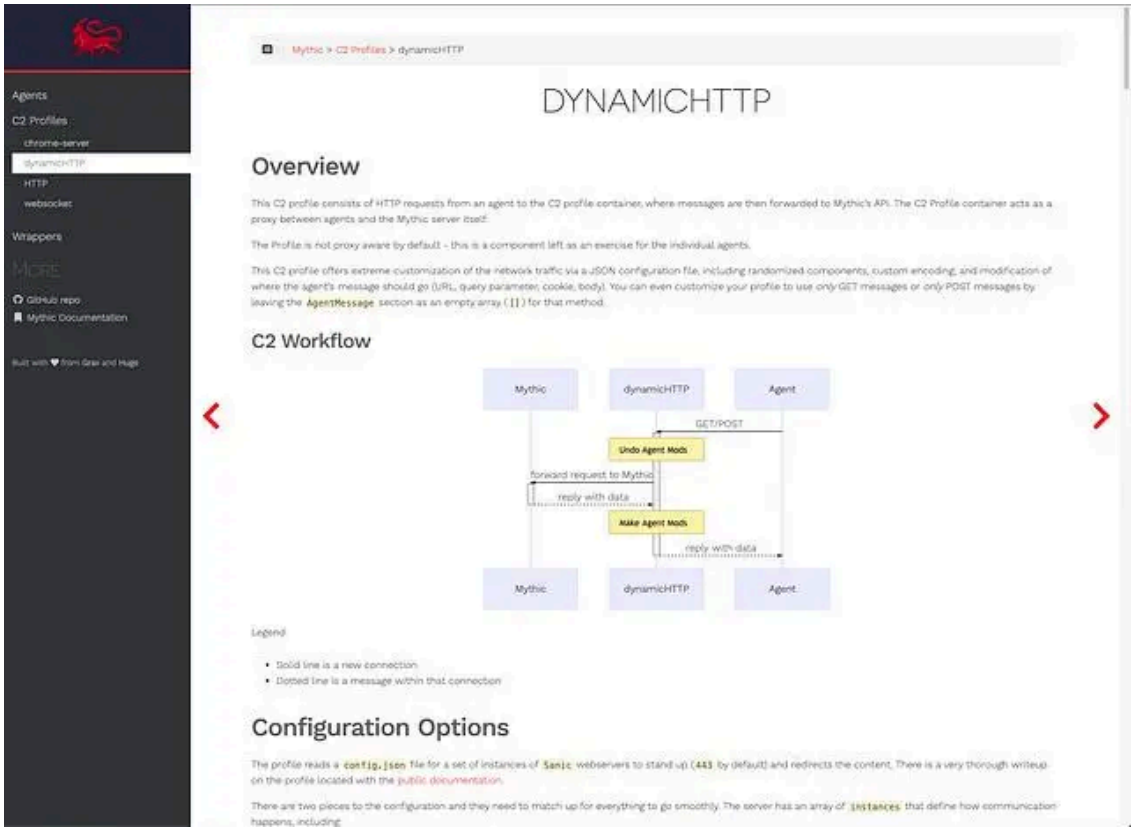
No one tool has all the functionality needed for an entire operation, and there's no reason for that to be the case. So, operators have a suite of tools they use for various purposes. The hard part is leveraging all of these tools through a framework's C2 channel. This is where SOCKS comes into play — you can tunnel all of your other tooling through the current C2 profile's traffic via SOCKS5. This is currently only supported by the `Poseidon` agent, but other agents will support it in time.

Spectator View:

During engagements, you occasionally need to give other people/teams access to your operator console for collaboration; however, you don't want to give them the same level of control that you have. In addition to Mythic allowing you to create `block lists` of commands per agent that people aren't allowed to execute, you can set an operator's mode to `spectator` which prevents them from making any changes across the entire operation. This includes payload creation, modifying C2 profiles, sending tasking, editing artifacts, modifying credentials, editing browser scripts, etc. Essentially, the operator will have read-only access to the operation.

Documentation:

Everybody loves to see documentation, but nobody wants to write it. It's also a difficult UI design decision on how to best show detailed agent and C2 documentation without cluttering up the main user interface. Mythic now has a `documentation` docker container that locally serves a mini Hugo WebServer with detailed documentation for all of the agents and C2 profiles. This includes notes on specific API calls used for functions, OPSEC considerations for agents and functions, how C2 looks on the wire, and more.



Operators can still type `help [command name]` to see some basic description and execution context in a helpful popup message, but the detailed descriptions will be in the documentation docker container.

Poseidon Bonus:

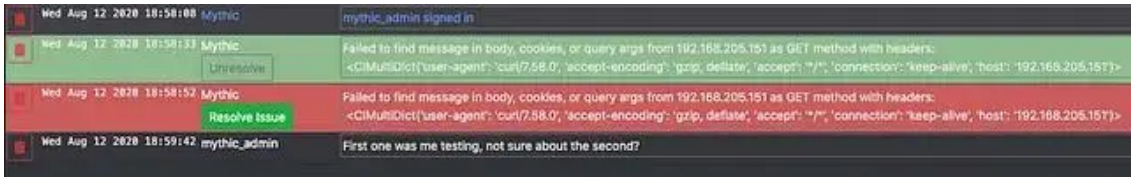
Many people commented on the Poseidon container being large. That container is now down to 3.5GB and the Poseidon agent now builds in about 5 seconds. The reason that container is still so large is because it has the build chains and SDKs for Linux and macOS so that the Poseidon Golang agent can utilize Objective C API calls in addition to Golang functionality.

Developer Bonus:

One of the big shifts in this update is for ease of development. The processing and ingestion of new Payload Types and C2 Profiles changed away from massive JSON config files. More power is given to developers to customize how processing of commands is handled, how payloads are created, and what is displayed to the user.

Search and Eventing Bonus:

The search capabilities within Mythic were expanded to include file browser data and the event log. In addition to the event log being a simple chat program between operators and a notification engine for Mythic, there's an additional feature for `warnings`. These are messages that an operator should pay attention to and from an OPSEC perspective, need to be "resolved":



Going Forward

There are still many things on the roadmap for Mythic to add such as more scripting hooks into the framework, more flexibility around C2 profiles, more analytics, integration of the tracked artifacts into deconfliction reports, updated MITRE ATT&CK mappings for the new sub-techniques, operating system specific tracking, additional views of an operation, and more.

There will also be more agents released and included as part of the base Mythic framework along with helpful videos on how to easily add in your own agent in the near future.
