Making Cobalt Strike harder for threat actors to abuse

By Greg Sinclair

Published: 2022-11-18 · Archived: 2026-04-05 17:23:42 UTC

Cobalt Strike, the popular tool used by red teams to test the resilience of their cyber defenses, has seen many iterations and improvements over the last decade. First released in 2012, it was originally the commercial spinoff of the open-source Armitage project that added a graphical user interface (GUI) to the Metasploit framework to help security practitioners detect software vulnerabilities more quickly.

It has since matured into a point-and-click system for the deployment of the Swiss Army Knife of remote access tools onto targeted assets. While the intention of Cobalt Strike is to emulate a real cyber threat, malicious actors have latched on to its capabilities, and use it as a robust tool for lateral movement in their victim's network as part of their second-stage attack payload.

Cobalt Strike vendor Fortra (until recently known as Help Systems) uses a vetting process that attempts to minimize the potential that the software will be provided to actors who will use it for nefarious purposes, but Cobalt Strike has been leaked and cracked over the years. These unauthorized versions of Cobalt Strike are just as powerful as their retail cousins except that they don't have active licenses, so they can't be upgraded easily.

We are releasing to the community a set of open-source YARA Rules and their integration as a VirusTotal Collection to help the community flag and identify Cobalt Strike's components and its respective versions. Since many threat actors rely on cracked versions of Cobalt Strike to advance their cyberattacks, we hope that by disrupting its use we can help protect organizations, their employees, and their customers around the globe.

Inside Cobalt Strike

Cobalt Strike is a collection of multiple software tools rolled into a single JAR file. An actor begins by activating the Team Server component, which sets up a centralized server that operates as both a Command and Control (C2) endpoint and a coordinating hub for multiple actors to control infected devices.

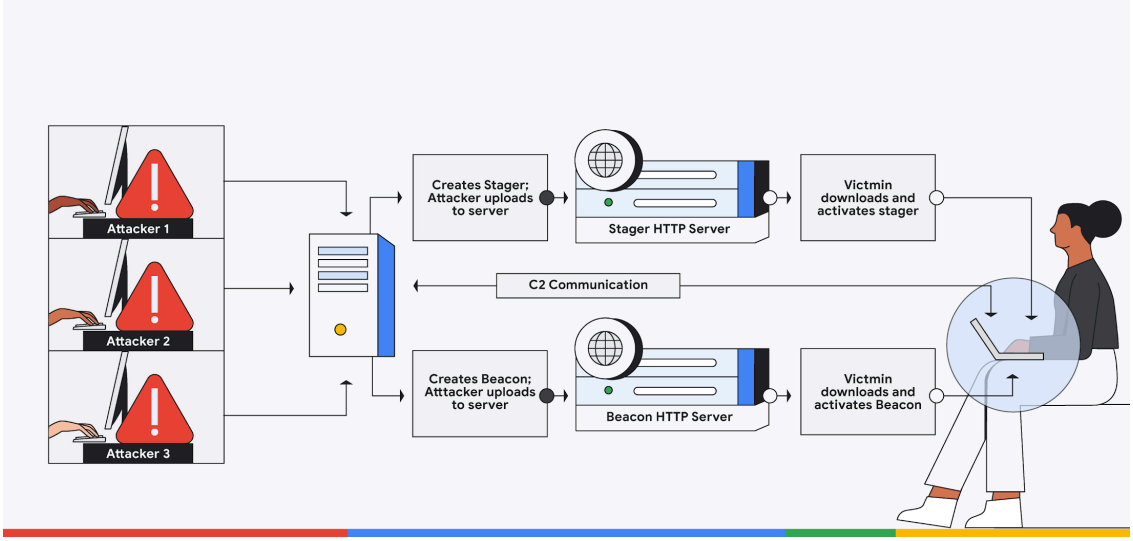

Figure 1: Typical Cobalt Strike infrastructure setup

Actors connect to the Team Server by activating the JAR as a Client. The Client serves the GUI from which the actor can control the Team Server and infected hosts. The Team Server generates a multitude of attack framework components that actors can deploy to infect and control remote endpoints.

Cobalt Strike contains several delivery templates for Javascript, VBA macros, and Powershell scripts which can deploy small shellcode (diskless) implants known as stagers. These stagers call back to the Team Server via one of the supported communication channels, including HTTP/HTTPS, SMB, and DNS to download the final stage implant known as the Beacon.

The Beacon is the core binary that gives the actor control over the infected computer. It supports multiple commands and operations, while also being extensible to enable downloading and execution of actor developed modules. The Team Server/Client model also allows multiple actors to collaborate on a collection of infected assets.

The stagers, templates, and beacon are contained within the Cobalt Strike JAR file. They are not created on the fly, nor are they heavily obfuscated before deployment from the Team Server. Cobalt Strike offers basic protection using a reversible XOR encoding.

Solving for hacked Cobalt Strike

We were able to locate versions of the Cobalt Strike JAR file starting with version 1.44 (circa 2012) up to version 4.7 (the latest version at the time of publishing this blog). We cataloged the stagers, templates, and beacons, including the XOR encodings used by Cobalt Strike since version 1.44.

With the set of Cobalt Strike components available, we built YARA-based detection across these malicious variants in the wild with a high degree of accuracy. Each Cobalt Strike version contains approximately 10 to 100

attack template binaries. We found 34 different Cobalt Strike release versions with a total of 275 unique JAR files across these versions. All told, we estimated a minimum of 340 binaries that must be analyzed and have signatures written to detect them.

For each release version of Cobalt Strike, we found that a new, unique beacon component is usually created. The stagers and templates, however, tend to be more constant across versions. Looking for unique stagers, templates, and beacons across the different versions, a total of 165 signatures were generated to detect these Cobalt Strike components across the versions of Cobalt Strike up to and including version 4.7.

Our goal was to make high-fidelity detections to enable pinpointing the exact version of particular Cobalt Strike components. Whenever possible, we built signatures to detect specific versions of the Cobalt Strike component.

Containing Cobalt Strike abuse

We decided that detecting the exact version of Cobalt Strike was an important component to determining the legitimacy of its use by non-malicious actors since some versions have been abused by threat actors.

We wanted to enable better detection of actions done by bad actors, and we needed a surgical approach to excise the bad versions while leaving the legitimate ones untouched. This required detecting the exact version of the Cobalt Strike component. By targeting only the non-current versions of the components, we can leave the most recent versions alone, the version that paying customers are using.

The leaked and cracked versions of Cobalt Strike are not the latest versions from Fortra, but are typically at least one release version behind. We focused on these versions by crafting hundreds of unique signatures that we integrated as a collection of community signatures available in VirusTotal. We also released these signatures as open source to cybersecurity vendors who are interested in deploying them within their own products, continuing our commitment to improving open source security across the industry.

Our intention is to move the tool back to the domain of legitimate red teams and make it harder for bad guys to abuse. For more on using YARA Rules to help stop the abuse of Cobalt Strike, you can listen to this special Google Cloud Security podcast.

Posted in

- Security & Identity
