4. Listeners

Listeners are Cobalt Strike's abstraction on top of payload handlers. A listener is a name attached to payload configuration information (e.g., protocol, host, port, etc.) and, in some cases, a promise to setup a server to receive connections from the described payload.

Listener API

Aggressor Script aggregates listener information from all of the team servers you're currently connected to. This makes it easy to pass sessions to another team server. To get a list of all listener names, use the &listeners (functions.html#listeners) function. If you would like to work with local listeners only, use &listeners_local (functions.html#listeners_local). The &listener_info (functions.html#listener_info) function resolves a listener name to its configuration information. This example dumps all listeners and their configuration to the Aggressor Script console:

```
command listeners {
        local('$name $key $value');
        foreach $name (listeners()) {
                println("== $name == ");
                foreach $key => $value (listener_info($name)) {
                        println("$[20]key : $value");
                }
        }
}
```

Creating Listeners

Use &listener_create (functions.html#listener_create) to create a listener and start a payload handler associated with it.

Choosing Listeners

Use &openPayloadHelper (functions.html#openPayloadHelper) to open a dialog that lists all available listeners. After the user selects a listener, this dialog will close, and Cobalt Strike will run a callback function. Here's the source code for Beacon's spawn menu:

```
item "&Spawn" {
        openPayloadHelper(lambda({
                binput($bids, "spawn $1");
                bspawn($bids, $1);
        }, $bids => $1));
}
```

Shellcode

Use &shellcode (functions.html#shellcode) to generate x86 shellcode for the specified listener name.

Executables and DLLs

Use &artifact (functions.html#artifact) to generate an executable or DLL for the specified listener name.

PowerShell

Use &powershell (functions.html#powershell) to generate a PowerShell one-liner to bootstrap a listener.