Variables

Overview

A variable is a value which may be changed throughout the program. It may be changed by operators, or compared within conditional statements to alter the program flow.

Variables contain unsigned integers with a values ranging from 0 to 65535. Booleans may be represented by the keywords `TRUE` and `FALSE`, or any non-zero integer for true and `0` for false.

All variables have a **global scope** — meaning it may be referred to anywhere within the payload.

VAR

In DuckyScript, variables are initialized using the `VAR` command.

```
REM Example Integer Variable
VAR $SPEED = 2000

REM Example Boolean (TRUE/FALSE or 1/0)
VAR $BLINK = TRUE
VAR $BLINK = 1
```

Unlike a constant (declared by `DEFINE`), a variable is prepended with the dollar sign ("`$`") sigil.

Example

```
REM Constant string which may not change throughout the payload
DEFINE FOO Hello, World!

REM Variable integer which may change throughout the payload
VAR $BAR = 1337
```

Result

The constant `FOO` will always be replaced with the string "`Hello, World!`" throughout the payload.

While the variable `$BAR` currently holds the value `1337`, this may change throughout the payload — which will be detailed shortly by using operators.

Internal Variables

In addition to creating your own variables using the `VAR` command, DuckyScript 3 provides many built-in internal variables. These variables exist automatically and are prepended with dollar sign underscore ("`$_`"). These internal variables will be described in full in sections ahead relevant to their individual usage. For a complete list you may find them listed in the quick reference.

Avoiding Errors

- Variable names should only contain letters, numbers and underscore ("`_`").
- Internal variables begin with an underscore, so it is best practice to avoid this style.
- Spaces cannot be used in naming a variable — however underscore makes for a suitable replacement. For example: `VAR $BLINK_ON_FINISH = TRUE`.
- Constants should be short and descriptive. For example, `$BLINK` is better than `$B`, and `$BLINK_ON_FINISH` is better than `$BLINK`.
- Be careful when using the uppercase letter `O` or lowercase letter `l` as they may be confused with the numbers `0` and `1`.
- Avoid using the names of commands or internal variables (e.g. `ATTACKMODE`, `STRING`, `WINDOWS`, `MAC`, `$_BUTTON_ENABLED`). See the full command and variable reference.