Malware Analysis Final Project Whitepaper

Github Repository: https://github.com/jpbrown46/MalwareAnalysisFinalProject

# Static Analysis of the Unknown Binary

`file` command output

unknown: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=5fcf1f5836b3354b3ad5c66a597540bb46973cc3, for GNU/Linux 3.2.0, stripped

Ghirda import results

Graphical user interface, application

Description automatically generated

`strings` command output

`invalid regular expression` - involves use of a regular expression

`shell` - likely runs in a terminal

`stdout` - can write to console

`failed to create temporary file in %s` - creates temporary files

`failed to open %s for` - reads/writes file input

`David MacKenzie

Jay Lepreau` - likely authors

` /home/hawkinsw/code/coreutils/install/share/locale` - modified by Professor Hawkins

` Try '%s --help' for more information.

Usage: %s [OPTION]... [FILE]...

With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.

-b, --before USAGE OMMITTED

-r, --regex USAGE OMMITTED

-s, --separator=STRING USAGE OMMITTED

--help display this help and exit

--version output version information and exit` - Usage statement

This `Usage` statement is extremely helpful, we know that `unknown` takes a file as input, with optional flags. The `regex` flag confirms that `unknown` uses regular expressions, and the `separator` flag likely indicates that the `FILE` input is a text file.

Additionally, there are numerous standard C library functions listed in the beginning of `unknown.strings`.

# Dynamic Analysis of the Unknown Binary

`unknown` takes a text file (or standard in by default).

`unknown` will print the input, in reverse order by the `separator` (which is a new line character by default) to standard out.

`unknown` takes a `before` flag, that appears to change where the `separator` is expected (i.e. beginning of the line, rather than the end of the line)

`unknown` takes a `regex` that appears to change the `separator` to a regular expression.

Using static and dynamic analysis, it was determined that the `unknown` binary is `tac`.