Running r2Pipe Python in batch interactive radare2 session. With a good understanding of the r2 commands explored throughout this

We've looked at how to pass le paths as arguments and how to run scripts, commands and macros without opening an

using Go to drive r2pipe. adding comments, and decoding strings, and we have navigated some of the complexities of dealing with

your script as an argument when you call r2pipe as follows:

For example, this will print out the result of the

option will tell r2 to quit after running the script.

Note that we didn't decode all encrypted strings in the binary. We could iterate over all strings (including non-encrypted

as an argument to the

At this point, since the

binary and at the prompt type:

[3], the "pd 1" command tells r2 to print one line of disassembly from the given address.

We next implement a function that will print out the disassembly along with the commented string to the r2 prompt. At

code is another work around for wrapping the string in quotes so that any special characters in the string do not get

As r2pipe's Go implementation doesn't support easy capture of

For each command we want to pass to the r2 session, we rst format the command as a string, then pass the command

see why shortly.

The

launchctl%

PHIL STOKES

Thus, [1] formats a command that returns the bytes at the current address as a string. At [2], we format a

!pipe /usr/

local

/bin/godec/decode

for that.

command is awkward to remember and type out every time, you might want to create

Phil Stokes is a Threat Researcher at SentinelOne, specializing in macOS threat intelligence,

Automated String Decryption in OSX.Fairytale

lets me hack scripts together rather haphazardly to achieve what I need. When scripting your own reversing sessions,

Other Tools and Plugins

To illustrate automated string decryption. Though I'll be using Go, you can easily

os /usr/local/bin/rxorb

for the reader.

To tackle three different challenges that are common to RE automation: decrypting strings, applying comments, and

other tools, or process les in batches. Most reverse engineering platforms have some kind of scripting engine to help

SECURITY RESEARCH

©2023 SentinelOne, All Rights Reserved.

Eventually, you'll want to tackle more complex automation, like dealing with encrypted strings. This is where a

After the imports, we declare a global variable

specify whether the cipher needs to be base64 decoded before being XOR'd with the given key. This alone will take care

Saving this as

should give us the option to do both.

Another work around to convert the string into a command that returns the bytes at the current address as a string is

Example: Running a script using Go to drive r2pipe:

```
import "github.com/radareorg/r2pipe-go"

package main

import (
    "fmt"
    "os"
    "strings"
)

func main() {
    r2p := r2pipe.NewPipe(os.Args[1])
    defer r2p.Close()

    cmd := fmt.Sprintf("!rxorb -b %s %s > /tmp/rxorb.txt", os.Args[2], os.Args[3])
    r2p.Cmd(cmd)
}
```

This script accepts a binary path, an encrypted string, and a key as arguments, and runs the "rxorb" command on it:

```
Usage: Provide path to a binary.
```

This can be useful for automating tasks, especially when dealing with encrypted strings. By using a script, you can

For example:

```
rxorb $0 @ $1
```


richtext