Writing and Monitoring Mutation Fuzzers

Dr. Jared DeMott
CTO AND FOUNDER

@jareddemott www.vdalabs.com

Overview

Mutation fuzzing

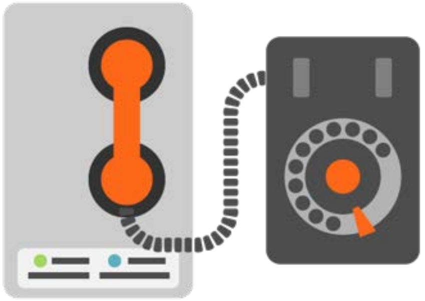
Monitoring

Demo

Demo

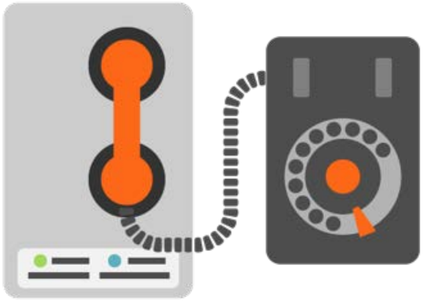
File Fuzzing

- Mac file fuzzing script



Mutation Fuzzing

- Need good samples and/or lots of tests
  - Bit flipping
    - DWORD sliding, etc.
    - Effective in the past



Network Packet Mutator

- Fuzz client or server
- Capture/replay based
- Old example: GPF
    - ./GPF –C ftp.pcap ftp.gpf
    - Command-line driven modes
    - ../GPF –P ftp.gpf client 192.170.1.105 21 ? Tcp 123456 1000 3 auto none short normal_ascii med quit

Debug Heap

Library interception

- Create a new library which exports the same symbols as libraries used by the application

Debug Heap

Electric Fence for Linux and Guard Malloc for BSD/Mac OS X

- Supplies its own version of malloc() and free()
- Puts each allocation on its own virtual memory page and places the end of the buffer at the end of this page
- The next virtual memory page is purposefully left unallocated

Debug Heap

Windows

- gflags –i blah.exe +hpa +ust
- The key benefit to this is that heap bugs that might not cause an exception right away (or ever) are caught immediately with a bus error
- Does slow down the application
  - Trade off between monitoring and testing time

Monitoring

How will we detect faults?

- Start by manually causing a fault
  - Shows you what fault will look like
- Fuzzing is pointless if you’re not sure how to detect errors
  - Debuggers, OS or application logs, network sniffs, crash files, etc.
- Some fuzzing tools can “bin” crashes
  - 5000 crashes could be the result of one bug

Gen 1

Attach to the process with debugger

Run attack

Wait for exceptions

Might not work well for two reasons:

- Doesn’t scale/requires constant manual work
- Catches too many 1st chance exceptions
    - IE is the king of exceptions that are “OK”
        - Exception types can of course be ignored

Gen 2

Wrapped by another program

```
crash.exe "C:\Program Files\QuickTime\QuickTimePlayer.exe" 5000 C:\bad-1.m4v
```

```
[*] crash.exe "C:\Program Files\QuickTime\QuickTimePlayer.exe" 5000 C:\bad.m4v
[*] Access Violation
[*] Exception caught at 6828e4fe mov edx,[edx+0x4]
[*] EAX:00005af4 EBX:00000000 ECX:00000004 EDX:00142ffc
[*] ESI:00142ffc EDI:00116704 ESP:001160fc EBP:00000000
```

Pydbg and Sulley Installation

Run the Sulley installer

- Installs pydbg
- Lots of setup tweaking
- Old, unmaintained project
- Still interesting and occasionally useful

Gen 3

```
import sys
from pydbg import *
from pydbg.defines import *
def handler_crash (pydbg):
    print pydbg.dump_context()
    return DBG_EXCEPTION_NOT_HANDLED
dbg = pydbg()
for (pid, name) in dbg.enumerate_processes():
    if name == sys.argv[1]:
        break
dbg.attach(pid)
dbg.set_callback(EXCEPTION_ACCESS_VIOLATION, handler_crash)
dbg.debug_event_loop()
```

Demo

Pydbg file fuzzer

Summary

Maturing fuzzers

Sulley framework coming up