Week 2 Report

Zachary Porter

2022-09-06

1 Todos

- ☑ Create detailed use case with goal of writing patch for closed souce driver
  ☒ Outline to identify an ”interesting” action for the fuzzer
  ☑ Fix reverse-continues in my library
  ☐ Implement scan->breakpoint->continue loop to test feasibility

1.1 Requirements

- The program must be run from user-space
- The user cannot debug code that is run on the other side of a interprocess socket
- Any generated patch will not work across updates

2 Adobe Acrobat Reader

2.1 Target Software

Adobe Acrobat Reader for Linux. Last supported version was 9.5.5 released in May 2013.

2.2 Reason

Since the last supported version there have been a large number of vulnerabilities in the software. None of these are patched in the Linux version

2.3 Particular CVEs

I’ve been looking at https://security.gentoo.org/glsa/200808-10 which is a RCE from an unprotected JavaScript method.

2.4 Difficulty

Hard

2.5 Detection

Unsure

2.6 Patch

Unsure

3 Test Binary with Buffer Overflow

3.1 Target Software

Vulnerable code that runs strncopy() or scanf() into a buffer of improper size causing a buffer overflow.

3.2 Reason

Lots of people make this mistake

3.3 Difficulty

Easier

3.4 Detection

Given a proper trace where a buffer overflow occurs, it should be possible to see as addresses and data on the stack gets overwritten and can be traced back to the input. (describe in person)

3.5 Patch

Alter strncopy or scanf to only read an acceptable number of chracters

4 Intel C++ Compiler

4.1 Target Software

Intel C++ Compiler

4.2 Reason

The software contains a check for vendor string ”Genuineintel” for the CPU where it spits out the optimal code. Otherwise it returns the slowest generated version of the code.

4.3 Difficulty

Medum

4.4 Detection

Check for comparisons with the string ”Genuineintel”

4.5 Patch

Replace the output of those comparisons to ensure they always pass or fail

5 Instrumentation Example

5.1 Target Software

Unspecified
: Talk in person

6 Ethics

Distributing a patched binary will never be acceptable, however, creating software to patch working but vulernable code is ethical as long as the original binary is not redistributed and the original creator is not harmed.