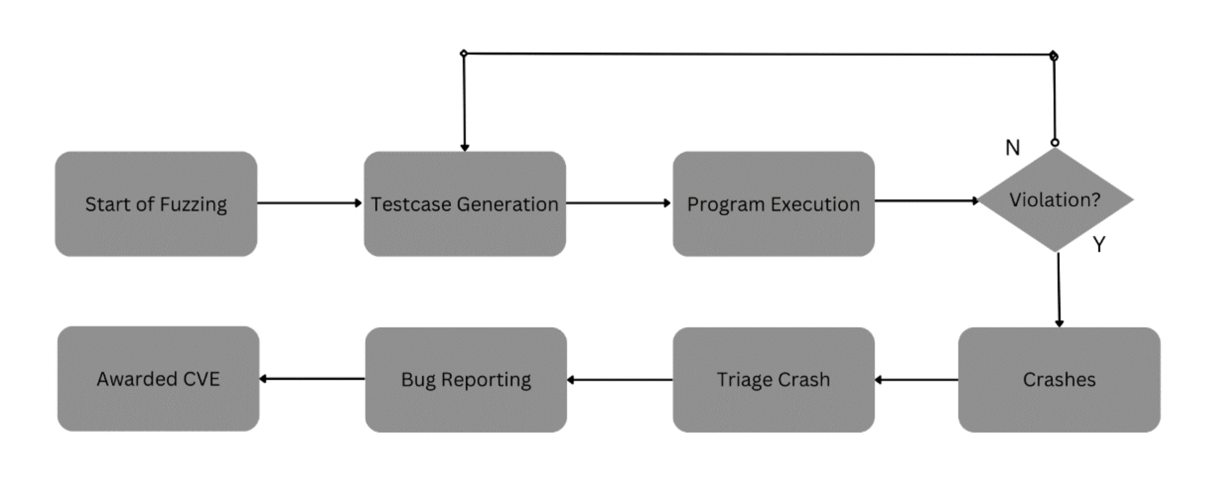
Fuzzing Linux Binaries with AFL++

Student: Lim Wei Cheng Supervisor: Dr Liu Yang

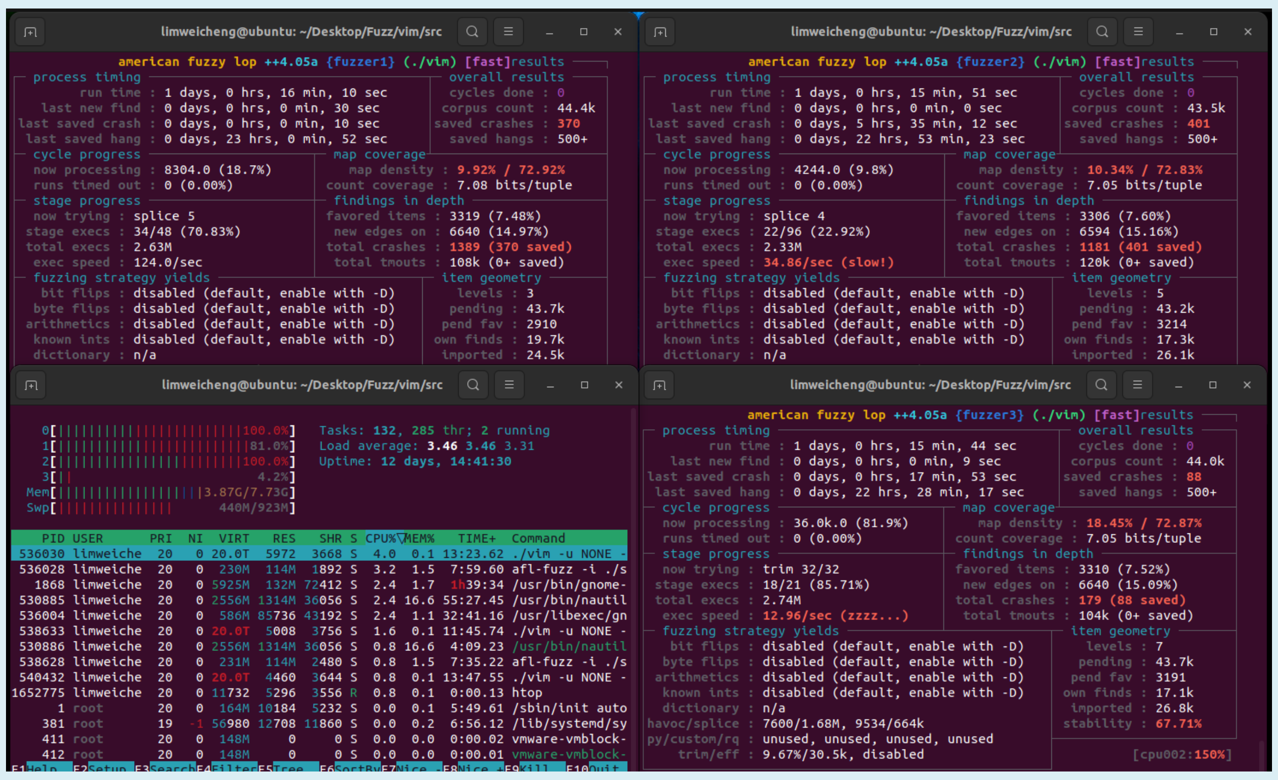
Project Objectives

The objective of this project is to demonstrate fuzzing using AFL++ on Linux programs for discovering zero-day vulnerabilities. AFL++ will be used to generate crashes, which would point towards the possibility of zero-day vulnerabilities being discovered. The plan is to triage the crashes and identify the underlying vulnerability to create accurate bug reports for the program developers. This will help the developers to mitigate the vulnerabilities and enhance the security of their program. The project also demonstrate the effectiveness of fuzzing as a tool to improve the security of Linux programs.



Flowchart of FYP: AFL++ apply various mutation strategies to input seeds to create new test cases that yield interesting crashes. Triaging the cashes and mitigate found vulnerability with developers. Found vulnerability will be awarded CVE.

AFL++ is an open-source fuzzer that can deliver state-of-the-art fuzzing results. It is a grey box fuzzer that injects additional code to instrument the target program and gather feedback on its execution. By using good-quality seeds and parallel fuzzing, AFL++ can generate crashes within days, which enables the discovery of new bugs more quickly than conventional bug hunting or software testing methods.



limweicheng@ubuntu: ~/Desktop/Fuzz/vim/src

```
            american fuzzy lop ++4.05a {fuzzer1} (./vim) [fast]results
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 1 days, 0 hrs, 16 min, 10 sec      │  cycles done : 0     │
│   last new find : 0 days, 0 hrs, 0 min, 30 sec       │ corpus count : 44.4k │
│last saved crash : 0 days, 0 hrs, 0 min, 10 sec       │saved crashes : 370   │
│ last saved hang : 0 days, 23 hrs, 0 min, 52 sec      │  saved hangs : 500+  │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 8304.0 (18.7%)     │    map density : 9.92% / 72.92%     │
│  runs timed out : 0 (0.00%)          │ count coverage : 7.08 bits/tuple    │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : splice 5               │ favored items : 3319 (7.48%)        │
│ stage execs : 34/48 (70.83%)         │  new edges on : 6640 (14.97%)       │
│ total execs : 2.63M                  │ total crashes : 1389 (370 saved)    │
│  exec speed : 124.0/sec              │  total tmouts : 108k (0+ saved)     │
├─ fuzzing strategy yields ────────────┴───────────┬─ item geometry ────────┤
│   bit flips : disabled (default, enable with -D) │    levels : 3          │
│  byte flips : disabled (default, enable with -D) │   pending : 43.7k      │
│ arithmetics : disabled (default, enable with -D) │  pend fav : 2910       │
│  known ints : disabled (default, enable with -D) │ own finds : 19.7k      │
│  dictionary : n/a                                │  imported : 24.5k      │
```

limweicheng@ubuntu: ~/Desktop/Fuzz/vim/src

```
            american fuzzy lop ++4.05a {fuzzer2} (./vim) [fast]results
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 1 days, 0 hrs, 15 min, 51 sec      │  cycles done : 0     │
│   last new find : 0 days, 0 hrs, 0 min, 0 sec        │ corpus count : 43.5k │
│last saved crash : 0 days, 5 hrs, 35 min, 12 sec      │saved crashes : 401   │
│ last saved hang : 0 days, 22 hrs, 53 min, 23 sec     │  saved hangs : 500+  │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 4244.0 (9.8%)      │    map density : 10.34% / 72.83%    │
│  runs timed out : 0 (0.00%)          │ count coverage : 7.05 bits/tuple    │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : splice 4               │ favored items : 3306 (7.60%)        │
│ stage execs : 22/96 (22.92%)         │  new edges on : 6594 (15.16%)       │
│ total execs : 2.33M                  │ total crashes : 1181 (401 saved)    │
│  exec speed : 34.86/sec (slow!)      │  total tmouts : 120k (0+ saved)     │
├─ fuzzing strategy yields ────────────┴───────────┬─ item geometry ────────┤
│   bit flips : disabled (default, enable with -D) │    levels : 5          │
│  byte flips : disabled (default, enable with -D) │   pending : 43.2k      │
│ arithmetics : disabled (default, enable with -D) │  pend fav : 3214       │
│  known ints : disabled (default, enable with -D) │ own finds : 17.3k      │
│  dictionary : n/a                                │  imported : 26.1k      │
```

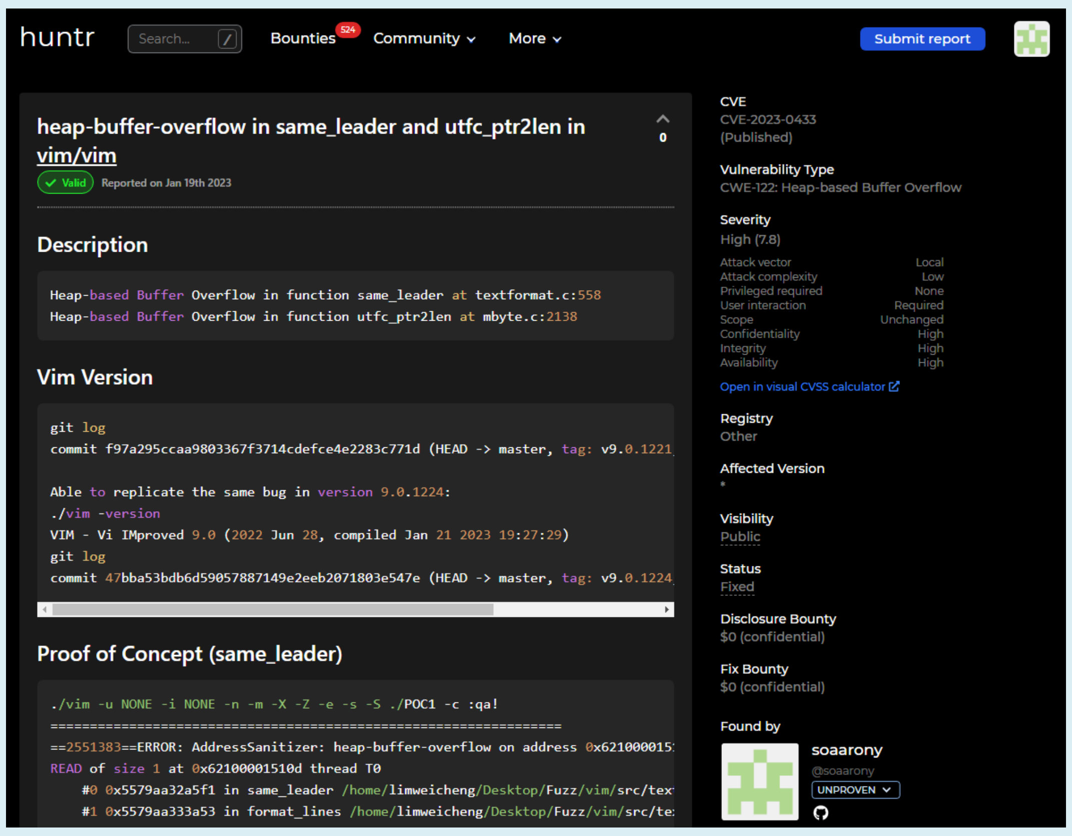
limweicheng@ubuntu: ~/Desktop/Fuzz/vim/src

```
    0[||||||||||||||||||||||||||100.0%]   Tasks: 132, 285 thr; 2 running
    1[|||||||||||||||||||||||||||81.0%]   Load average: 3.46 3.46 3.31
    2[||||||||||||||||||||||||||100.0%]   Uptime: 12 days, 14:41:30
    3[||                          4.2%]
  Mem[||||||||||||||||||||3.87G/7.73G]
  Swp[||||||||||||||         440M/923M]

    PID USER       PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME+  Command
 536030 limweiche   20   0 20.0T  5972  3668 S  4.0  0.1 13:23.62 ./vim -u NONE -
 536028 limweiche   20   0  230M  114M  1892 S  3.2  1.5  7:59.60 afl-fuzz -i ./s
   1868 limweiche   20   0 5925M  132M 72412 S  2.4  1.7  1h39:34 /usr/bin/gnome-
 530885 limweiche   20   0 2556M 1314M 36056 S  2.4 16.6 55:27.45 /usr/bin/nautil
 536004 limweiche   20   0  586M 85736 43192 S  2.4  1.1 32:41.16 /usr/libexec/gn
 538633 limweiche   20   0 20.0T  5008  3756 S  1.6  0.1 11:45.74 ./vim -u NONE -
 530886 limweiche   20   0 2556M 1314M 36056 S  0.8 16.6  4:09.23 /usr/bin/nautil
 538628 limweiche   20   0  231M  114M  2480 S  0.8  1.5  7:35.22 afl-fuzz -i ./s
 540432 limweiche   20   0 20.0T  4460  3644 S  0.8  0.1 13:47.55 ./vim -u NONE -
1652775 limweiche   20   0 11732  5296  3556 R  0.8  0.1  0:00.13 htop
      1 root        20   0  164M 10184  5232 S  0.0  0.1  5:49.61 /sbin/init auto
    381 root        19  -1 56980 12708 11860 S  0.0  0.2  6:56.12 /lib/systemd/sy
    411 root        20   0  148M     0     0 S  0.0  0.0  0:00.02 vmware-vmblock-
    412 root        20   0  148M     0     0 S  0.0  0.0  0:00.01 vmware-vmblock-
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

limweicheng@ubuntu: ~/Desktop/Fuzz/vim/src

```
            american fuzzy lop ++4.05a {fuzzer3} (./vim) [fast]results
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 1 days, 0 hrs, 15 min, 44 sec      │  cycles done : 0     │
│   last new find : 0 days, 0 hrs, 0 min, 9 sec        │ corpus count : 44.0k │
│last saved crash : 0 days, 0 hrs, 17 min, 53 sec      │saved crashes : 88    │
│ last saved hang : 0 days, 22 hrs, 28 min, 17 sec     │  saved hangs : 500+  │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 36.0k.0 (81.9%)    │    map density : 18.45% / 72.87%    │
│  runs timed out : 0 (0.00%)          │ count coverage : 7.05 bits/tuple    │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : trim 32/32             │ favored items : 3310 (7.52%)        │
│ stage execs : 18/21 (85.71%)         │  new edges on : 6640 (15.09%)       │
│ total execs : 2.74M                  │ total crashes : 179 (88 saved)      │
│  exec speed : 12.96/sec (zzzz...)    │  total tmouts : 104k (0+ saved)     │
├─ fuzzing strategy yields ────────────┴───────────┬─ item geometry ────────┤
│   bit flips : disabled (default, enable with -D) │    levels : 7          │
│  byte flips : disabled (default, enable with -D) │   pending : 43.7k      │
│ arithmetics : disabled (default, enable with -D) │  pend fav : 3191       │
│  known ints : disabled (default, enable with -D) │ own finds : 17.1k      │
│  dictionary : n/a                                │  imported : 26.8k      │
│havoc/splice : 7600/1.68M, 9534/664k              │ stability : 67.71%     │
│py/custom/rq : unused, unused, unused, unused     ├────────────────────────┘
│    trim/eff : 9.67%/30.5k, disabled              │          [cpu002:150%]
└──────────────────────────────────────────────────┘
```



heap-buffer-overflow in same_leader and utfc_ptr2len in vim/vim

✓ Valid Reported on Jan 19th 2023

Description

```
Heap-based Buffer Overflow in function same_leader at textformat.c:558
Heap-based Buffer Overflow in function utfc_ptr2len at mbyte.c:2138
```

Vim Version

```
git log
commit f97a295ccaa9803367f3714cdefce4e2283c771d (HEAD -> master, tag: v9.0.1221

Able to replicate the same bug in version 9.0.1224:
./vim -version
VIM - Vi IMproved 9.0 (2022 Jun 28, compiled Jan 21 2023 19:27:29)
git log
commit 47bba53bdb6d59057887149e2eeb2071803e547e (HEAD -> master, tag: v9.0.1224
```

Proof of Concept (same_leader)

```
./vim -u NONE -i NONE -n -m -X -Z -e -s -S ./POC1 -c :qa!
=================================================================
==2551383==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62100015
READ of size 1 at 0x62100001510d thread T0
    #0 0x5579aa32a5f1 in same_leader /home/limweicheng/Desktop/Fuzz/vim/src/tex
    #1 0x5579aa333a53 in format_lines /home/limweicheng/Desktop/Fuzz/vim/src/te
```

CVE
CVE-2023-0433
(Published)

Vulnerability Type
CWE-122: Heap-based Buffer Overflow

Severity
High (7.8)

Attack vector	Local
Attack complexity	Low
Privileged required	None
User interaction	Required
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

Open in visual CVSS calculator

Registry
Other

Affected Version
*

Visibility
Public

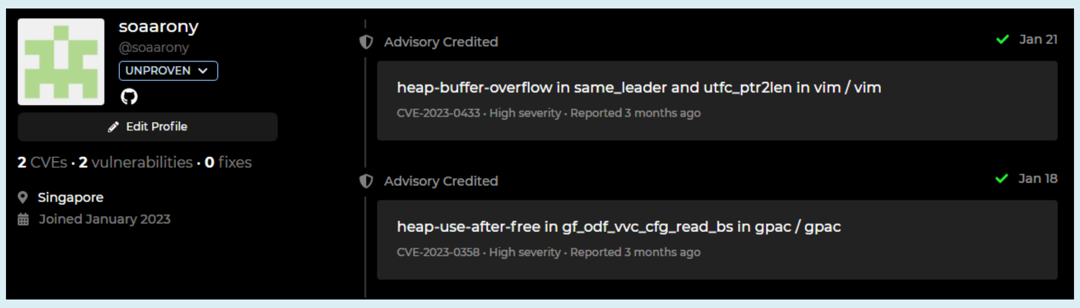
Status
Fixed

Disclosure Bounty
$0 (confidential)

Fix Bounty
$0 (confidential)

Found by
soaarony
@soaarony
UNPROVEN

Bug Reporting on huntr.dev



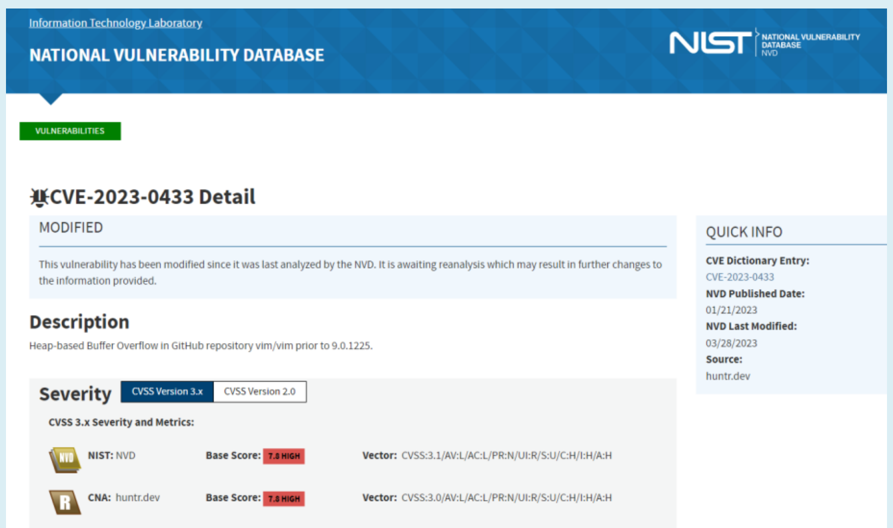
soaarony
@soaarony

2 CVEs · 2 vulnerabilities · 0 fixes

Singapore
Joined January 2023

Advisory Credited ✓ Jan 21
heap-buffer-overflow in same_leader and utfc_ptr2len in vim / vim
CVE-2023-0433 · High severity · Reported 3 months ago

Advisory Credited ✓ Jan 18
heap-use-after-free in gf_odf_vvc_cfg_read_bs in gpac / gpac
CVE-2023-0358 · High severity · Reported 3 months ago


Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE
NIST NATIONAL VULNERABILITY DATABASE NVD
VULNERABILITIES
CVSS Version 3.x
CVSS Version 2.0

CVE-2023-0433 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Description

Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1225.

Severity

CVSS 3.x Severity and Metrics:

NIST: NVD Base Score: 7.8 HIGH Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

CNA: huntr.dev Base Score: 7.8 HIGH Vector: CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

QUICK INFO
CVE Dictionary Entry:
CVE-2023-0433
NVD Published Date:
01/21/2023
NVD Last Modified:
03/28/2023
Source:
huntr.dev

2 CVEs awarded for this FYP Project