IS THE LINUX DESKTOP LESS SECURE THAN WINDOWS 10?

OR HOW SUPER MARIO MUSIC CAN OWN YOUR SYSTEM.

Hanno Böck

https://hboeck.de

This was too easy . It should not be possible to find a serious memory corruption vulnerability in the default Linux desktop attack surface with just a few minutes of looking. Although it’ s hard to say it, this is not the kind of situation that occurs with a latest Windows 10 default install. Is it possible that Linux desktop security has rotted? (Chris Evans)

NINTENDO SOUND FILES (1)

Exploit against Gstreamer in Ubuntu 12.04 (LTS).

Thumbnail parser.

NINTENDO SOUND FILES (2)

NSF players are mini-emulators - the attacker can execute code in an emulator.

Easier to bypass modern exploit mitigation techniques.

FIX

The fix is to delete the affected NSF gstreamer plugin.

No problem: Ubuntu shipped two different NSF player plugins.

FLIC EXPLOIT

AUTOMATIC DOWNLOADS

Some browsers automatically download files to ~/Downloads.

Any webpage can create files on your filesystem.

(Chrome/Chromium, Epiphany, ... - not Linux specific)

TRACKER

GNOME Desktop search tool automatically indexes all new files in a user's home - including ~/Downloads.

REACTION FROM TRACKER DEVELOPER

Furthermore, the GStreamer guys were extremely fast in fixing it. You could claim that other libraries used for metadata extraction are just as insecure, but that'd really be bugs in these libraries to fix. (Carlos Garnacho)

TRACKER PARSERS (1)

Gstreamer, ffmpeg, flac, totem-pl-parser, tiff, libvorbis, taglib, libpng, libexif, giflib, libjpeg-turbo, libosinfo, poppler, libxml2, exempi, libgxps, ghostscript, libitpcdata

TRACKER PARSERS (2)

If you can exploit any of them you can exploit many Linux desktop users from the web without user interaction.

NOT JUST TRACKER

KDE has Baloo.

Thumbnail tools from file managers have similar issues.

PROBLEMS

Automation: Non-interactive downloads and automatic indexing creates a huge attack surface.

Support for a vast variety of file formats by using many libraries of varying quality.

WHAT CAN BE DONE?

SANDBOXING

Isolated parser processes are good targets for sandboxing.

After these events Tracker implemented sandboxing based on libseccomp (KDE/Baloo hasn't yet).

EXPLOIT MITIGATION

Stack Canaries, nonexecutable memory, Address Space Layout Randomization, Code-Flow Integrity.

LINUX AND ASLR (1)

ASLR is one of the strongest exploit mitigation techniques available.

Linux has ASLR support since kernel 2.6.12.

LINUX AND ASLR (2)

Proper ASLR needs position-independent code and executables (-fpic -pie).

Linux distributions have been extremely slow in adopting ASLR.

STATUS ASLR / PIE

Ubuntu: Introduced it in 16.10 (2016)

Fedora: Introduced it in 23 (2015).

Debian: Work in progress (Stretch / 2017).

openSUSE: No (only for few packages).

Gentoo: Only hardened Gentoo.

AND WINDOWS?

Microsoft introduced ASLR in Vista.

Modern Windows already has next-level mitigations like Code-Flow Integrity.

However: Exploit mitigations depend on applications and configuration.

BURN ALL C?

So let's rewrite everything in Rust or other memory safe languages?

Gstreamer already supports plugins written in Rust.

OR CAN WE JUST FIX ALL THE BUGS?

Gstreamer is extremely prone to memory safety bugs - C code, parsers for many different file formats.

Similar cases: ffmpeg, ImageMagick, browsers, wireshark, tcpdump, ...

LET'S DO SOME FUZZING

Most of these bugs can be trivially found with modern coverage-based fuzzing and sanitizing tools.

If they're still there it means nobody is trying to find and report them.

American Fuzzy Lop, LibFuzzer, Address Sanitizer.

FUZZING GSTREAMER

Result: 20 memory safety issues (crashes, invalid memory reads, not necessarily exploitable).

This is quite a bit, but it's doable.

BUT THERE ARE THE DEPENDENCIES...

- libopus, flac, libvpx, libtheora, ffmpeg
- wavpack, game-music-emu, schroedinger, libsidplay, faad, a52dec, libcdio.

FUZZING GSTREAMER CONCLUSION

I think we can fix most of the security bugs in Gstreamer.

Not sure if the same is true for its dependencies.

IS LINUX LESS SECURE THAN WINDOWS?

Automatic indexing of files with a lot of questionable quality parser code.

Does something similar exist in Windows? Not by default, but there's Antivirus software.

BUG IN APPORT (1)

Donncha O'Cearbhaill found a code injection vulnerability in apport, an Ubuntu tool to handle crashes.

No automation, requires user to click on .crash file.

BUG IN APPORT (2)

An exploit dealer company offered the bug finder $ 10.000 for this bug.

There's someone out there who thinks it's worth $ 10.000 to exploit some Ubuntu users.

LINUX DESKTOP SECURITY MATTERS. WE HAVE TO FIX THIS!