USENIX Security '25 Artifact Appendix: Narrowbeer: A Practical Replay Attack Against the Widevine DRM

Florian Roudot
Univ Rennes, CNRS, IRISA

Mohamed Sabt
Univ Rennes, CNRS, IRISA

A Artifact Appendix

A.1 Abstract

Streaming services like Netflix, Prime Video, and HBO Max rely on DRM solutions to ward off piracy. By enabling the distribution of encrypted content, DRM systems prevent subscribed users from downloading the streamed content, as well as unauthorized users from having access to it.

Google Widevine, one of the most deployed DRMs, provides a fully software-based solution on desktop platforms to ensure portability.

Following our findings of new flaws in the Widevine license acquisition process, we developed Narrowbeer, a practical replay attack that allows legitimate users to generate never-expiring licenses and enables unauthorized users to reuse these licenses to access premium content without subscription. As an artifact of our paper, we provide all the necessary components to reproduce the attack on free content.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

Executing the attack does not present any security risks to the user's device. The exploited vulnerability has been responsibly disclosed to Widevine and fixed on the newer versions of the CDM. The licenses provided are extracted from Shaka Player demonstration website[1] and only allow for the playback of free content hosted on Google's servers. The video playback happens in normal conditions and does not represent any risk to Google's servers.

A.2.2 How to access

The artifact is accessible on https://doi.org/10.5281/zenodo.15525617.

A.2.3 Hardware dependencies

Since the licenses were generated on a computer with an x86-64 CPU, they can only be used on systems with the same architecture.

[1] https://shaka-player-demo.appspot.com/

A.2.4 Software dependencies

The attack can be performed on Linux (verified on Debian 12 and Ubuntu 24.04.2) and Windows (verified on Windows 11 Enterprise 24H2), whether real or virtual machines, with software dependencies varying for each system.

On Linux.

- GCC[2] 13.3.0 and Make[3] 4.3 (any recent version should work.)
- Firefox[4] ESR-128 (any recent version should work.)
- Libavcodec[5] from FFmpeg, as some content is distributed using proprietary codecs.

On Windows.

- Visual Studio 2022[6] 17.14.0 with the *Desktop development with C++* kit (any recent version should work.)
- Firefox[4] 131.0.3. The attack does not work with another version as the provided licenses contain a hash of the binary and its signature.

A.2.5 Benchmarks

None.

A.3 Set-up

A.3.1 Installation

On Linux.

1. Install GCC, Make, Firefox ESR and libavcodec using a package manager.
2. Compile the attack executable by running `make` in the `Narrowbeer/Linux/` directory.

[2] https://gcc.gnu.org/
[3] https://www.gnu.org/software/make/
[4] https://www.mozilla.org/firefox/
[5] https://packages.debian.org/bookworm/libavcodec59
[6] https://visualstudio.microsoft.com/

On Windows.

1. Download (https://visualstudio.microsoft.com/downloads/) and install Visual Studio. When asked which additional components to install, select *Desktop development with C++* and continue.
2. Download (https://ftp.mozilla.org/pub/firefox/releases/131.0.3/win64/en-US/) and install Firefox 131.0.3.
3. Double click on `Narrowbeer/Windows/narrowbeer.sln` to open it in Visual Studio then build the solution by going to the *Build* menu and selecting *Build Solution*.

On Firefox.

1. On Linux, type `about:config` in the address bar, then set `dom.ipc.forkserver.enable` to `false`.
2. On Windows, disable the auto-update in Firefox's settings.
3. Make sure Widevine is installed by visiting https://bitmovin.com/demos/drm/ and enabling DRMs if asked.
4. Overwrite the newly installed Widevine CDM with the older vulnerable version by copying the file(s) in `WidevineCDM/<system>/` to Widevine's installation folder `gmp-widevine/<version>/`. This folder can be found by typing `about:profiles` in Firefox's address bar and opening the *Root Directory* of the profile in use.

A.3.2 Basic Test

The easiest way to test that everything is functioning correctly is to perform the first and only experiment **(E1)** of this artifact, described in A.4.2

A.4 Evaluation workflow

A.4.1 Major Claims

Our artifact allows for the reproduction of the Narrowbeer attack on Widevine, achieving three main goals:

(C1): Time Manipulation. We hook Firefox time system/API calls and manage to deceive Widevine into thinking a past time is present.

(C2): Randomness Manipulation. We hook Widevine random system/API calls to fix the random values used in the generation of the license request.

(C3): License Replay. Given a previously received and used license, we can manipulate the Widevine CDM to generate the appropriate license request. This allows an attacker to reuse the license and play protected content without contacting the license server, thus bypassing the need to authenticate on premium streaming services.

A.4.2 Experiments

(E1): [Replay Attack] [A few seconds]: Run the Narrowbeer binary to perform the attack on Widevine and play protected content without contacting the license server.
Preparation: On Linux, configure `ptrace` scope by setting the content of the `/proc/sys/kernel/yama/ptrace_scope` file to `1`. Finally, make sure to close all instances of Firefox.
Execution: On Linux, open a terminal and run:
`./narrowbeer firefox-esr`
On Windows, open a terminal and run:
`./narrowbeer <path/to/firefox.exe>`
On both systems, once Firefox is opened, browse to `file://<artifact_root>/Website/index.html`. Click on *Browse* and select one of the license files available in the `Licenses/<system>/` directory, then click on *Play*.
Results: The video should start playing. If the video doesn't stop after a few seconds, it means that Narrowbeer successfully hooked Firefox **(C1)** and Widevine **(C2)**, thus forcing the CDM to generate a fixed request compatible with the reused license **(C3)**. Please note that Firefox needs to be closed and the Narrowbeer binary ran again to perform the attack another time.
If the video stops after a few seconds, the attack failed. Our script can print valuable information in Firefox's console to help troubleshoot the setup. Moreover, the attack on Windows does not work every time and might need to be rerun a few times.

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.