[Artifact] Vault Raider: Stealthy UI-based Attacks Against Password Managers in Desktop Environments

[Artifact] Vault Raider: Stealthy UI-based Attacks Against Password Managers in Desktop Environments - Andrea Infantino, Mir Masood Ali, Kostas Solomos, Jason Polakis, University of Illinois Chicago / Zenodo.

- Published: date not stated
- Original: <https://zenodo.org/records/16996391>
- Preserved from: https://zenodo.org/records/16996391 (manual-import) on 2026-09-13
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

[Artifact] Vault Raider: Stealthy UI-based Attacks Against Password Managers in Desktop Environments

Authors: Andrea Infantino, Mir Masood Ali, Kostas Solomos and Jason Polakis

Source: https://zenodo.org/records/16996391

Original public artifact record and files created: 2025-08-29. The documentation names the subsequent NDSS 2026 publication.

Original artifact README

Artifacts

This artifact repository accompanies the NDSS 2026 paper, **Vault Raider: Stealthy UI-based Attacks Against Password Managers in Desktop Environments**.

Source Code

The repository includes the source code for each attack and denfese per operating system.:

- `code/Attacks/` — Source code for each variation of the credential-stealing attacks in different environments.
- `code/Defenses/` — Source code for countermeasures introduced in **Section VII** of the paper, focusing on mitigating phishing attacks and strengthening password manager validation mechanisms.

Video Demos

We include demonstration videos of our credential harvesting attacks on various password managers. Each video illustrates how different variations of the attack are executed. The **Stealthy** version shows the attack performed in the background, remaining hidden from the user's view. The **Walkthrough** version showcases the attack in action, highlighting the exploit process in detail.

1Password

macOS -- Credential Harversting

- **Stealthy:** Stealthy Attack
- **Walkthrough:** Walkthrough Attack

macOS -- Harvesting System Password

- **Stealthy:** Stealthy Attack
- **Walkthrough:** Walkthrough Attack

macOS -- CLI Attack

- **Stealthy:** Stealthy Attack
- **Walkthrough:** Walkthrough Attack

Windows

- **Stealthy:** Stealthy Attack
- **Walkthrough:** Walkthrough Attack

Keeper

macOS

- **Stealthy:** Stealthy Attack

Windows

- **Stealthy:** Stealthy Attack
- **Walkthrough:** Walkthrough Attack

LastPass

Windows

- **Stealthy:** Stealthy Attack
- **Walkthrough:** Walkthrough Attack

KeePassXC

macOS

- **Stealthy:** Stealthy Attack
- **Walkthrough:** Walkthrough Attack

Windows

- **Stealthy:** Stealthy Attack
- **Walkthrough:** Walkthrough Attack

MacPass

macOS

- **Stealthy:** Stealthy Attack
- **Walkthrough:** Walkthrough Attack

Citation

If you use artifacts from this repository in your research, please cite our NDSS 2026 publication. You can use the following BibTeX.

@inproceedings{vaultraider2026ndss, author = {Infantino, Andrea and Ali, Mir Masood and Solomos, Kostas and Polakis, Jason}, title = {Vault {Raider}: {Stealthy} {UI}-based {Attacks} {Against} {Password} {Managers} in {Desktop} {Environments}}, booktitle = {Network and {Distributed} {System} {Security} ({NDSS}) {Symposium} 2026}, address = {San Diego, CA, USA}, year = {2026}, pages = {1--18}, }

License

Original documentation: code/Attacks/MacOS/1password-credentials-attack/README.md

1Password Attack

Modes

We have designed our attack so that it can be run in three modes:

- Stealth: the malicious operations are completely hidden from the user sight.
- Semi Showcase: the malicious operations are shown, but not completely.
- Full Showcase: the malicious operations are completely shown.

To switch mode, change the variable named `STEALTH_LEVEL on the top of main.js` file in this way:

- Stealth: `STEALTH_LEVEL = 2`
- Semi Showcase: `STEALTH_LEVEL = 1`
- Full Showcase: `STEALTH_LEVEL = 0`

Target Apps The targeted apps/websites can be changed by modifying the constant `credentialsNoOtp into credentials.js`.

It is treated like a json object, so you can add a new target by adding an object with this structure:

```
{
   target: 'target_name',
   possible_names: ['possible_name_1', 'possible_name_2', ...]
}
```

where `target_name can be any string, and possible_name_x` are the possible names under which the credentials for the target could have been saved into 1Password vault (i.e. the stealing attempts will be made on this array).

Stealing OTPs (One-Time-Passwords) The attack is designed with the possibility of stealing also OTPs from 1Password vault. In order to do it change the value of the variable named `INCLUDE_OTP on the top of main.js to true, and set up the targeted apps/websites by modifying the constant credentialsOtp, instead of credentialsNoOtp (located into the file credentials.js`).

Build Instructions

Follow these steps to build and run the malicious applications:

1. Extract the content of the zip file into a folder.
2. Open a terminal at the folder when you extracted the files.
3. Compile in this way:

```
npm install
npm run make
```

1. Navigate to `/out/1password-attack-xxxx` where *xxxx* is your own architecture.

2. Run the executable `1password-attack.app`