CS-501

Introduction to Malware, Threat Hunting
& Offensive Capabilities Development

Lecture 17: Stealers

Stealers

What do stealers do?

Stealers

What do stealers do?

They steal stuff.

Stealers

They steal stuff.

More formally, stealers in this class will refer to any malware that is

1) Part of a Post Exploitation task
2) Uploads sensitive victim data to a C2

Stealers

Browser data: Cookies, history, passwords

Secrets: password manager databases, ssh keys, RDP profiles, Crypto Wallet Private keys, Certificates...etc

Documents …really anything

Example Stealers:

- Racoon
- Azorult
- Vidar
- Loki
- Redline

Stealers: Browsers

Most browsers (firefox, Chrome, Edge, Opera, Brave...etc) store all of the user data in an encrypted SQLite database

The path for that database is predictable, and can be read

These values include cookies, usernames, passwords, autofill data, history, ...etc

Luckily, enough people complained that the values were stored in plaintext, and most browsers encrypt the values in the database! So we are SOL right?

Windows: Doing weird crypto stuff since 1985

Time permitting, we will discuss the Data Protection API (DAPI) in greater depth, but for now, a few important points:

- Most browsers will use the DPAPI to encrypt symmetric keys used to decrypt passwords
- The DAPI is Token based→ ergo if the browser is used by User “John Doe”, then any processes with the “John Doe”’s token can decrypt values that were encrypted with the DAPI that targeted that user’s token
- In other words, if we trick John into running our program, we can recover his encrypted browser passwords and decrypt them!

How to Loot chrome (old)

The data is stored at C:\Users\User\AppData\Local\Google\Chrome\User Data\default\Login Data

The value itself is encrypted with the DPAPI, and can be easily recovered by calling CryptUnprotectData from a process that has the same token as the logged in user.

To do this, we must parse the SQLite database and decrypt each value

How to Loot Chrome (new)

Chrome stores the secret key that was used to encrypt passwords and other data inside of a configuration file

C:\Users\<User>\AppData\Local\Google\Chrome\User Data\Local State

This is a JSON file that contains the secret key, that is encrypted with the DAPI.

To decrypt the values in the SQLite database, we must

- First parse the encrypted key
- Second, decrypt the key with the DPAPI
- Finally, we decrypt the data inside of the SQLite database

Encryption Libraries

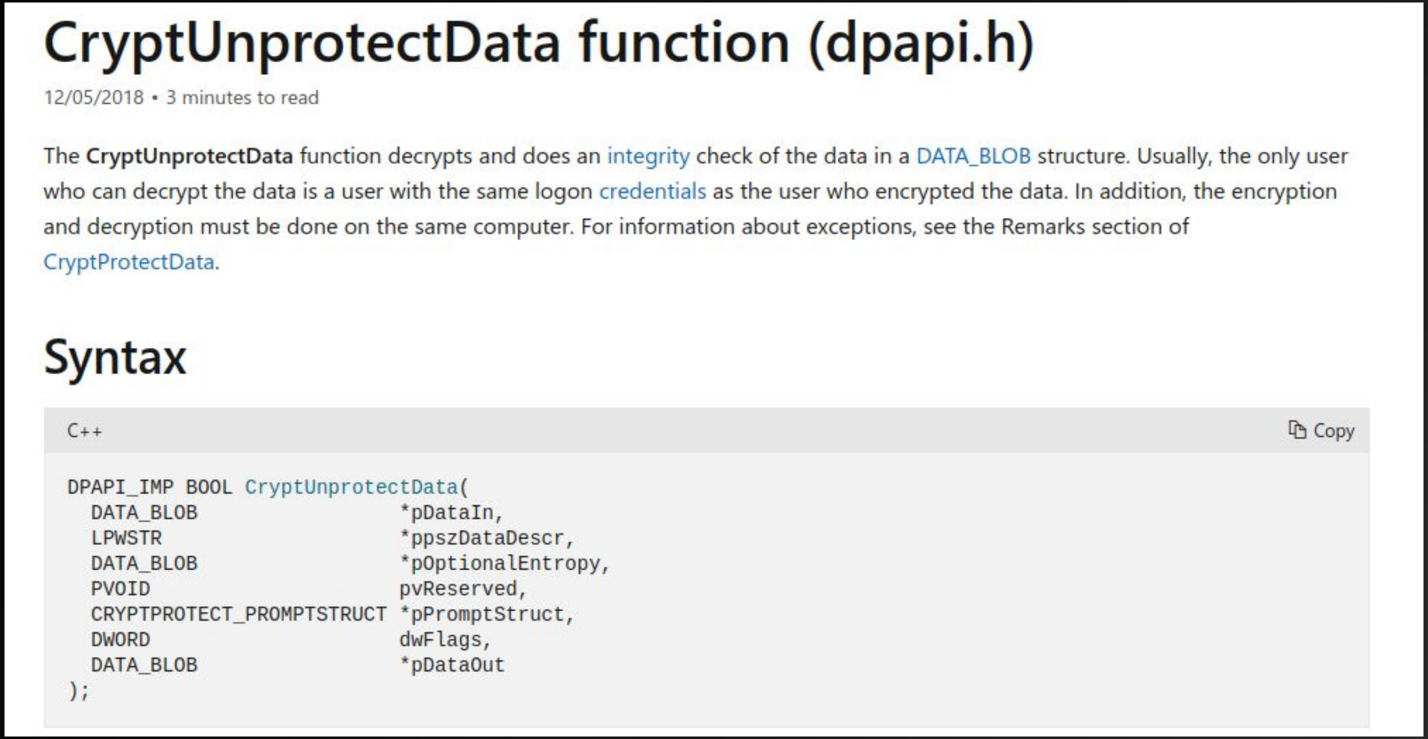
Python: pycryptodome, cryptography (prefered)

C: To name a few: OpenSSL (Pain), Libsodium, NACL, LibHydrogen, Monocyper.

Windows: Crypto API (Sus)

When it comes to crypto, you probably want to use someone else’s implementation. We will talk more about this next week.

Read the Documentation!



CryptUnprotectData function (dpapi.h)

12/05/2018 • 3 minutes to read

The **CryptUnprotectData** function decrypts and does an integrity check of the data in a DATA_BLOB structure. Usually, the only user who can decrypt the data is a user with the same logon credentials as the user who encrypted the data. In addition, the encryption and decryption must be done on the same computer. For information about exceptions, see the Remarks section of CryptProtectData.

Syntax

```
DPAPI_IMP BOOL CryptUnprotectData(
  DATA_BLOB                 *pDataIn,
  LPWSTR                    *ppszDataDescr,
  DATA_BLOB                 *pOptionalEntropy,
  PVOID                     pvReserved,
  CRYPTPROTECT_PROMPTSTRUCT *pPromptStruct,
  DWORD                     dwFlags,
  DATA_BLOB                 *pDataOut
);
```

https://docs.microsoft.com/en-us/windows/win32/api/dpapi/nf-dpapi-cryptunprotectdata

Example: Using Python to Recover Passwords

Demo

Other Useful Files

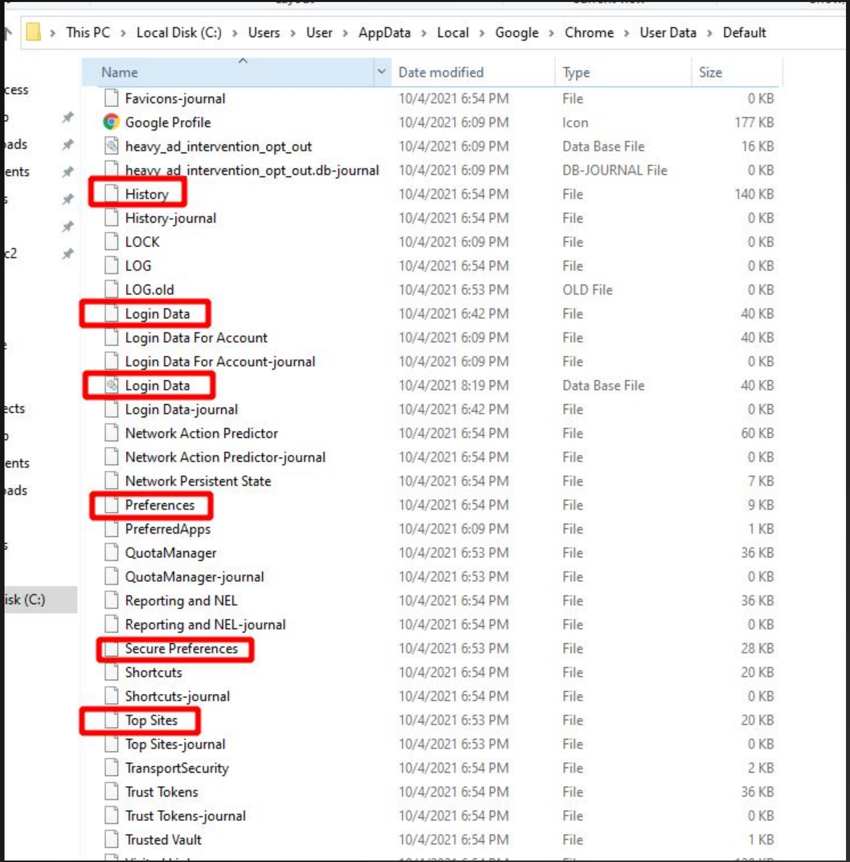
History

Cookies

Top Sites

Preferences

Secure Preferences



> This PC > Local Disk (C:) > Users > User > AppData > Local > Google > Chrome > User Data > Default

Name	Date modified	Type	Size
Favicons-journal	10/4/2021 6:54 PM	File	0 KB
Google Profile	10/4/2021 6:09 PM	Icon	177 KB
heavy_ad_intervention_opt_out	10/4/2021 6:09 PM	Data Base File	16 KB
heavy_ad_intervention_opt_out.db-journal	10/4/2021 6:09 PM	DB-JOURNAL File	0 KB
History	10/4/2021 6:54 PM	File	140 KB
History-journal	10/4/2021 6:54 PM	File	0 KB
LOCK	10/4/2021 6:09 PM	File	0 KB
LOG	10/4/2021 6:54 PM	File	0 KB
LOG.old	10/4/2021 6:53 PM	OLD File	0 KB
Login Data	10/4/2021 6:42 PM	File	40 KB
Login Data For Account	10/4/2021 6:09 PM	File	40 KB
Login Data For Account-journal	10/4/2021 6:09 PM	File	0 KB
Login Data	10/4/2021 8:19 PM	Data Base File	40 KB
Login Data-journal	10/4/2021 6:42 PM	File	0 KB
Network Action Predictor	10/4/2021 6:54 PM	File	60 KB
Network Action Predictor-journal	10/4/2021 6:54 PM	File	0 KB
Network Persistent State	10/4/2021 6:54 PM	File	7 KB
Preferences	10/4/2021 6:54 PM	File	9 KB
PreferredApps	10/4/2021 6:09 PM	File	1 KB
QuotaManager	10/4/2021 6:53 PM	File	36 KB
QuotaManager-journal	10/4/2021 6:53 PM	File	0 KB
Reporting and NEL	10/4/2021 6:54 PM	File	36 KB
Reporting and NEL-journal	10/4/2021 6:54 PM	File	0 KB
Secure Preferences	10/4/2021 6:53 PM	File	28 KB
Shortcuts	10/4/2021 6:54 PM	File	20 KB
Shortcuts-journal	10/4/2021 6:54 PM	File	0 KB
Top Sites	10/4/2021 6:53 PM	File	20 KB
Top Sites-journal	10/4/2021 6:53 PM	File	0 KB
TransportSecurity	10/4/2021 6:54 PM	File	2 KB
Trust Tokens	10/4/2021 6:54 PM	File	36 KB
Trust Tokens-journal	10/4/2021 6:54 PM	File	0 KB
Trusted Vault	10/4/2021 6:54 PM	File	1 KB

Interview Question:

You come across malware that is using a statically linked Sqlite client. What kind of malware is it most likely?

Valid answers:

It is probably a stealer targeting browsers.

More generally, it is doing something that requires interacting with a SQLite database

The most common targets for commodity malware are browsers! Why?

Questions?