KCon 2024

# The Dark Side of the Browser

## ——Unveiling HackBrowserData and a New Approach to Cookie Extraction

演讲人：moonD4rk & SlimWang

时间：2024.08.25

# 目录

CONTENT



KCon 2024

KCon

PART ONE

01

# 浏览器安全概览

# 浏览器是事实上的操作系统

KCon

01

Kcon 2024

## 应用程序入口

- 电子邮件
- 购物
- 银行 / 数字钱包
- 社交媒体
- 游戏
- 文档 / 表格

02

Kcon 2024

## 资源管理和隔离

- 进程管理
- 内存管理
- 权限管理
- 操作系统 API
- 访问硬件

04

Kcon 2024

## 独特安全机制

- Cookie 机制
- 同源策略
- 内容安全策略
- 沙箱环境
- 隔离存储

# 浏览器真的安全吗?

KCon

## MEGA 浏览器恶意扩展导致密码泄漏

2018年9月，MEGA网盘的Chrome扩展被篡改，导致使用此扩展的用户，其亚马逊、谷歌、微软账户面临泄漏风险。

## 恶意 Npm 包窃取浏览器数据

2021年9月，Npm 安全团队发现某恶意 JavaScript 库窃取浏览器的 LevelDB 数据用于数据分析。


## 浏览器历史记录扫描泄漏数亿用户隐私

2021年1月，某软件扫描并读取用户的浏览器历史记录，将数据上传到其服务器，影响数亿用户。

## 浏览器钓鱼插件导致百万美元虚拟币损失

2024年6月，某交易所用户安装了恶意浏览器插件，导致网页Cookies被劫持，损失金额达百万美元。

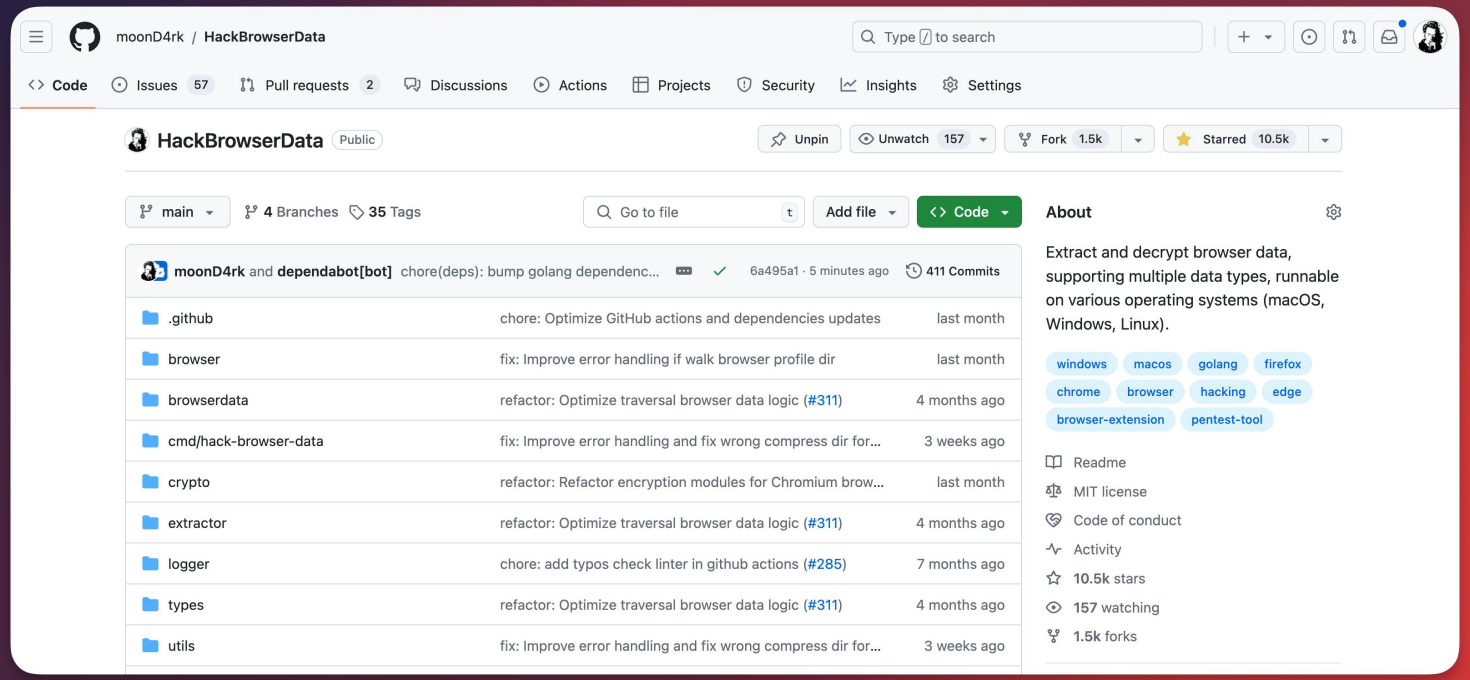
KCon

PART Two

02

# 如何窃取浏览器数据

# HackBrowserData 是什么

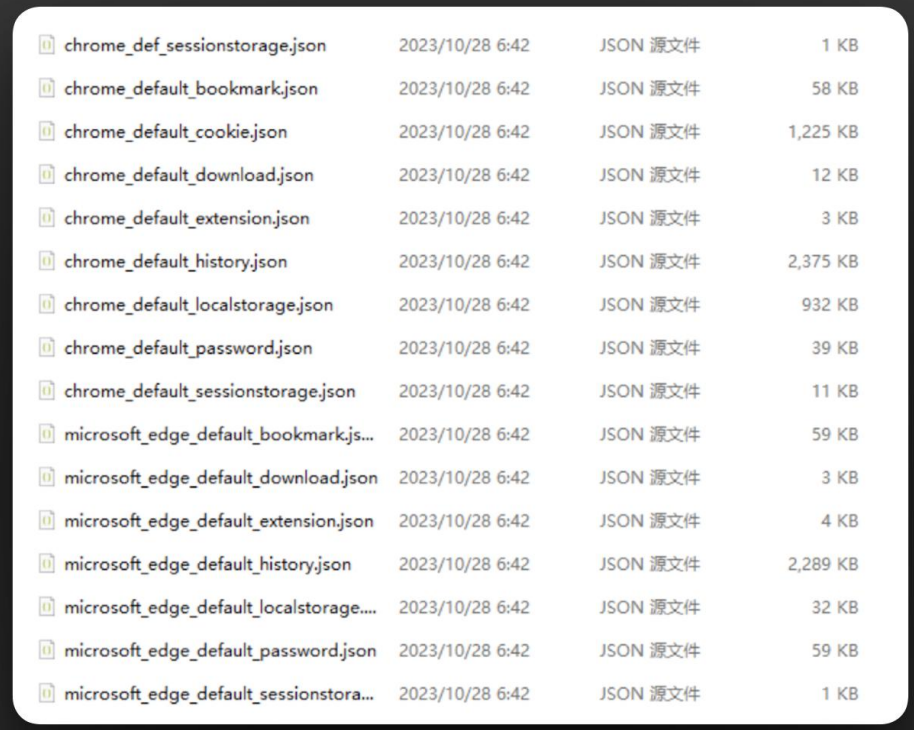



HackBrowserData 是一个浏览器数据（密码|历史记录|Cookie|书签|信用卡|下载记录|localStorage|浏览器插件）解密导出工具，支持全平台主流桌面浏览器。

# HackBrowserData 是什么

```
PS C:\Users\moond4rk\Desktop> .\hack-browser-data.exe -h
NAME:
   hack-browser-data - Export passwords|bookmarks|cookies|history|credit cards|download hist
USAGE:
   [hack-browser-data -b chrome -f json --dir results --zip]
   Export all browsing data (passwords/cookies/history/bookmarks) from browser
   Github Link: https://github.com/moonD4rk/HackBrowserData
VERSION:
   0.4.6

GLOBAL OPTIONS:
   --verbose, --vv                  verbose (default: false)
   --compress, --zip                compress result to zip (default: false)
   --browser value, -b value        available browsers: all|360|brave|chrome|chrome-beta|ch
   --results-dir value, --dir value export dir (default: "results")
   --format value, -f value         output format: csv|json (default: "csv")
   --profile-path value, -p value   custom profile dir path, get with chrome://version
   --full-export, --full            is export full browsing data (default: true)
   --help, -h                       show help
   --version, -v                    print the version
```



| | | | |
|---|---|---|---|
| chrome_def_sessionstorage.json | 2023/10/28 6:42 | JSON 源文件 | 1 KB |
| chrome_default_bookmark.json | 2023/10/28 6:42 | JSON 源文件 | 58 KB |
| chrome_default_cookie.json | 2023/10/28 6:42 | JSON 源文件 | 1,225 KB |
| chrome_default_download.json | 2023/10/28 6:42 | JSON 源文件 | 12 KB |
| chrome_default_extension.json | 2023/10/28 6:42 | JSON 源文件 | 3 KB |
| chrome_default_history.json | 2023/10/28 6:42 | JSON 源文件 | 2,375 KB |
| chrome_default_localstorage.json | 2023/10/28 6:42 | JSON 源文件 | 932 KB |
| chrome_default_password.json | 2023/10/28 6:42 | JSON 源文件 | 39 KB |
| chrome_default_sessionstorage.json | 2023/10/28 6:42 | JSON 源文件 | 11 KB |
| microsoft_edge_default_bookmark.js... | 2023/10/28 6:42 | JSON 源文件 | 59 KB |
| microsoft_edge_default_download.json | 2023/10/28 6:42 | JSON 源文件 | 3 KB |
| microsoft_edge_default_extension.json | 2023/10/28 6:42 | JSON 源文件 | 4 KB |
| microsoft_edge_default_history.json | 2023/10/28 6:42 | JSON 源文件 | 2,289 KB |
| microsoft_edge_default_localstorage.... | 2023/10/28 6:42 | JSON 源文件 | 32 KB |
| microsoft_edge_default_password.json | 2023/10/28 6:42 | JSON 源文件 | 59 KB |
| microsoft_edge_default_sessionstora... | 2023/10/28 6:42 | JSON 源文件 | 1 KB |

只需简单双击或者使用对应命令行工具，可以自动导出并解密存储在浏览器中的敏感数据。
目前在 404 星链项目中 Star 数排行第一 (10.5k stars)

# Chrome 如何存储数据

KCon

## 数据存储在浏览器用户目录下

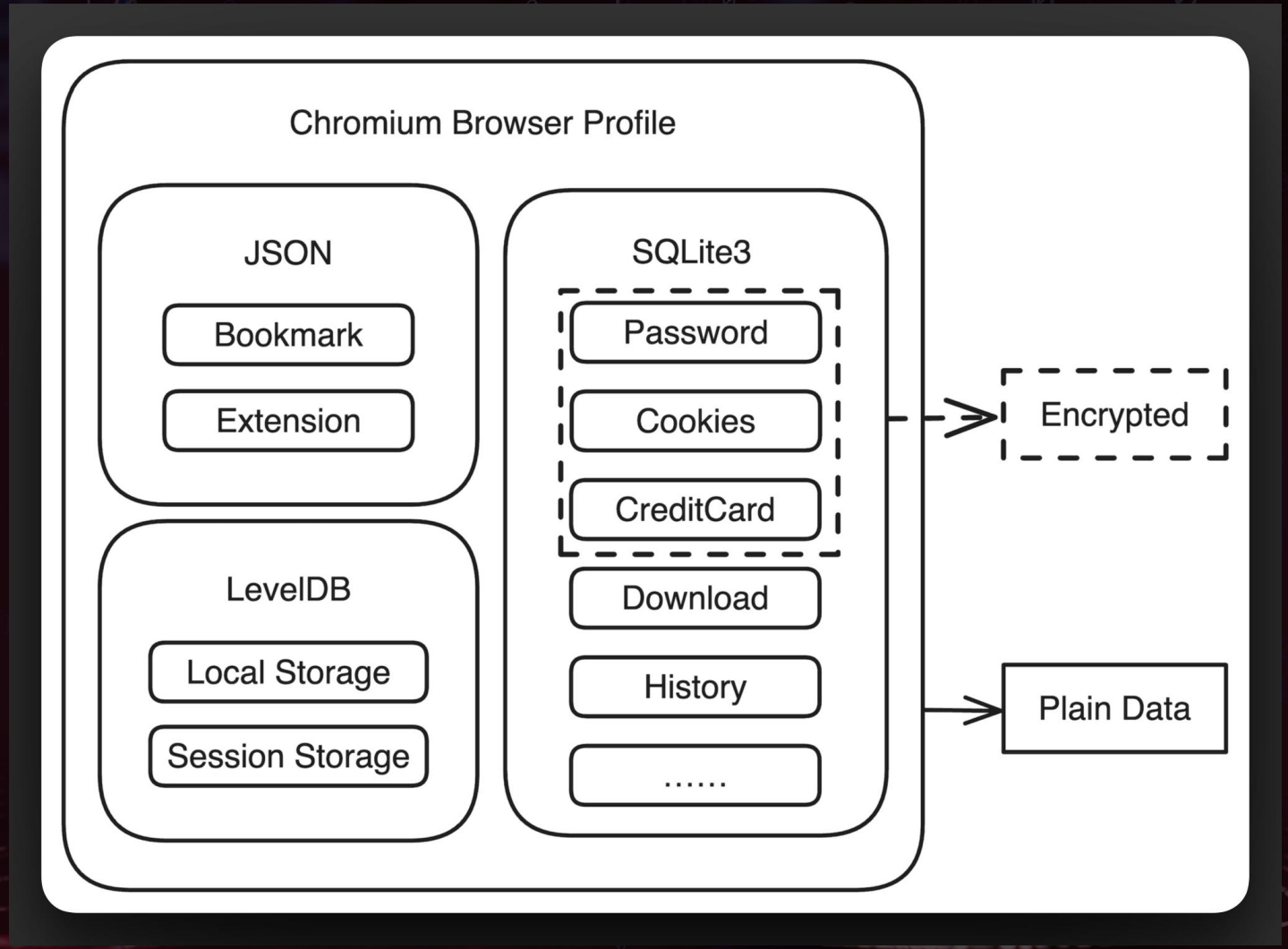
- chrome://version/ 下 Profile Path 目录

## 三类文件存储形式

- SQLite
- JSON
- LevelDB

## 部分数据被加密

- Password
- Cookies
- Credit Card

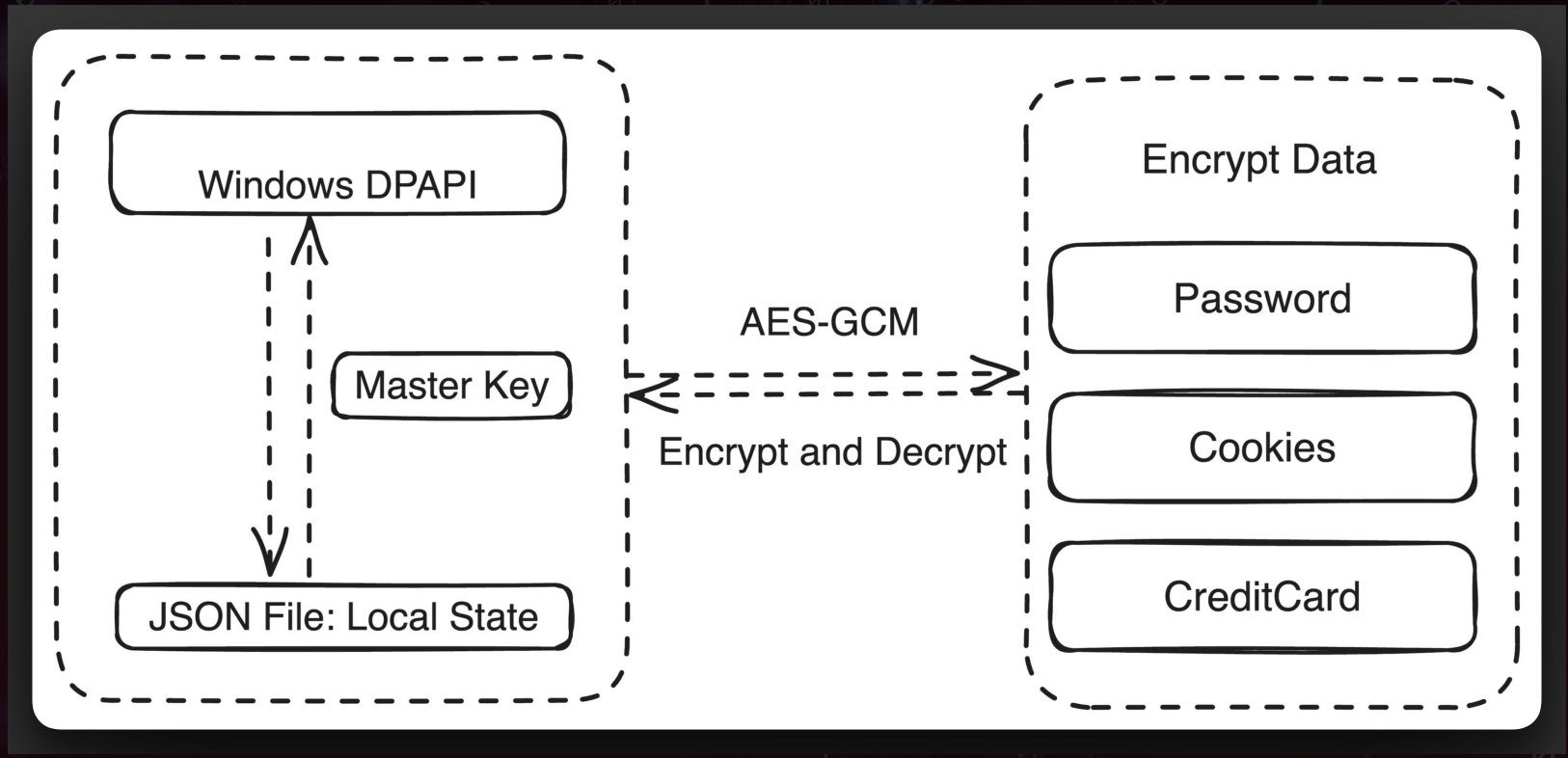


# Chrome 如何加密敏感数据

KCon

## 数据加解密流程

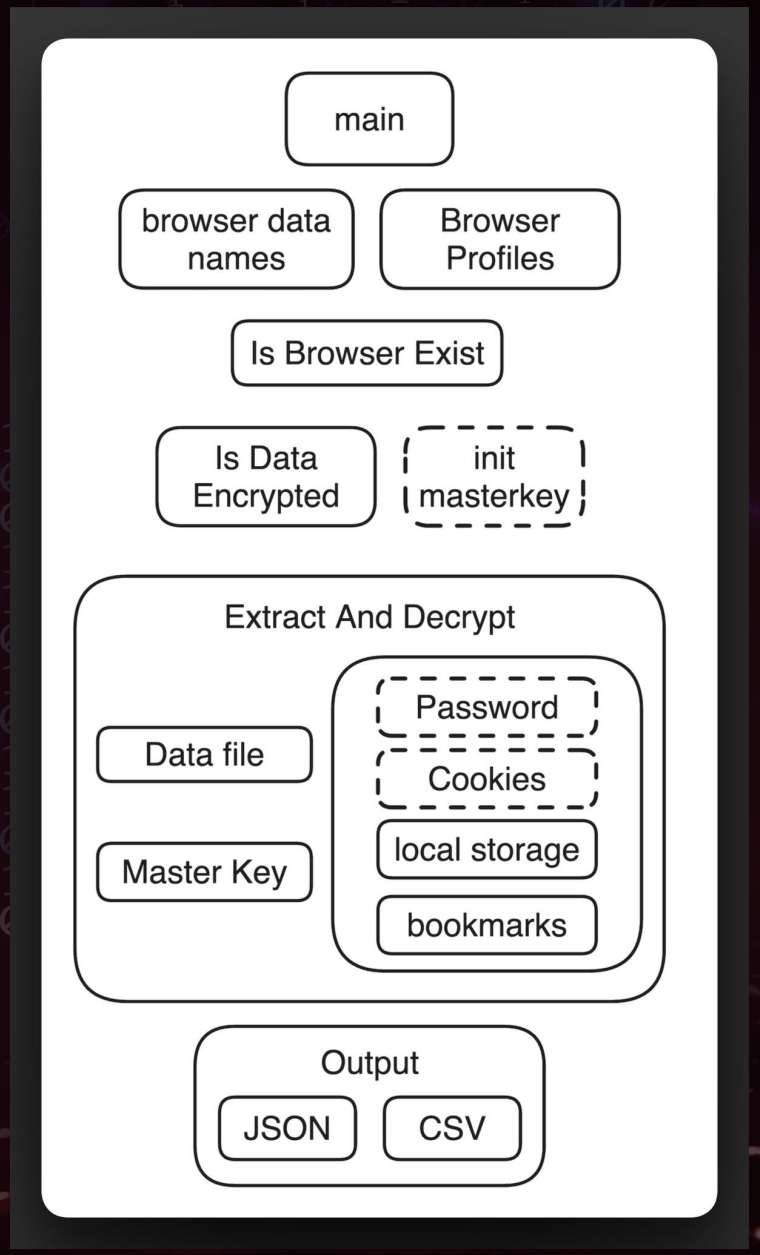
- 调用 Windows DPAPI 生成密钥
- JSON用密钥 MasterKey 对称加密
- 钥存储在本地 Local State 文件



# HackBrowserData 如何窃取数据

## 导出并解密浏览器数据的流程

- 描浏览器目录，判断浏览器是否存在
- 获取数据存储文件列表
- 调用数据 DPAPI 获取 Master Key
- 使用密钥 Master Key 解密加密数据
- 输出明文为 JSON 或 CSV 文件格式



KCon

PART Three

03

# Chromium的安全补丁

# Cookies 的重要性

HTTP Cookie（也称 Web Cookie 或浏览器 Cookie）是服务器发送到用户浏览器并保存在本地的一小块数据。
浏览器会存储 cookie 并在下次向同一服务器再发起请求时携带并发送到服务器上。通常，它用于告知服务端两个请求是否来自同一浏览器——如保持用户的登录状态。

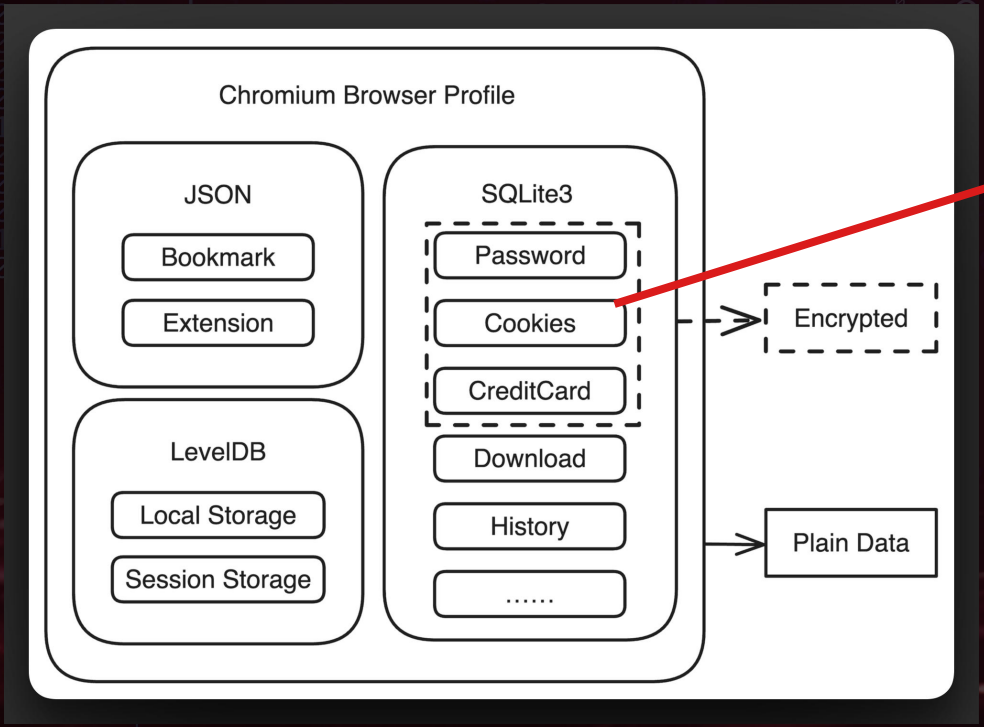
——MDN Web Docs – Mozilla

## 利用Cookies绕过双因子认证（会话劫持）

- HackBrowserData可以窃取浏览器中保存的账号密码
- 通常情况下，利用账号密码即可接管目标用户的各种Web资产，如（网盘、OA、虚拟币应用等）
- 但某些Web资产存在双因子认证（手机短信、Google验证器、微软验证器），此时只凭账号密码无法登录这些资产
- 利用HackBrowserData导出的Cookies可以进行会话劫持，绕过双因子认证，正常登录
- 同时，对于没有保存密码的网站，也可以通过这种方式登录

# Chromium的安全补丁

level=ERROR
source=chromium.go:99
msg="copy item to local error“
path="C:\Users\Username\AppData\\Local\Google\Chrome\User Data\Default\Network\Cookies"
filename=C:\Users\Username\AppData\Local\Temp\Cookies_2.temp
err="open C:\Users\Username\AppData\Local\Google\Chrome\User Data\Default\Network\Cookies: The process cannot access the file because it is being used by another process."




Sqlite3 数据库文件 Cookies 无法访问

无法提取基于新的 Chromium 内核的浏览器 Cookies

如：Google Chrome、Microsoft Edge、Brave等

# Chromium的安全补丁

```go
// CopyFile copies the file from the source to the destination
func CopyFile(src, dst string) error {
    s, err := os.ReadFile(src)
    if err != nil {
        return err
    }
    err = os.WriteFile(dst, s, 0o600)
    if err != nil {
        return err
    }
    return nil
}
```

需要找到新的方式来读取该文件：

C:\Users\Username\AppData\Local\Google\Chrome\User Data\Default\Network\Cookies

KCon

PART Three

04

# 绕过该补丁提取Cookies

# Copy Locked Files

## Volume Shadow Copy Service (VSS)

- VSS 是微软Windows操作系统中的一个功能，它允许创建磁盘的快照副本。VSS可以用来备份和恢复系统或应用程序的数据，即使在数据正在被使用或被修改的情况下。
- 要求文件系统必须为NTFS，噪音大，需要管理员权限

VS

## NTFS RawCopy

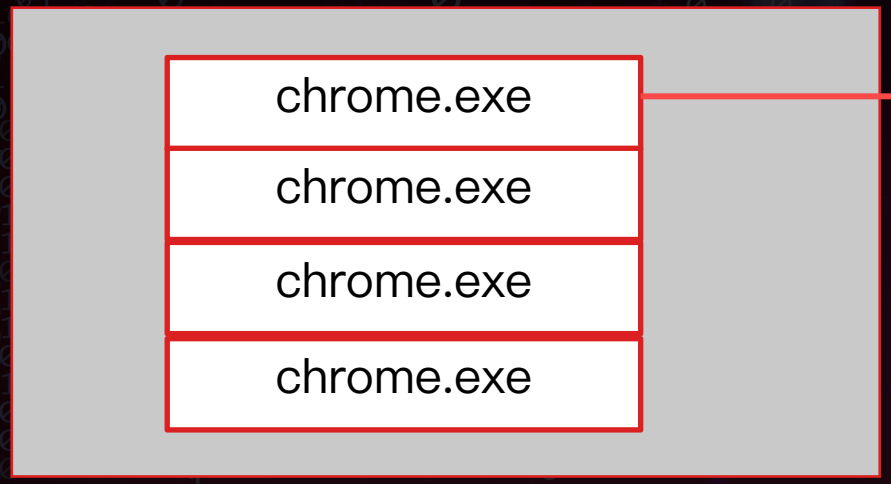
- 直接从磁盘读取文件
- 要求文件系统必须为NTFS
- 需要管理员权限

VS

## DuplicateHandle

- 通过复制文件句柄方式直接读取文件内容
- 通过文件映射方式间接读取文件内容
- 只需要普通用户权限

# DuplicateHandle 基本原理




以“--utility-sub-type=network.mojom.NetworkService” 参数启动，为网络服务相关进程，该进程拥有 Cookies文件的句柄

该文件已被 chrome.exe 进程打开，无法再被其它进程打开（无法通过此方式获取该文件的句柄）

| Type | Name | Handle |
|---|---|---|
| File | C:\Users\ \AppData\Local\Google\Chrome\User Data\Default\Shared Dictionary\db | 0x414 |
| File | C:\Users\ \AppData\Local\Google\Chrome\User Data\Default\Cache\Cache_Data\index | 0x430 |
| File | C:\Users\ \AppData\Local\Google\Chrome\User Data\Default\Cache\Cache_Data\index | 0x434 |
| File | C:\Users\ \AppData\Local\Google\Chrome\User Data\Default\Cache\Cache_Data\data_1 | 0x43c |
| File | C:\Users\ \AppData\Local\Google\Chrome\User Data\Default\Cache\Cache_Data\data_2 | 0x444 |
| File | C:\Users\ \AppData\Local\Google\Chrome\User Data\Default\Cache\Cache_Data\data_2 | 0x448 |
| File | C:\Users\ \AppData\Local\Google\Chrome\User Data\Default\Cache\Cache_Data\data_3 | 0x450 |
| File | C:\Users\ \AppData\Local\Google\Chrome\User Data\Default\Cache\Cache_Data\data_3 | 0x454 |
| File | C:\Users\ \AppData\Local\Google\Chrome\User Data\Default\Network\Cookies | 0x460 |
| File | \Device\KsecDD | 0x464 |
| File | \Device\NamedPipe\mojo.15396.28224.5009487067500727918 | 0x494 |
| File | \Device\Afd | 0x5a8 |
| File | \Device\Afd | 0x5ac |

# DuplicateHandle

A. 找到进程名为“chrome.exe”且参数中含有“--utility-sub-type=network.mojom.NetworkService”的进程 Pid

B. 使用 OpenProcess 函数以 PROCESS_DUP_HANDLE 权限打开该进程，获取目标进程句柄

C. 使用 NtQuerySystemInformation 函数获取操作系统的 SystemHandleInformation 信息，存入 handles

D. 遍历所有 handles：

找到 UniqueProcessId == Pid 的句柄

以 DUPLICATE_SAME_ACCESS 权限复制该句柄到当前进程

使用 GetFileType 函数判断其是否为 FILE_TYPE_DISK

使用 GetFinalPathNameByHandle 函数判断其文件名是否以`\Network\Cookies`结尾

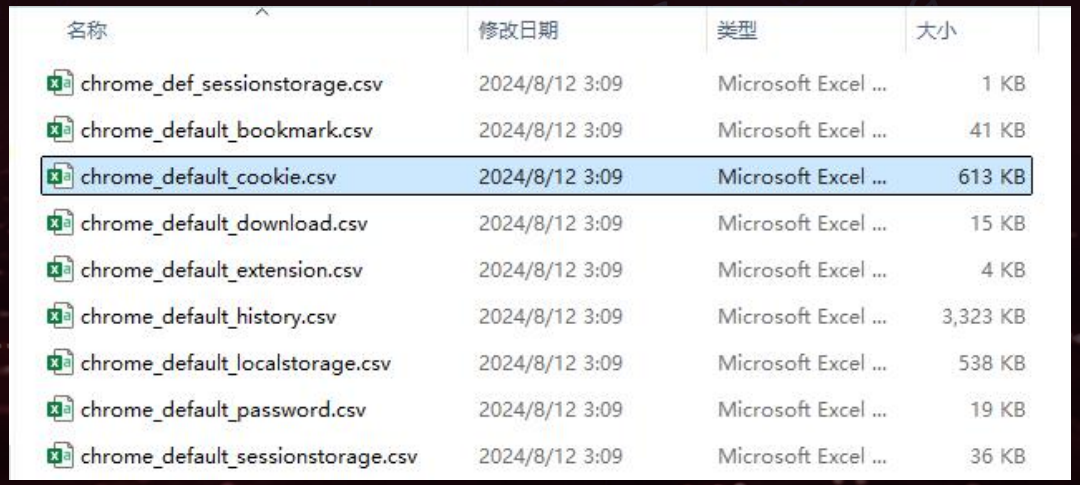
使用 GetFileInformationByHandle 函数获取其文件信息，解析文件大小 fileSize

创建 fileSize 大小的缓冲区，并使用 ReadFile 函数以异步方式读取文件内容（需传入 Overlapped 参数）

将读取到的内容写入新文件

# DuplicateHandle 结果

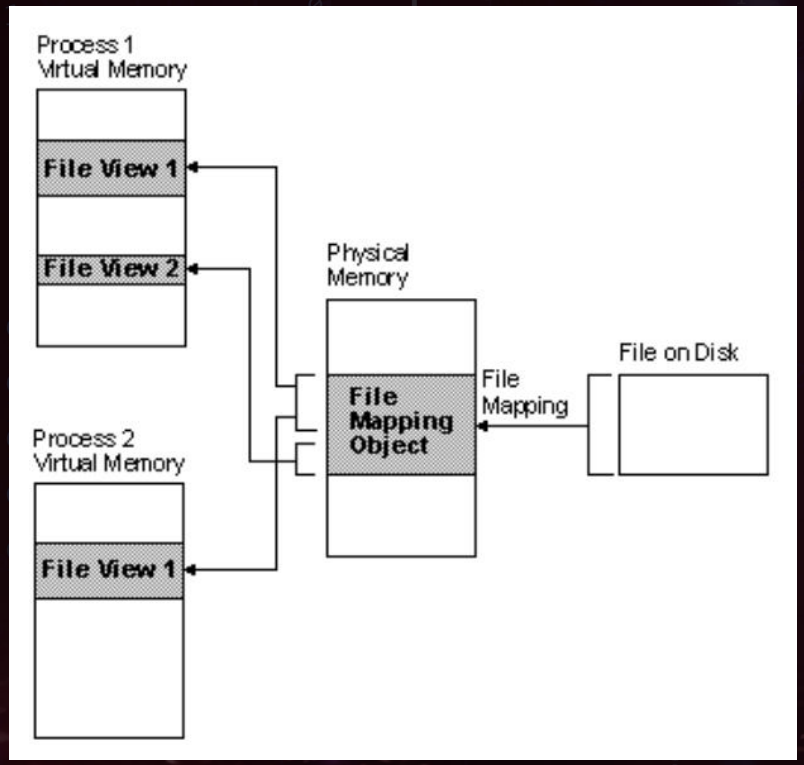
```
go build -trimpath -ldflags="-s -w" -o test/hack-browser-data.exe ./cmd/hack-browser-data/main.go
Test build completed. Output: test/hack-browser-data.exe
cd test && ./hack-browser-data.exe -b chrome -f csv
level=WARN source=browser.go:73 msg="find browser success" browser=chrome_default
level=WARN source=browser.go:73 msg="find browser success" browser=chrome_allowcookies
level=WARN source=browser.go:73 msg="find browser success" browser=chrome_def
 CopyWinLockedFile
FileSize:  1376256
BytesRead:  1376256
File copied successfully!
level=WARN source=browserdata.go:56 msg="export success" filename=chrome_default_download.csv
level=WARN source=browserdata.go:56 msg="export success" filename=chrome_default_localstorage.csv
level=WARN source=browserdata.go:56 msg="export success" filename=chrome_default_password.csv
level=WARN source=browserdata.go:56 msg="export success" filename=chrome_default_cookie.csv
level=WARN source=browserdata.go:56 msg="export success" filename=chrome_default_history.csv
level=WARN source=browserdata.go:56 msg="export success" filename=chrome_default_sessionstorage.csv
level=WARN source=browserdata.go:56 msg="export success" filename=chrome_default_bookmark.csv
level=WARN source=browserdata.go:56 msg="export success" filename=chrome_default_extension.csv
level=WARN source=browserdata.go:56 msg="export success" filename=chrome_def_sessionstorage.csv
Test completed.
```



| 名称 | 修改日期 | 类型 | 大小 |
|---|---|---|---|
| chrome_def_sessionstorage.csv | 2024/8/12 3:09 | Microsoft Excel ... | 1 KB |
| chrome_default_bookmark.csv | 2024/8/12 3:09 | Microsoft Excel ... | 41 KB |
| chrome_default_cookie.csv | 2024/8/12 3:09 | Microsoft Excel ... | 613 KB |
| chrome_default_download.csv | 2024/8/12 3:09 | Microsoft Excel ... | 15 KB |
| chrome_default_extension.csv | 2024/8/12 3:09 | Microsoft Excel ... | 4 KB |
| chrome_default_history.csv | 2024/8/12 3:09 | Microsoft Excel ... | 3,323 KB |
| chrome_default_localstorage.csv | 2024/8/12 3:09 | Microsoft Excel ... | 538 KB |
| chrome_default_password.csv | 2024/8/12 3:09 | Microsoft Excel ... | 19 KB |
| chrome_default_sessionstorage.csv | 2024/8/12 3:09 | Microsoft Excel ... | 36 KB |

# FileMapping

## 某些版本的浏览器中，无法 通过 ReadFile 方式直接读取文件内容




可以考虑使用创建文件映射的方式间接读取文件内容（只需要普通用户权限）

文件映射对象允许一个或多个进程将文件的内容映射到内存地址空间，从而可以像访问内存一样访问文件内容。

- CreateFileMapping 函数创建一个文件映射对象，该对象表示整个文件或文件的一部分在虚拟内存中的映射。文件映射对象可以被同一进程的多个线程或不同进程中的多个线程共享。
- MapViewOfFile 函数用于将文件映射对象的一个部分或全部映射到调用进程的地址空间中，使进程可以像操作内存一样直接访问文件内容。这个函数通常与 CreateFileMapping 一起使用，来实现高效的文件 I/O 操作或进程间通信。

# FileMapping

A. ...复制文件句柄

B. 使用 CreateFileMapping 函数创建文件映射

C. 使用 MapViewOfFile 函数创建文件视图，返回的内存指针存入 buffer

D. 使用 WriteFile 函数将该 buffer 写入新文件

E. 成功读取 Cookies 文件内容

TONGDAO

KCon 2024

# KCon 2024

# THANKS

演讲人：moonD4rk & SlimWang

时间：2024.08.25