

# Pragmatic Web Security

Security training for developers



## SECURITY PATTERNS FOR KEEPING SECRETS IN THE BROWSER

# DR. PHILIPPE DE RYCK

- Deep understanding of the web security landscape
- Google Developer Expert (not employed by Google)
- Author of the *primer on client-side web security*
- Course curator of the  **SecAppDev** course  
(<https://secappdev.org>)



@PHILIPPEDERYCK

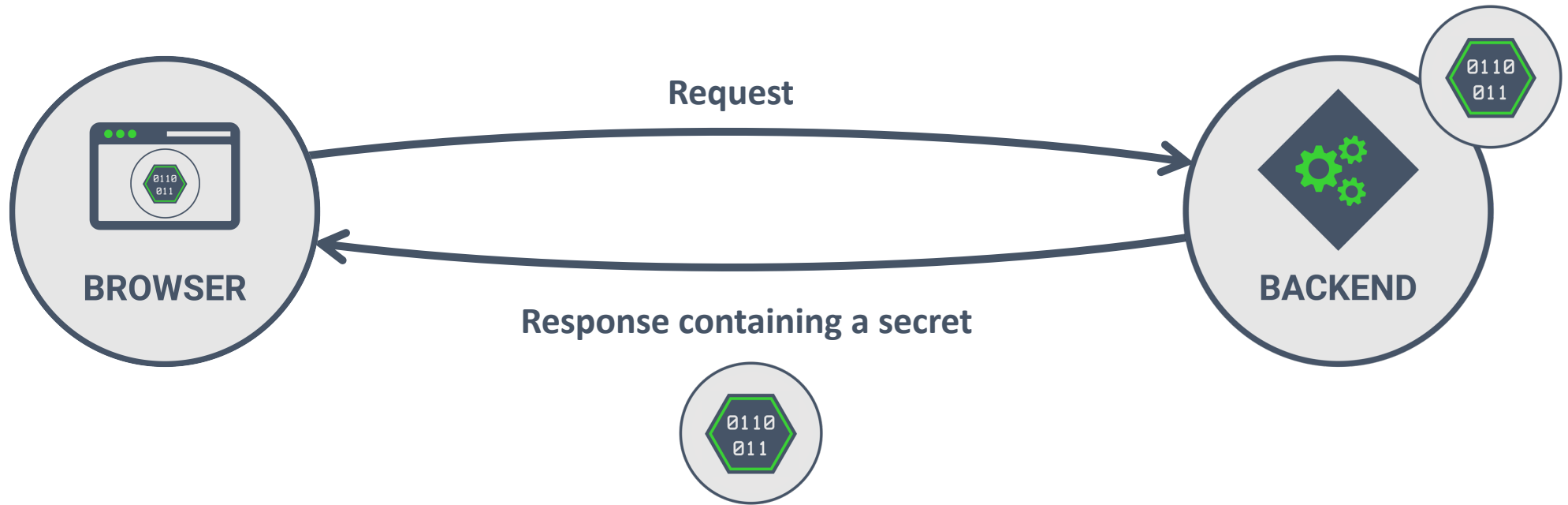
[HTTPS://PRAGMATICWEBSECURITY.COM](https://pragmaticwebsecurity.com)



## Pragmatic Web Security

High-quality security training for developers and managers

Custom courses covering web security, API security, Angular security, ...



JavaScript libraries  
and services

User-provided  
content



3rd party components



# The Ticketmaster breach – what happened and what to do

28 JUN 2018

4



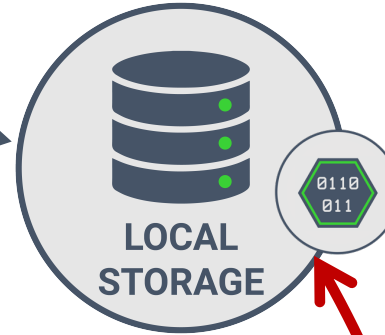
## Hacking Fortnite Accounts

January 16, 2019

**Research by:** Alon Boxiner, Eran Vaknin and Oded Vanunu



(https, restograde.com, 443)



# LOCALSTORAGE



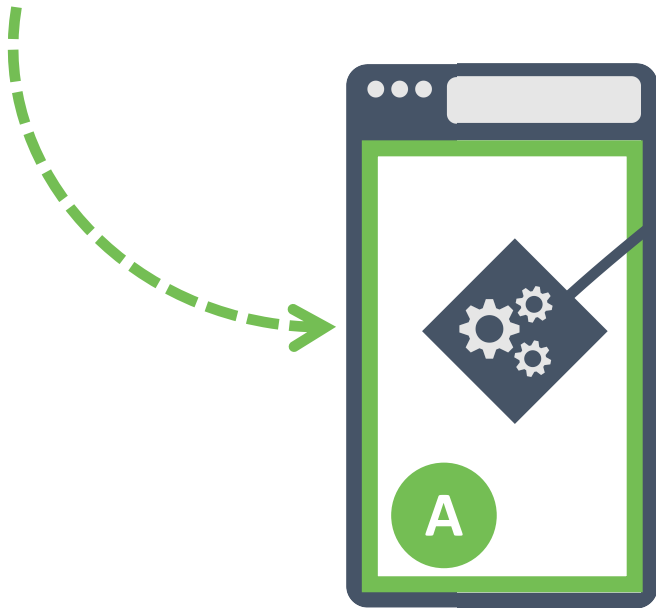
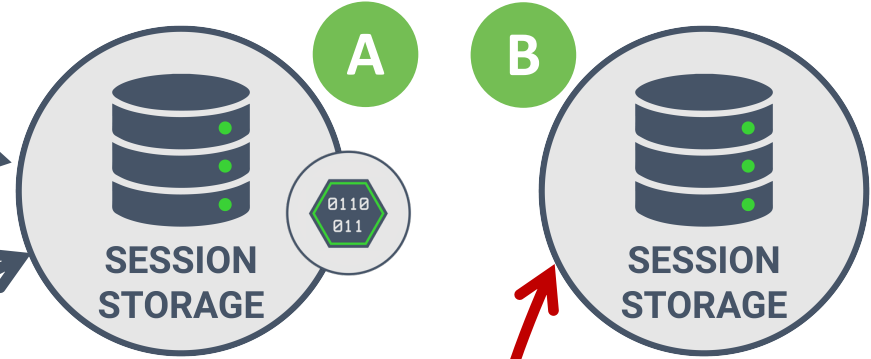
*LocalStorage is accessible for every context in the origin*

*Useful to store long-term non-sensitive data*





(https, restograde.com, 443)



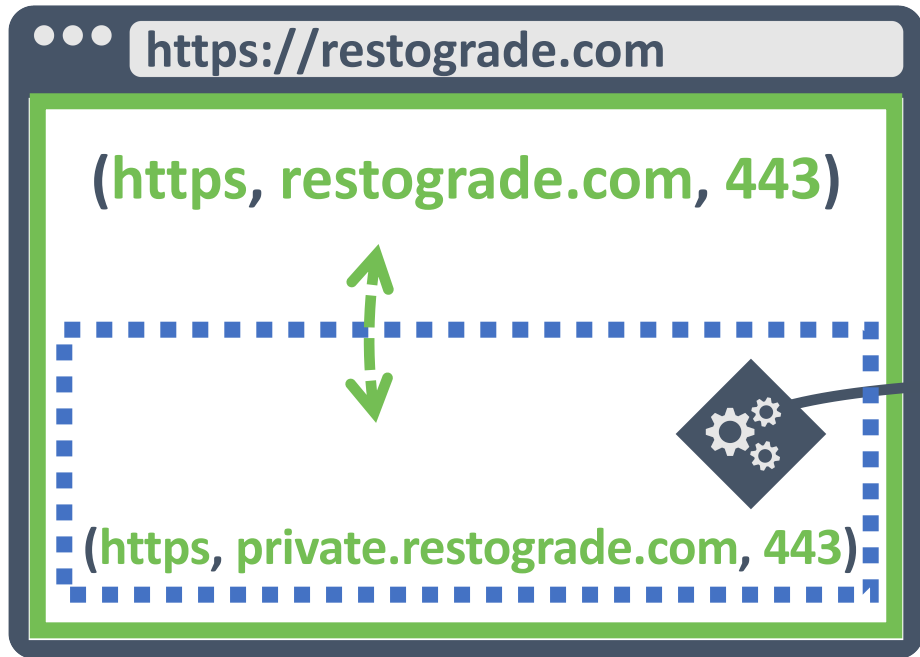
# SESSIONSTORAGE



*SessionStorage is accessible to a set of browsing contexts within an origin*

*Useful to store client-accessible data during a “session”*

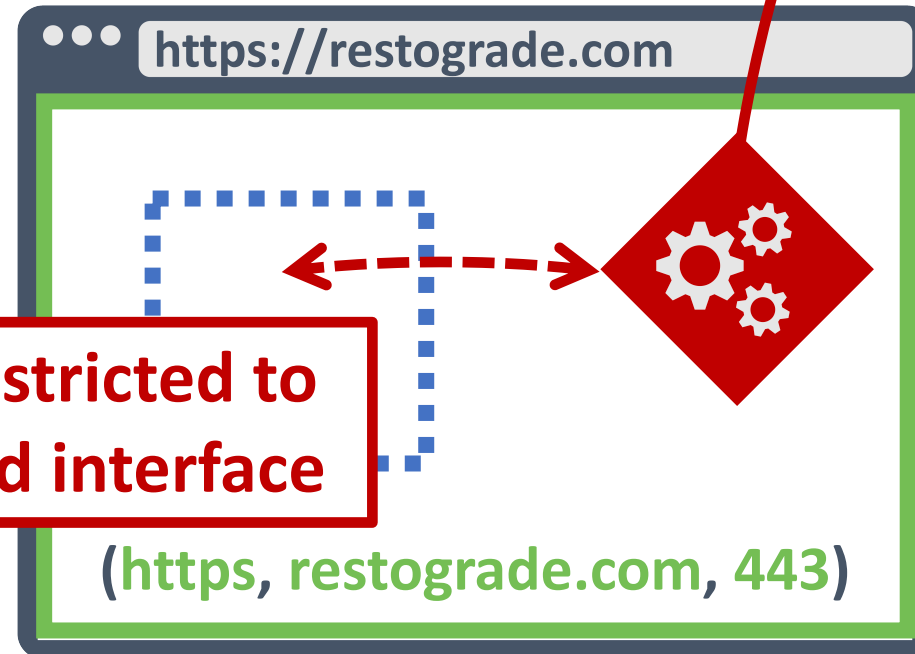




`(https, private.restograde.com, 443)`



`(https, restograde.com, 443)`



**Abuse is restricted to  
the exposed interface**

# ISOLATING LOCALSTORAGE PER ORIGIN



*Origin-based isolation limits access to LocalStorage*  
*Exposed interface can implement access control rules*

*Useful to store sensitive data*



## KEY TAKEAWAY

# ORIGIN-BASED ISOLATION IS A STRONG ISOLATION MECHANISM

- 🛡️ *Create an origin without third-party code or user data*
- 🛡️ *Expose a limited set of operations (e.g., read-only)*
- 🛡️ *Enforce an origin-based access control policy on operations*



```
philippe@substitute p71kgx6k.default $ sqlite3 webappsstore.sqlite
SQLite version 3.24.0 2018-06-04 14:10:15
Enter ".help" for usage hints.
sqlite> .tables
webappsstore2
sqlite> .schema
CREATE TABLE webappsstore2 (originAttributes TEXT, originKey TEXT, scope TEXT, key TEXT, value TEXT)
;
CREATE UNIQUE INDEX origin_key_index ON webappsstore2(originAttributes, originKey, key);
sqlite> select * from webappsstore2 ;
|moc.ebutuoy.www.:https:443|moc.ebutuoy.www.:https:443|yt-remote-connected-devices|{"data":"[]","exp
iration":1550764746173,"creation":1550678346173}
|moc.ebutuoy.www.:https:443|moc.ebutuoy.www.:https:443|yt-remote-device-id|{"data":"ff4e4b35-d3c0-41
2a-8c66-e4b4488f34c4","expiration":1582214346152,"creation":1550678346152}
|moc.ytirucesbewcitamgarp.:https:443|moc.ytirucesbewcitamgarp.:https:443|secret|treasure!
sqlite>
```



Filter output

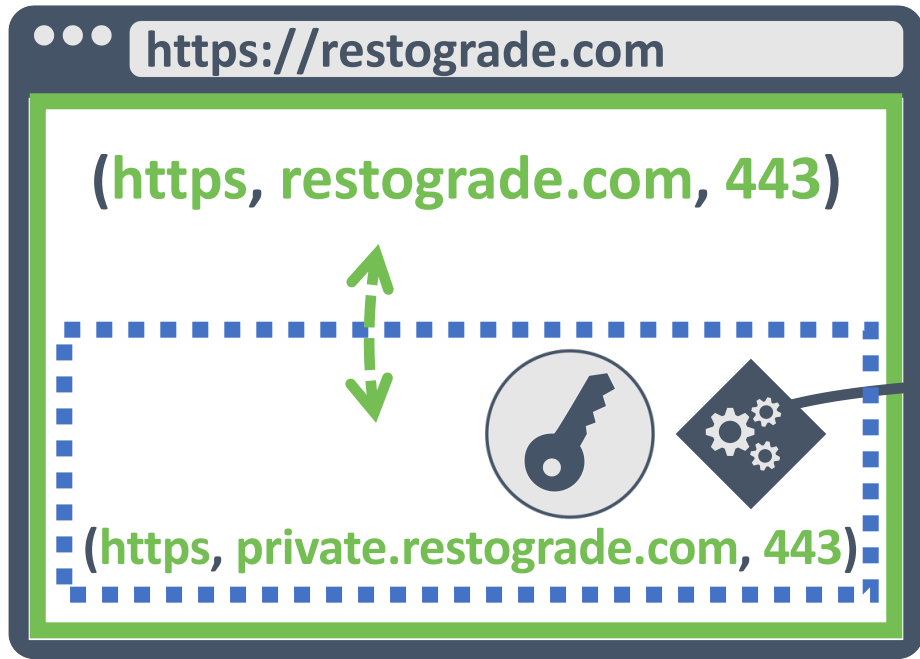
☐ Persist Logs

&gt;&gt; localStorage.setItem("secret", "treasure!")

&lt;- undefined

&gt;&gt;

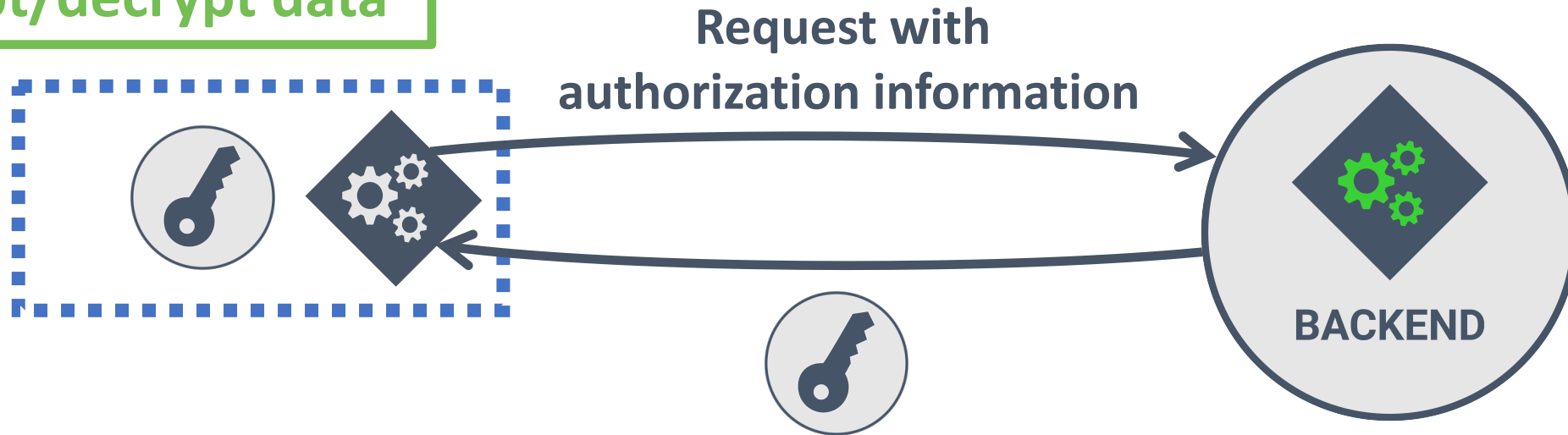




`(https, private.restograde.com, 443)`



Use WebCrypto to  
load the key and  
encrypt/decrypt data



Retrieve user-specific  
key from backend

# ENCRYPTION WITH A SERVER-PROVIDED KEY

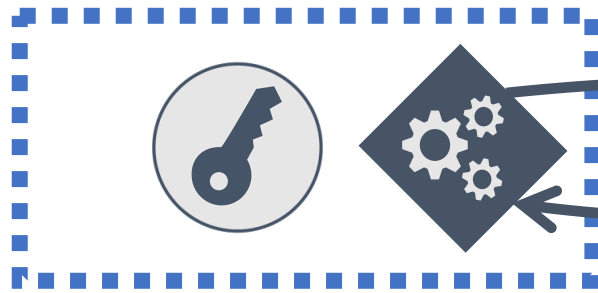


*The server provides a user/device-specific key*

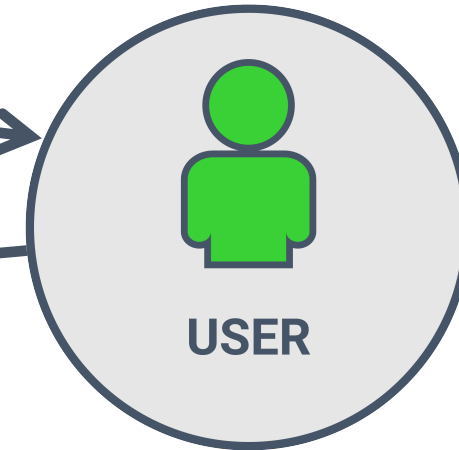
*Useful to handle encrypted data in online scenarios*



Use WebCrypto to transform password into a key



Request password



\*\*\*\*\*

Use WebCrypto for encrypting/decrypting data

Request the application password from the user

# ENCRYPTION WITH A PASSWORD-DERIVED KEY



*The user provides an application password, which is transformed into an encryption/decryption key*

*Useful to handle encrypted data in offline scenarios*





## THE BROWSER IS NO PLACE FOR SECRETS

*Avoid storing sensitive data in the browser if you can*



## STORE SECRETS IN A DEDICATED ORIGIN

*Origin-based isolation is a strong browser security mechanism*



## CONSIDER ENCRYPTING SENSITIVE DATA

*Browser data is easily accessible from outside the browser*




Security patterns for keeping s x +

← → ↻ https://pragmaticwebsecurity.com/talks/browsersecrets 🔍 📄 ☰

**Pragmatic Web Security** Security training The instructor Courses & Talks Testimonials Contact

This page contains the resources for the talk titled "Security patterns for keeping secrets in the browser". Next to the slides, a cheat sheet provides an overview of the different patterns, their pros and cons. The GitHub repository contains code examples for each of the patterns.

**Pragmatic Web Security**  
Security training for developers



**SECURITY PATTERNS FOR  
KEEPING SECRETS IN THE BROWSER**

**PHILIPPE DE RYCK** @PhilippeDeRyck - philippe@PragmaticWebSecurity.com

[DOWNLOAD SLIDES](#) [GET THE CHEAT SHEET](#) [REVIEW THE CODE](#)

<https://pragmaticwebsecurity.com/talks/browsersecrets>



# Pragmatic Web Security

Security training for developers



**`/in/PhilippeDeRyck`**



**`@PhilippeDeRyck`**

**`philippe@pragmaticwebsecurity.com`**