

Security in
knowledge

DID YOU READ THE
NEWS?

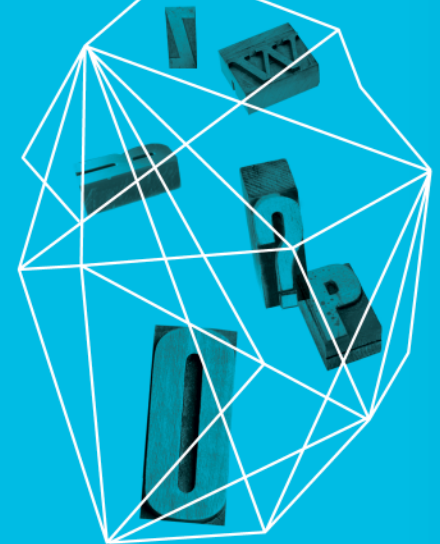
HTTP REQUEST HIJACKING

Yair Amit (@YairAmit)

CTO & co-founder, Skycure

Adi Sharabani (@AdiSharabani)

CEO & co-founder, Skycure



RSACONFERENCE
EUROPE 2013

© 2013 Skycure

Session ID: HTA-T07

Session Classification:

About the Presenters

▶ Yair Amit

- ▶ CTO & co-founder of Skycure
- ▶ Web, network and mobile researcher
- ▶ Inventor of 15+ patents
- ▶ Former manager of the Application Security & Research group at IBM

▶ Adi Sharabani

- ▶ CEO & co-founder of Skycure
- ▶ Watchfire's research group [Acquired by IBM]
- ▶ Lead the security of IBM software
- ▶ Teacher at Ohel Shem high-school

— Agenda

- ▶ Background
- ▶ The Skycure Journal
- ▶ HTTP Request Hijacking
 - ▶ Demonstration
- ▶ Impact
- ▶ Extensions to the attack
 - ▶ Malicious Profiles
 - ▶ Captive Networks
- ▶ Remediation

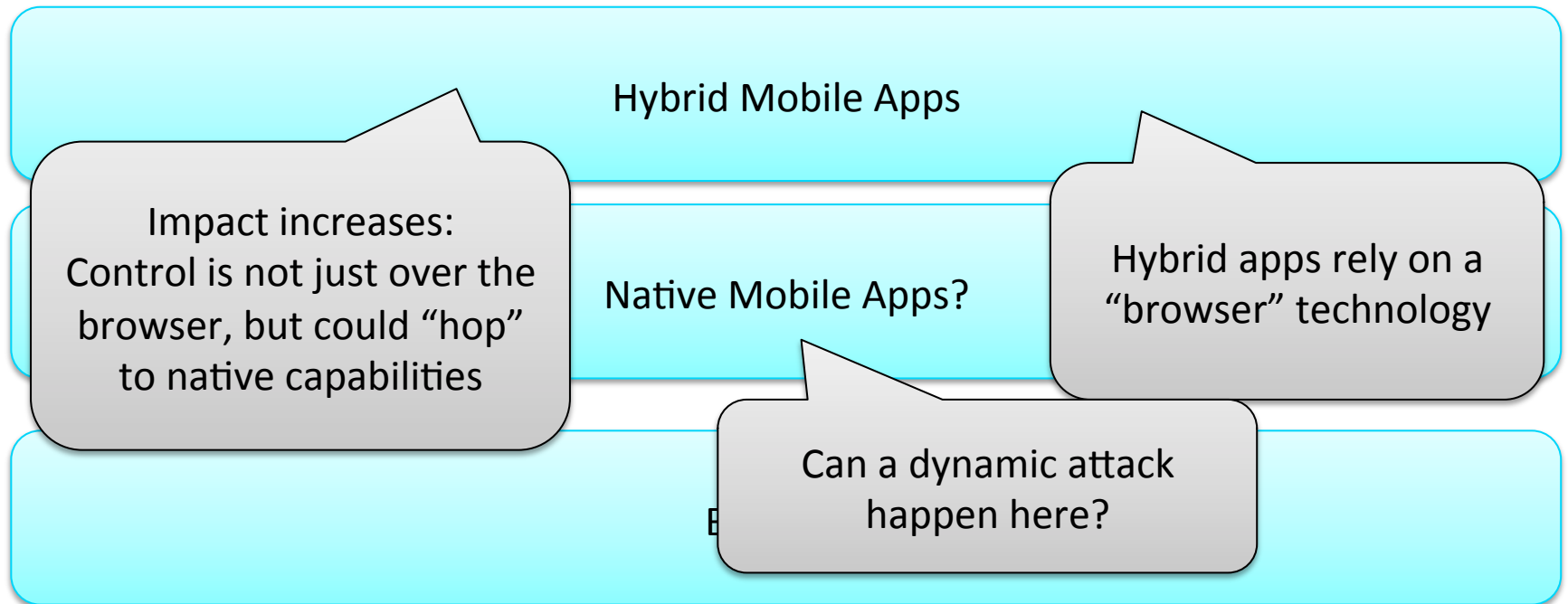
— Want to participate?

- ▶ Ever wanted to be an editor of a newspaper?
- ▶ Tweet with **#skycure** during this presentation

Man in the Middle Challenges



Man in the Middle Challenges



HTTP REQUEST HIJACKING

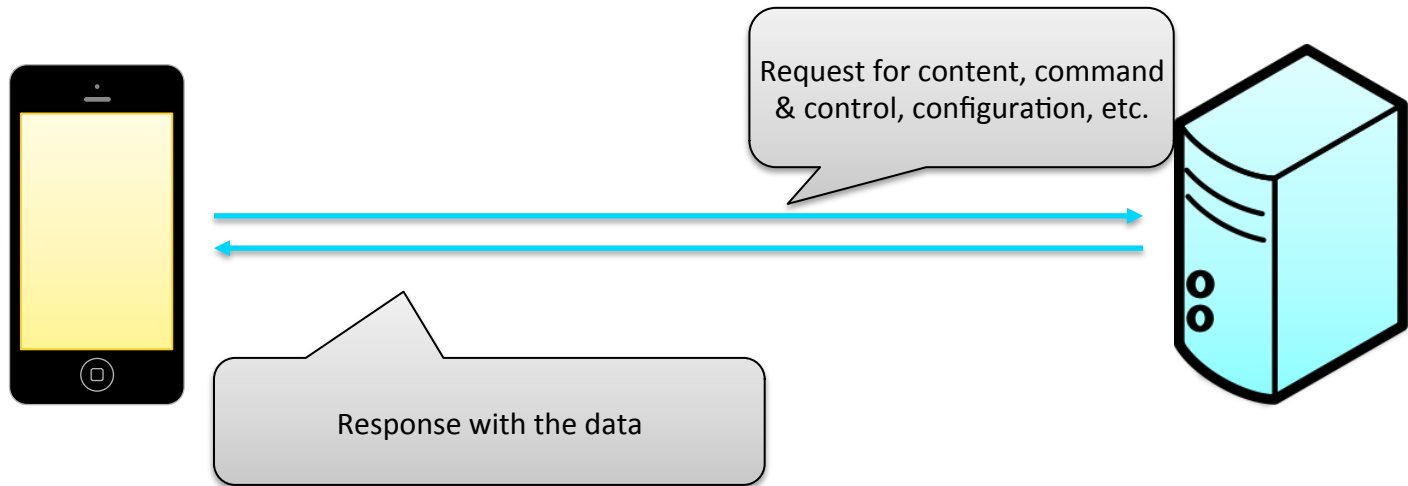
“We accept the reality of the world with which we are presented, it's as simple as that.”

- Christof, The Truman show



RSAC CONFERENCE
EUROPE 2013

Native Mobile Apps

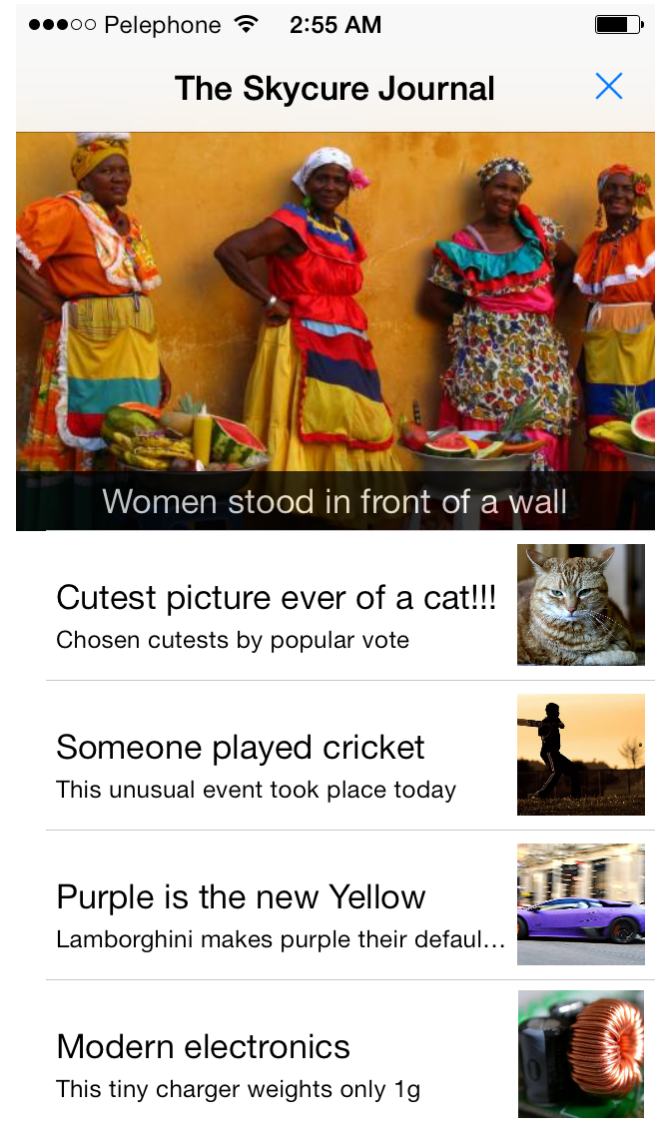


The Skycure Journal

- ▶ While very basic, *The Skycure Journal* operates in a similar way to most major news apps:
 - ▶ Load A JSON formatted feed
 - ▶ Parse it
 - ▶ Display to the reader...

Code available at:

https://github.com/skycure/Skycure_news



Into the Code / Objective-C

```
- (void)fetchArticles
{
    NSURL *serverUrl =
        [NSURL URLWithString:@"http://journal.skycure.com"];

    NSMutableURLRequest *request =
        [NSMutableURLRequest requestWithURL:serverUrl];

    [request setValue:@"application/json"
        forHTTPHeaderField:@"Content-Type"];

    self.connection =
        [[NSURLConnection alloc] initWithRequest:request delegate:self];
}
```

— HTTP Request Hijacking

Let's look at the actual network traffic...

HTTP Request Hijacking

```
- (void)fetchArticles  
{
```

```
    NSURL *serverUrl =  
        [NSURL URLWithString:@"http://journal.skycure.com"];
```

```
    NSMutableURLRequest *request =  
        [NSMutableURLRequest requestWithURL:serverUrl];
```

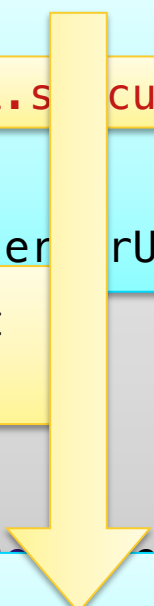
```
    [request setValue:@"application/javascript"  
        forHTTPHeaderField:@"Content-Type"];
```

```
    self.connection =
```

```
    NSURL *serverUrl =  
        [NSURL URLWithString:@"http://attacker.site/skycureJournal"];
```

```
    NSMutableURLRequest *request =  
        [NSMutableURLRequest requestWithURL:serverUrl];
```

HTTP Request
Hijacking



— HTTP Request Hijacking

Question: How is that done?

Answer: Very simple!

RFC of 301 *Moved Permanently*

“10.3.2 301 Moved Permanently

*The requested resource has been assigned a new **permanent URI** and **any future references** to this resource **SHOULD** use one of **the returned URIs**.*

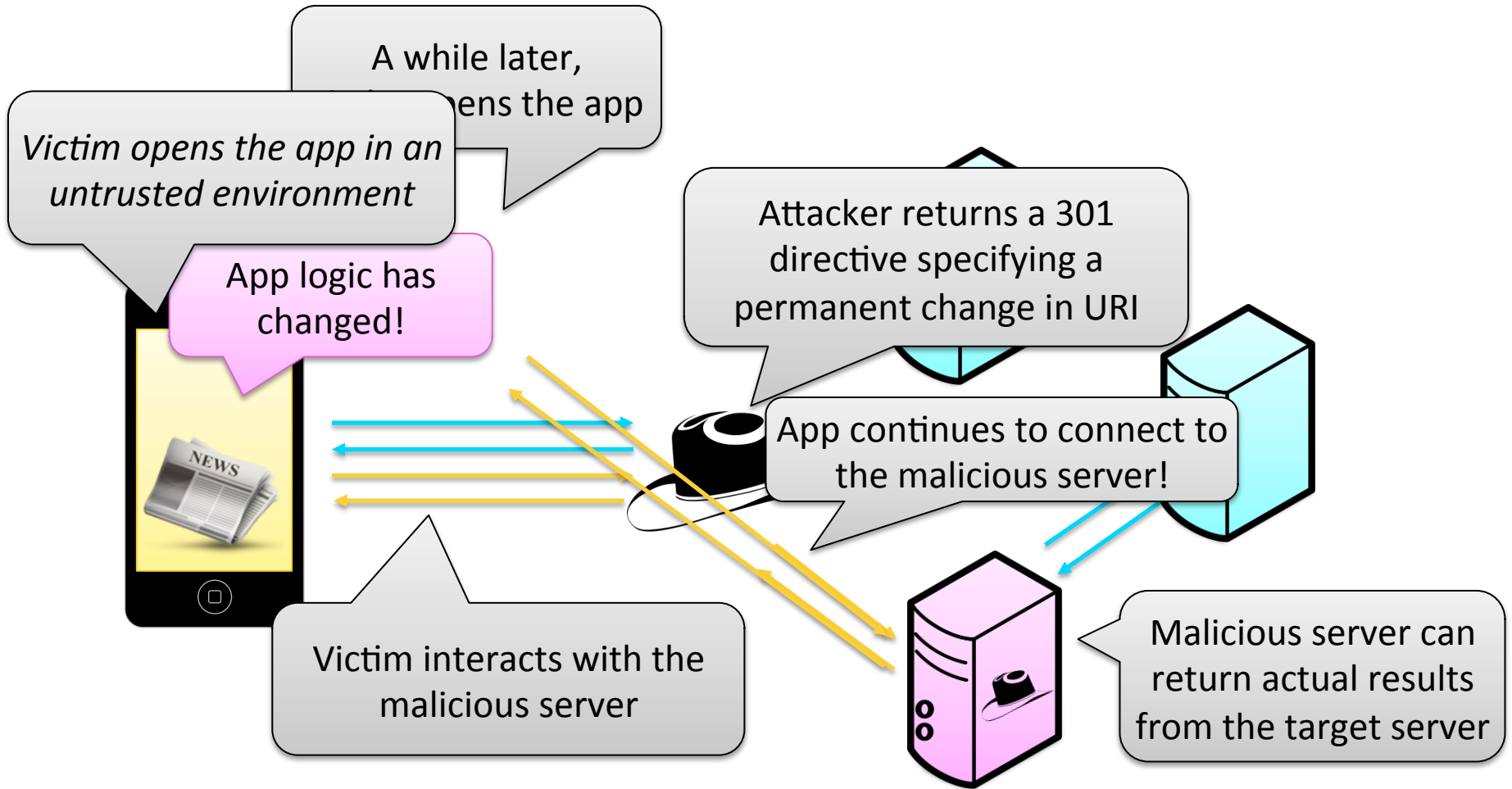
*Clients with link editing capabilities ought to automatically re-link references to the Request-URI to one or more of the new references returned by the server, where possible. **This response is cacheable unless indicated otherwise.**”*

Source: [RFC 2616 Fielding, et al](#)

301 Moved Permanently

- ▶ “If you need to change the URL of a page as it is shown in search engine results, we recommend that you use a server-side 301 redirect.” [\(Google help page\)](#)
- ▶ Usage examples:
 - ▶ Moving to a new domain
 - ▶ Merging two websites
- ▶ **Useful for the web, bad for mobile apps**

HRH – Attack Flow



— HTTP Request Hijacking

- ▶ **Requests can be hijacked!**
 - ▶ **Seamlessly**
 - ▶ **Permanently**
- ▶ **App logic practically changes**
 - ▶ **Hidden**
 - ▶ **Hard to remove**

HRH - Demo

This is where you get to dictate the news...

Impact - Example

AP The Associated Press 
@AP

 Following

Breaking: Two Explosions in the White House and Barack Obama is injured

 Reply  Retweet  Favorite  More

3,063
RETWEETS

144
FAVORITES



12:07 PM - 23 Apr 13



Challenges [1-2]

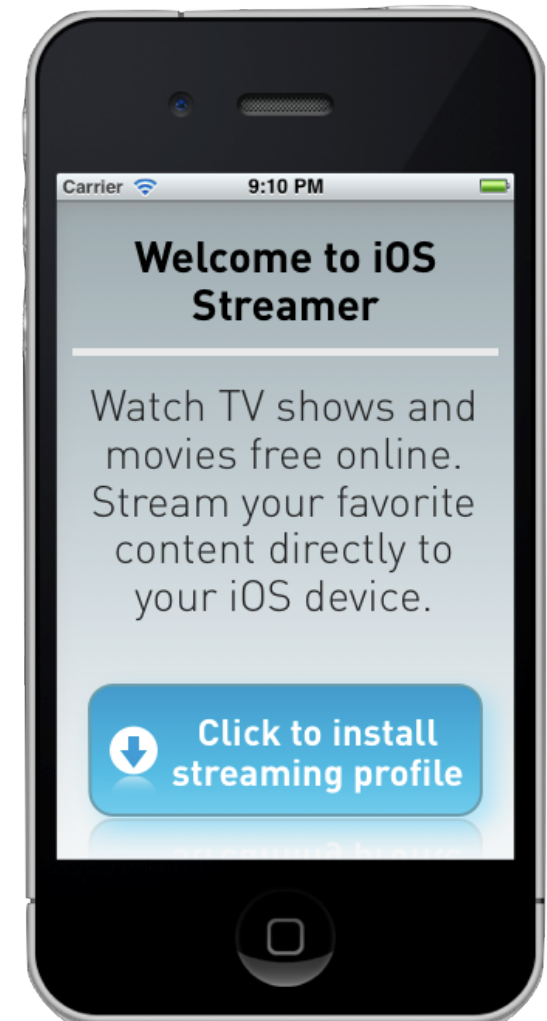
- ▶ Some limitations:
 - ▶ Attack is only effective on HTTP
 - ▶ Attacker has to be nearby the victim

Answer: **Malicious profiles!**

Malicious Profiles

- ▶ Configuration profiles are a great mechanism to configure iOS devices
- ▶ However... they might be used for bad deeds:
 - ▶ Tunnel all traffic through remote servers
 - ▶ Install a root certificate

- ▶ **Recently uncovered by Skycure**



Malicious Profiles

Demo

— HRH + Malicious profiles

▶ Impact

- ▶ HRH Extended to HTTPS
- ▶ HRH can be carried from a different continent

Challenge [3]

▶ Challenge:

- ▶ Victim has to open the vulnerable app in the malicious network

Answer: **Captive!**

Captive to the Rescue

- ▶ Captive to the rescue:
 - ▶ Victim auto-connects to a network
 - ▶ Pineapple, WifiGate, etc.
 - ▶ Attacker remotely opens an arbitrary application
 - ▶ Vulnerable app gets infected

Demo

Putting it All Together

- ▶ Generalizing the attack:
 - ▶ Automate an attack on a huge amount of people
 - ▶ Plant malicious code today, perform the attack tomorrow



HRH-Persister – A Testing Tool

▶ Step 1 (Infect):

- ▶ Set up a proxy
- ▶ Open *Tested app*
- ▶ Return 301 for every GET request with an identification
 - ▶ Identification could be a domain, sub-domain, path, etc.
- ▶ For the second “redirected” request, return the original response

▶ Step 2 (detect):

- ▶ Close the *Tested app*
- ▶ Open it again
- ▶ Look for requests with the identification value

Vulnerable Apps

- ▶ We tested a bunch of high profile apps
 - ▶ Almost half of them were susceptible to HRH
- ▶ Responsible disclosure challenge: we cannot identify and contact all affected vendors

— Developers - Best Practice

- ▶ Communicate via HTTPS instead of HTTP:
 - ▶ Highly recommended
 - ▶ Effective mitigation, not a fix

— Developers - Best Practice

- ▶ Caching helps with performance
 - ▶ However, it could lead to security threats
- ▶ Refrain from using caching in critical logic of the application when not needed*

```
NSURL *serverUrl =  
    [NSURL URLWithString:@"http://journal.skycure.com"];  
  
NSMutableURLRequest *request =  
    [NSMutableURLRequest requestWithURL:serverUrl  
        cachePolicy:NSURLRequestReloadIgnoringLocalAndRemoteCacheData  
        timeoutInterval:60.0];
```

— Developers – Remediation

- ▶ Change your cache policies to prevent 301 caching
- ▶ Impact
 - ▶ Practically, remove support for 301 handling in mobile applications
- ▶ In most cases this is a acceptable
 - ▶ We are talking about apps. If the developer wants to move to a new a page moves permanently, the developer can always update the app itself.

Developers – Remediation (cont.)

```
@interface HRHResistantURLCache : NSURLCache
@end
```

```
@implementation HRHResistantURLCache

- (void) storeCachedResponse:(NSCachedURLResponse *)cachedResponse
    forRequest:(NSURLRequest *)request
{
    NSInteger statusCode =
        [(NSHTTPURLResponse *)cachedResponse.response statusCode];

    if (301 == statusCode)
    {
        return;
    }
    [super storeCachedResponse:cachedResponse forRequest:request];
}
@end
```


— Developers – Remediation (cont.)

- ▶ Set the new cache policy to be used by the app
- ▶ Making sure you place the initialization code before any request in your code.

```
HRHResistantURLCache *myCache = [[HRHResistantURLCache alloc]
    initWithMemoryCapacity:512000
    diskCapacity:10000000
    diskPath:@"MyCache.db"];

[NSURLCache setSharedURLCache:myCache];
```

Source: <http://www.skycure.com/blog/http-request-hijacking/>

— End Users

- ▶ Let us know if you think you have been under attack
 - ▶ Remove and reinstall app to ensure removal of the attack
- ▶ Always be sure to update your apps
 - ▶ Especially in the near future
 - ▶ Auto-update in iOS 7

— Organizations/CISOs/IT

- ▶ Implement a mobile security tool that provides both visibility and protection for mobile related threats

— Future work

- ▶ Research:
 - ▶ 308, 302 + Cache-Control
 - ▶ Other operating systems
- ▶ RFC:
 - ▶ Create a mobile application specific RFC

Summary



Summary

- ▶ Tip of the iceberg
- ▶ App-level security

- ▶ Check out our blog for more information:
 - ▶ <http://www.skycure.com/blog/http-request-hijacking/>

Thank you!

Yair Amit, Adi Sharabani

Skycure

twitter: YairAmit, AdiSharabani

{yair,adi}@skycure.com

<http://www.skycure.com>



RSAC CONFERENCE
EUROPE 2013