


Let's talk Security

Beyond Scanning



Dheeraj Joshi
 @dheerajhere

About Me

- Front-End @ *Wingify*
- Previously @ **zOHO**
- Open Source ([medium-cli](#))
- Ambidextrous TT Player

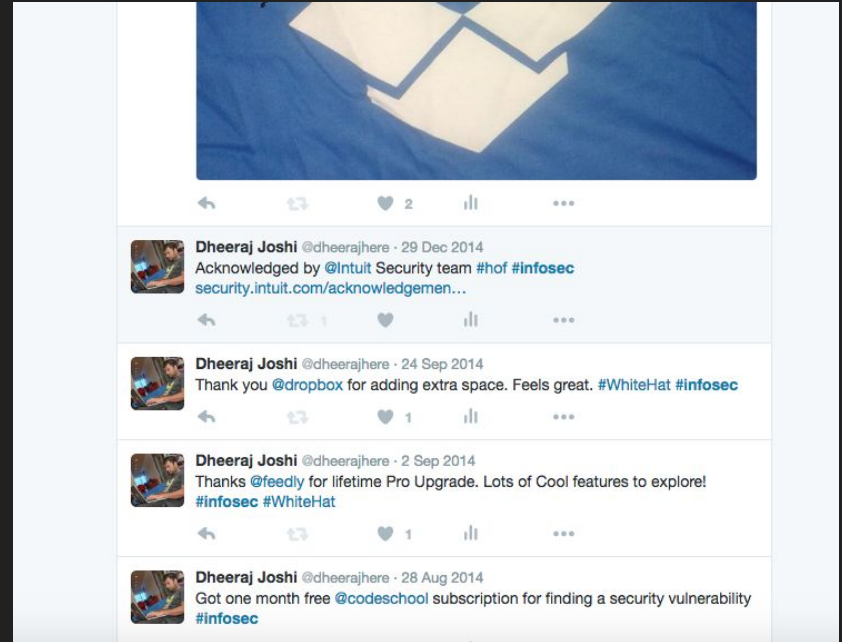
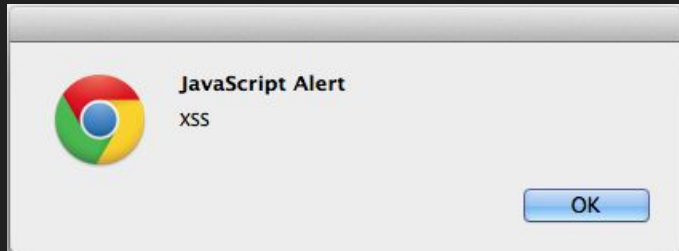


More...



Uber, CKEditor, Dropbox,
MailChimp, Recruiterbox, InVision,
DigitalOcean, Intuit, Groupon, etc.

What makes me happy?



In this talk...

- Why ?
- Cross-site Scripting (**XSS**)
- Cross-site Request Forgery (**CSRF**)
- Content Security Policy (**CSP**)
- HTTP Security Headers
- Best Practices & Demo



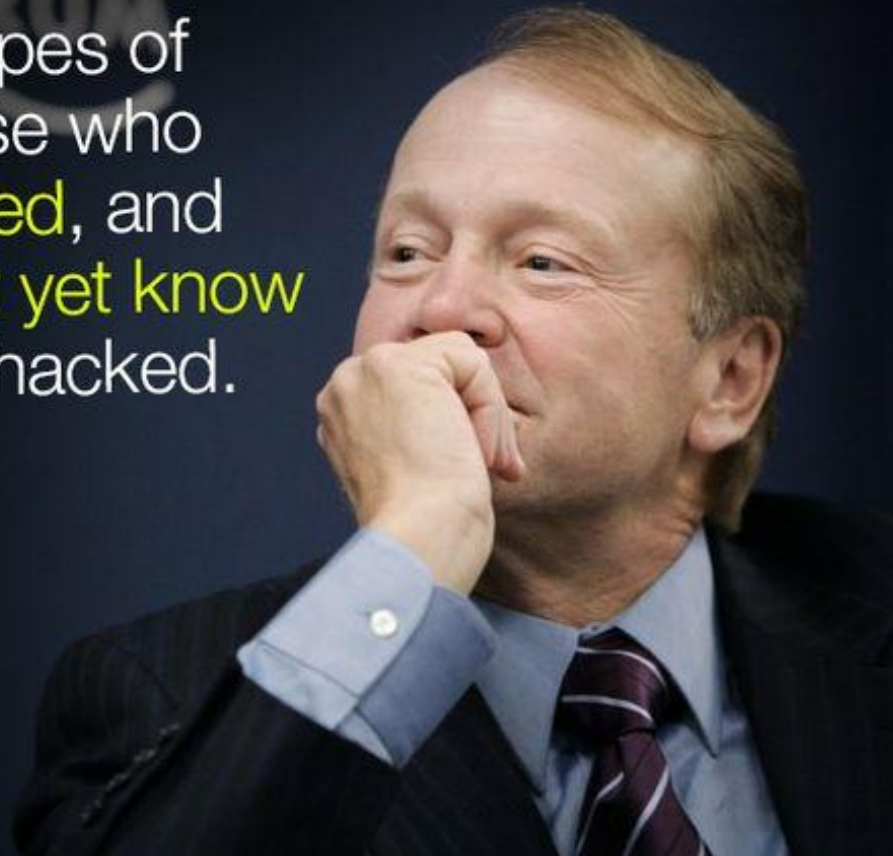
**LESS
TALK
AND
MORE
ACTION**

Why should we Care about Security?

Startups & SMEs are known to cut corners. One of the first things they cut is 'Security'.

There are two types of companies: those who **have been hacked**, and those who **don't yet know** they have been hacked.

John Chambers
Chief Executive Officer of Cisco







Password Reuse Attacks



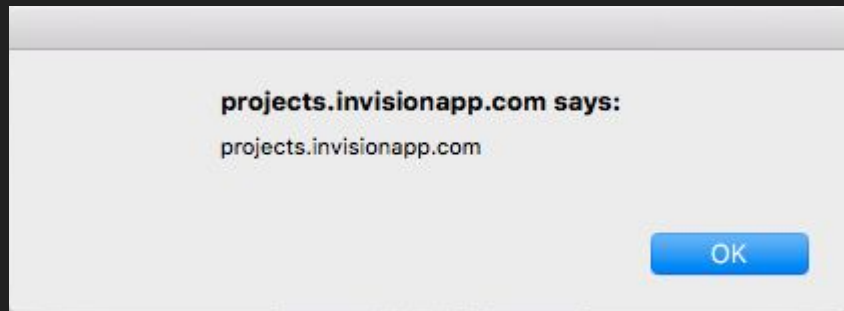
The Shutdown



HACKER PUTS HOSTING SERVICE "CODE SPACES" OUT OF BUSINESS

CROSS SITE SCRIPTING - XSS

- XSS attack users
- Inject Malicious content
- Exploits can be real bad



What is XSS?

`http://www.websearch.com/search?q=<script>document.write('
...
</html>
```

# Hunt...

- Data <-> Code
- Input Validation
- Check HTML Encoding
- Sanitizers
- Analyze places where DOM elements are created





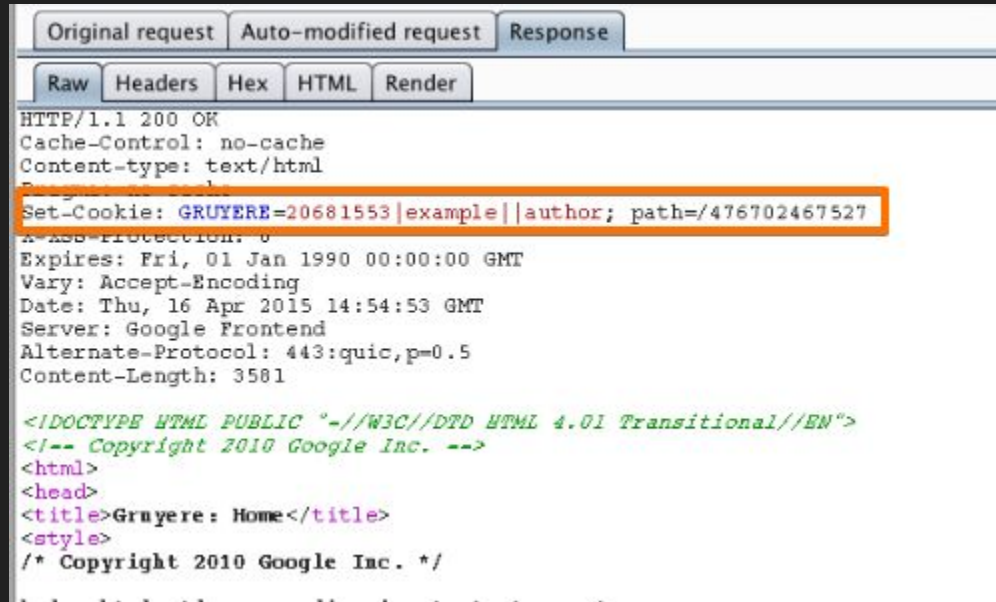


# XSS via template injection

Using Sandbox Bypasses

<http://blog.portswigger.net/2016/04/adapting-angularjs-payloads-to-exploit.html>

- Check for **HTTPOnly**, **Secure** flag on Session Cookie

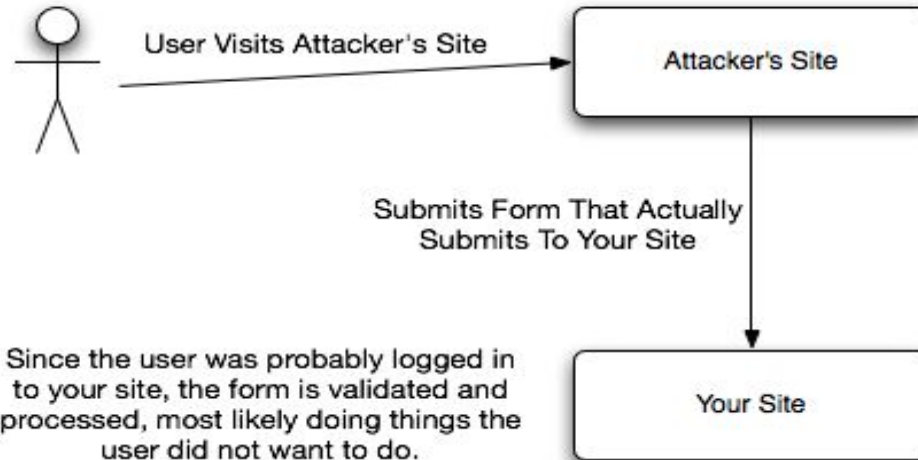


The screenshot shows a web browser's developer tools window with the 'Response' tab selected. The 'Headers' sub-tab is active, displaying the response headers for an HTTP 200 OK status. The 'Set-Cookie' header is highlighted with an orange box, showing the cookie name 'GRUYERE' and its value '20681553|example||author; path=/476702467527'. Below the headers, the raw HTML content is visible, starting with the DOCTYPE declaration and the copyright notice for Google Inc. 2010.

```
Original request Auto-modified request Response
Raw Headers Hex HTML Render
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-type: text/html
Set-Cookie: GRUYERE=20681553|example||author; path=/476702467527
Expires: Fri, 01 Jan 1990 00:00:00 GMT
Vary: Accept-Encoding
Date: Thu, 16 Apr 2015 14:54:53 GMT
Server: Google Frontend
Alternate-Protocol: 443:quic,p=0.5
Content-Length: 3581

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<!-- Copyright 2010 Google Inc. -->
<html>
<head>
<title>Gruyere: Home</title>
<style>
/* Copyright 2010 Google Inc. */
```

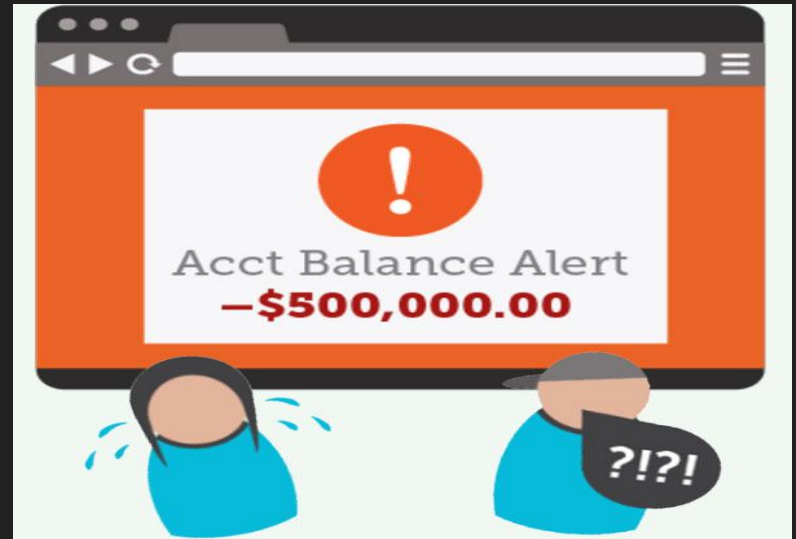
# CROSS-SITE REQUEST FORGERY (CSRF)



# CSRF Attacks

Because the attack is carried out by the victim, CSRF can bypass:

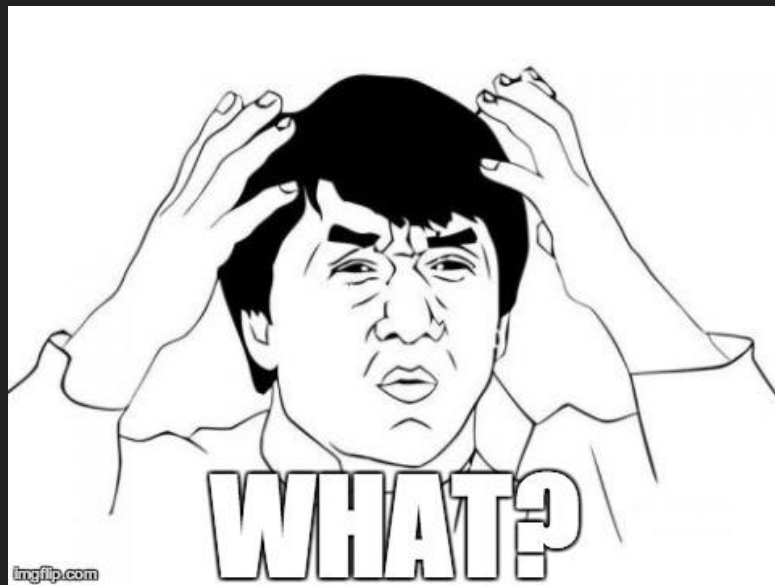
- HTTP Auth
- Session-based auth
- Firewalls



# “CSRF Myths”

## Preventions that Won't work

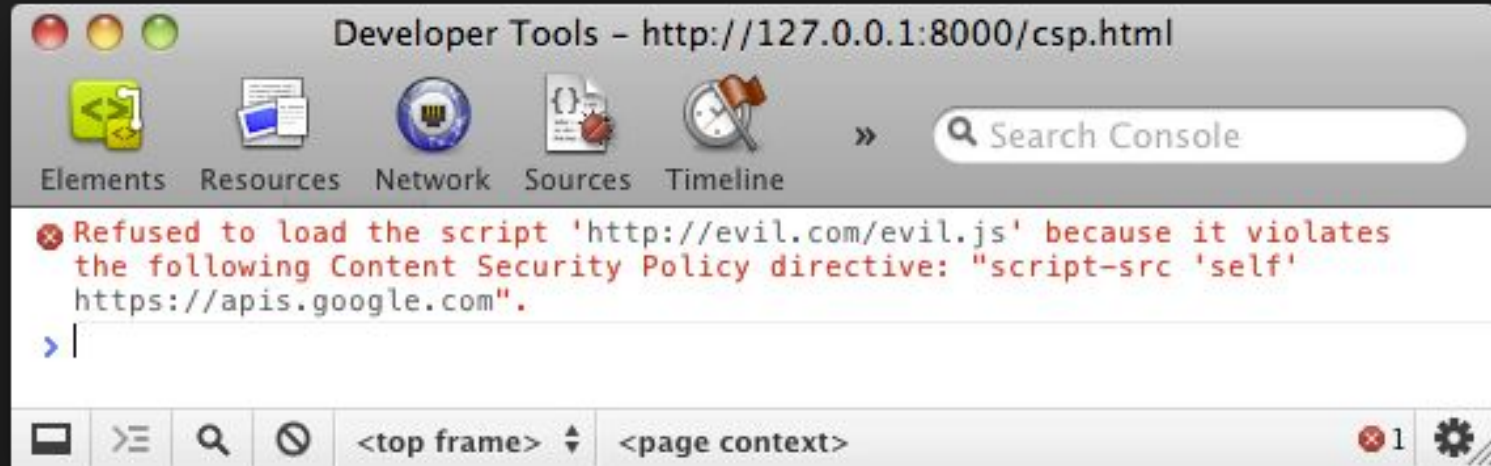
- Only accepting POST requests
- Referrer Protection
- Multi-Step Transactions
- URL Rewriting
- application/json



XSS + CSRF = ?



# Content Security Policy (CSP)



CSP Evaluator  
(<https://csp-evaluator.withgoogle.com>)



# HTTP Security headers

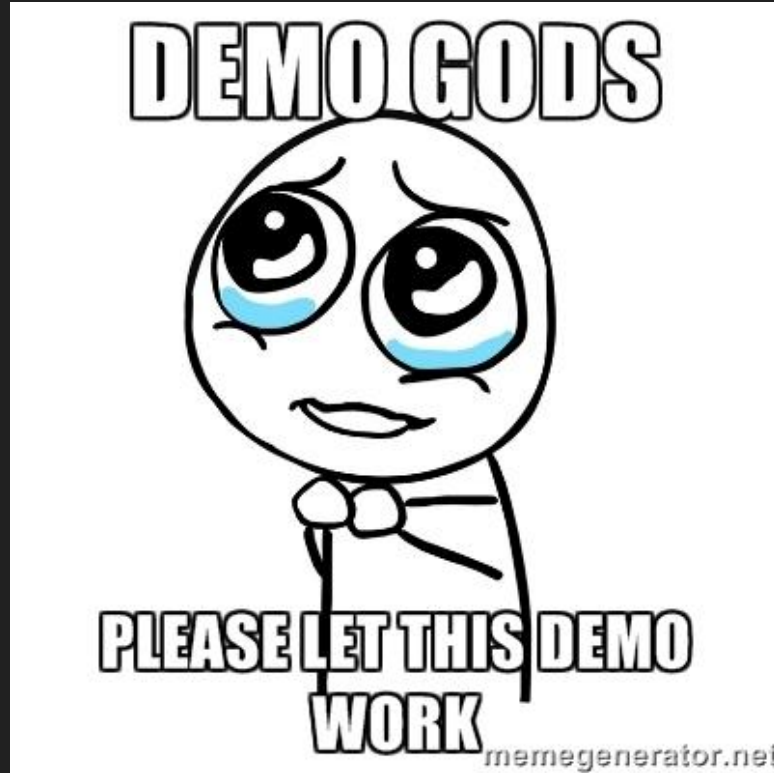
- Strict-Transport-Security:  
max-age=16070400; includeSubDomains
- X-Frame-Options: deny
- X-XSS-Protection: 1; mode=block
- X-Content-Type-Options: nosniff

# Defense

- **Strategy** - Integrate into SDLC
- Static Code Analysis
- Security Audits
- CTFs



# Show Time !



# Questions ?



# Thank you



@dheerajhere



@djadmin