OWASP
Open Web Application Security Project

# HTTP 101

## Security considerations

# Agenda

Encrypt all the things!

Public key pinning

Information leaks

Secure headers

# Encrypt all the things!

# Mywebsite.com:80 – Turn it off!

**What was th**
**unable to u**



# Mywebsite.com:80 – Turn it off!

## Encrypted Websites Protect Our Privacy and are Significantly Faster.

## Try it yourself @ httpvshttps.com

HTTP vs HTTPS — Test th

www.httpvshttps.com

OWASP
Open Web Application Security Project

# HTTP vs HTTPS Test

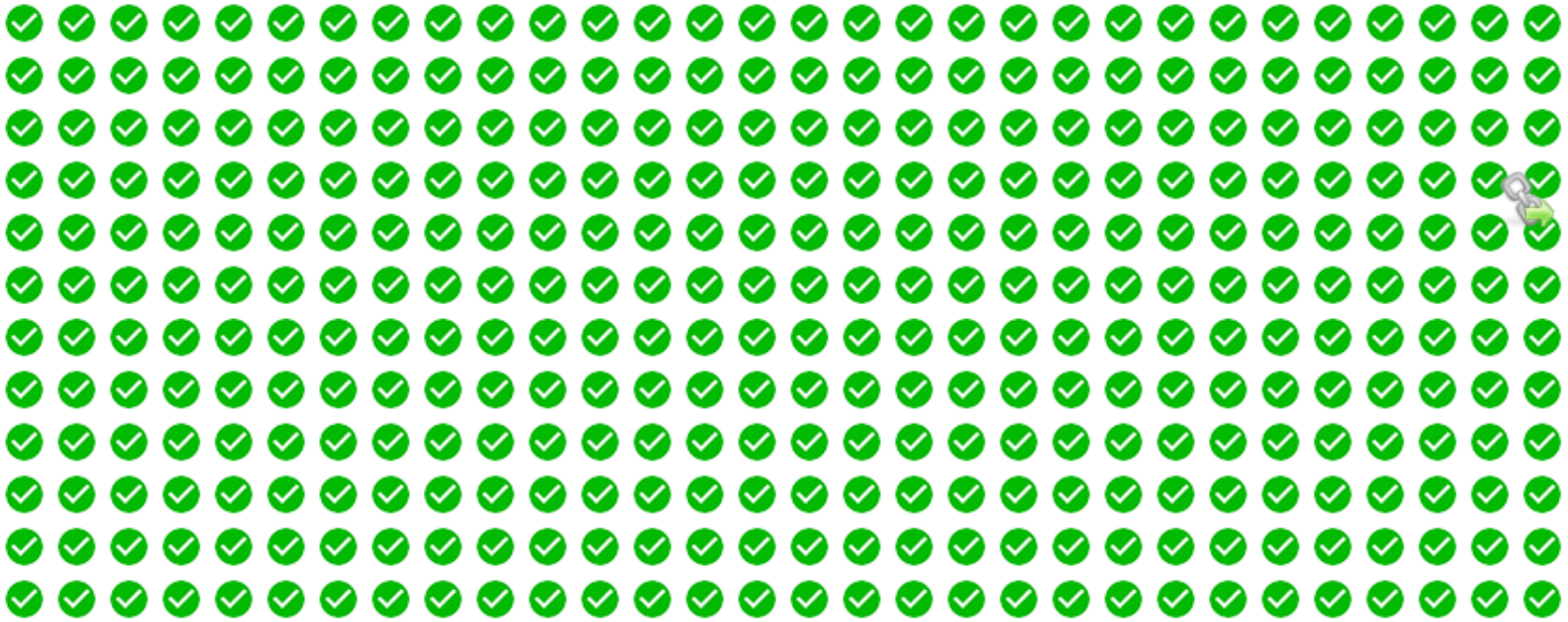
HTTP | HTTPS

**Encrypted Websites Protect Our Privacy and are Significantly Faster**

Compare load times of the unsecure HTTP and encrypted HTTPS versions of this page. Each test loads 360 unique, non-cached images (0.62 MB total). For fastest results, run each test 2-3 times in a private/incognito browsing session.

**21.918 s**

301% slower than HTTPS



ENG 4:43 PM 3/23/2017

HTTP vs HTTPS — Test th

Secure | https://www.httpvshttps.com

OWASP
Open Web Application Security Project

# HTTP vs HTTPS Test

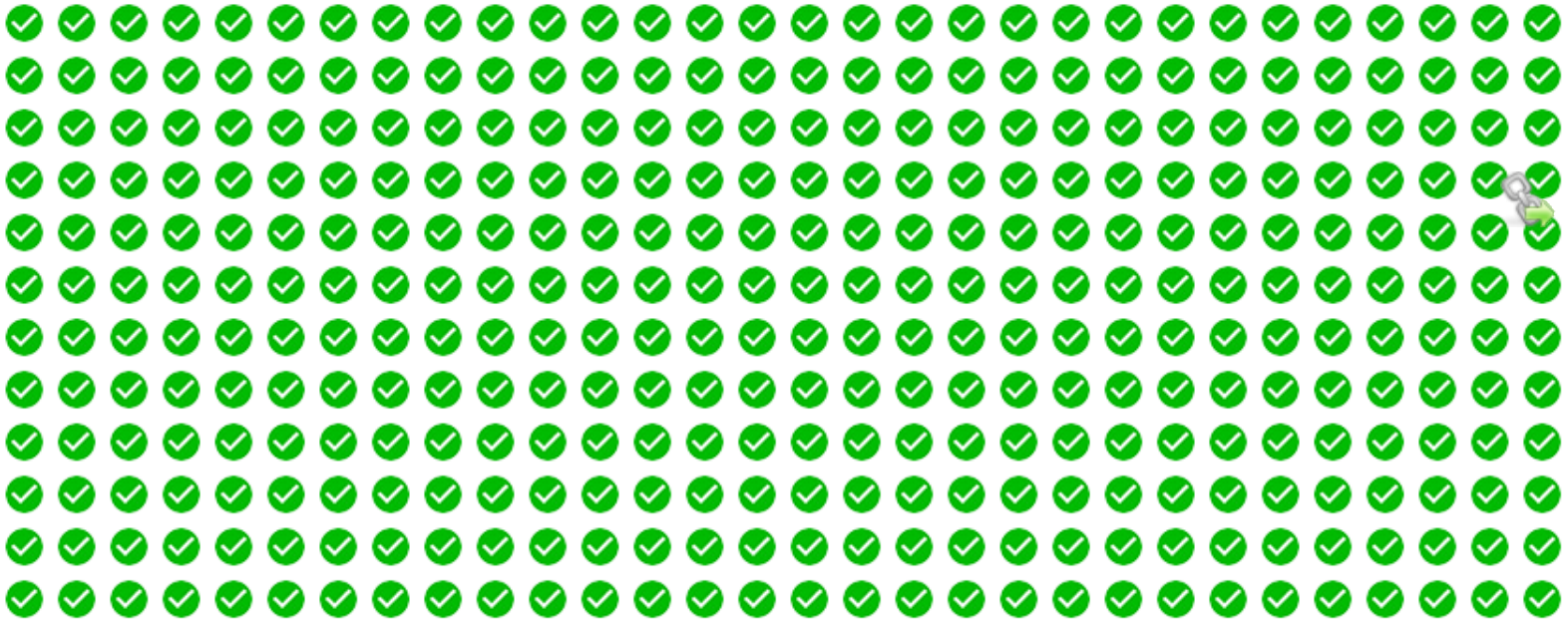
HTTP | HTTPS

**Encrypted Websites Protect Our Privacy and are Significantly Faster**

Compare load times of the unsecure HTTP and encrypted HTTPS versions of this page. Each test loads 360 unique, non-cached images (0.62 MB total). For fastest results, run each test 2-3 times in a private/incognito browsing session.

**4.646 s**

79% faster than HTTP



ENG 4:43 PM 3/23/2017

OWASP
Open Web Application Security Project

# Unable to pay for certificate?

## LetsEncrypt.org offers free certificates trusted by all major browsers.

Let's Encrypt

For more info visit: https://letsencrypt.org/blog/

OWASP
Open Web Application Security Project

# Getting certificate is not enough

## You need proper set up, aka A+ score on ssl labs.


https://www.ssllabs.com/ssltest/

# Certificate pinning?




**Any one of the Certificate Authorities (CAs) in your trust store can issue a certificate for your hostname and the browser will trust it implicitly. This is great in the sense that you can obtain a certificate from any CA of your choosing, but not so great when one gets compromised.**

January 23, 2017

# Symantec caught issuing illegit certificates for second time in two years

Independent researcher Andrew Ayer spotted Symantec once again improperly issuing 108 invalidated transport layer security certificates.

The credentials were in strict violation of industry guidelines.

The credentials were in strict violation of industry guidelines with nine of the improper certs reportedly issued without the knowledge or permission of the affected domain orders and 99 were issued to companies with data that was obviously fraudulent, according to Jan. 19 blog post.

Ayer reported the issue to the firm and was told by Symantec Policy Manager Steven Medin that the company was investigating and would report on the resolution, cause analysis, and corrective actions once they're completed. Many of the improperly issued certifications were revoked within an hour of being issued but still represent a major violation on Symantec's part.

While the investigation is still ongoing, a Symantec spokesperson told SC Media the certificates in question were issued by the firm's of our WebTrust audited partners.

January 23, 2017


others leaked

**2.** Report: Dark web vendor selling millions of Gmail and Yahoo accounts

**3.** Google searchers beware Police might soon be searchi for you

**4.** Cybersecurity made simp

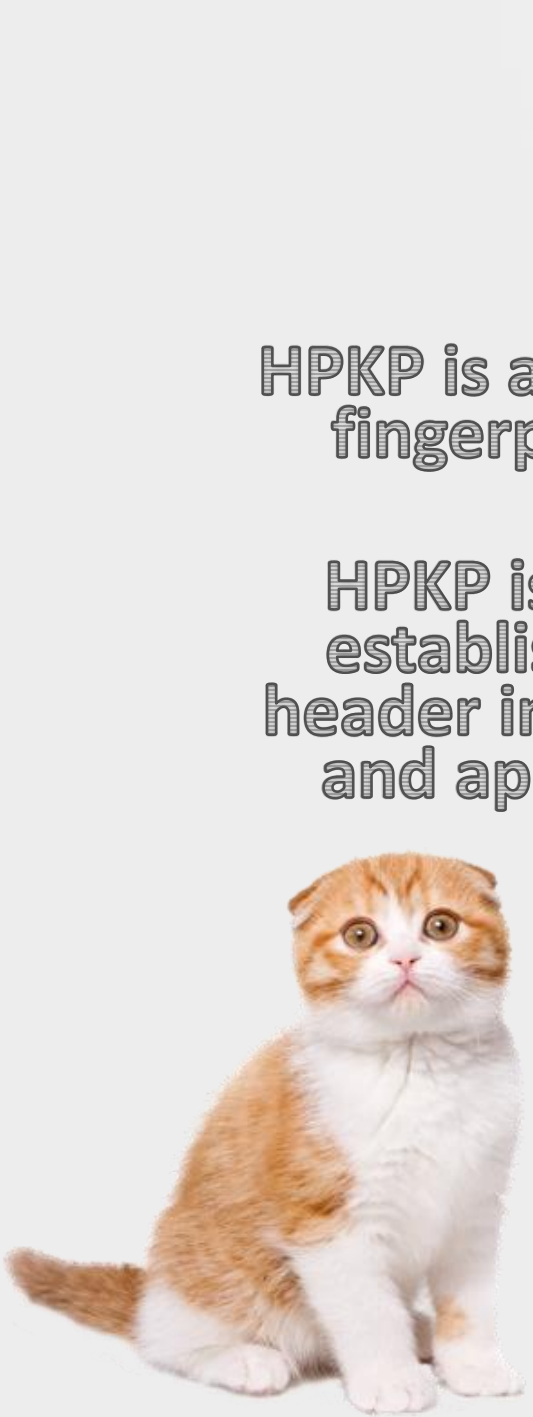
**5.** Demi Lovato nudes leake scammer targets private citizens for sex shows



others leaked

**2.** Report: Dark web vendor

# Certificate pinning?




**HPKP is a HTTP response header that allows a host to deliver a set of fingerprints to cryptographic identities that the User Agent (UA) should accept for the site going forwards.**

**HPKP is a Trust On First Use (TOFU) mechanism and the UA must establish at least one secure session with the host to receive the header intact. Once the header has been received, the UA will cache and apply the policy as per the directives included in the header.**

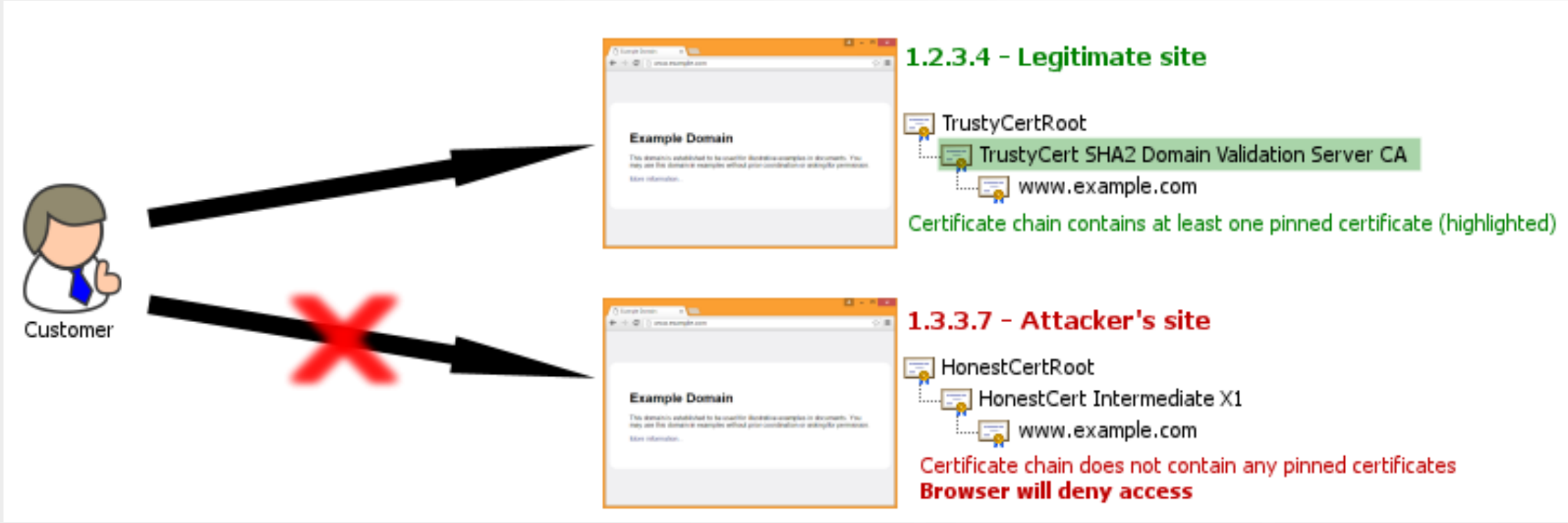


# Certificate pinning?




```
Public-Key-Pins: max-age=2592000;
pin-sha256="E9CZ9INDbd+2eRQozYqqbQ2yXLVKB9+xcprMF+44U1g=";
pin-sha256="LPJNul+wow4m6DsqxbninhsWHlwfp0JecwQzYpOLmCQ=";
report-uri="http://example.com/pkp-report"
```

# Certificate pinning



# Certificate pinning?


The most ideal solution is to include the fingerprint of your current TLS certificate and at least one backup.

The backup can be the fingerprint of a Certificate Signing Request so that you don't have to purchase a backup certificate.

If the private key of your certificate were ever compromised, you could use the CSR to request the signing of a new public key.

For this to work, the CSR has to be created with a brand new RSA key pair and stored securely offline.

As the fingerprint of the CSR was already in the HPKP header, you can switch out to the new certificate without a problem.

CSR: https://en.wikipedia.org/wiki/Certificate_signing_request

# Agenda

~~Encrypt all the things!~~

~~Public key pinning~~

Information leaks

Secure headers

# Information leaks


**Remove the following headers:**

**Server**

**X-Powered-By**

**X-Generator**

**and anything else that leaks information about the state or the technology behind the app.**

http://projects.webappsec.org/w/page/13246936/Information%20Leakage

# Information leaks – ping of death

OWASP
Open Web Application Security Project

**Data Centre ▸ Servers**

## Sysadmins, patch now: HTTP 'pings of death' are spewing across web to kill Windows servers

### Patch Tuesday bug reverse engineered by Thursday

16 Apr 2015 at 21:41, Iain Thomson

The SANS Institute has warned Windows IIS web server admins to get patching as miscreants are now exploiting a flaw in the software to crash websites.

The security bug (CVE-2015-1635) allows attackers to knock web servers offline by sending a simple HTTP request. Microsoft fixed this denial-of-service vulnerability on Tuesday with a patch numbered MS15-034.

However, within hours of the update going live, people reverse engineered the new code to find out where the hole is and how to exploit it, and have started sending out the pings of death.

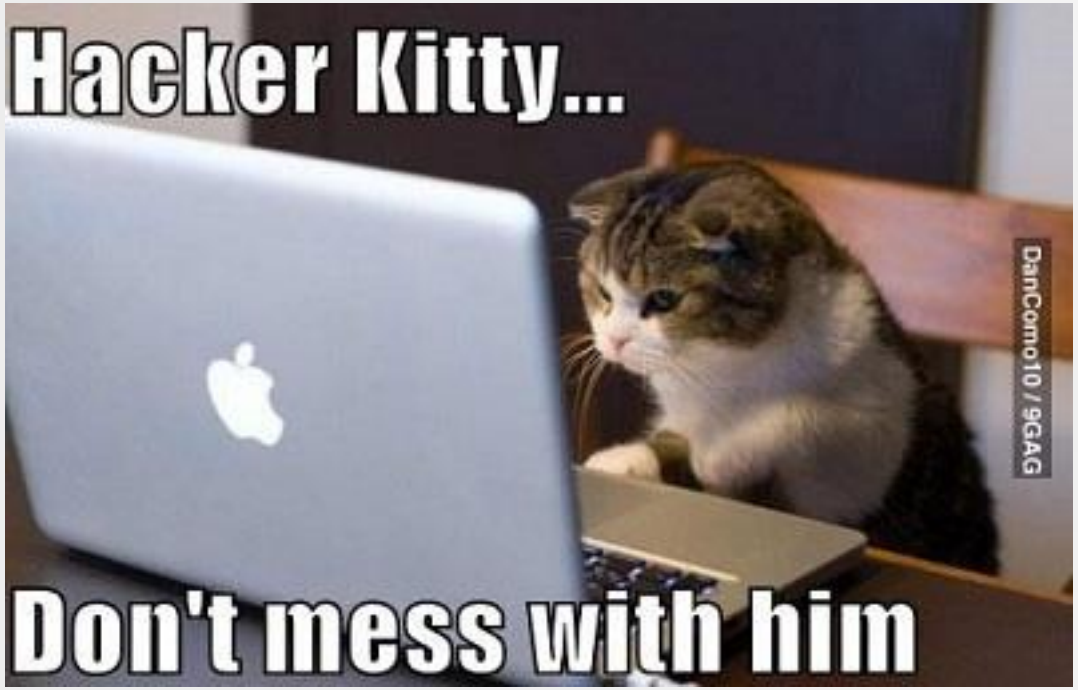
# Ping of death example




**curl -v 10.0.1.1/welcome.png**

**-H "Host: irrelevant"**

**-H "Range: bytes=20-18446744073709551615"**




https://ma.ttias.be/remote-code-execution-via-http-request-in-iis-on-windows/

# Agenda

~~Encrypt all the things!~~

~~Public key pinning~~

~~Information leaks~~

Secure headers

OWASP
Open Web Application Security Project

# Secure Headers

## HTTP Strict Transport Security (HSTS)

## X-Frame-Options

## X-XSS-Protection

## X-Content-Type-Options

## Content-Security-Policy

## Secure cookies

OWASP Secure Headers Project : https://www.owasp.org/index.php/OWASP_Secure_Headers_Project

# HTTP Strict Transport Security

OWASP
Open Web Application Security Project

**HSTS allows a site to request that it always be contacted over HTTPS.**

**Strict-Transport-Security: max-age=16070400; includeSubDomains**

# X-Frame-Options

OWASP
Open Web Application Security Project

**The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a <frame>, <iframe> or <object> .**

**Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.**

# X-Frame-Options

OWASP
Open Web Application Security Project

## There are three possible values for the X-Frame-Options header:

1. DENY, which prevents any domain from framing the content. The "DENY" setting is recommended unless a specific need has been identified for framing.
2. SAMEORIGIN, which only allows the current site to frame the content.
3. ALLOW-FROM uri, which permits the specified 'uri' to frame this page. (e.g., ALLOW-FROM http://www.example.com)

For more info visit: https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet#X-Frame-Options_Header_Types

# X-XSS-Protection


**This response header can be used to configure a user-agent's built in reflective XSS protection.**

# X-XSS-Protection

OWASP
Open Web Application Security Project

- 0 - Disables the XSS Protections offered by the user-agent.
- 1 - Enables the XSS Protections
- 1; mode=block - Enables XSS protections and instructs the user-agent to block the response in the event that script has been inserted from user input, instead of sanitizing.
- 1; report=http://site.com/report - A Chrome and WebKit only directive that tells the user-agent to report potential XSS attacks to a single URL.
  Data will be POST'd to the report URL in JSON format.

# X-Content-Type-Options

OWASP
Open Web Application Security Project

**This header can be set to protect against MIME type confusion attacks.**

**nosniff - This is the only valid setting, it must match nosniff.**

Example: http://lcamtuf.coredump.cx/squirrel/

# *Hello, squirrel fans!*

This is an embedded landing page for an image. You can link to this URL and get the HTML document you are viewing right now (soon to include essential squirrel facts); or embed the exact same U
image on your own squirrel-themed page:

```
<a href="http://lcamtuf.coredump.cx/squirrel/">Click here!</a>

<img src="http://lcamtuf.coredump.cx/squirrel/">
```

No server-side hacks involved - the magic happens in your browser. Let's try embedding the current page as an image right now (INCEPTION!):

# X-Content-Type-Options


Some browsers try to be smart (it's called content sniffing) and guess the type of file being transferred by default.

This allows the browser to render an HTML file if the content looks right even if the server says that the file is plaintext.

This can be used as an attack vector for untrusted JavaScript code.

# Content-Security-Policy


Content Security Policy (CSP) header lists all authorized domains and resources your app is allowed to use.

Careful, policy misconfiguration and too permissive policies can do more harm than good.

# Content-Security-Policy

OWASP
Open Web Application Security Project

**Allow everything but only from the same origin:**

**default-src 'self';**

**Allow Google Analytics, Google AJAX CDN and Same Origin:**

**script-src 'self' www.google-analytics.com ajax.googleapis.com;**

**Starter Policy:**

**default-src 'none'; script-src 'self'; connect-src 'self'; img-src 'self'; style-src 'self';**

# Content-Security-Policy


**Content Security Policy Level 3**
**W3C Working Draft, 13 September 2016**
**https://www.w3.org/TR/CSP3/**

# Content-Security-Policy


**Why do we need this? Because attacks can come in many shapes and your modern framework can't always protect you.**

XSS without HTML: Client-Side Template Injection with AngularJS
http://blog.portswigger.net/2016/01/xss-without-html-client-side-template.html

CSP Reference
https://content-security-policy.com/

# Bonus: subsource integrity


**Subresource Integrity (SRI) is a security feature that enables browsers to verify that files they fetch are delivered without unexpected manipulation. It works by allowing you to provide a cryptographic hash that a fetched file must match.**

```
<script src="https://example.com/example-framework.js"
integrity="sha384oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl1k
PzQho1wx4JwY8wC"
crossorigin="anonymous"> </script>
```

# Secure cookies


## Use httpOnly and secure flags.

## HttpOnly will forbid JavaScript from accessing cookies which will prevent an XSS attack from being able to send your user's cookies to the attacker.

## The Secure attribute will only allow the cookies to be transferred over an HTTPS connection and not over HTTP, so an attacker with access to your network won't be able to read unencrypted cookies.

# Recap:


I SHOULD REALLY START TO...
ENCRYPT EVERYTHING!
memegenerator.net


THANKS FOR LISTENING
ANY QUESTIONS
memegenerator.net