This Talk Was Brought To You By

The Etsy Security Team
What’s an Etsy?
Security Headers?

Why Security Headers?
Security Headers

Fundamentally, a user security issue
Security Headers

Fundamentally, a user security issue
Changes are browser-impacting
Security Headers

Fundamentally, a user security issue
Changes are browser-impacting
Unfortunately, browsers != users
Security Headers

Fundamentally, a user security issue
Changes are browser-impacting
Unfortunately, browsers != users
Often requires non-trivial changes
Security Headers

Strategies for deployment
Security Headers

Strategies for deployment

Lessons learned from our bug bounty
Overview

HTTP Strict Transport Security (HSTS)
Overview

HTTP Strict Transport Security (HSTS)
Content Security Policy (CSP)
Overview

HTTP Strict Transport Security (HSTS)
Content Security Policy (CSP)
X-Frame-Options (XFO)
Overview

HTTP Strict Transport Security (HSTS)
Content Security Policy (CSP)
X-Frame-Options (XFO)
Miscellaneous
HSTS --What is it?

A guarantee to visit the url using HTTPS
HSTS -- What is it?

A guarantee to visit the url using HTTPS

You have to have seen the site before
What’s the Attack?

The Classic Man-in-the-Middle Attack
What’s the Attack?

The Classic Man-in-the-Middle Attack

Let’s just turn on TLS/SSL for everything
What’s the Attack?

The Classic Man-in-the-Middle Attack
Let’s just turn on TLS/SSL for everything
Make HTTPS canonical for your site

Hosted by OWASP & the NYC Chapter
HTTP/HTTPS Traffic
HTTP/HTTPS Traffic
HSTS Background

Infrastructure changes needed for SSL
HSTS Background

Infrastructure changes needed for SSL
Bundle HSTS as part of an SSL preference for users
The Old Ways

Split Architecture
The Old Ways

Split Architecture
Most pages HTTP, “secure” ones HTTPS
The Old Ways

Split Architecture
Most pages HTTP, “secure” ones HTTPS
Load balancers constrained rollout
On Load Balancers

HTTP->HTTPS logic handled by the LB
On Load Balancers

HTTP-> HTTPS logic handled by the LB

Difficult and slow to change
On Load Balancers

HTTP->HTTPS logic handled by the LB
Difficult and slow to change
Broke HTTPS plugins
Refactoring

HTTP->HTTPS logic handled by the app
Refactoring

HTTP->HTTPS logic handled by the app
Make it easy to add new secure pages
Refactoring

HTTP->HTTPS logic handled by the app
Make it easy to add new secure pages
Transparency for developers
How Do I HTTPS

Ramp it up!
How Do I HTTPS

Ramp it up!

Enabled HSTS if SSL preference “on”
How Do I HTTPS

Ramp it up!

Enabled HSTS if SSL preference “on”

Bail-out Mechanism:

![Image of Security Settings]

Hosted by OWASP & the NYC Chapter

Wednesday, November 20, 13
The HSTS Header

Enabled header when full-site SSL “on”
The HSTS Header

Enabled header when full-site SSL “on”

Strict-Transport-Security: max-age=631138520; includeSubDomains
HSTS Part 2

Strict-Transport-Security: max-age=631138520; includeSubDomains

All subdomains get HSTS that match the host
HSTS Part 3

Note the difference: HSTS on ‘www.etsy.com’
HSTS Part 3

Note the difference: HSTS on ‘www.etsy.com’

Query domain

Input a domain name to query the current HSTS set:

Domain: www.etsy.com  Query

Found: mode: STRICT  sts_include_subdomains: true
HSTS Part 3

Note the difference: HSTS on ‘www.etsy.com’

Query domain

Input a domain name to query the current HSTS set:

Domain: www.etsy.com  Query

Found: mode: STRICT  sts_include_subdomains: true

Query domain

Input a domain name to query the current HSTS set:

Domain: subdomain.etsy.com  Query

Not found
HSTS Part 2

Check out Chrome’s HSTS settings

chrome://net-internals/#hsts
HSTS Rollout

Implement HTTPS management on app level
HSTS Rollout

Implement HTTPS management on app level
Rolled out to admins -> sellers -> buyers
HSTS Rollout

Implement HTTPS management on app level
Rolled out to admins -> sellers -> buyers
Code-based “SSL wrangler” in repo
SSL Wranglin’

Controller to handle SSL transition
SSL Wranglin’

Controller to handle SSL transition

Skipped for users with full-site SSL pref on
SSL Wranglin’

Controller to handle SSL transition
Skipped for users with full-site SSL pref on
On sign-out, set HSTS max-age=0
Wins

Fixes on-domain mixed content
Wins

Fixes on-domain mixed content
Browser transparently 302 redirects
SSL Concerns

Do your CDNs support it?
SSL Concerns

Do your CDNs support it?

What about 3rd party content providers?
SSL Concerns

Do your CDNs support it?
What about 3rd party content providers?
Can your servers/LBs handle it?
Kill Mixed Content

You still need to fix off-domain HTTP
Kill Mixed Content

You still need to fix off-domain HTTP

Browser mixed content warnings
Kill Mixed Content

You still need to fix off-domain HTTP

Browser mixed content warnings
Mobile

HSTS supported on mobile browsers
Mobile

HSTS supported on mobile browsers
Notably absent from others
Mobile

HSTS supported on mobile browsers
Notably absent from others
HSTS: Be Ready

Not a crutch for fixing routing problems!
HSTS: Be Ready

Not a crutch for fixing routing problems!

There will be outliers
HSTS: Be Ready

Not a crutch for fixing routing problems!
There will be outliers
SSL/TLS errors confuse users
HSTS: Be Ready

Not a crutch for fixing routing problems!
There will be outliers
SSL/TLS errors confuse users
Have a process for managing HSTS
X-Frame-Options

Problem: Clickjacking
X-Frame-Options

Framing sucks, get rid of framing!
X-Frame-Options

How do you prevent this type of attack?
X-Frame-Options

How do you prevent this type of attack?

<script>
if (top!=self) top.location.href=self.location.href
</script>
X-Frame-Options

How do you prevent this type of attack?

<script>
    if (top!=self) top.location.href=self.location.href
</script>

Not really a defense at all
How Do I Use XFO?

Figure out when you’re being framed
How Do I Use XFO?

Figure out when you’re being framed
Log the framing attempts
How Do I Use XFO?

Figure out when you’re being framed
Log the framing attempts
Whitelist specific framing sites (search engines)
How Do I Use XFO?

Figure out when you’re being framed
Log the framing attempts
Whitelist specific framing sites (search engines)
Only allow whitelisted sites to frame
Be Careful

Thoroughly vet your whitelist
Be Careful

Thoroughly vet your whitelist

Read about XFO’s options
Be Careful

Thoroughly vet your whitelist
Read about XFO’s options
Test thoroughly
Non-Whitelisted sites
Non-Whitelisted sites
Don’t Forget...

If you’re taking away framing, warn your users
Don’t Forget...

If you’re taking away framing, warn your users

Whitelisting will break everyone else
Let’s Talk CSP

Policies can grow fairly large
Let’s Talk CSP

Policies can grow fairly large

Doesn’t like inline javascript by default
Let’s Talk CSP

Policies can grow fairly large

 Doesn’t like inline javascript by default

Where do I start?
CSP 1.0

Most websites have inline JS
CSP 1.0

Most websites have inline JS

Removing/refactoring some of it just isn’t possible
CSP 1.0

Most websites have inline JS

Removing/refactoring some of it just isn’t possible

FF & Chrome use unprefixed ‘Content-Security-Policy’
CSP 1.1

Will have browser javascript API support
CSP 1.1

Will have browser javascript API support
Support for inline CSP in a `<meta>` tag
CSP 1.1

CSP 1.1 will allow for script-nonce and script-hash

Hosted by OWASP & the NYC Chapter
CSP Lessons

CSP introduced the idea of a reporting mechanism
CSP Lessons

CSP introduced the idea of a reporting mechanism

Identify pages with inline scripts => smaller policy size
CSP Lessons

CSP introduced the idea of a reporting mechanism
Identify pages with inline scripts => smaller policy size
Log, aggregate reports to find mixed content
CSP Lessons

CSP introduced the idea of a reporting mechanism
Identify pages with inline scripts => smaller policy size
Log, aggregate reports to find mixed content
Some interesting results
Hosted by OWASP & the NYC Chapter

Wednesday, November 20, 13
How Do I Deploy CSP?

Organize and assess your existing javascript
How Do I Deploy CSP?

Organize and assess your existing javascript
Have specific template logic for handling javascript
How Do I Deploy CSP?

Organize and assess your existing javascript
Have specific template logic for handling javascript
Give devs an ‘opt-out’ mechanism for inline js
How Do I Deploy CSP?

Organize and assess your existing javascript
Have specific template logic for handling javascript
Give devs an ‘opt-out’ mechanism for inline js
Deploy to specific parts/subdomains of your site
CSP Compliance

Actively monitor the # of inline scripts you have left
Some CSP Tools

Some tools for CSP Generation
Some CSP Tools

Some tools for CSP Generation

http://cspisawesome.com/
Some CSP Tools

Some tools for CSP Generation

http://cspisawesome.com/
https://github.com/Kennysan/CSPTools
CSP Tools

Browser proxy, automated browser, and csp parser
CSP Tools

Browser proxy, automated browser, and csp parser

Lets you create/test a CSP for your prod environment
CSP Tools

Browser proxy, automated browser, and csp parser

Lets you create/test a CSP for your prod environment

https://github.com/Kennysan/CSPTools
X-XSS-Protection

Originally IE XSS blocking mechanism
X-XSS-Protection

Originally IE XSS blocking mechanism

Looks for parameter arguments in response
X-XSS-Protection

Originally IE XSS blocking mechanism
Looks for parameter arguments in response
Side effect: Clients can break your javascript
X-XSS-Protection

X-XSS-Protection: 1; mode=block
X-XSS-Protection: 1; mode=block

Reflected XSS protection, but now...
X-XSS-Protection

X-XSS-Protection: 1; mode=block

Reflected XSS protection, but now...

Chrome lets you specify a report url
X-XSS-Protection

X-XSS-Protection: 1; mode=block

Reflected XSS protection, but now...

Chrome lets you specify a report url

Clientside protection; serverside reporting
XSS Logging

X-XSS-Protection: 1; mode=block; report-uri=/log.php
XSS Logging

X-XSS-Protection: 1; mode=block; report-uri=/log.php

Allows Chrome reflected XSS logging, ala CSP-style
XSS Logging

X-XSS-Protection: 1; mode=block; report-uri=/log.php

Allows Chrome reflected XSS logging, ala CSP-style

Other browsers: Implement server-side XSS-Auditor
XSS Logging

X-XSS-Protection: 1; mode=block; report-uri=/log.php

Allows Chrome reflected XSS logging, ala CSP-style

Other browsers: Implement server-side XSS-Auditor

Look for this functionality in CSP 1.1
X-Content-Type-Options

X-Content-Type-Options: nosniff
X-Content-Type-Options

X-Content-Type-Options: nosniff

Older versions of IE will guess response content-type
X-Content-Type-Options

X-Content-Type-Options: nosniff

Older versions of IE will guess response content-type

Ignores Content-Type specified!
X-Content-Type-Options

X-Content-Type-Options: nosniff

Older versions of IE will guess response content-type

Ignores Content-Type specified!

Example: query parameter lets you specify .html
X-Content-Type-Options

X-Content-Type-Options: nosniff

Older versions of IE will guess response content-type

Ignores Content-Type specified!

Example: query parameter lets you specify .html

IE will consider the content to be text/html!
Final Thoughts

Treat header deployment like any other code
Final Thoughts

Treat header deployment like any other code

Be agile with header development
Final Thoughts

Treat header deployment like any other code

Be agile with header development

Can’t deploy everywhere? Have a plan--deploy in part
Final Thoughts

Treat header deployment like any other code

Be agile with header development

Can’t deploy everywhere? Have a plan—deploy in part

Starting with security is easier than baking it in later
Final Thoughts

Treat header deployment like any other code
Be agile with header development
Can’t deploy everywhere? Have a plan--deploy in part
Starting with security is easier than baking it in later
Log early and often--you learn a lot
Thanks for Listening!

@kennysan

klee@etsy.com

github.com/kennysan