When Security Meets Compatibility

Emily Stark, Google Chrome
estark@chromium.org
@estark37
Your challenge:

Motivate as many web servers as possible to migrate to a new protocol version.

Eventually, remove client support for the old insecure version.

The sooner the better.
Your challenge:

Motivate as many web servers as possible to migrate to a new protocol version.

Eventually, remove client support for the old insecure version.

The sooner the better.

Outreach  
Documentation  
Enlist CDNs and hosting providers  
Browser UI
Security-sensitive breaking changes...
Security-sensitive breaking changes...

- Can be **urgent**
  - React to new attacks or vulnerabilities
Security-sensitive breaking changes...

- Can be **urgent**
  - React to new attacks or vulnerabilities
- Can introduce **risk**
  - Warning fatigue
Security-sensitive breaking changes...

- Can be **urgent**
  - React to new attacks or vulnerabilities
- Can introduce **risk**
  - Warning fatigue
- Might be subject to a **lowest common denominator effect**
  - If users move to different browsers or platforms, they may not be protected
When security meets compatibility

The process of breaking the web

The science of outreach

Experimentation with implementation
When security meets compatibility

The process of breaking the web

> Assessing the damage

> Approval from the powers that be

The science of outreach

Experimentation with implementation
When security meets compatibility

The process of breaking the web

- Assessing the damage
- Approval from the powers that be

The science of outreach

Experimentation with implementation
Measuring web incompatibility

- % affected page loads
- % affected connections
- % affected users
- # affected sites
“There are around 771 billion web pages viewed in Chrome every month (not counting other Chromium-based browsers). So seriously breaking even 0.0001% still results in someone being frustrated every 3 seconds.”

- “Blink Principles of Web Compatibility”
% of HTTP Archive URLs with mixed images in 2018
https://chromestatus.com/metrics/feature/timeline/popularity/614

Too common to block outright => autoupgrade to HTTPS instead
Measuring HTTPS autoupgrading

How much does it decrease breakage compared to blocking outright?

We have to actually roll it out to know.
Measuring HTTPS autoupgrading

Are http:// and https:// the same resource?

Analyze crawled resources to gain confidence.
Measuring HTTPS autoupgrading

Does a particular broken resource actually “count”?

Who knows?!
When security meets compatibility

The process of breaking the web

> Assessing the damage
> Approval from the powers that be

The science of outreach

Experimentation with implementation
Intent to Remove: <feature name>

Body:

**Primary eng (and PM) emails**
@chromium.org preferred over @google.com

**Summary**
Give a high-level description of your change.

**Motivation**
Explain why this feature should be removed.

**Interoperability and Compatibility Risk**
Describe the degree of interoperability and compatibility risk. For a feature that is also supported in some other engine, do they support eventual removal?

Edge: Supported/not supported, positive/neutral/negative to removal
Firefox: Supported/not supported, positive/neutral/negative to removal
Safari: Supported/not supported, positive/neutral/negative to removal

Please include links where possible.

**Alternative implementation suggestion for web developers**
If this feature goes away, what other techniques can developers use to achieve the same effects?

**Usage information from UseCounter**
How much of the web are you going to break? How seriously will the removal break sites?
If possible, please link to usage details on chromestatus.com/metrics (example link)
If you haven’t instrumented this feature yet, say so.

**Entry on the feature dashboard**
The feature dashboard is used to keep track of web-facing changes in Blink (and V8) that matter to developers. Make sure your change has an entry if you think it merits outreach to developers (e.g. inclusion in the Chromium Blog Beta posts). If there’s no entry, please explain why you think this change doesn’t need one (e.g. “small change”, “fits under an existing entry”). You may be asked to create one.
Mixed Content: The page at 'https://mixed.badssl.com/' was loaded over HTTPS, but requested an insecure image 'http://mixed.badssl.com/image.jpg'. This content should also be served over HTTPS.
Incompatible changes are feasible, if they’re carefully measured and considered.
When security meets compatibility

The process of breaking the web

> Assessing the damage
> Approval from the powers that be

The science of outreach

Experimentation with implementation
When security meets compatibility

The process of breaking the web

The science of outreach

Experimentation with implementation
“Make Notifications Great Again: Learning How to Notify in the Age of Large-Scale Vulnerability Scanning” Cetin et al.

“You’ve Got Vulnerability: Exploring Effective Vulnerability Notifications” Li et al.

“Remedying Web Hijacking: Notification Effectiveness and Webmaster Comprehension” Li et al.

“Didn’t You Hear Me? Towards More Successful Web Vulnerability Notifications” Stock et al.

“Hey, You Have a Problem: On the Feasibility of Large-Scale Web Vulnerability Notification” Stock et al.

“Do Malware Reports Expedite Cleanup? An Experimental Study” Vasek et al.

“The Matter of Heartbleed” Durumeric et al.
1 issue detected
Google has detected harmful content on some of your site's pages. We recommend that you remove it as soon as possible. Until then, browsers such as Google Chrome will display a warning when users visit or download certain files from your site.

Detected Issues
Malware

Description: These pages direct users to a site that serves malware. Learn more
Sample URLs: N/A

VS.
WHOIS emails
Down and to the right faster is better
Distrust of the Symantec PKI: Immediate action needed by site operators

March 7, 2018

Posted by Devon O’Brien, Ryan Sleevi, Emily Stark, Chrome security team

*Update October 17, 2018:* Chrome 70 has now been released to the Stable Channel, and users will start to see full screen interstitials on sites which still use certificates issues by the Legacy Symantec PKI. Initially this change will reach a small percentage of users, and then slowly scale up to 100% over the next several weeks.
Targeted outreach isn’t a standalone strategy for motivating server operators to remediate security risks.

Fixing HTTPS Misconfigurations at Scale: An Experiment with Security Notifications.
Eric Zeng, Frank Li, Emily Stark, Adrienne Porter Felt, Parisa Tabriz
When security meets compatibility

The process of breaking the web

The science of outreach

Experimentation with implementation
This site can’t provide a secure connection

rc4.badssl.com uses an unsupported protocol.

ERR_SSL_VERSION_OR_CIPHER_MISMATCH

Your connection is not private

Attackers might be trying to steal your information from expired.badssl.com (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_DATE_INVALID

Back to safety

This server could not prove that it is expired.badssl.com; its security certificate expired 2,062 days ago. This may be caused by a misconfiguration or an attacker intercepting your connection. Your computer’s clock is currently set to Wednesday, December 2, 2020. Does that look right? If not, you should correct your system’s clock and then refresh this page.

Proceed to expired.badssl.com [unsafely]

http.badssl.com

More aggressive

Less aggressive
Choosing a deprecation UI

- What types of sites affected
- How often the warning appears
- Risk to user
- How quickly servers will remediate
Faster remediation

Less warning fatigue
“TLS 1.0 and 1.1 will be disabled altogether in Chrome 81”

Modernizing Transport Security
October 15, 2018

Posted by David Benjamin, Chrome networking

*Updated on October 17, 2018 with details about changes in other browsers*

TLS (Transport Layer Security) is the protocol which secures HTTPS. It has a long history stretching back to the nearly twenty-year-old TLS 1.0 and its even older predecessor, SSL. Over that time, we have learned a lot about how to build secure protocols.
Phase 1

Phase 2

Days since warning UI launched
When security meets compatibility

The process of breaking the web

The science of outreach

Experimentation with implementation
Incompatible changes are possible.
Security-motivated deprecations are an active research area.
When **Security** Meets **Compatibility**

Emily Stark, Google Chrome

[estark@chromium.org](mailto:estark@chromium.org)

[@estark37](https://twitter.com/estark37)