

You are what you include:

Large-scale evaluation of remote
JavaScript inclusions

Nick Nikiforakis, Luca Invernizzi, Alexandros Kapravelos, Steven Van Acker,
Wouter Joosen, Christopher Kruegel, Frank Piessens, Giovanni Vigna



Introduction: my USB stick



Introduction: browsers don't care



Distrinet

You are what you include:

Large-scale evaluation of remote
JavaScript inclusions

Nick Nikiforakis, Luca Invernizzi, Alexandros Kapravelos, Steven Van Acker,
Wouter Joosen, Christopher Kruegel, Frank Piessens, Giovanni Vigna



Outline

- JavaScript in a browser

- ... and motivation for an experiment

- Our experiment

- Our results

- Some unsurprising results

- Some weirdness

- Countermeasures

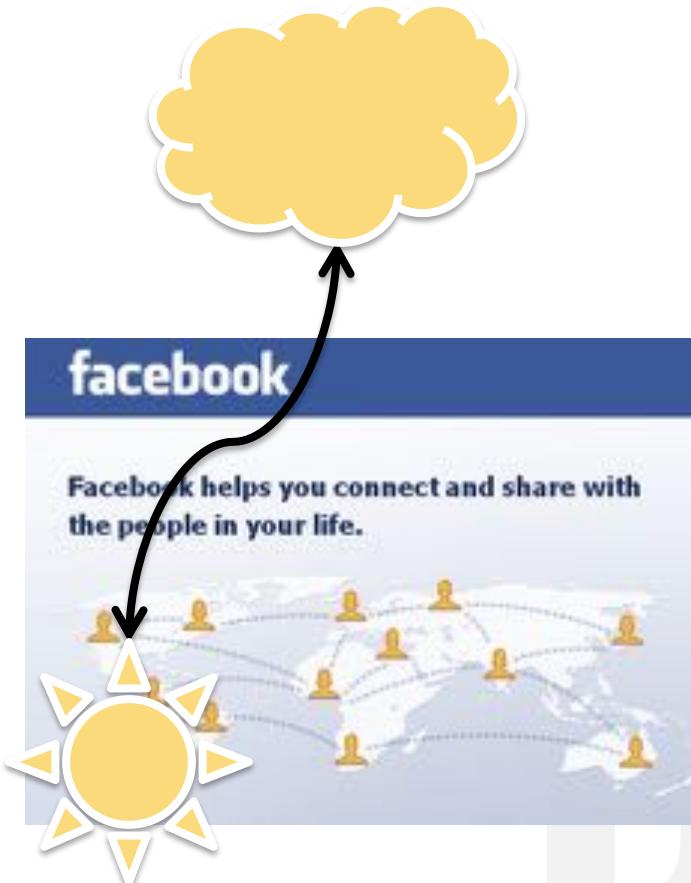


JavaScript in the browser

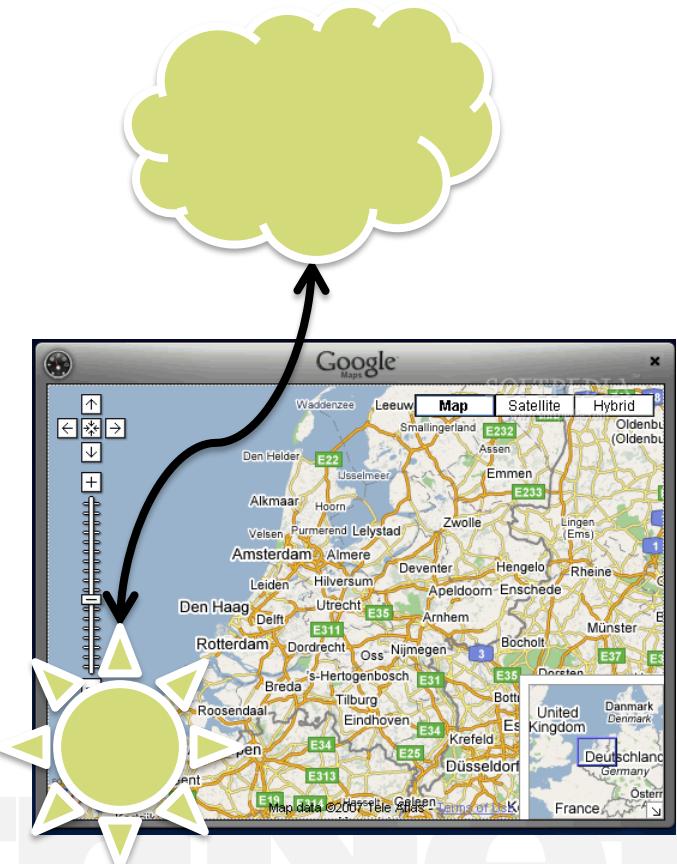
DistriNet

JavaScript in a browser: origins

Origin: http://facebook.com, 80



Origin: http://google-maps.com, 80

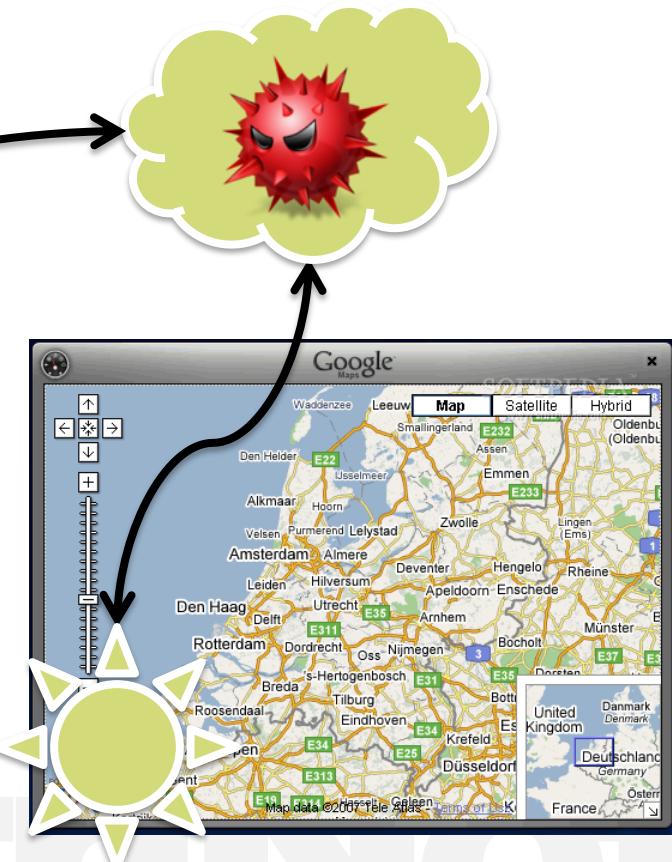


JavaScript in a browser: inclusions

Origin: http://facebook.com, 80



Origin: http://google-maps.com, 80



Motivation...

The screenshot shows the official qTip website at craigworks.com/qtip/. A red banner at the top states: "Warning: qTip 1.0 is no longer actively developed, and is superseded by qTip2. Click here to check it out!" Below this, the qTip logo is displayed, along with a green box containing the text: "qTip is a tooltip plugin for the jQuery framework. It's cross-browser, customizable and packed full of features! Do what are you waiting for? Join the qTip-community!" To the right, there's a graphic of a speech bubble with the word "neighborhood" and a list of features: Stylish, Customizable, Cross-browser, Degradable, Small Noise, each with a green checkmark. A navigation bar below includes Home, Features, Demos, Download, Documentation, and Forum. A large red box in the center contains a warning message: "If you downloaded the qTip2 library between 8th December 2011 and 10th of January 2012, please make sure to re-download the library as the site was compromised between these dates due to malicious code injected via a Wordpress bug. Apologies for any inconvenience caused by this, but as usual vulnerabilities like this can only be pro-actively remedied as they occur." Below this, a section titled "Download latest: 1.0.0-rc3" lists four options: Production - Uncompressed source code (38KB), Development - Uncompressed source code (43KB), Debugger - qTip debug plugin for easier development (5KB), and jQuery 1.3.2 - Tested and recommended for qTip (56KB). A large green button labeled "qTip 1.0.0-rc3" has an arrow pointing to it from the text "32 days...". To the right, there's a "Download" button with a file size of 56KB. Further down, sections for "GitHub Repository" and "License" are visible.

32 days...

qTip 1.0.0-rc3

Production - Uncompressed source code - 38KB

Development - Uncompressed source code - 43KB

Debugger - qTip debug plugin for easier development - 5KB

jQuery 1.3.2 - Tested and recommended for qTip - 56KB

GitHub Repository

Check out current and past releases of qTip 1.0 at the [GitHub repository](#).

License

What you need to know:

qTip is licensed under the open source [MIT license](#), which allows you to use the script for whatever you want, whenever you want, however you want.

The MIT License

Copyright © 2009 Craig Thompson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files, to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

1. The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

2. The name "Craig Thompson" may not be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



Our experiment

DistriNet



Our experiment: questions

Given that remote JS inclusions happen...

... Should sites be trusting remote providers?

- Which third-party vendors do they currently trust?
- Are JS providers capable of securing their website? What is the quality of maintenance profile of each JS provider?
 - Could a provider be attacked as a way of reaching a harder-to-get target?
- Are there attack vectors, in relation to remote inclusions, that we were not aware of ?
- How can one protect his web application?
 - Are coarse-grained sandboxes sufficient?



Our experiment: crawler

- Crawler requirements:
 - Download webpages
 - Log JavaScript inclusions
 - Execute JavaScript for dynamic inclusions
- HTMLUnit: JS-enabled headless browser in Java
- Queried Bing for max 500 pages of Alexa top 10000



Our experiment: some numbers

- Crawled over 3,300,000 pages belonging to the Alexa top 10,000
- Discovered:
 - 8,439,799 remote inclusions
 - 88.45% of Alexa top 10k uses at least 1 remote JS library
 - 301,968 unique JS files
 - 20,225 uniquely-addressed remote hosts

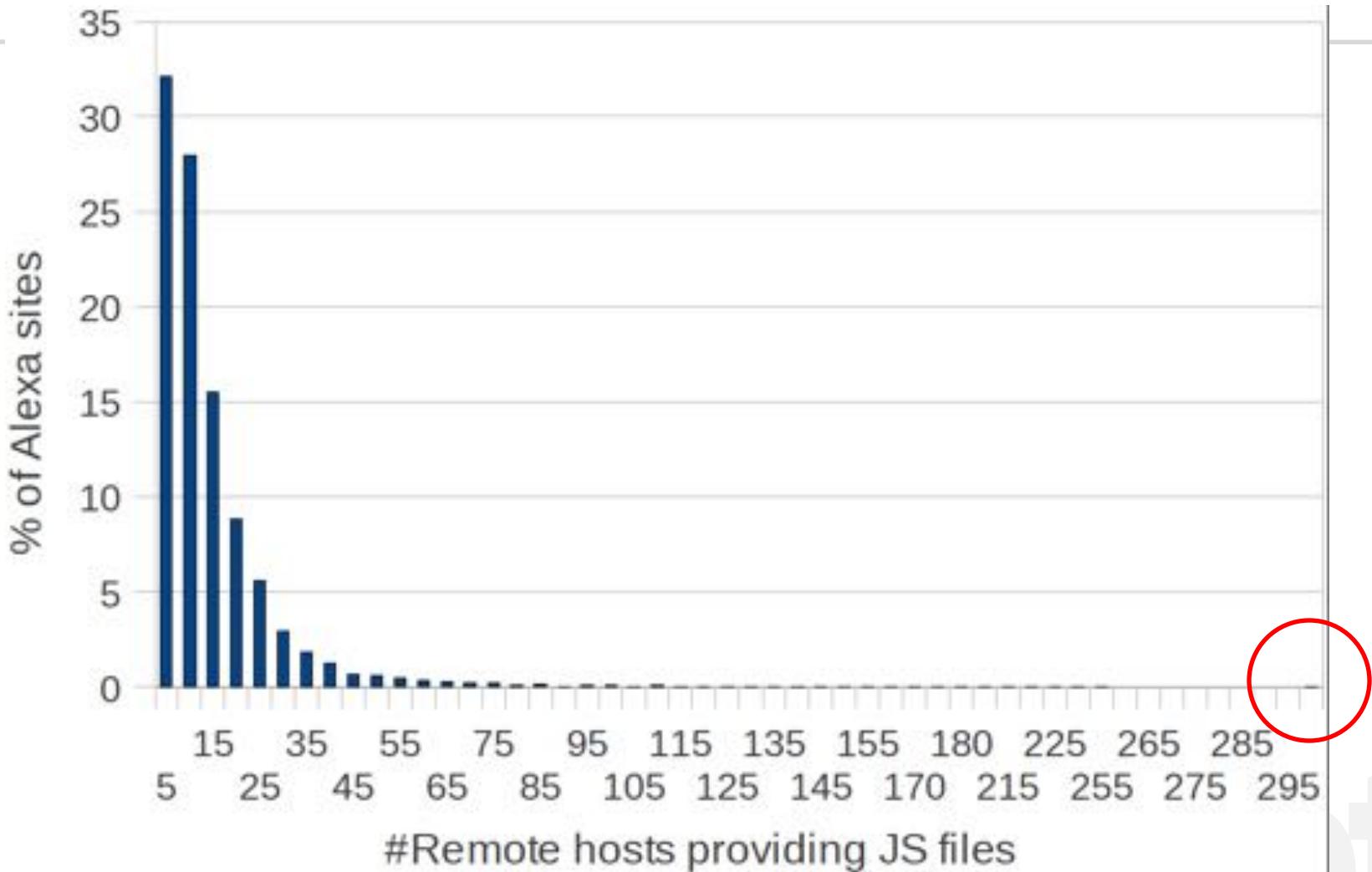




Results: unsurprisingly...

DistriNet

Results: how many remote hosts?



Results: Popular JavaScript includes

Offered service	JavaScript file	% Top Alexa
Web analytics	www.google-analytics.com/ga.js	68.37%
Dynamic Ads	pagead2.googlesyndication.com/pagead/show_ads.js	23.87%
Web analytics	www.google-analytics.com/urchin.js	17.32%
Social Networking	connect.facebook.net/en_us/all.js	16.82%
Social Networking	platform.twitter.com/widgets.js	13.87%
Social Networking & Web analytics	s7.addthis.com/ja/250/addthis_widget.js	12.68%
Web analytics & Tracking	edge.quantserve.com/quant.js	11.98%
Market Research	b.scorecardresearch.com/beacon.js	10.45%
Google Helper Functions	www.google.com/jsapi	10.14%
Web analytics	ssl.google-analytics.com/ga.js	10.12%

DistriNet

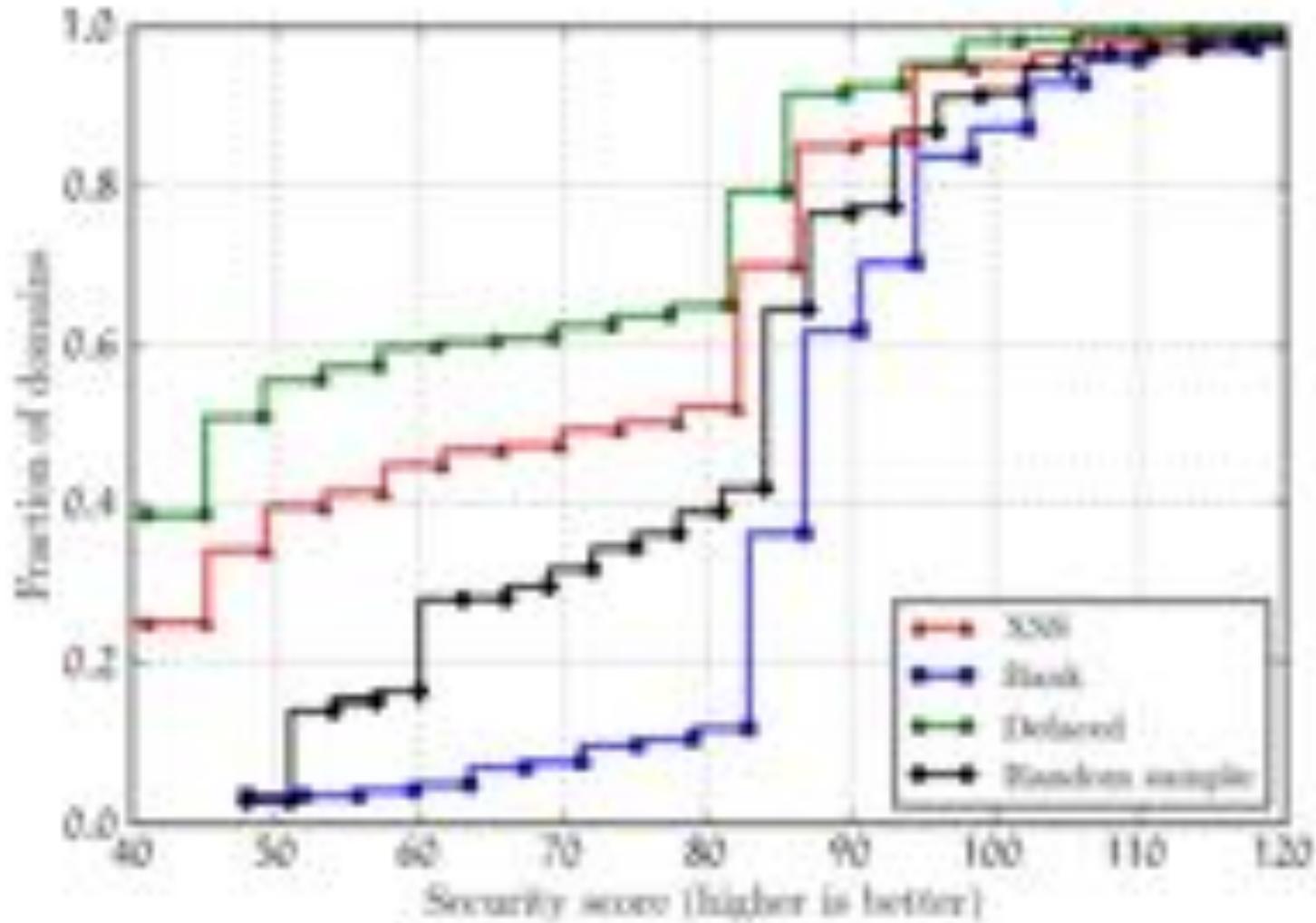


Results: quality of maintenance?

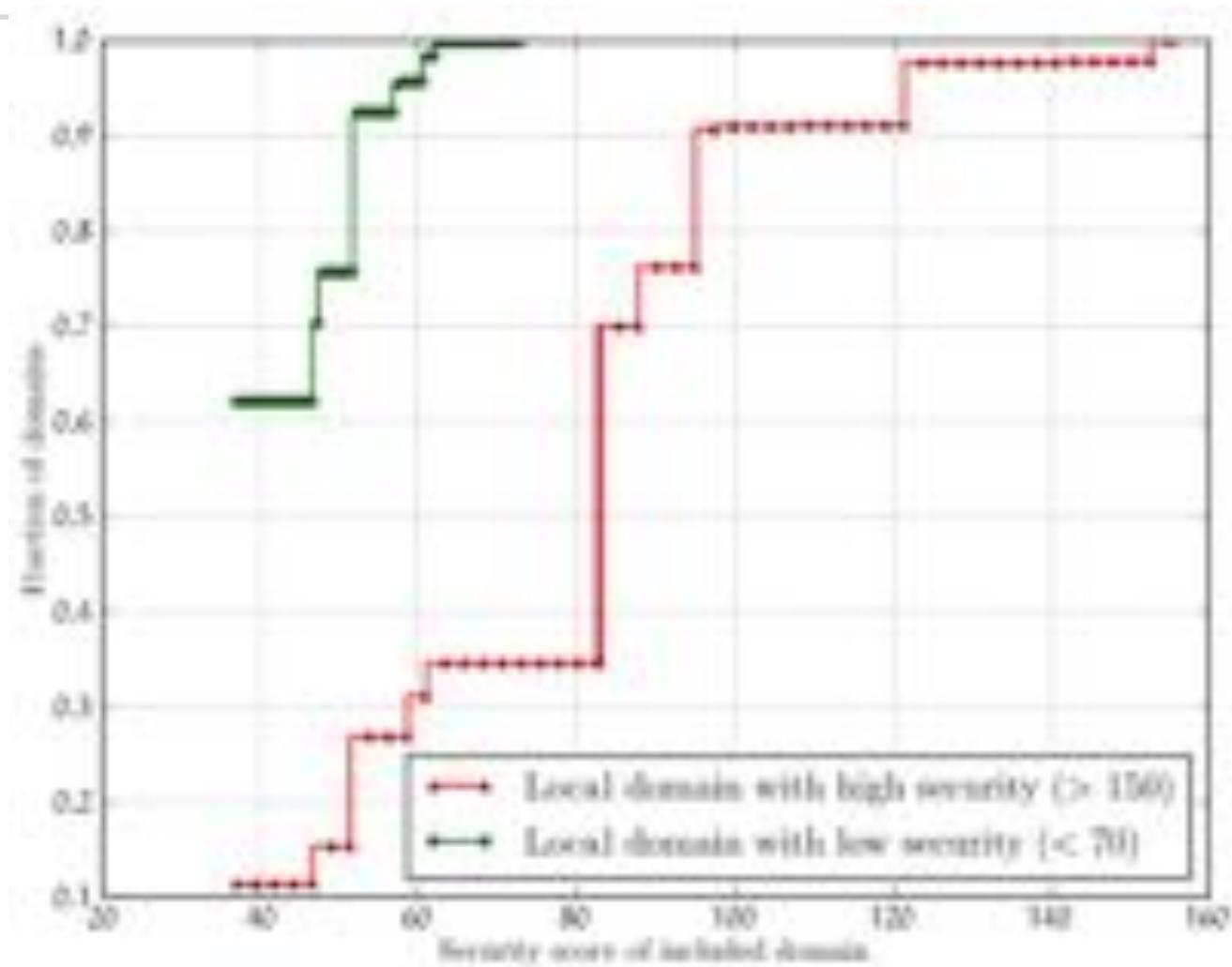
- Assumption: Unmaintained websites are easier to attack
- QoM indicator comprised of these factors:
 - Availability: DNS not expired, publicly-routable IP address
 - Cookies (at least one):
 - HttpOnly?
 - Secure?
 - Path & Expiration?
 - Anti-XSS & Anti-Clickjacking headers?
 - TLS/SSL implementation
 - Weak ciphers
 - Valid certificates
 - Strict Transport Protocol
 - Cache control when using TLS/SSL?
 - Outdated web servers?



Results: QoM in color!



Results: like attracts like



Results: weirdness!

Distrinet



Results: weirdness?

- In about 8.5 million records of remote inclusions, is there something that we didn't know?
- 4 Things! ☺
 - Cross-user & Cross-network Scripting
 - Stale domain-based inclusions
 - Stale IP-based inclusions
 - Typo-squatting Cross-Site Scripting



Weirdness: Cross-user Scripting

■ <script src=http://localhost/script.js>

- 133 records were found
- 131 specified a port (localhost:12345), always greater than 1024
- Attack:
 - Setup a web-server, listen to high ports, hack other users



Weirdness: Cross-network Scripting

- <script src=http://192.168.2.3/script.js>
 - 68 of them
 - Same as before, but now you just need to be in the same local network
- Who is doing that?
 - akamai.com
 - virginmobileusa.com
 - gc.ca (Government of Canada)



Weirdness: Stale IP-based remote inclusions

- What if the IP address of the host which you trust for JavaScript, changes?
 - The including page's scripts must also change
 - Do they?
- Manual analysis of the 299 pages
 - 39 addresses had:
 - a) Not changed
 - b) no longer provided JavaScript
 - a) In 89.74%, we got a “Connection Timeout”



Weirdness: Stale domain-based inclusions

- What happens when you trust a remote site and the domain of that site expires?
 - Anyone can register it, and start serving malicious JS
 - Equal in power to the, almost extinct, stored XSS
 - Try proving in court that someone hacked you with that
- 56 domains found, used in 47 sites
 - 6 were identified as special cases (TXSS)

Scared yet?



Weirdness: Typo-squatting XSS (TXSS)

- Unfortunately... developers are humans
 - <script src=[http://googlesyndicatio.com/...](http://googlesyndicatio.com/)>
- Typo-squatting
 - registering domains that are mistypes of popular domains
 - Serve ads, phishing, drive-by downloads etc. to users that mistype the domain



Weirdness: TXSS examples found...

Intended domain	Actual domain
googlesyndication.com	googlesyndicatio_.com
purdue.edu	purude_.edu
worldofwarcraft.com	worldofwaircraft.com
lesechos.fr	lessechos.fr
onegrp.com	onegrp.nl

Googlesyndicatio.com (15 days)	
Unique visitors	163,188
Including domains	1185
Including pages	21,830



Countermeasures

Distrinet



Countermeasures

- Problems with remote inclusions
 - Never the visitor's fault
 - A developer can mess up
 - Cross-user, cross-network and TXSS
 - The remote host can mess up
 - Low security, expiration of domain names
- How to protect one's self?
 - i. Sandbox remote scripts
 - ii. Download them locally



Countermeasures: sandboxing

- Is it feasible?
- What are the current requirements of legitimate scripts?
- Study the top 100
 - Automatically study each script
 - JavaScript wrappers + stack trace
 - Find out what sensitive resources they access
 - Cookies, Storage, Geolocation, Eval, document.write
 - Is containment possible?



... sandboxing: Access to resources

JS Action	# of Top scripts
Reading Cookies	41
document.write()	36
Writing Cookies	30
eval()	28
XHR	14
Accessing LocalStorage	3
Accessing SessionStorage	0
Geolocation	0

Coarse-grained sandboxing is useless here,
legitimate scripts and attackers act the same way ☹



Countermeasures: local copies

- Study the frequency of script modifications
 - Discover overhead for administrator
- Top 1,000 most-included scripts (803)
 - Download every script three consecutive times and remove the ones that changed all three times
 - Study the rest for a week
- 10.21% were modified
 - 6.97% were modified once
 - 1.86% were modified twice
 - 1.83% were modified three or more

89.79% was never modified!
96.76% at most once



Conclusions

DistriNet



Conclusions

- Remote inclusions mean, almost unconditional, trust
 - Think twice before including something from a remote host
- **Do NOT:**
 - Include from 127.0.0.1 or private networks
 - Include from IP addresses
 - Include from stale domains
 - Include from typodomains
 - Include from questionable JS providers
- **Do:**
 - Make local copies
 - Sandbox 3rd party JS if it is feasible
 - Have hope: sleep sound tonight



Thank you!

Questions?

