Mobile Friendly or Attacker Friendly? A Large-scale Security Evaluation of Mobile-first Websites

Tom Van Goethem

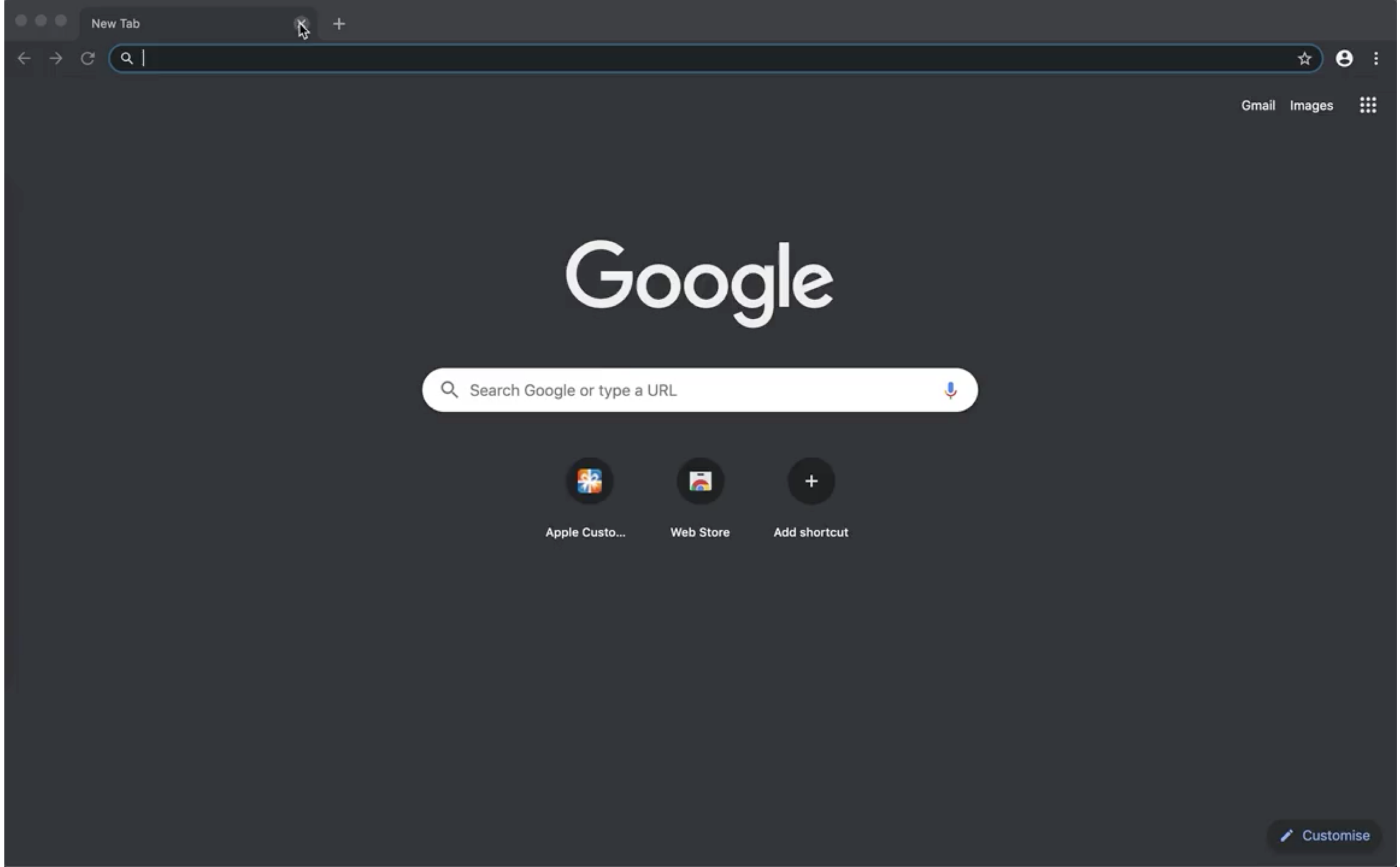
KU LEUVEN

@tomvangoethem

DistriNet

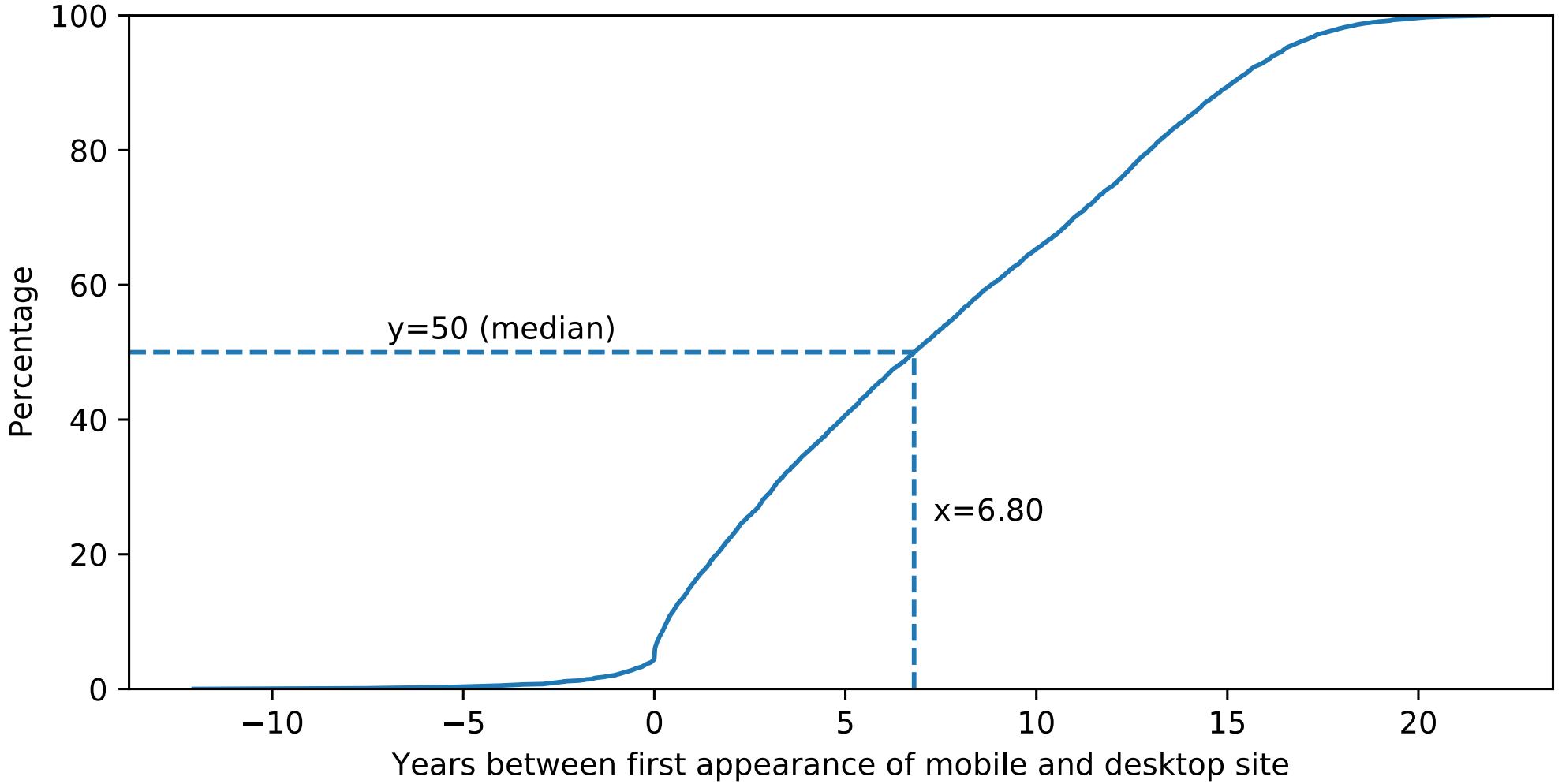
Mobile-first websites

- Websites developed specifically for mobile users
- Typically different sub-domain (mostly m.example.com)
  - Encountered various cases of abuse: phishing (m-twitter.com); scams/malware (m-amazon.com); malware (m-norton.com)


New Tab
Gmail
Images
Google
Search Google or type a URL
Apple Custo...
Web Store
Add shortcut
Customise

Mobile-first websites

- Websites developed specifically for mobile users
- Typically different sub-domain (mostly m.example.com)
  - Encountered various cases of abuse: phishing (m-twitter.com); scams/malware (m-amazon.com); malware (m-norton.com)
- Mostly developed several years after desktop site



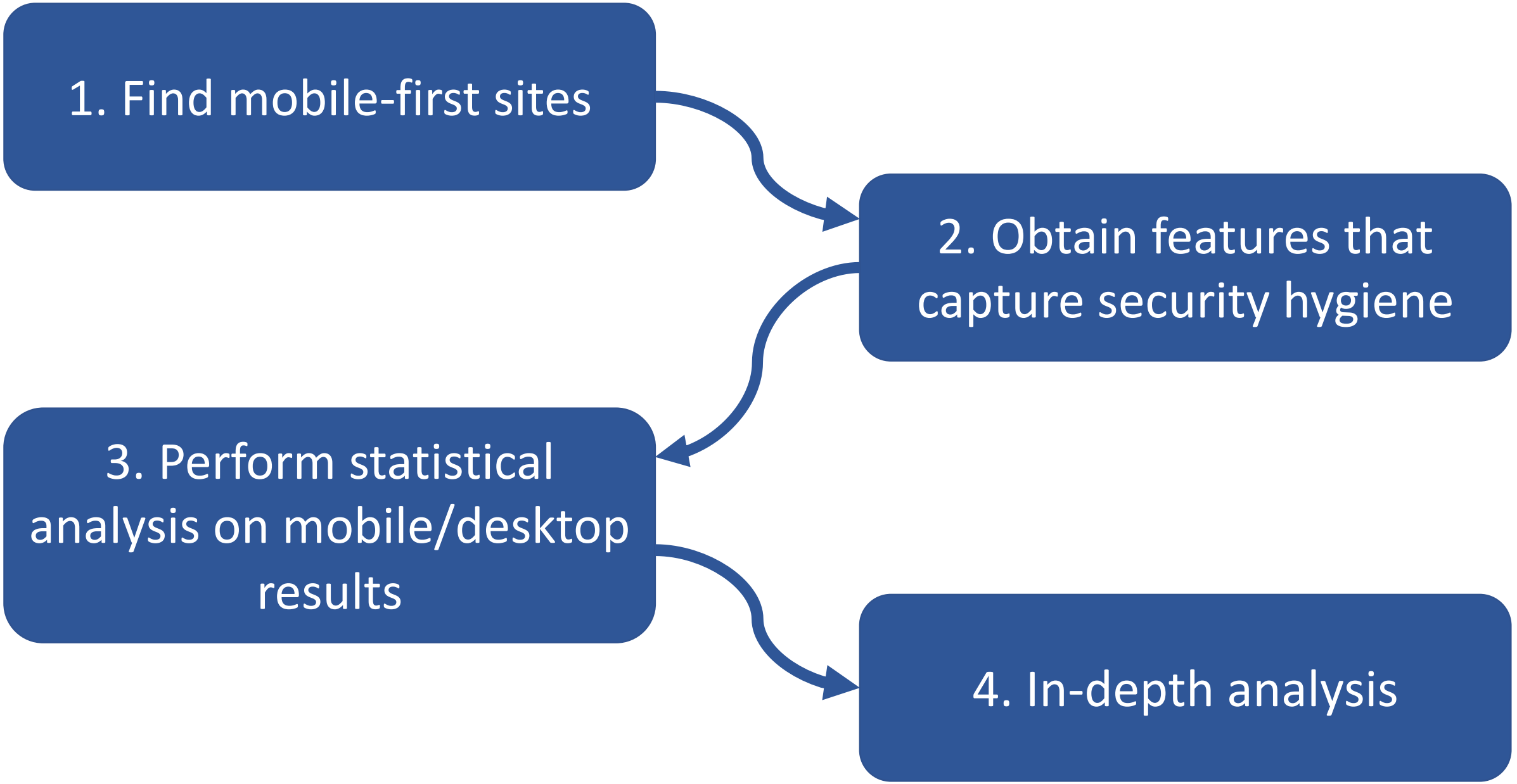
Based on Mementos (http://timetravel.mementoweb.org/)

Mobile-first websites

- Websites developed specifically for mobile users
- Typically different sub-domain (mostly m.example.com)
  - Encountered various cases of abuse: phishing (m-twitter.com); Bitcoin scam (m-amazon.com); malware (m-norton.com)
- Mostly developed several years after desktop site
- Provide unique viewpoint on how security is handled
  - Assumption: if security features are considered during design-time, mobile should have higher adoption on certain (newer) features

Web security

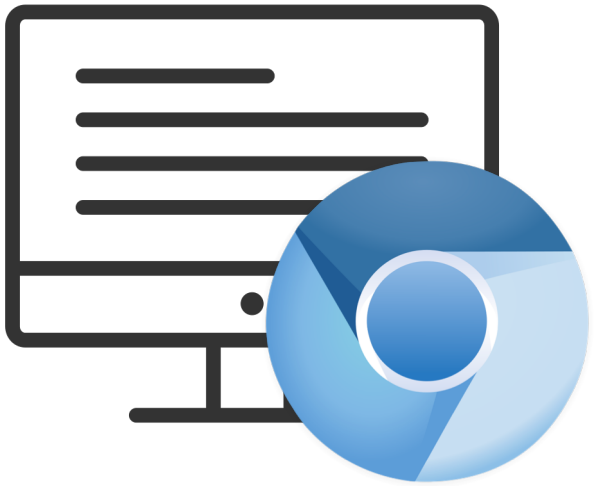
- Despite good security practices, websites may still suffer from vulnerabilities
- Several defense mechanisms exist to prevent vulnerabilities or limit their impact
  - X-Frame-Options to prevent clickjacking attacks
  - HttpOnly attribute to prevent cookies to be stolen in XSS attack
- Are these mechanisms applied ad hoc, or universally across all assets?
- Are defense mechanisms considered during design time, or only applied reactively?


1. Find mobile-first sites
2. Obtain features that capture security hygiene
3. Perform statistical analysis on mobile/desktop results
4. In-depth analysis

1. Find mobile-first sites

if desktop browser is redirected to different (sub)domain than mobile browser

=> possible mobile-first site



headless Chromium

headless Chromium, instrumented to emulate mobile device

(screen dimensions, UA, built-in mobile emulator, …)

Finding mobile-first sites

- Visited home page of Tranco top 1M sites
- 15,541 domains redirected to different (sub-)domain
- Filter out irrelevant websites
  - 45 redirected to Google Play store
  - 268 did not have accessible mobile/desktop site
  - 2,173 empty sites
  - 820 duplicate sites (e.g. google.com/google.nl) => found by perceptual hash
  - 1,471 non-unique mobile-first sites (redirect to domain already in dataset)
  - 512 were compromised and redirected mobile users to suspicious domain
- In total: 10,222 mobile-first sites

Finding mobile-first sites

- Discovered 512 compromised sites
  - Most likely caused by vulnerable WordPress plugin
  - Attacker uploaded .htaccess file
  - If visitor's user agent string was mobile: redirect to malicious site (wwwtype.ru) -> redirected to porn site with referral in URL
  - Compromise is less likely to be detected (only if administrator visits site on mobile device)

2. Obtain features that capture security hygiene

Obtaining features that capture security hygiene

- Website's security: vulnerabilities + defense mechanisms that defend against them
- Finding vulnerabilities at large scale
  - Often requires intrusive techniques
  - Might be difficult to detect
  - Detection site-specific (=> no universal method)
- Defense mechanisms
  - Usage recommended to completely defend or limit attack consequences
  - Communicated to browser => easier to detect (response headers)

Security features (1)

- XSS in-depth defenses
  - HttpOnly attribute on cookies (=> cookie not accessible from JS)
  - Content-Security-Policy (=> defines which sources can execute JS)
- CSRF defense
  - Unique token in form (in our study: only against login-CSRF)
- Clickjacking defense
  - X-Frame-Options (value: DENY/SAMEORIGIN)
  - Content Security Policy (frame-ancestors directive)

Security features (2)

- Mime-sniffing defense
  - X-Content-Type-Options (nosniff: instruct browser to not try to determine content type)
- Man-in-the-Middle defenses/issues
  - Presence of HTTPS
  - Secure attribute on cookie (cookie not sent over insecure connections)
  - Strict-Transport-Security (all following connections are made over HTTPS)
  - Mixed content (HTTP resources on HTTPS page)
  - SSL stripping (form to HTTPS on HTTP page)
  - Insecure content submission (form to HTTP on HTTPS page)

Security features (3)

- Defense of including dynamic content
  - sandbox attribute on iframe (determine what iframe is allowed to do)
  - integrity attribute for scripts (scripts served without unexpected changes)
- Prevent leaking potentially sensitive information
  - Referrer-Policy header: control the referrer information sent to 3rd parties

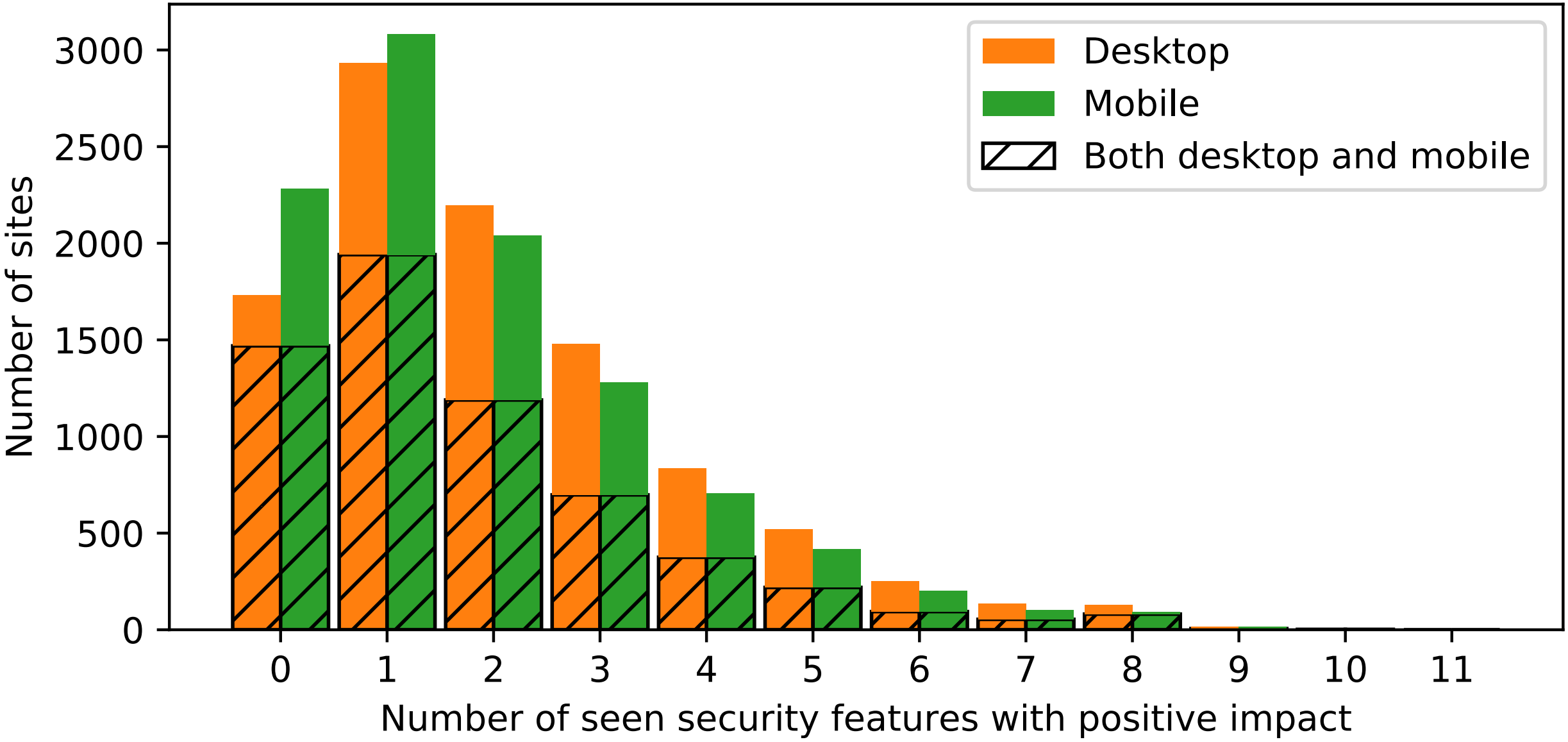
Security features

- Use of defense mechanisms is not always required
  - Depends on the mechanism and application
  - Mobile/desktop are likely to offer similar functionality
  - Comparison should be pair-wise (m.example.com vs example.com)
- In our case: we want to estimate *security consciousness*
  - Presence of security features is a good indicator
  - At least website administrator considered it
- Total: 11 defense mechanisms & 4 potential weaknesses

Obtaining data

- Up to 20 page visits per site per browser type
  - 191,237 for mobile
  - 195,487 for desktop
- Instrumented headless Chrome with/without mobile emulation
  - Used customized distributed crawler

3. Perform statistical analysis


Desktop
Mobile
Both desktop and mobile
Number of sites
3000
2500
2000
1500
1000
500
0
0
1
2
3
4
5
6
7
8
9
10
11
Number of seen security features with positive impact

Statistical analysis: questions

- Are mobile sites more secure than desktop sites?
- Which security features are more prevalent on mobile sites compared to their desktop counterpart?
- Are the features introduced because of security effort made by web developer?

Statistical analysis: approach

- Wilcoxon signed-rank test
  - Statistical test for paired samples (mobile vs desktop) 👍
  - Does not rely on a priori assumptions on the distribution of data 👍
- Mediation analysis
  - Determine what the effect of a web app's complexity is on its security feature usage
  - Complexity is quite vague; we consider it feature-specific e.g. # HttpOnly cookies compared to total number of cookies


$\lim\left(1+\frac{1}{n}\right)^n = e$

Table 2: Summary of the results of our statistical analysis on pairs of desktop and mobile sites. Stars indicate statistical significance of the test scores (*: p<0.05; **: p<0.01; ***: p<0.001; ****: p<0.0001). The sign of the (in)direct effect indicates whether it goes in the same (+) or opposite (-) direction as the total effect. A mediator is present if zero lies outside the confidence interval of the indirect effect.

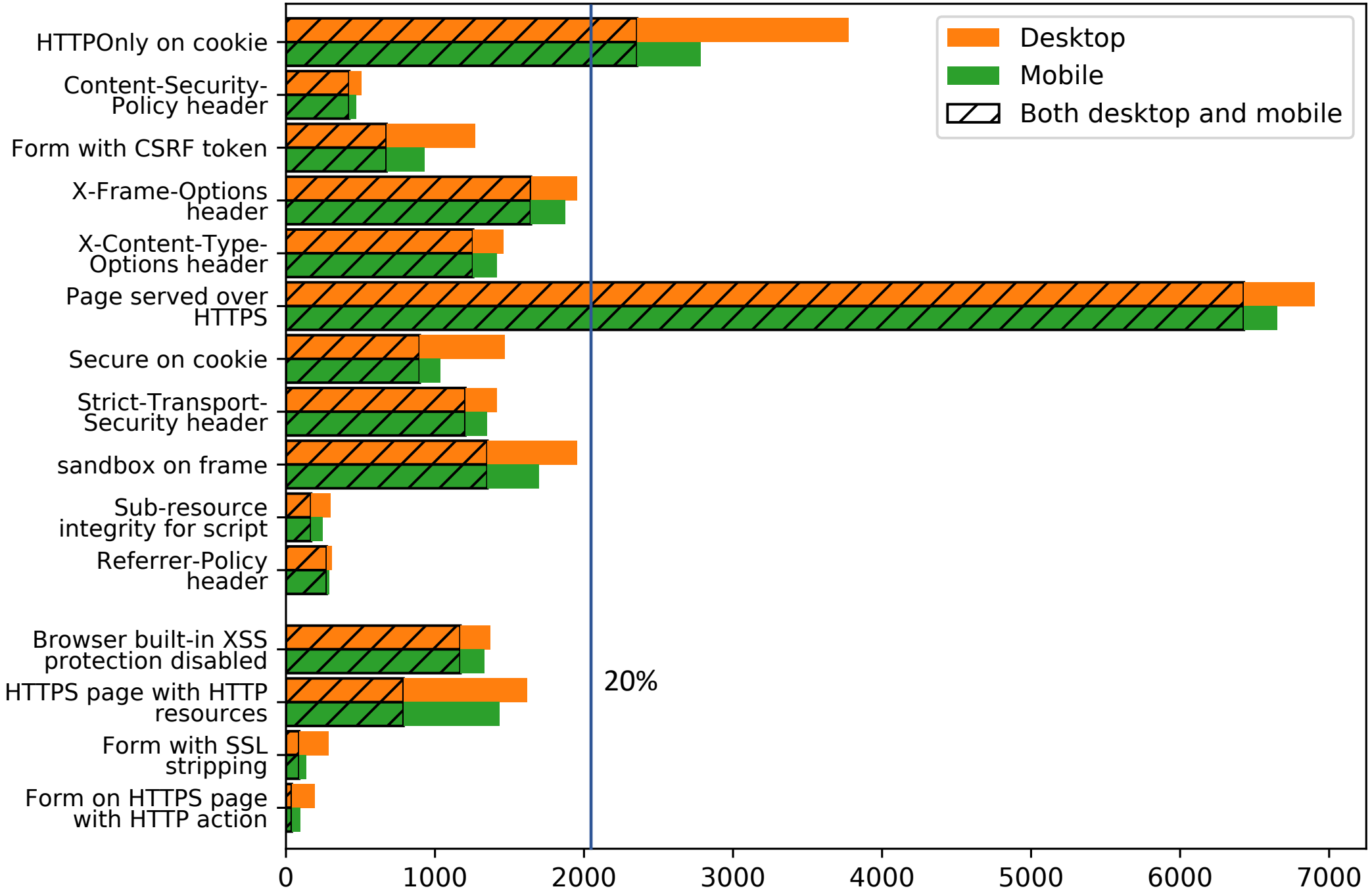
							Mediation analysis						
					Wilcoxon		Total effect		Direct effect		Indirect effect		
Feature		Pairs	Different	Direction	Corr.	p	Size	p	Size	p	Size	Confidence interval	Mediator
Cross-site scripting													
`HTTPOnly` on cookie	(+)	8231	3273	desktop	0.142	0.000 (****)	0.47%	0.055	+1.13%	0.000 (****)	-0.66%	[-0.83, -0.51]	Yes
`Content-Security-Policy` header	(+)	10222	299	desktop	0.013	0.847	0.14%	0.105	+0.11%	0.209	+0.03%	[+0.00, +0.06]	Yes
Browser built-in XSS protection disabled	(-)	10222	725	desktop	0.053	0.220	0.10%	0.486	+0.08%	0.578	+0.02%	[-0.02, +0.06]	No
Cross-site request forgery													
Form with CSRF token	(+)	7697	1195	desktop	0.026	0.440	0.16%	0.422	+0.12%	0.567	+0.04%	[-0.11, +0.22]	No
Clickjacking													
`X-Frame-Options` header	(+)	10222	1146	desktop	0.070	0.041 (*)	0.35%	0.034 (*)	+0.32%	0.054	+0.03%	[-0.02, +0.08]	No
Content-sniffing													
`X-Content-Type-Options` header	(+)	10222	772	desktop	0.055	0.189	0.14%	0.320	+0.12%	0.415	+0.02%	[-0.02, +0.07]	No
Man-in-the-middle attacks													
Page served over HTTPS	(+)	10222	2063	desktop	0.222	0.000 (****)	2.19%	0.000 (****)	+2.11%	0.000 (****)	+0.08%	[+0.01, +0.16]	Yes
`Secure` on cookie	(+)	8231	1312	desktop	0.263	0.000 (****)	0.85%	0.000 (****)	+1.17%	0.000 (****)	-0.32%	[-0.41, -0.24]	Yes
`Strict-Transport-Security` header	(+)	6428	639	desktop	0.139	0.002 (**)	1.09%	0.000 (***)	+1.13%	0.000 (***)	-0.03%	[-0.15, +0.06]	No
HTTPS page with HTTP resources	(-)	6428	1833	mobile	0.031	0.245	0.55%	0.038 (*)	+0.61%	0.020 (*)	-0.07%	[-0.15, +0.01]	No
Form with SSL stripping	(-)	5183	201	desktop	0.203	0.012 (*)	0.25%	0.039 (*)	+0.25%	0.047 (*)	+0.00%	[-0.03, +0.03]	No
Form on HTTPS page with HTTP action	(-)	6428	192	desktop	0.391	0.000 (****)	0.35%	0.001 (***)	+0.38%	0.000 (***)	-0.03%	[-0.07, +0.00]	No
Including untrusted content													
`sandbox` on frame	(+)	6893	1986	mobile	0.080	0.002 (**)	0.82%	0.000 (****)	+0.53%	0.010 (*)	+0.29%	[+0.15, +0.48]	Yes
Sub-resource integrity for script	(+)	10180	374	mobile	0.060	0.312	0.01%	0.360	+0.01%	0.557	+0.00%	[+0.00, +0.01]	Yes
Information leakage													
`Referrer-Policy` header	(+)	10222	120	desktop	0.429	0.000 (****)	0.19%	0.000 (***)	+0.18%	0.000 (***)	+0.02%	[-0.00, +0.04]	No

Statistical analysis results

Table 2: Summary of the results of our statistical analysis on pairs of desktop and mobile sites. Stars indicate statistical significance of the test scores (*: $p<0.05$; **: $p<0.01$; ***: $p<0.001$; ****: $p<0.0001$). The sign of the (in)direct effect indicates whether it goes in the same (+) or opposite (-) direction as the total effect. A mediator is present if zero lies outside the confidence interval of the indirect effect.

					Wilcoxon		Mediation analysis					
							Total effect		Direct effect		Indirect effect		
Feature		Pairs	Different	Direction	Corr.	*p*	Size	*p*	Size	*p*	Size	Confidence interval	Mediator
Cross-site scripting													
`HTTPOnly` on cookie	(+)	8231	3273	desktop	0.142	0.000 (****)	0.47%	0.055	+1.13%	0.000 (****)	-0.66%	[-0.83, -0.51]	Yes
`Content-Security-Policy` header	(+)	10222	299	desktop	0.013	0.847	0.14%	0.105	+0.11%	0.209	+0.03%	[+0.00, +0.06]	Yes
Browser built-in XSS protection disabled	(-)	10222	725	desktop	0.053	0.220	0.10%	0.486	+0.08%	0.578	+0.02%	[-0.02, +0.06]	No
Cross-site request forgery													
Form with CSRF token	(+)	7697	1195	desktop	0.026	0.440	0.16%	0.422	+0.12%	0.567	+0.04%	[-0.11, +0.22]	No
Clickjacking													
`X-Frame-Options` header	(+)	10222	1146	desktop	0.070	0.041 (*)	0.35%	0.034 (*)	+0.32%	0.054	+0.03%	[-0.02, +0.08]	No
Content-sniffing													
`X-Content-Type-Options` header	(+)	10222	772	desktop	0.055	0.189	0.14%	0.320	+0.12%	0.415	+0.02%	[-0.02, +0.07]	No
Man-in-the-middle attacks													
Page served over HTTPS	(+)	10222	2063	desktop	0.222	0.000 (****)	2.19%	0.000 (****)	+2.11%	0.000 (****)	+0.08%	[+0.01, +0.16]	Yes
`Secure` on cookie	(+)	8231	1312	desktop	0.263	0.000 (****)	0.85%	0.000 (****)	+1.17%	0.000 (****)	-0.32%	[-0.41, -0.24]	Yes
`Strict-Transport-Security` header	(+)	6428	639	desktop	0.139	0.002 (**)	1.09%	0.000 (***)	+1.13%	0.000 (***)	-0.03%	[-0.15, +0.06]	No
HTTPS page with HTTP resources	(-)	6428	1833	mobile	0.031	0.245	0.55%	0.038 (*)	+0.61%	0.020 (*)	-0.07%	[-0.15, +0.01]	No
Form with SSL stripping	(-)	5183	201	desktop	0.203	0.012 (*)	0.25%	0.039 (*)	+0.25%	0.047 (*)	+0.00%	[-0.03, +0.03]	No
Form on HTTPS page with HTTP action	(-)	6428	192	desktop	0.391	0.000 (****)	0.35%	0.001 (**)	+0.38%	0.000 (***)	-0.03%	[-0.07, +0.00]	No
Including untrusted content													
sandbox on frame	(+)	6893	1986	mobile	0.080	0.002 (**)	0.82%	0.000 (****)	+0.53%	0.010 (*)	+0.29%	[+0.15, +0.48]	Yes
Sub-resource integrity for script	(+)	10180	374	mobile	0.060	0.312	0.01%	0.360	+0.01%	0.557	+0.00%	[+0.00, +0.01]	Yes
Information leakage													
`Referrer-Policy` header	(+)	10222	120	desktop	0.429	0.000 (****)	0.19%	0.000 (***)	+0.18%	0.000 (***)	+0.02%	[-0.00, +0.04]	No

- Security features more prevalent on desktop
  - Effect of device most outspoken for MitM-related features
- For most features: effect of device limited & often statistically insignificant
  - Indicates consistent application of security features across desktop & mobile sites
- Mediation analysis: complexity of website has significant indirect effect on cookie/frame/HTTPS related features
  - E.g. desktop sites have more cookies => more likely to have HttpOnly/Secure cookie


HTTPOnly on cookie
Content-Security-Policy header
Form with CSRF token
X-Frame-Options header
X-Content-Type-Options header
Page served over HTTPS
Secure on cookie
Strict-Transport-Security header
sandbox on frame
Sub-resource integrity for script
Referrer-Policy header
Browser built-in XSS protection disabled
HTTPS page with HTTP resources
Form with SSL stripping
Form on HTTPS page with HTTP action
20%
Desktop
Mobile
Both desktop and mobile
0
1000
2000
3000
4000
5000
6000
7000

4. In-depth analysis

In-depth analysis

- Content Security Policy
  - Complex mechanism: many different directives, has impact on many site features
  - Comparison with prior study [1]
- HTTPS adoption
  - Most effect of website type (mobile/desktop)
  - Most prevalent feature on desktop + mobile

[1] Weichselbaum et al. CSP is dead, long live CSP! On the insecurity of whitelists and the future of content security policy. CCS'16

In-depth analysis: Content Security Policy

- 502 desktop sites, 482 mobile sites
- If enabled: most pages covered
  - 78.94% for desktop, 82.01% for mobile
  - Typically (90+%) same policy on all pages
- Almost all suffer from high-severity issues
  - Based on Google's CSP evaluator
  - Only 2 desktop and 3 mobile sites without high-severity issues
  - Mostly due to unsafe-inline

Directive	mobile %	desktop %	[1] %
report-uri	41.27	42.60	41.42
default-src	34.62	33.94	85.71
block-all-mixed-content	33.73	27.08	1.20
script-src	29.76	25.99	86.78
frame-ancestors	28.17	22.74	8.12
referrer	27.78	25.27	1.61
img-src	26.59	25.99	77.58
style-src	23.41	22.02	78.22
font-src	19.84	16.62	66.55
connect-src	19.44	20.22	54.37

NOTE: [1] studied different dataset

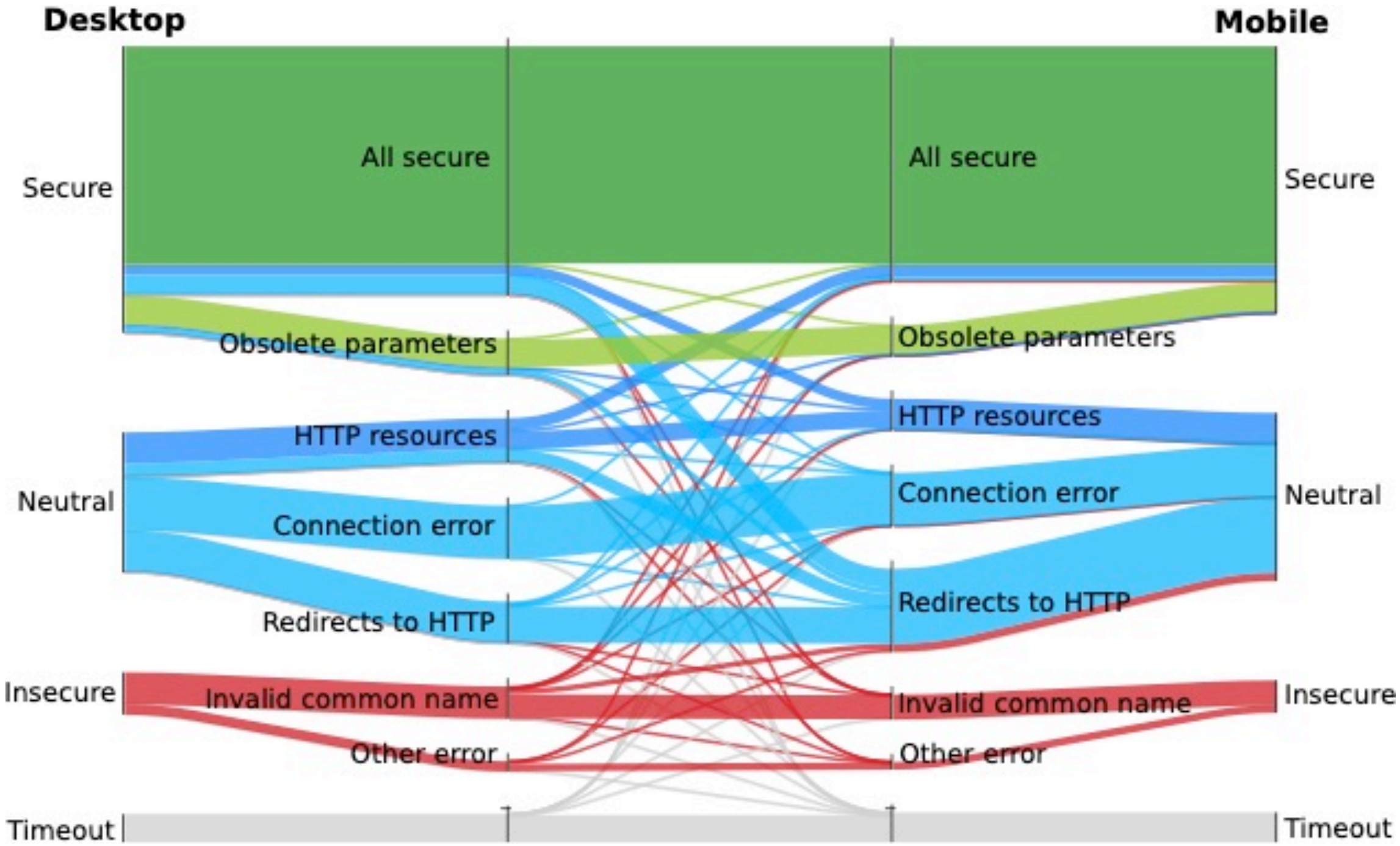
- default-src: used much less than in 2016 study
- New directives: more adoption

  (block-all-mixed-content, referrer, frame-ancestors)

[1] Weichselbaum et al. CSP is dead, long live CSP! On the insecurity of whitelists and the future of content security policy. CCS'16

In-depth analysis: HTTPS adoption

- 6,428 (62.88%) websites adopt HTTPS on mobile & desktop
- Most (69.59%) have a secure implementation on both sites
- 665 desktop sites are secure, whereas mobile version is not
  - Mobile redirects to HTTP
- 386 mobile sites are secure, whereas desktop version is not
  - Mixed content on desktop


Desktop
Mobile
Secure
Neutral
Insecure
Timeout
All secure
Obsolete parameters
HTTP resources
Connection error
Redirects to HTTP
Invalid common name
Other error
All secure
Obsolete parameters
HTTP resources
Connection error
Redirects to HTTP
Invalid common name
Other error
Secure
Neutral
Insecure
Timeout

What does this tell us?

Conclusions from analysis

- Adoption of security mechanisms is similar on mobile & desktop
  - In terms of type of security mechanism
  - In terms of usage/implementation of security mechanism
- => security mechanisms likely not considered at design time
- Overall, mobile sites have slightly fewer security mechanisms
  - Less need for it? Less interest in securing them?
- Adoption of security features is low (for most features: 5-20%)

Attribution of feature presence

- If a site contains an iframe with a sandbox attribute, is this because of efforts made by the web developer?
- We performed an analysis for several features
  - Group instances together based on common characteristic (e.g. resource location or HTML attributes)
- Presence of certain features is highly related to 3rd parties or libraries
  - Certain <iframe>s are always served with sandbox attribute (e.g. recaptcha challenge)
  - Almost all ASP.NET_SessionId cookies had HttpOnly attribute
- Not all though: only 16.4% of SRI usages can be attributed to 3rd parties

How do we move on from here?

The power of defaults

- Almost all sites stick to secure defaults (e.g. HttpOnly cookie)
- When creating a new application, what if…
  - We start with all security features enabled
  - And only make exceptions if needed
  - Making an exception would force web developer to learn about the security feature and the possible consequences of disabling it
- Still some limitations (e.g. what with new features), but still significantly better than nothing…

What else can we do?

- Your application will most likely have vulnerabilities
- Security features can make exploitation impossible or at least more difficult
  - Increase effort/costs of the attacker
- Try to **reduce threat surface** for users as much as possible
- For mobile-first sites: automatically **redirect desktop users to desktop site**; mobile users to mobile site
  - Vulnerabilities in desktop site shouldn't affect mobile users and vice versa

Conclusion

Conclusion

- We performed a large-scale comparative study on mobile-first sites
  - Provides unique viewpoint on security adoption of organization
- Desktop sites have slightly higher adoption of security features
- Security features typically applied universally across all website assets
  - Indicates defenses not applied at design time
- Complexity has influence on prevalence of features
  - Requires mediation analysis

Conclusion

- Usage of security features can not always be attributed to conscious choices made by web developers
- Overall usage of security mechanisms is quite low
- Secure defaults can be very effective
- Don't expose users to unnecessary vulnerabilities
  - Desktop users => desktop site
  - Mobile users => mobile site

For more details: take a look at our paper

Mobile Friendly or Attacker Friendly? A Large-scale Security Evaluation of Mobile-first Websites

Tom Van Goethem*
imec-DistriNet, KU Leuven
tom.vangoethem@cs.kuleuven.be

Victor Le Pochat*
imec-DistriNet, KU Leuven
victor.lepochat@cs.kuleuven.be

Wouter Joosen
imec-DistriNet, KU Leuven
wouter.joosen@cs.kuleuven.be

ABSTRACT

In the last few years, traffic generated by mobile devices has surpassed desktop visits. In order to provide users with the best browsing experience, many website owners specifically tailor their site to mobile devices. While some websites make use of reactive designs, many others opt to create an entirely new "mobile-first" website, typically hosted on a subdomain of the desktop site. These mobile-first sites provide a unique viewpoint on how organizations handle allow us to make more general observations on how security is applied on the web. More precisely, mobile-first sites are developed by and/or for the same organization as the desktop version, but at a much later time than the desktop version; by analyzing the similarities and differences, we obtain insights on whether security features are typically retroactively applied in an ad-hoc manner, or whether they are the result of a more structured and thorough approach during the initial development phase.

https://tom.vg/papers/mobile-first.pdf

† Signed copies available for purchase after session (while stocks last)

Questions?

@tomvangoethem

DistriNet