Upgrading HTTPS in Midair:
An Empirical Study on Strict Transport Security (HSTS) and Public Key Pinning (HPKP) in the Wild

Michael Kranch
Joseph Bonneau

NDSS 2015
February 9 2015

Takeaways Up Front

- HSTS and HPKP are being used
  - *Used by over 12,500 sites (~1% of top million)*
  - *500% increase in preload list in the past 4 months*

  Still better than HTTPS only
- Many errors in implementation
  - *59.5% of sites set HSTS incompletely*
  - *47.8% still leak cookies due to HSTS/HPKP hole*

- Standards contribute to issues
  - *Better defaults*
  - *Developer testing during process*

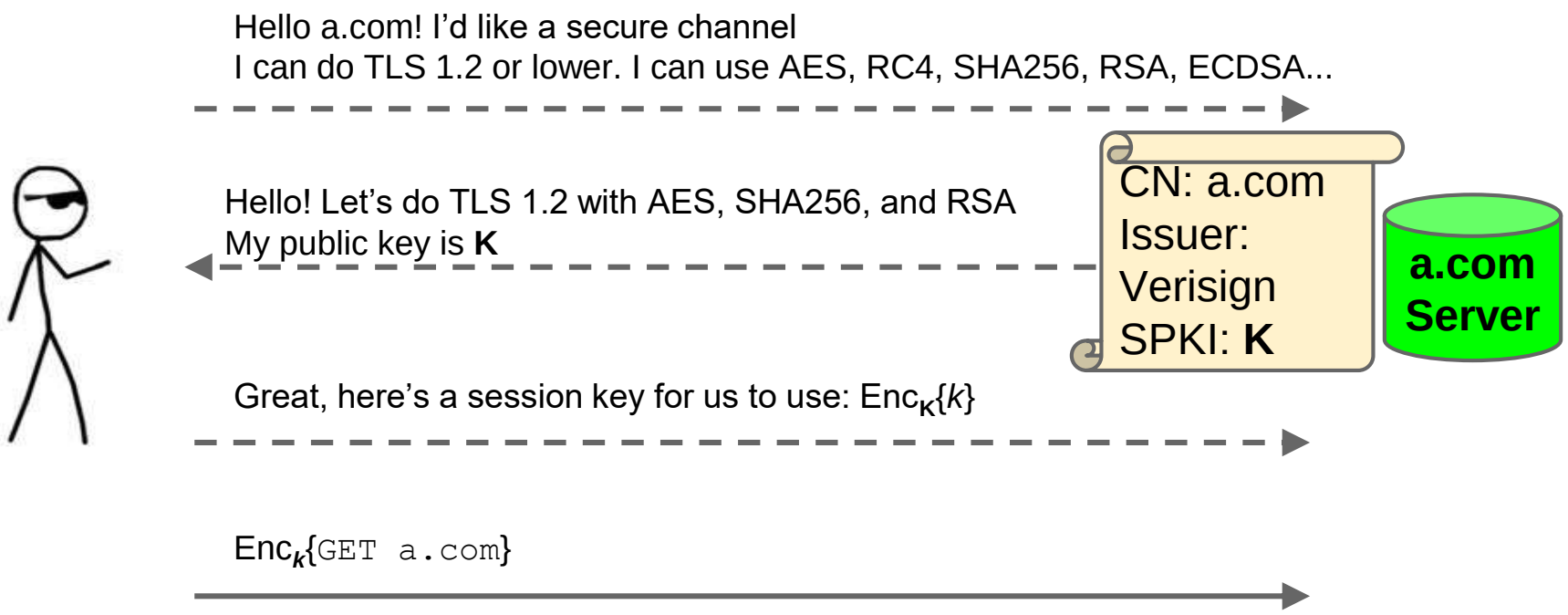
Agenda

- Background on HSTS and Pinning
- Study methodology
- Current deployment
- *High-level overview of errors*
- Takeaways from study

Check out the paper for more details:

http://www.jbonneau.com/doc/KB15-NDSS-hsts_pinning_survey.pdf

TLS in one slide



HTTPS attacks in *practice*

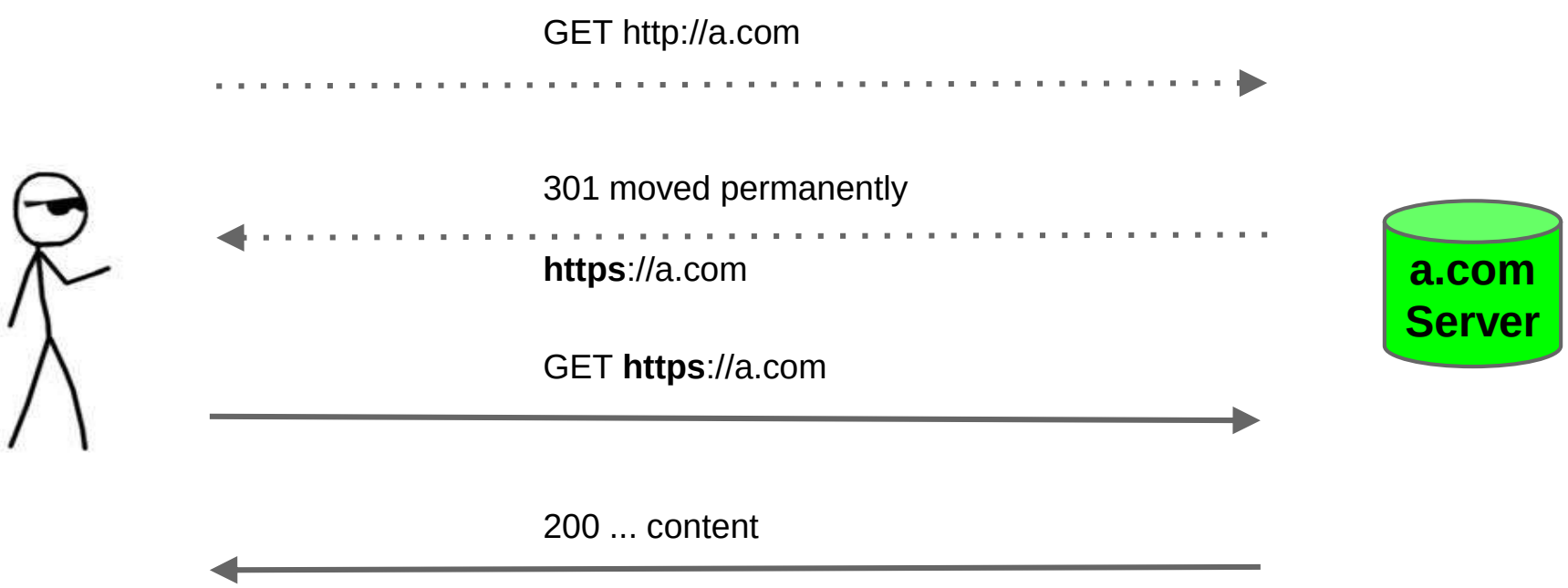
- Attacks against TLS
  - *Implementation attacks*
  - *Protocol flaws*
  - *Compromise of private keys*

POODLE

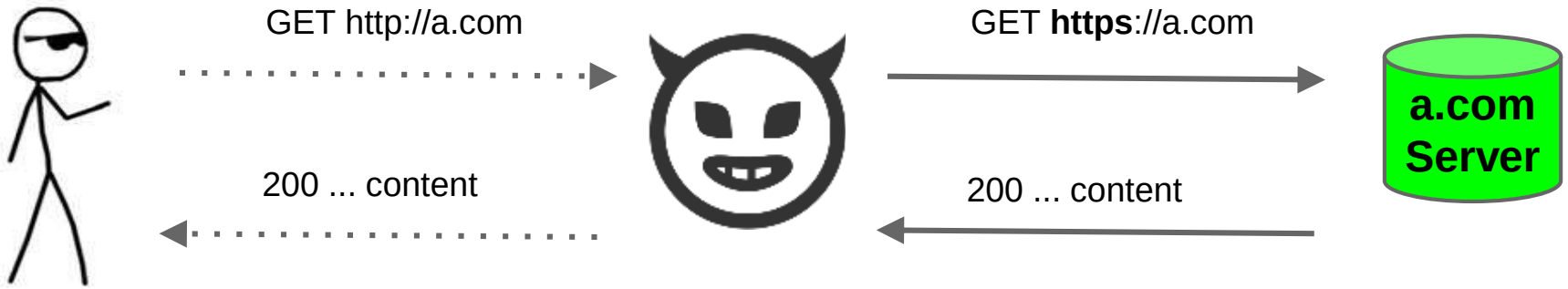
- Inconsistent or incomplete deployment
  - *Mixed content*
  - *Stripping attacks* → *HSTS*

- Failures by Certificate Authorities
  - *Rogue certificates* → *HPKP*

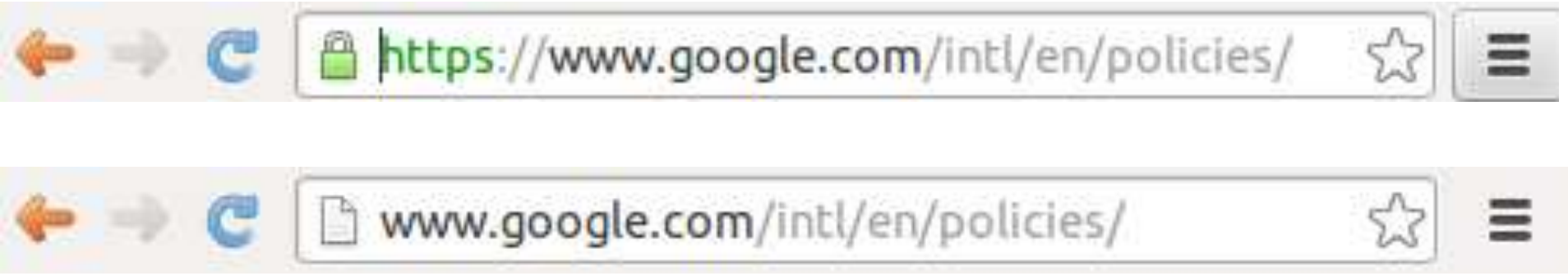
Problem 1: HTTPS stripping



Problem 1: HTTPS stripping



Users do not notice HTTPS stripping

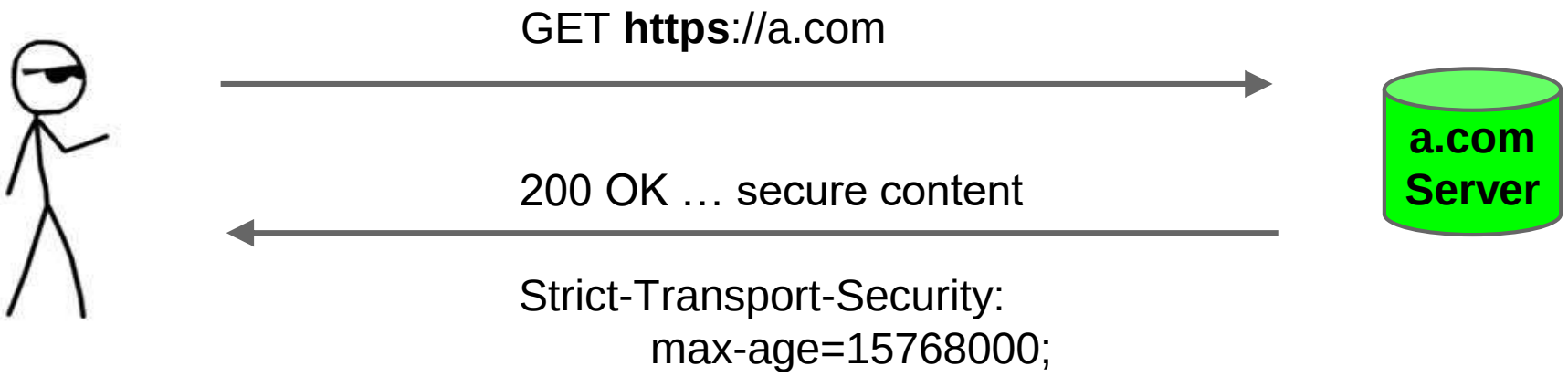


- **<10% notice** [Schechter et al. 2007] and others
- Automated tools available – can even do lock

Solution #1: HSTS (HTTP Strict Transport Security)

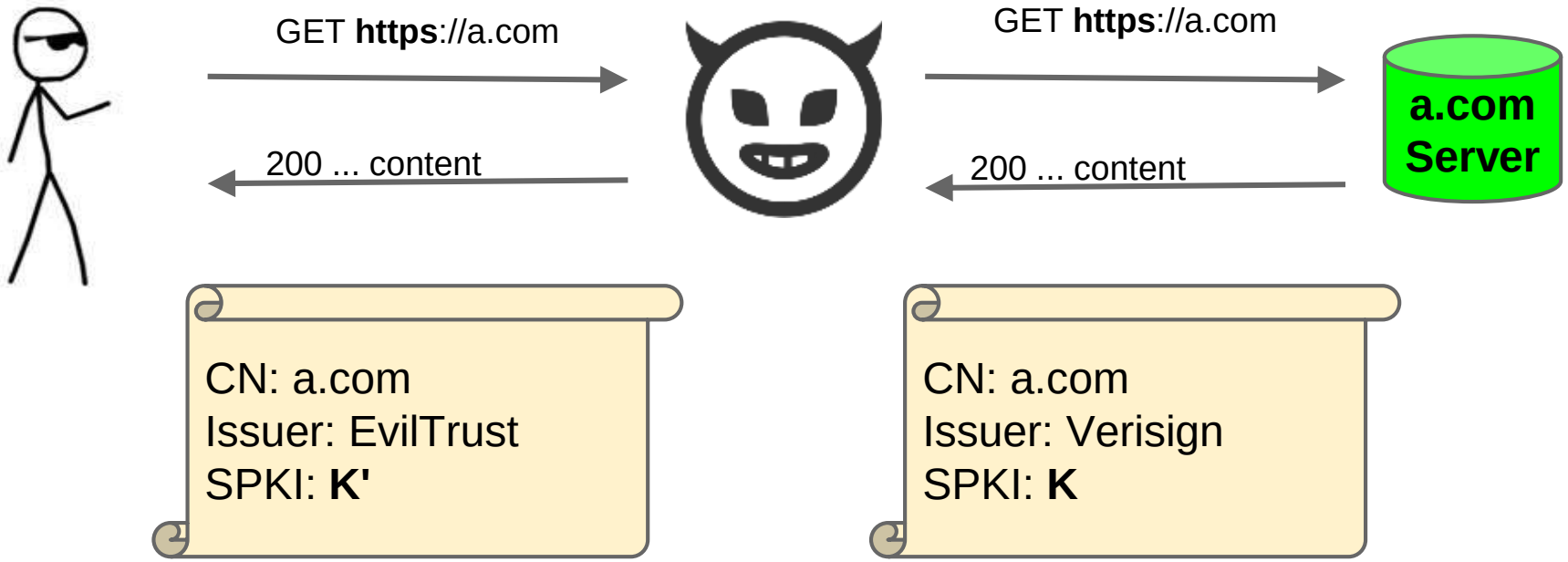
- *Mandatory* HTTPS at "HSTS domains“
  - Upgraded by browser in initial request
  - Converts HTTPS soft errors into hard errors
- Two methods of enabling
  - *Preloaded* via embedded browser list
  - *Dynamically* via HTTP Header
    - Must be set over HTTPS (trust on first use)
    - Policy expires based on included age
- Can set `includeSubdomains` token

HSTS in Action:



HSTS in Action

Problem 2: Rogue certificates



Rogue certificates in the wild

- March 2011: Comodo registrar hacked
  - 9 certs: mail.google.com, login.live.com, www.google.com, login.yahoo.com, login.skype.com, addons.mozilla.org
- July 2011: DigiNotar hacked
  - 531+ certs issued: *.google.com detected first
- ~2011: TürkTrust issues 2 intermediate CAs
  - One returned, one used in 2012 to proxy traffic...

Survey: Niemann, Brendel 2014

Solution #2: HPKP (HTTP Public Key Pinning)

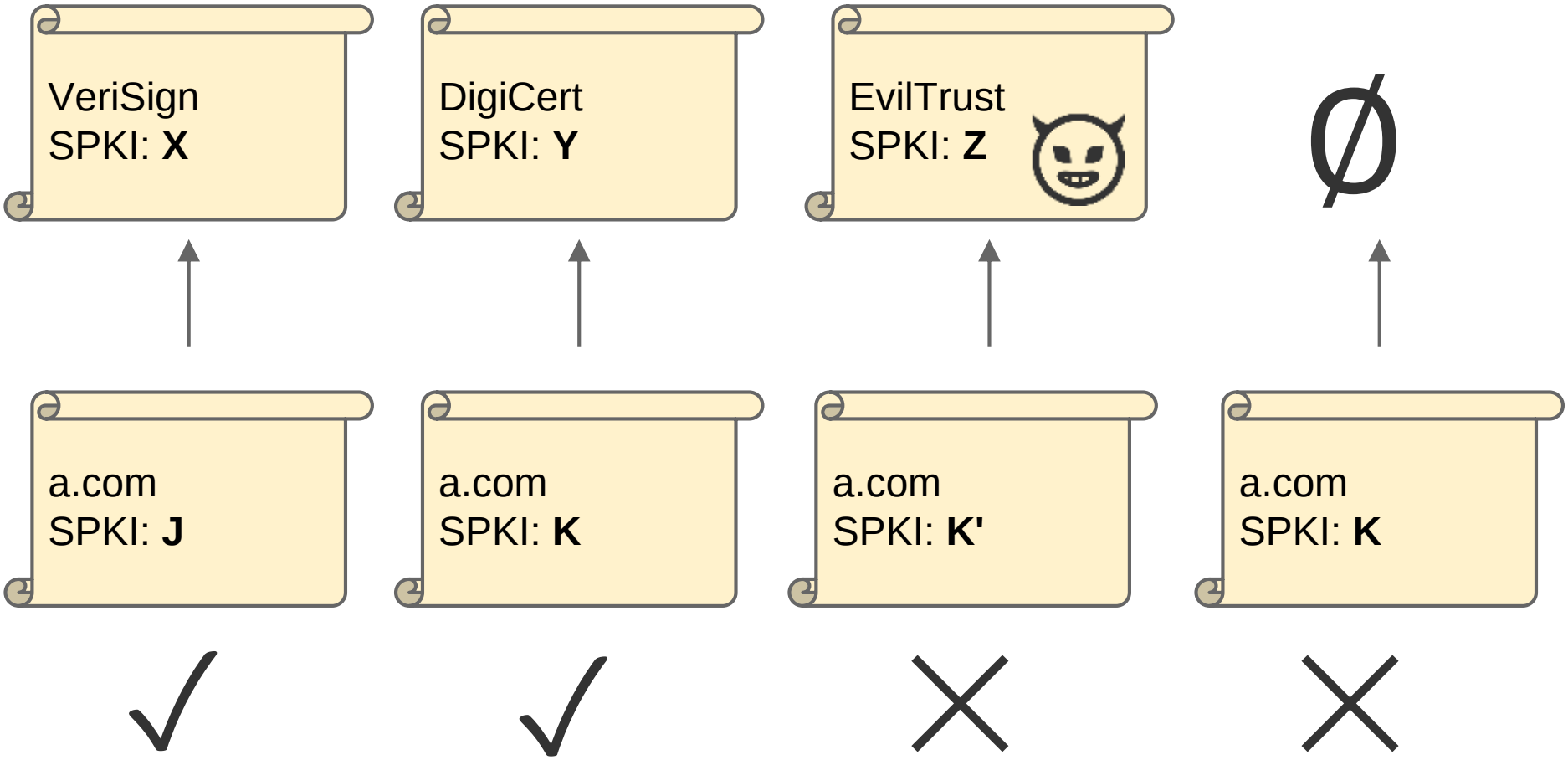
- Specified key hash must be present
  - Hard fail if hash not found
  - Hash can be in end-entity or CA cert

- Two methods of enabling
  - *Preloaded* via embedded browser list
  - *Dynamically* via HTTP Header*
    - Must be set over HTTPS (TOFU)
    - Policy expires based on included age
    - SHA1 or SHA256 Hash

**dynamic pinning not currently supported by browsers (proposed RFC)*

Solution #2: Key pinning

Pinset: {K, X}



Study Methodology

Infrastructure:

- OpenWPM*
  - Module for Static Resources (A tags, objects, etc.)
  - Firefox Extension for Dynamic Resources (Ajax)
- ZMAP

Span:

- Headers from Alexa Top Million
- Depth crawl of all HSTS domains
- Logged-in depth crawl of HPKP domains

*Visit our github page for more information

https://github.com/citp/OpenWPM/

Study Methodology

Infrastructure:

- OpenWPM*
  - Module for Static Resources (A tags, objects, etc.)
  - Firefox Extension for Dynamic Resources (Ajax)
- ZMAP

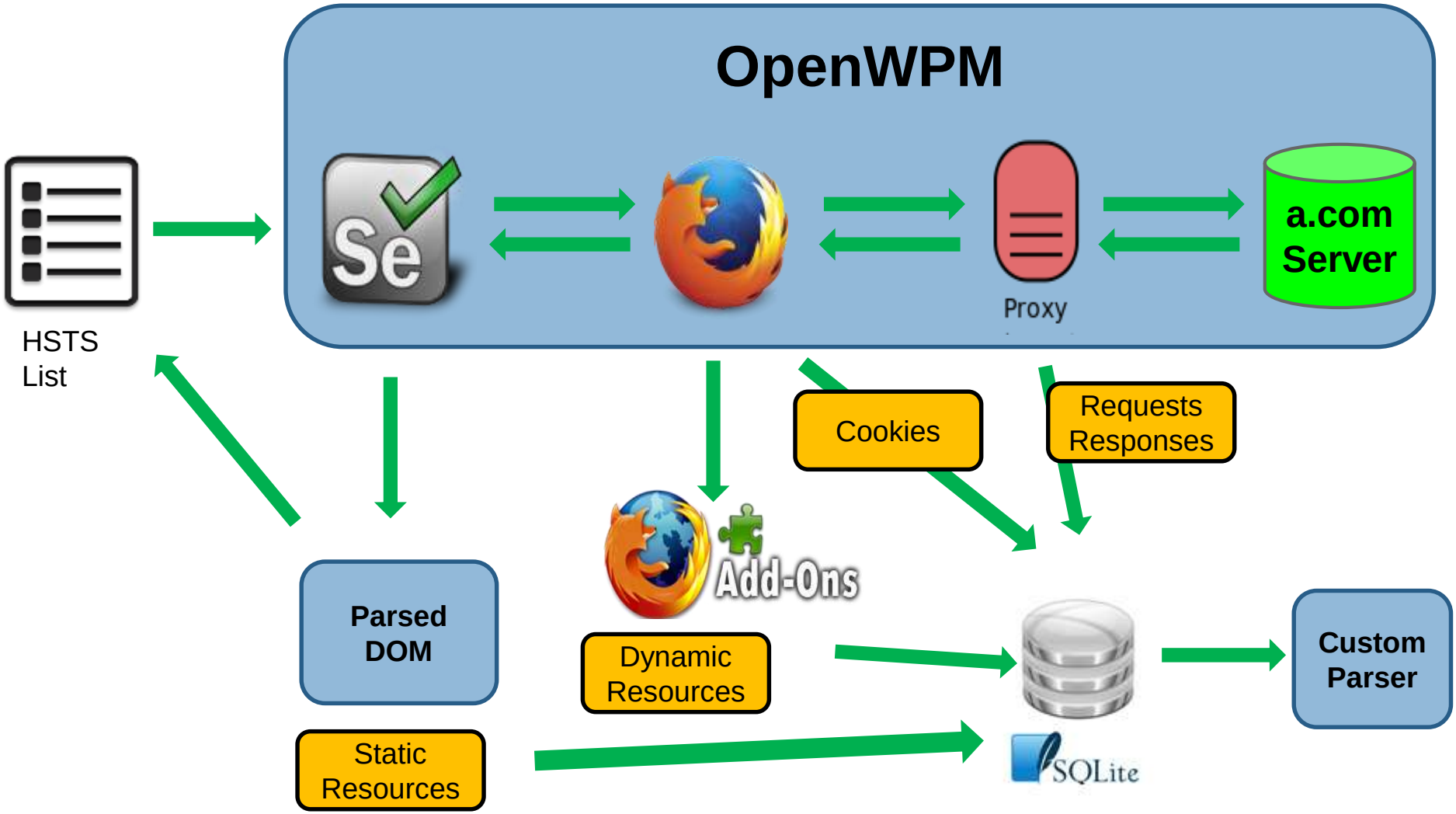
Span:

- Headers from Alexa Top Million
- Depth crawl of all HSTS domains
- Logged-in depth crawl of HPKP domains

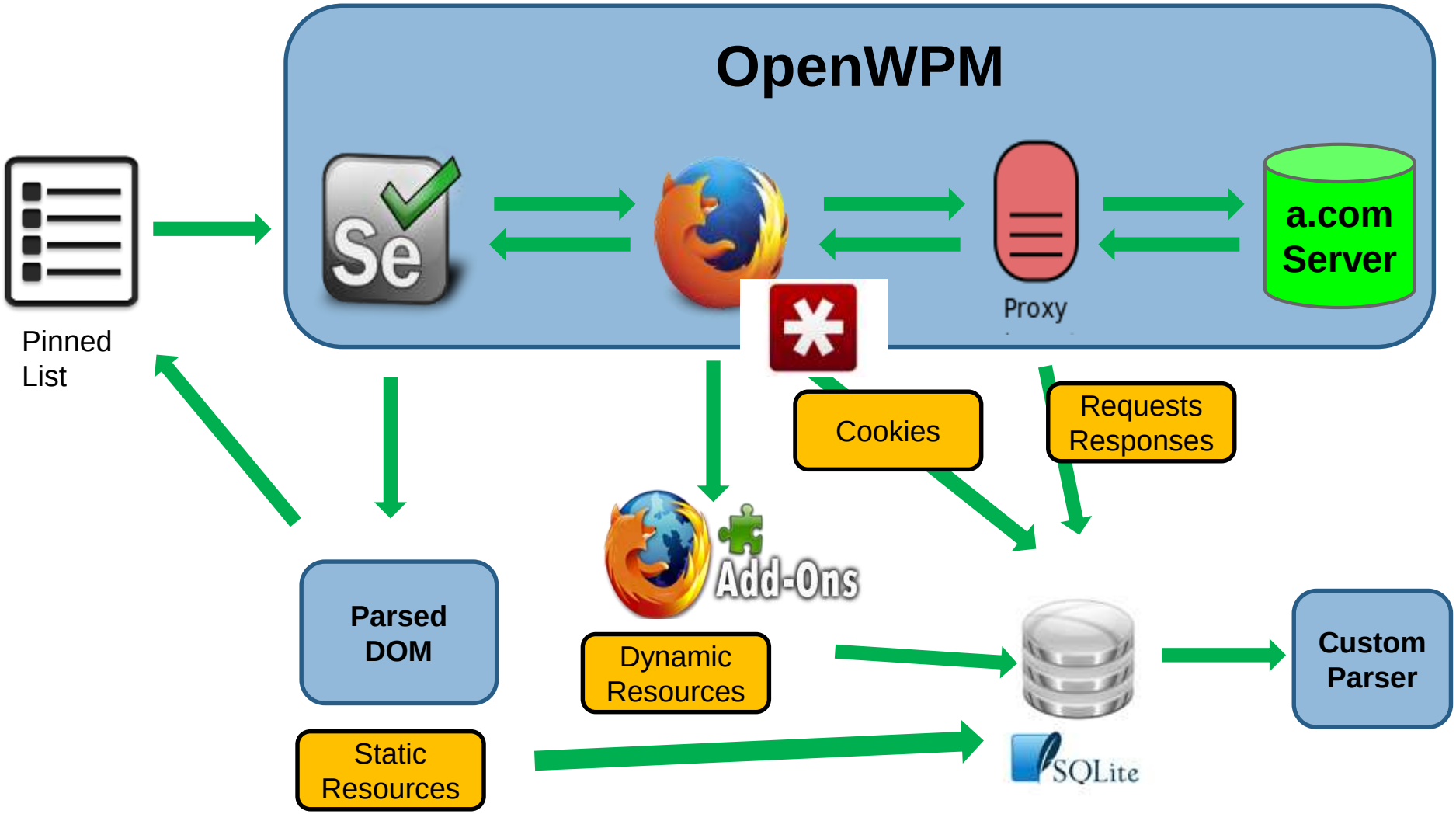
*Visit our github page for more information

https://github.com/citp/OpenWPM/

Study Methodology



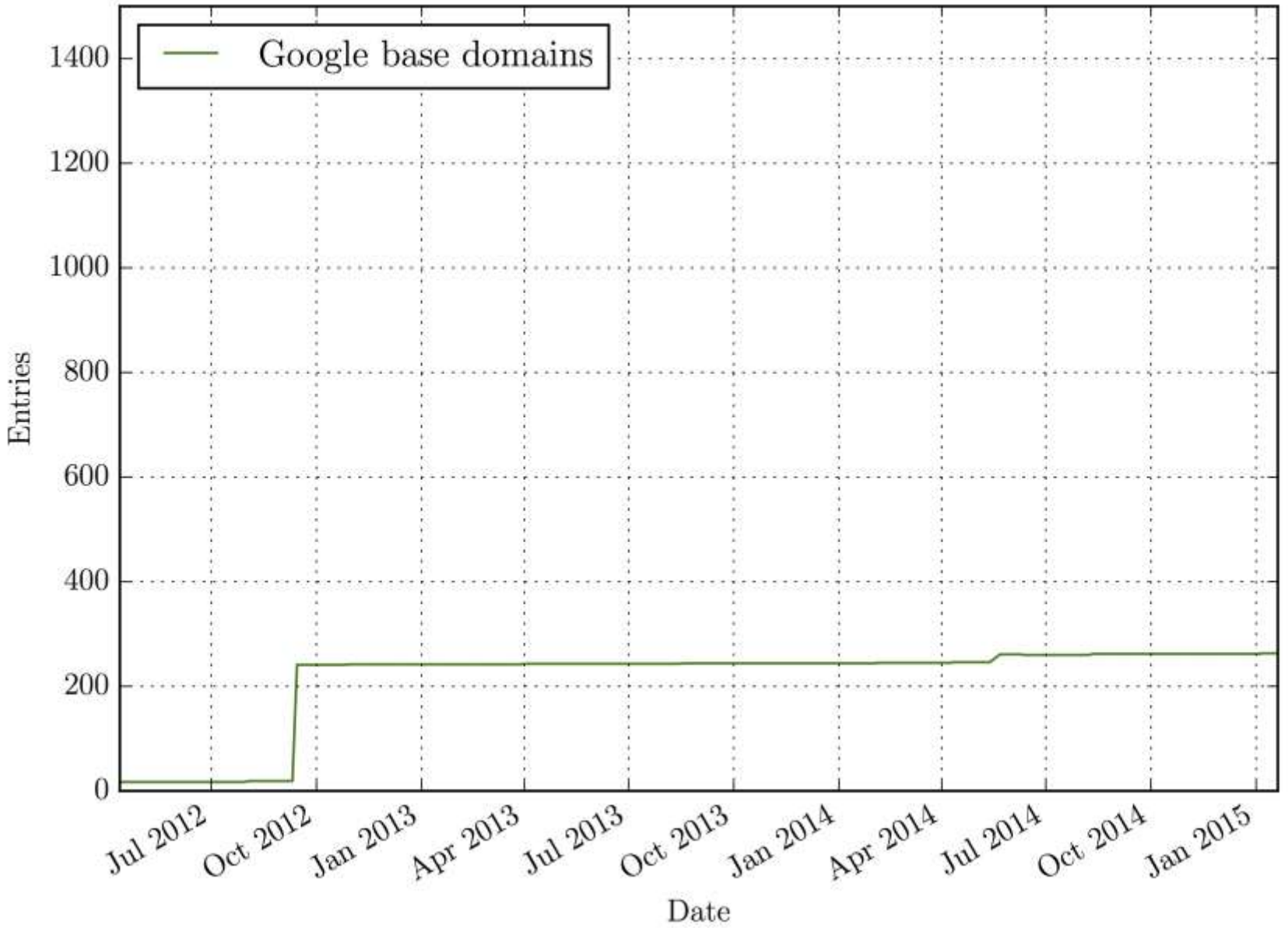
Study Methodology



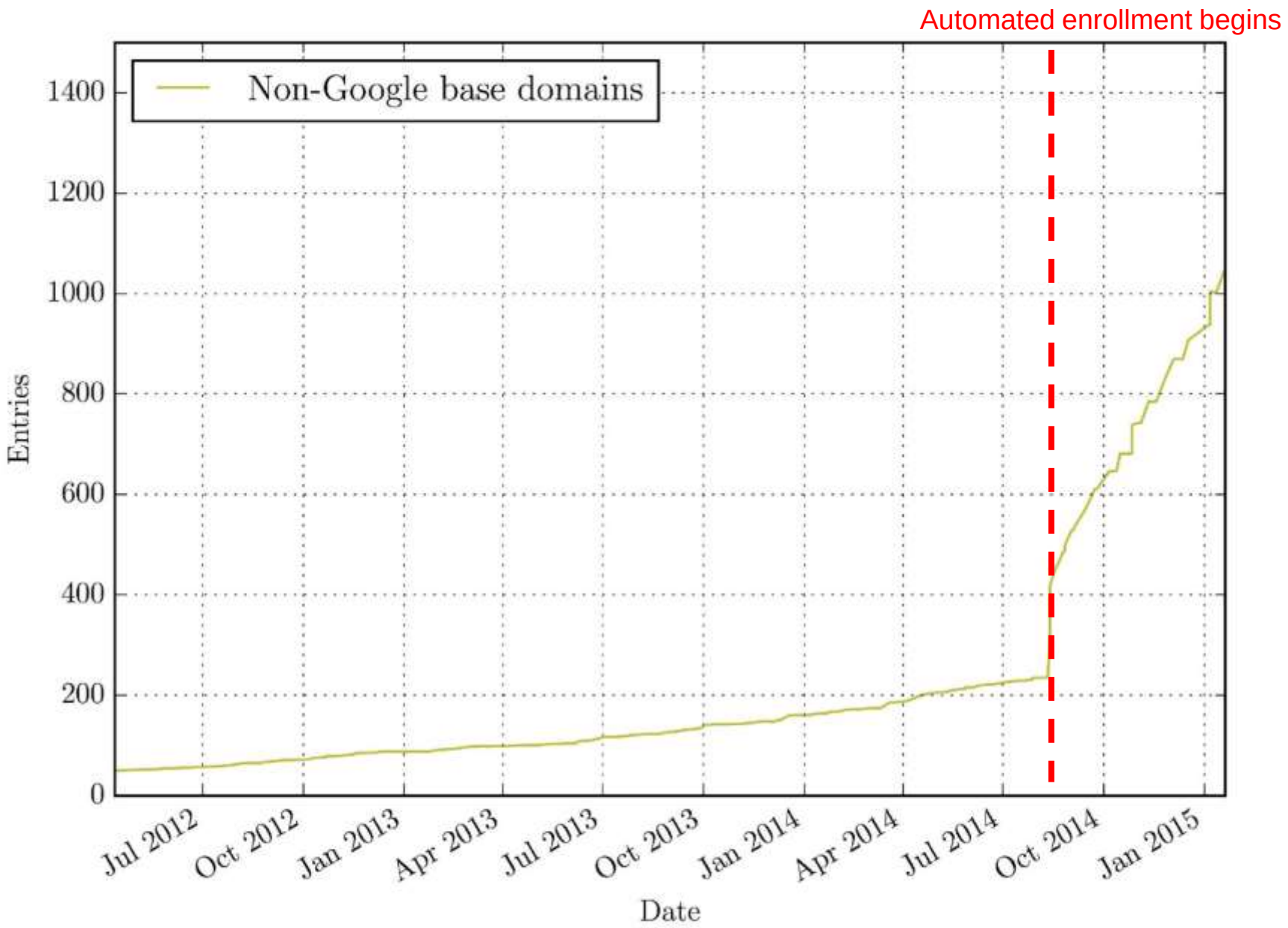
Deployment Summary

- Many sites are using HSTS
  - 12,593 of the Top 1M set HSTS headers
  - 1,021 Preloaded HSTS domains
- Many sites SHOULD be setting HSTS
  - 60% of Top 1M have active HTTPS sites
  - Of those, 10% redirect from HTTP to HTTPS
- Preloaded List has scalability issues
  - Started automated entry with manual review Aug 14
  - Surprisingly stale (10% return 404 or redirect to HTTP)

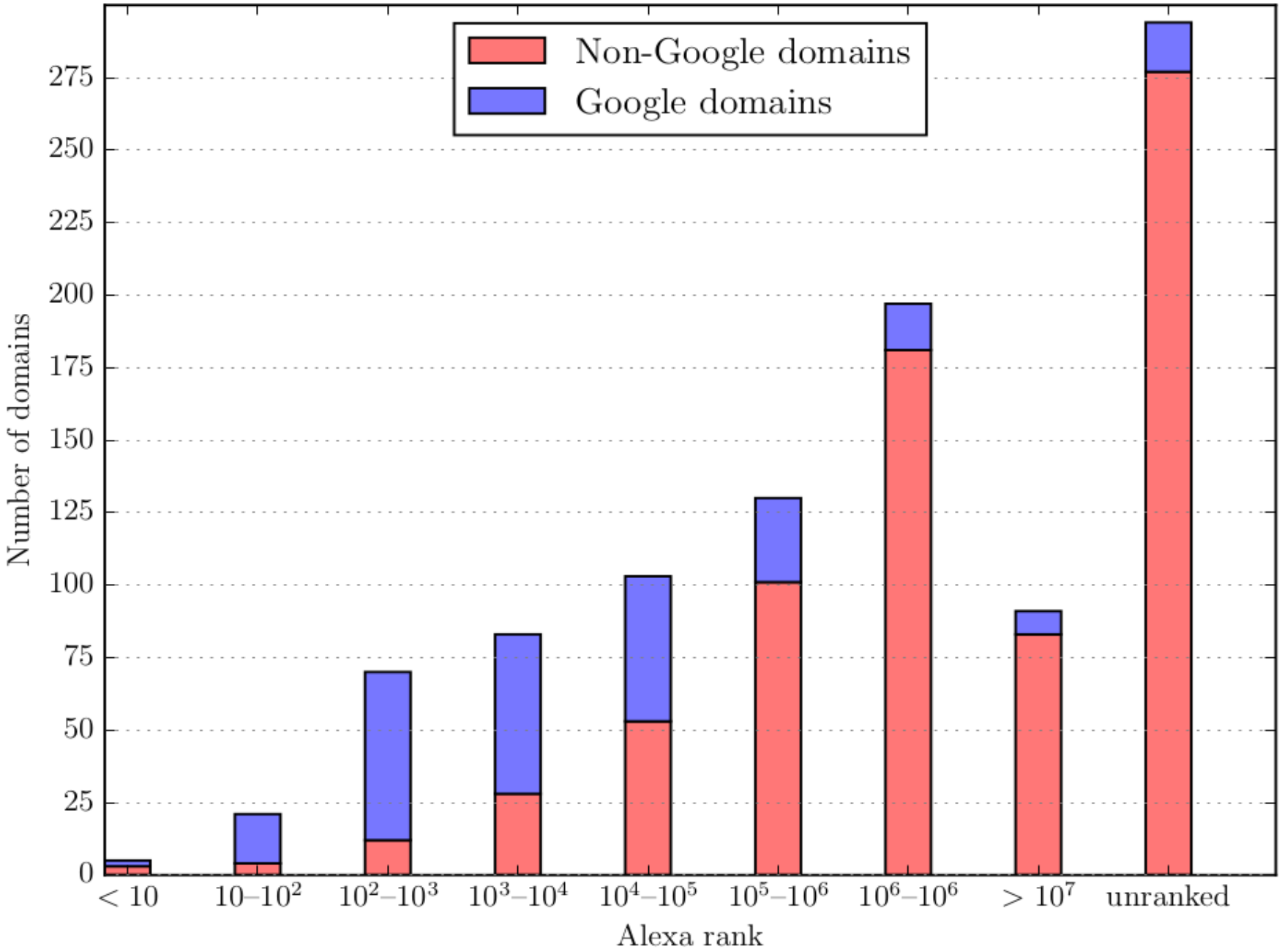
Chrome Preloaded List Growth



Chrome Preloaded List Growth



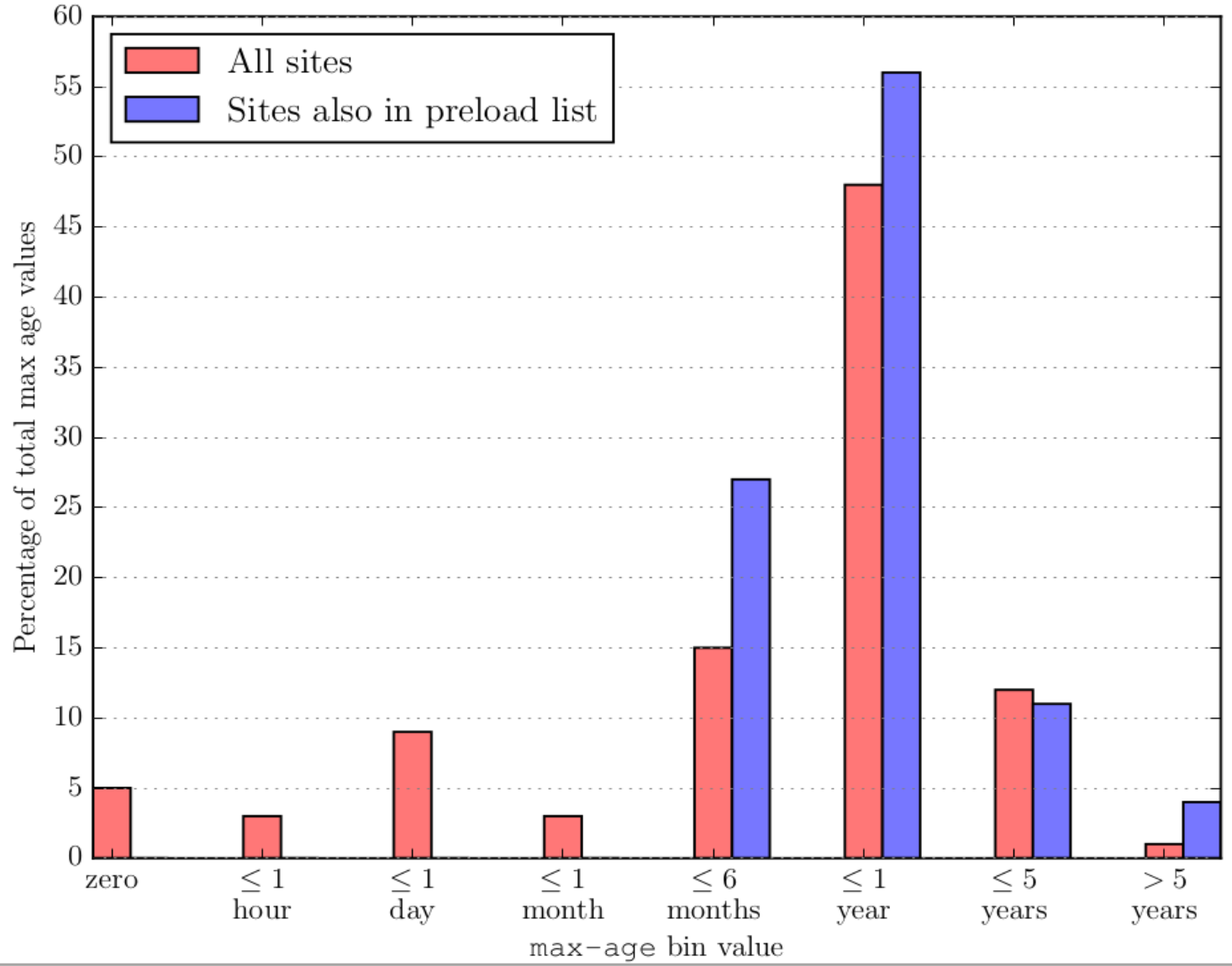
Many low-traffic sites preloaded



Error 1: Configuration Issues

- 5,099 of 12,593 (40%) set HSTS correctly according to the specification
  - 44% do not redirect from HTTP to HTTPS
  - 4% set ONLY via HTTP (does nothing)
  - 5% malformed headers
  - 18% set max-age less than 86400 (a day)

Max-age values vary significantly



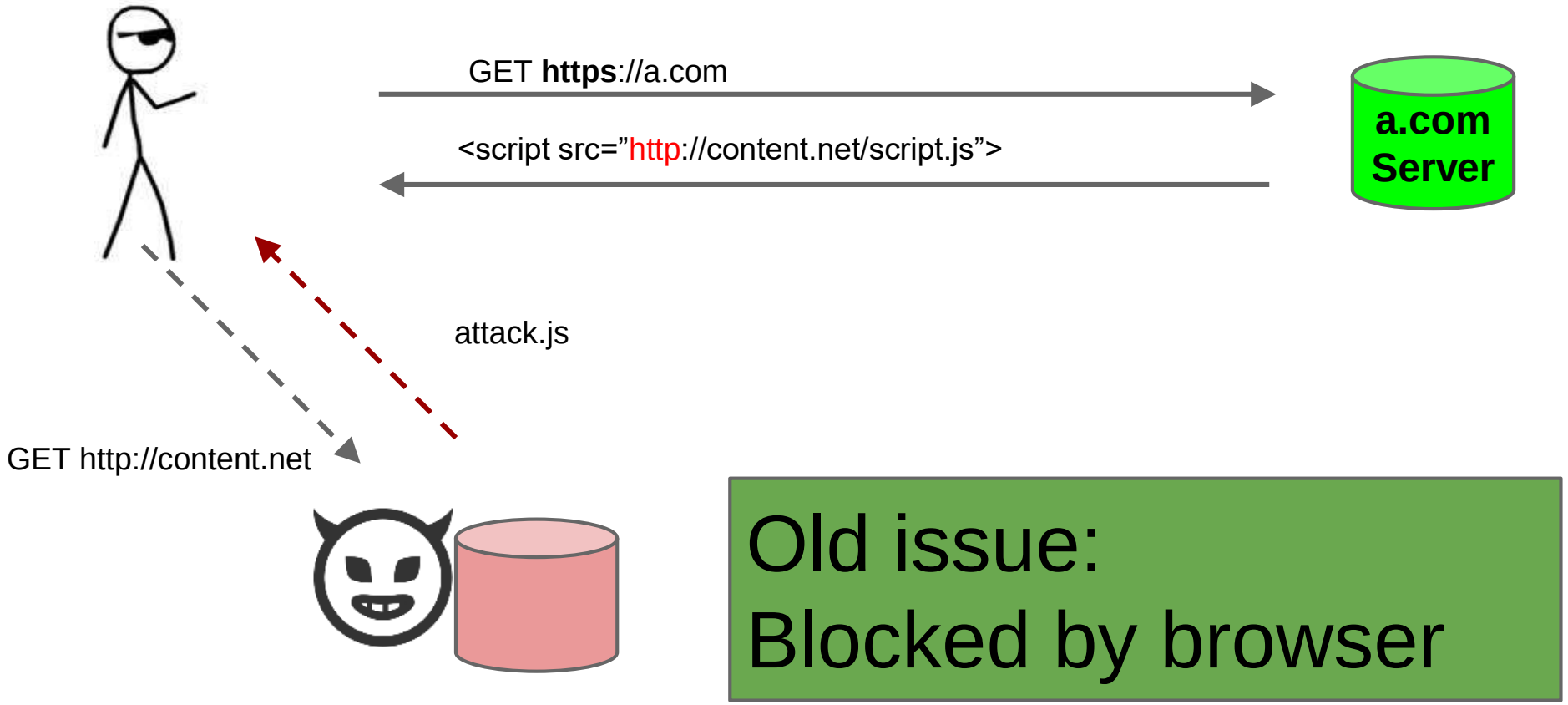
Error 1: Configuration Issues

- 5,099 of 12,593 (40%) set HSTS correctly according to the specification
  - 44% do not redirect from HTTP to HTTPS
  - 4% set ONLY via HTTP (does nothing)
  - 5% malformed headers
  - 18% set max-age less than 86400 (a day)

- Specification difficult to use
  - Max-age unit (seconds) is difficult to reason about
  - No clear list of steps

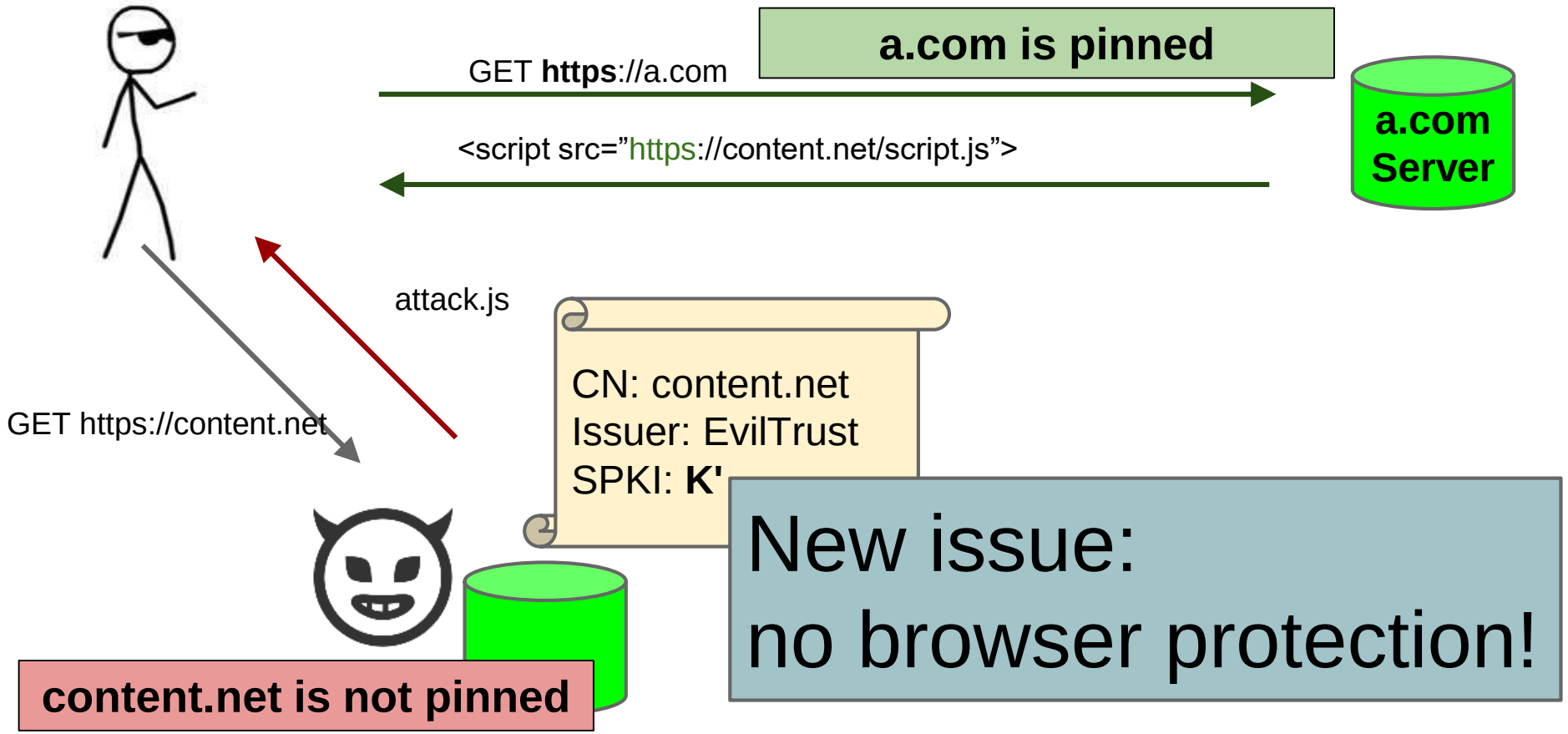
Error 2: New Mixed Content

Traditional Mixed Content



Old issue:
Blocked by browser

Pinned Mixed content



Pinned mixed content is common

- All pinsets include passive content
- 50% of pinsets include active content
  - 63% of active content from scripts
  - Examples: Twitter, Tor, and Dropbox
- Causes of mixed content
  - External services (Akamai and Doubleclick)
  - Self-referencing not pinned subdomains
  - ***Pinning is limited so its difficult to avoid***

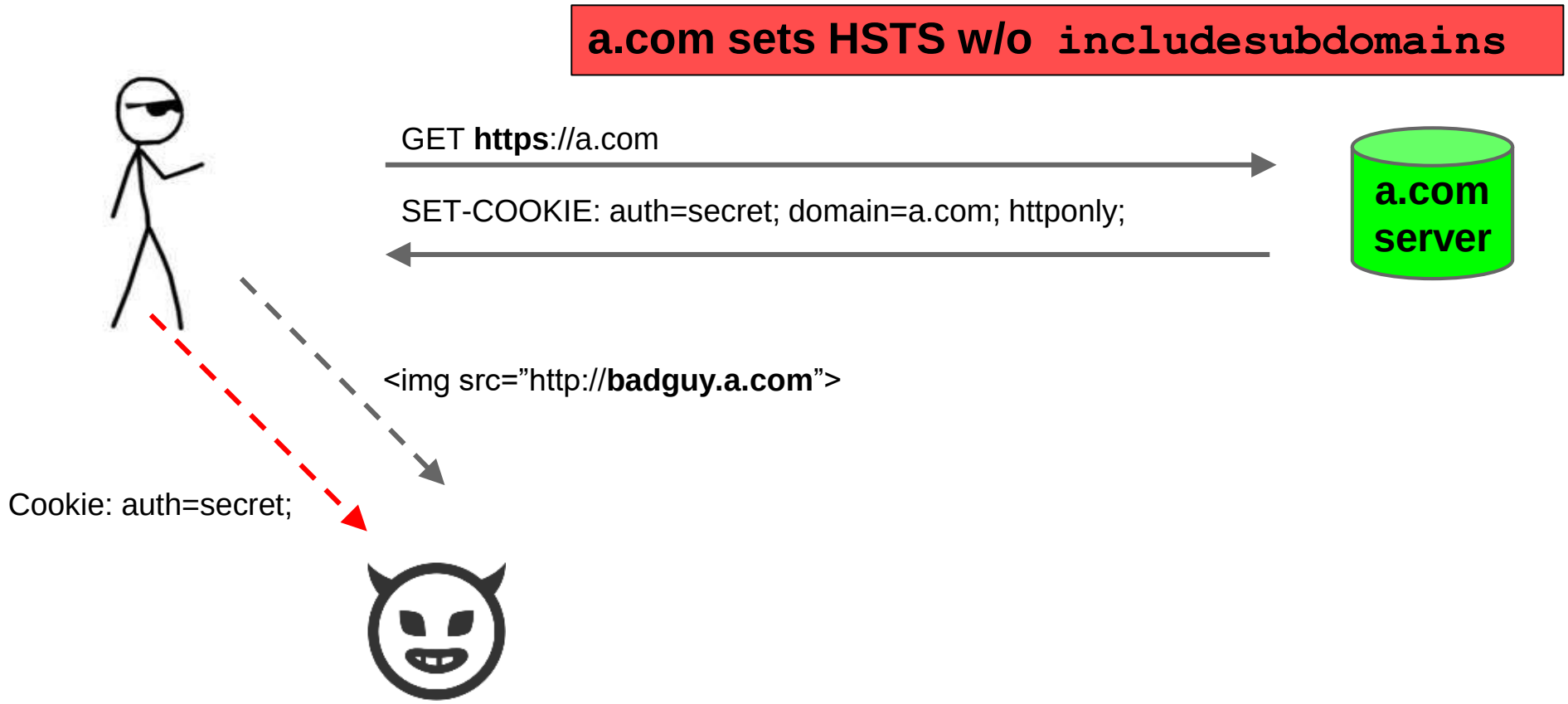
Error 3: Leakable Cookies

Default Subdomain Policies for HSTS/HPKP and Cookies are Different

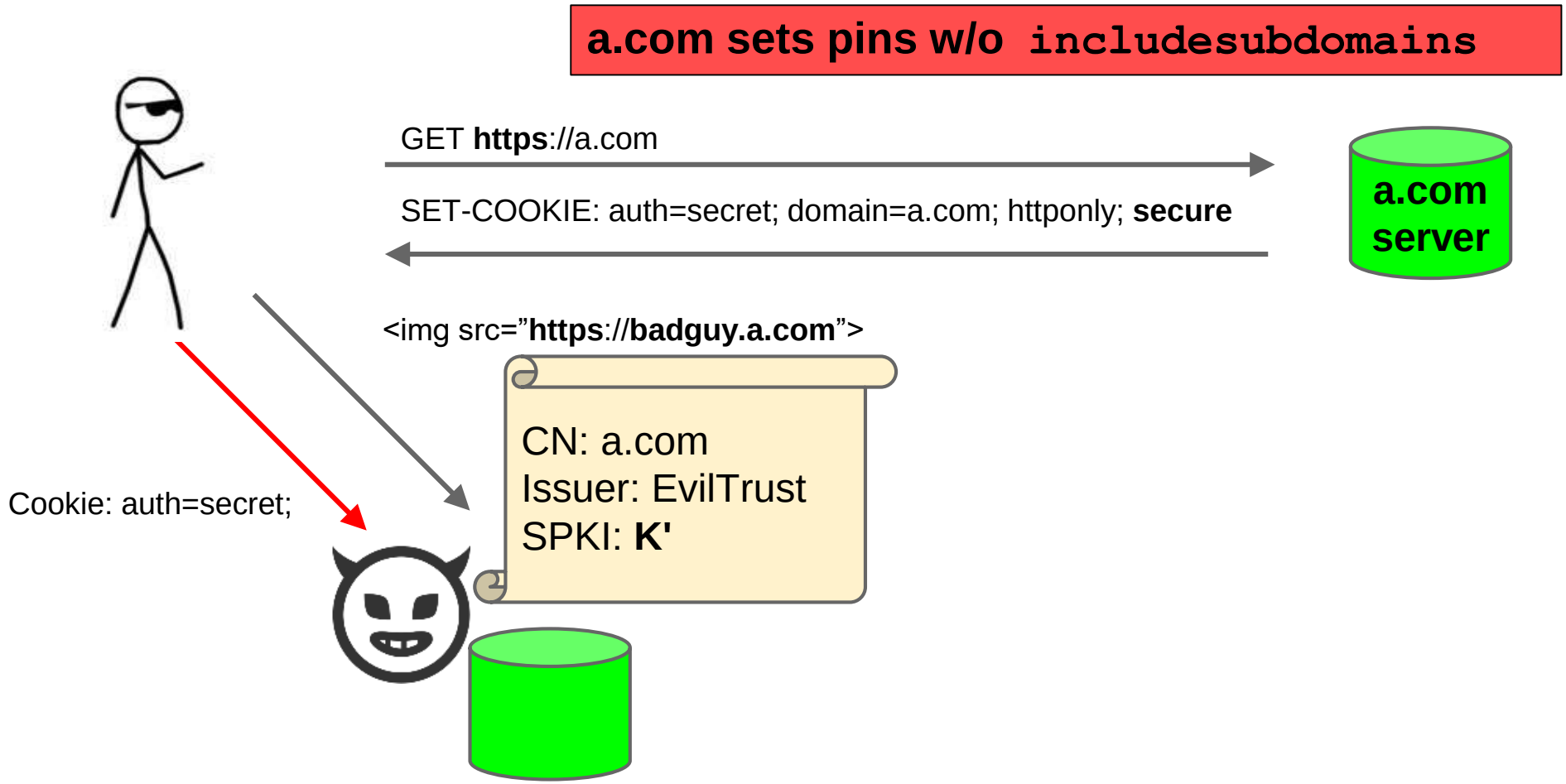
- HSTS and HPKP
  - By default **exclusive** on subdomains
  - Must specifically add `include_subdomains` directive to include subdomains

- Cookies
  - Most common case **inclusive** on subdomains
  - Must specific omit `domain` parameter from common case to do an exact domain (except on Internet Explorer)

Cookie-stealing attack (HSTS)



Cookie-stealing attack (pinning)



Many vulnerable cookies in the wild

- 49% of HSTS domains vulnerable
  - Any site w/o `includesSubdomains` is vulnerable
  - **10,174** cookies at **2,460** vulnerable domains
  - 98% **NOT** marked secure
  - Mostly tracking cookies (sites were not crawled logged in)

- 44% of Non-Google pinned domains vulnerable
  - Facebook, Twitter auth cookies vulnerable (known issue)
  - Bypasses purpose of pinning

Takeaways:

- Even simple upgrades are complicated in practice
  - Web platform is very large/complex
  - Standards do not necessarily reflect reality
- Better standards would help
  - Summaries, guidelines, and defaults would help
  - Consider testing with developers during process
- HSTS and HPKP are better than just HTTPS
  - Significant growth in the past 6 months
  - Some sites already setting HPKP

Thank you

jbonneau@princeton.edu
mkranch@princeton.edu

Error 1: Configuration Errors

	Alexa top 1M		Preloaded domains	
	Domains	%	Domains	%
Attempts to set dynamic HSTS	12,593	—	751	—
Doesn't redirect HTTP→HTTPS	5,554	44.1%	23	3.1%
Sets HSTS header only via HTTP	517	4.1%	3	0.4%
Redirects to HTTP domain	774	6.1%	9	3.1%
HSTS Redirects to non-HSTS	74	0.6%	3	0.4%
Malformed HSTS header	322	2.6%	12	1.6%
`max-age` = 0	665	5.3%	0	0%
0 < `max-age` <= 1 day	2,213	17.6%	5	0.7%
Sets HSTS securely w/o errors	5,099	40.5%	659	87.7%

Takeaways: standards not holistic

- Standards not Holistic
  - Different formats for headers, preloads (DANE different as well)
  - Preload format not standardized and is changing
- Better Defaults may help
  - Pinning, HSTS default should be `includeSubdomains`
  - Minimum `max-age` values
- HSTS and Key Pinning are used and growing
  - 500% Non-Google growth in the past 6 months
  - Sites already setting HPKP (*errors more costly*)

Preloaded HSTS

```
{...
 "entries":
   [
     {"name": "www.paypal.com", "mode": "force-https" },
     {"name": "www.elanex.biz", "mode": "force-https" },
     {"name": "jottit.com", "include_subdomains": true,
      "mode": "force-https" },
     {"name": "sunshinepress.org", "include_subdomains":
      true, "mode": "force-https" },
     {"name": "www.noisebridge.net", "mode":
      "force-https" },
...
   ]
}
```

transport_security_static.json (Chromium project)

Preloads: HPKP

```
{
  "pinsets": [
    {"name": "tor",
     "static_spki_hashes":
       ["RapidSSL",
        "DigiCertEVRoot",
        "Tor1",
        "Tor2",
        "Tor3"
       ]
    }, ...
  ]
  "entries": [
    {"name":"torproject.org",
     "mode": "force-https",
     "pins": "tor" }, ...
  ]
}
```

transport_security_static_state.json

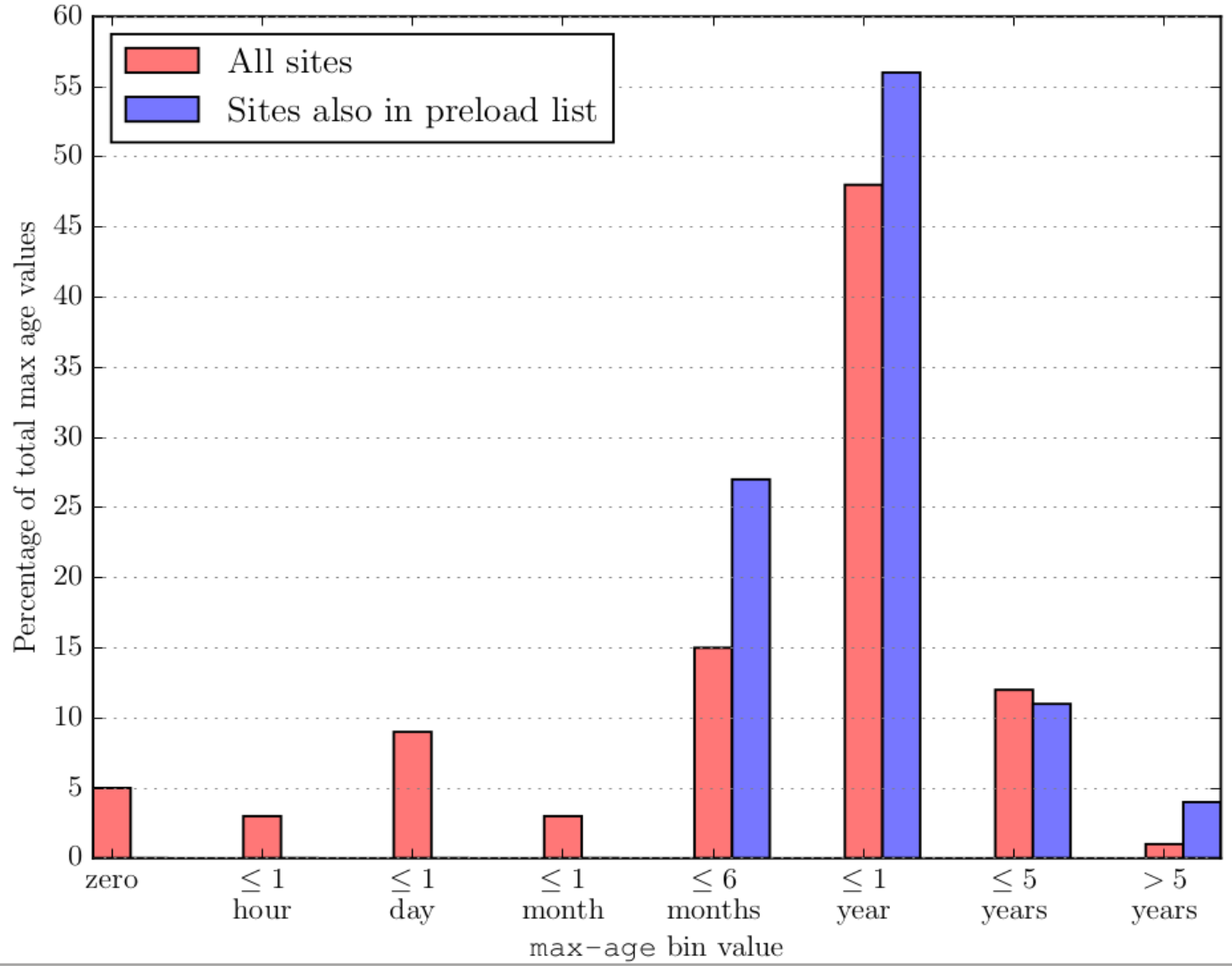
```
RapidSSL
-----BEGIN CERTIFICATE-----
MIID1TCCAr2gAwIBAgIDAjbRMA0GC
SqGSIb3DQEBBQUAMEIxCzAJBgNVBA
YTAlVTMRYwFAYDVQQKEw1HZW9UcnV
zdCBJbmMuMRswGQYDVQQDExJHZW9U
cnVzdCBHbG9iYWwgQ0EwHhcNMTAwM
...
-----END CERTIFICATE-----

Tor1
sha1/juNxSTv9UANmpC9kF5GKpmWN
x3Y=

Tor2
sha1/lia43lPolzSPVIq34Dw57uYc
LD8=
...
```

transport_security_static_state.cert

Max-age values vary significantly



Preventing cookie-stealing (Pinning)

- Set pins with **`include_subdomains`**
- Set cookies to more specific domain with **`include_subdomains`**

  dropbox.com **does not** include but *www.dropbox.com* **does**

No equivalent for preloaded pinning!

Proposed Addition Preload Token:
include_subdomains_for_pinning_only

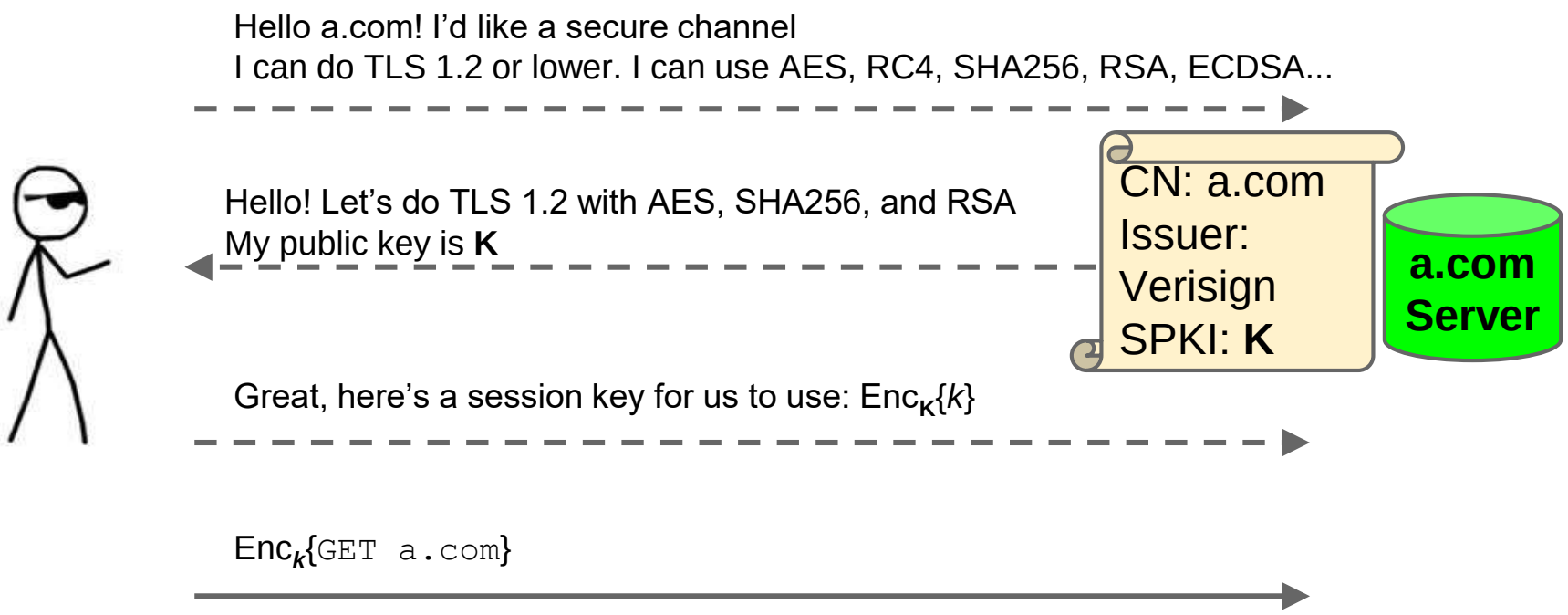
HTTPS: where web-sec meets TLS

HTTP (≈ web browsing)

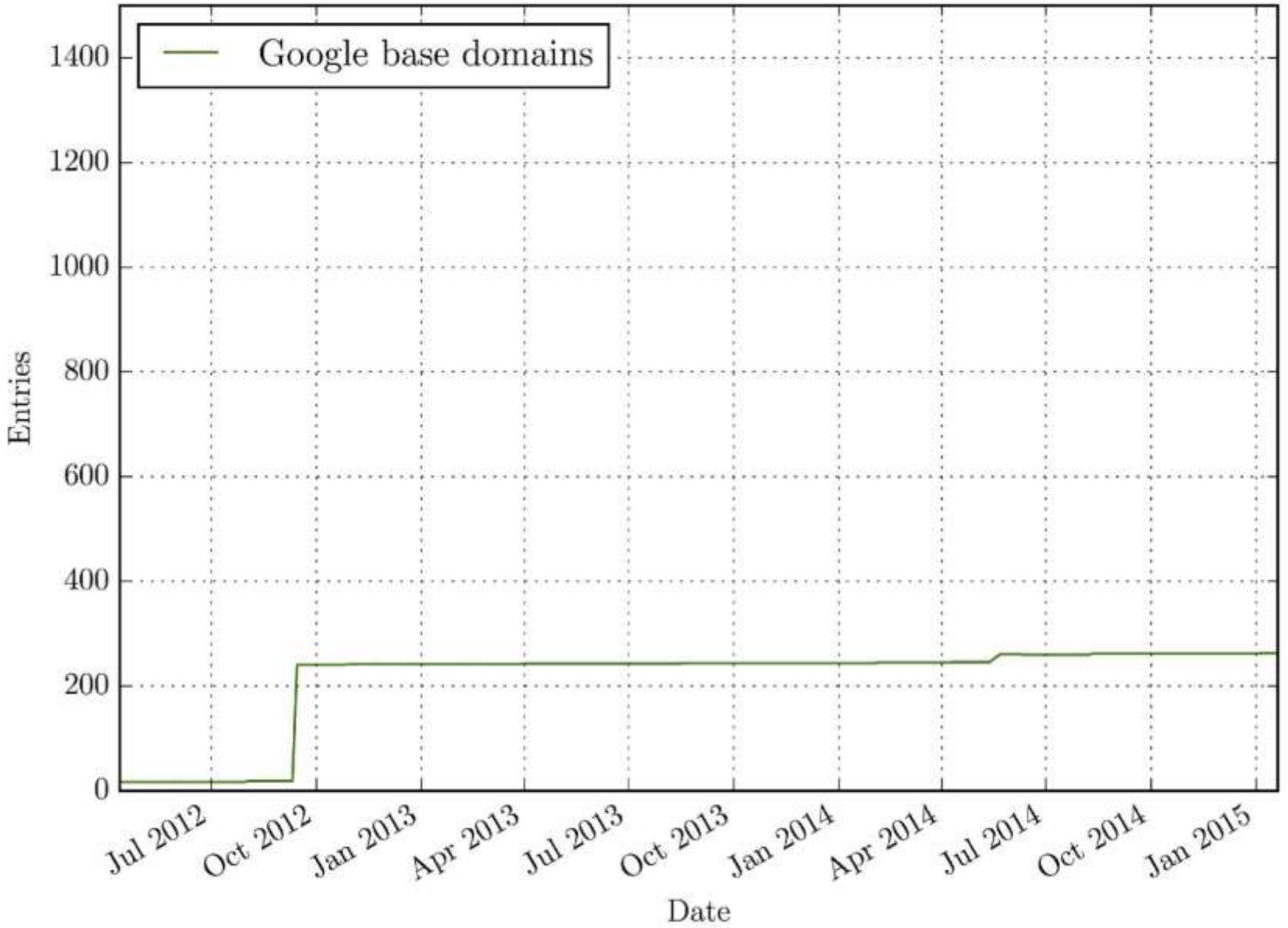
over

Secure **S**ockets **L**ayer (SSL)
or
Transport **L**ayer **S**ecurity (TLS)

TLS in one slide



Chrome Preloaded List Growth



Chrome Preloaded List Growth