

“Domain Fronting in 2025: A retro analysis”





- DEFCON33 @ Malware Village (2025-08-08) ~ Presented by Tom Cope

(warning: this presentation and content was entirely human generated)

Hello World!

- Tom Cope (<https://tomcope.com>)
- Principal Consultant at Control Plane
 - Ex-CSO of Ava Security (now part of Motorola)
 - Security Engineer / SOC Analysis for UK leading asset manager
 - IAM SME at IBM
- Working in Cyber Security ~12 years
- Part time Security Researcher (CVE-2020-5014 + CVE-2021-29707)
 - CodeGate AI Write up coming soon!
- Blue team by day, red team by night



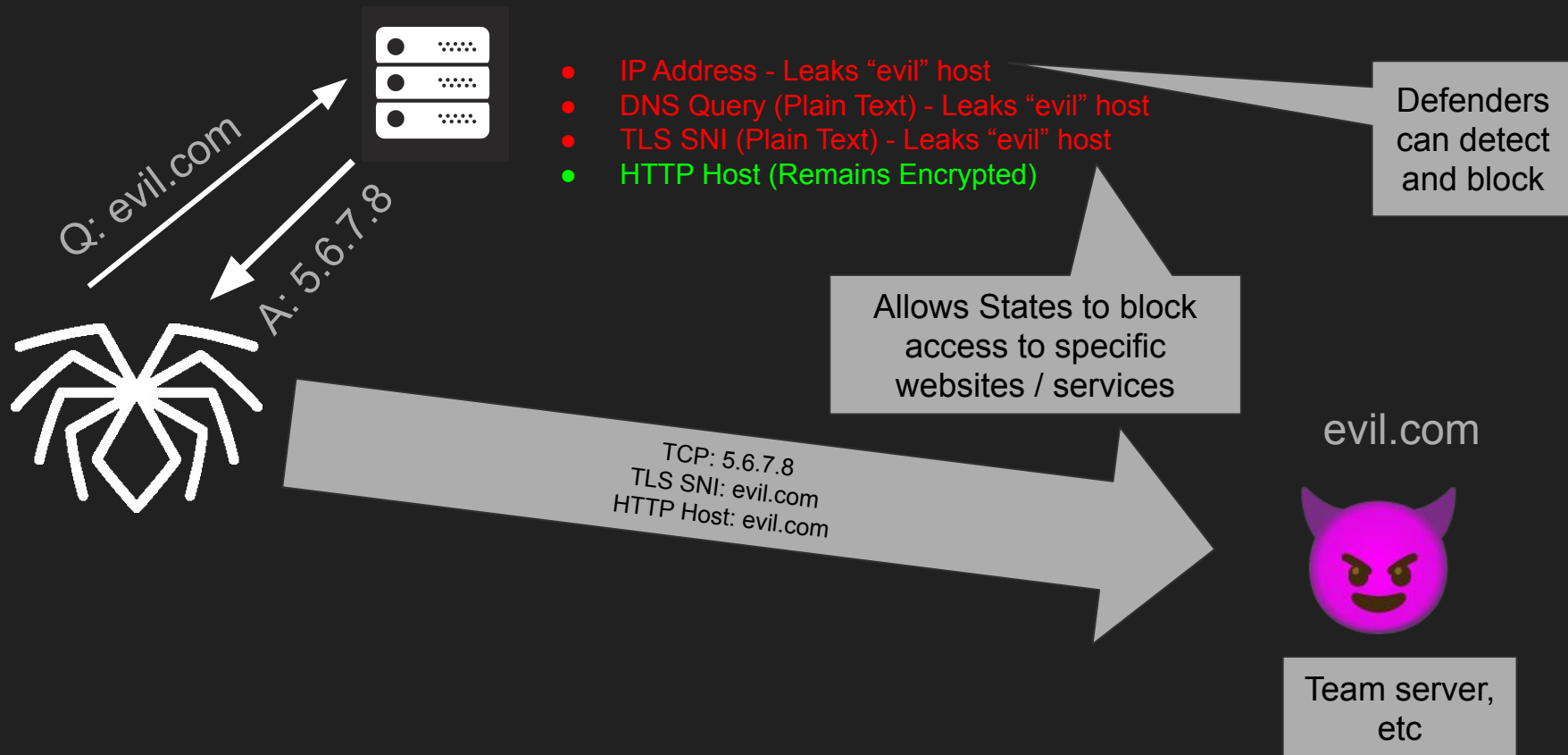
 <p>AWS Certified Solutions Architect – Associate Amazon Web Services Training and Certification</p>	 <p>CompTIA Security+ Certification CompTIA</p>	 <p>Certified Information Systems Security Professional (CISSP) (ISC)²</p>	 <p>Security and Privacy by Design Foundations IBM</p>	 <p>Think Like a Hacker IBM</p>	 <p>Government Silver Industry Insights & Solutions IBM</p>
 <p>IBM Mentor IBM</p>	 <p>Docker Essentials: A Developer Introduction IBM</p>	 <p>IBM Recognized Teacher/ Educator IBM</p>	 <p>IBM Security Essentials for Architects IBM</p>	 <p>Get started with Kubernetes and IBM Cloud Container Service IBM</p>	 <p>Cloud Security Architect and Engineer Fundamentals IBM</p>

Contents

- Network Artifacts produced when hiding a C2
- Define “Domain Fronting”
- Recap “Traditional Domain Fronting”
- Review other Domain Fronting Methods
 - Handy matrix included!
- State of CDNs “Cooperation” in 2023
- Research methodology
- Retrospective: State of CDNs “Cooperation” in **2025**

The Problem

The Problem: Network Artifacts



A popular solution:

Domain Fronting

What is “Domain Fronting”

Domain Fronting

“Domain fronting is a technique used to conceal the true destination of network traffic by disguising it as traffic to a different domain, by abusing both hosts being hosted behind the same CDN (Content Delivery Network) Service.”

Legitimate Use Cases:

Censorship bypass
(Telegram / Signal / Tor Bridge)

Malicious Use Cases:

Hiding C2 Traffic

CDN (Content Delivery Network)

“A Content Delivery Network (CDN) is a collection of globally distributed servers that speed up web content delivery by serving it from locations closer to users, thereby reducing load times and enhancing reliability”

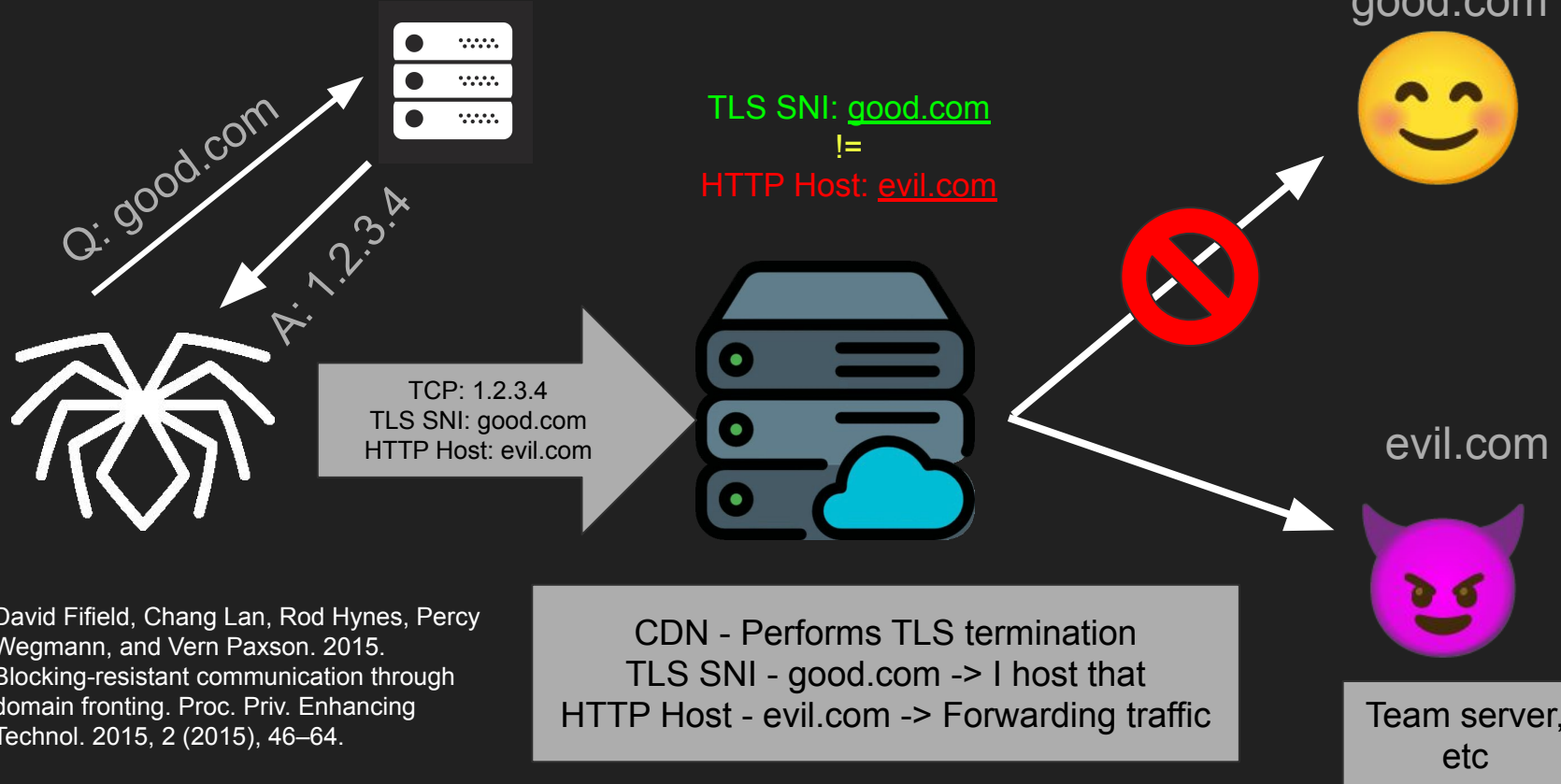
“Host your website behind us and we’ll make it faster and more secure”

Popular vendors: Cloudflare, Fastly, Bunny.net



Classical “Domain Fronting”

The Classic: TLS SNI and Host Mismatch



David Fifield, Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson. 2015. Blocking-resistant communication through domain fronting. Proc. Priv. Enhancing Technol. 2015, 2 (2015), 46–64.

Curl Example

Desired Backend

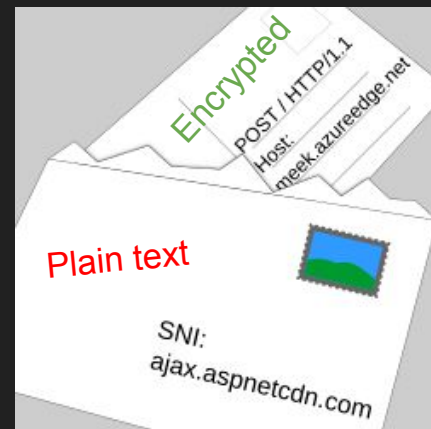
```
[tom@AwesomeShark ~]$ curl -s https://ipfs.tech/_nuxt/DlAUqK2U.js | md5sum
25e3a5dcaf00fb2b1ba0c8ecea6d2560 -
[tom@AwesomeShark ~]$ curl -s -H "Host: ipfs.tech" https://calculadora.now/_nuxt/DlAUqK2U.js | md5sum
25e3a5dcaf00fb2b1ba0c8ecea6d2560
[tom@AwesomeShark ~]$ date
Sat  2 Aug 13:33:54 BST 2025
[tom@AwesomeShark ~]$
```

Same Resource Served

Fronting domain

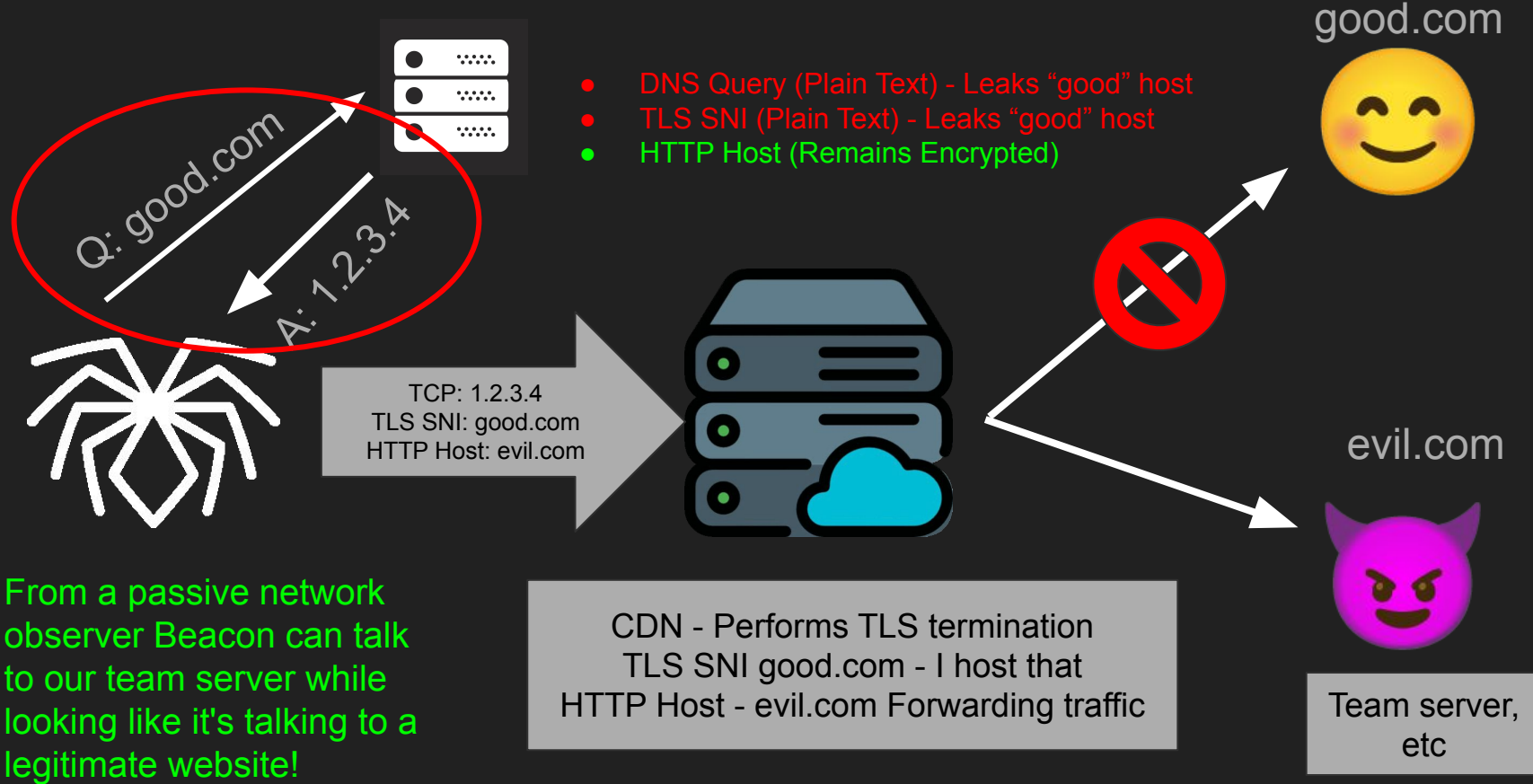
```
[tom@AwesomeShark ~]$ curl -v -H "Host: ipfs.tech" https://calculadora.now/_nuxt/DlAUqK2U.js
* Host calculadora.now:443 was resolved.
* IPv6: 2400:52e0:1e03::1205:1
* IPv4: 143.244.38.136
* Trying [2400:52e0:1e03::1205:1]:443...
* Immediate connect fail for 2400:52e0:1e03::1205:1: Network is unreachable
* Trying 143.244.38.136:443...
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* CAfile: /etc/ssl/certs/ca-certificates.crt
* CApath: none
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 / x25519 / RSASSA-PSS
* ALPN: server accepted h2
* Server certificate:
* subject: CN=calculadora.now
* start date: Jul 15 09:11:14 2025 GMT
* expire date: Oct 13 09:11:13 2025 GMT
* subjectAltName: host "calculadora.now" matched cert's "calculadora.now"
* issuer: C=US; O=Let's Encrypt; CN=R11
* SSL certificate verify ok.
* Certificate level 0: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 2: Public key type RSA (4096/152 Bits/secBits), signed using sha256WithRSAEncryption
* Connected to calculadora.now (143.244.38.136) port 443
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://calculadora.now/_nuxt/DlAUqK2U.js
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: ipfs.tech]
* [HTTP/2] [1] [:path: /_nuxt/DlAUqK2U.js]
* [HTTP/2] [1] [user-agent: curl/8.15.0]
* [HTTP/2] [1] [accept: */*]
> GET /_nuxt/DlAUqK2U.js HTTP/2
> Host: ipfs.tech
> User-Agent: curl/8.15.0
> Accept: */*
>
```

Connected to fronted domain



Traffic routed to desired domain

The Classic: Network Artifacts



Changing Times - CDNs Cracking down on Domain Fronting

08 APR 2019 - AWS

AWS Blogs Home Blogs ▾ Editions ▾

Networking & Content Delivery

Continually Enhancing Domain Security on Amazon CloudFront

by Woodrow Arrington | on 08 APR 2019 | in [Advanced \(300\)](#), [Amazon CloudFront](#), [AWS Certificate Manager](#), [Expert \(400\)](#), [Foundational \(100\)](#), [Intermediate \(200\)](#), [Networking & Content Delivery](#), [Security](#), [Security, Identity, & Compliance](#) | [Permalink](#) | [Share](#)

Last year, a colleague of mine wrote a blog post about new [security measures](#) that Amazon CloudFront was implementing to enhance the security of how domains are used on CloudFront distributions. This included mitigations to prevent the abusive use of domain fronting practices by not allowing SSL handshake requests and subsequent requests over the secured HTTP protocol to be made between two unrelated accounts. That work also included new mechanisms to clearly warn customers about inadvertently creating “dangling” DNS entries (which could lead to other security concerns) when they are removing an alternate domain name or deleting a CloudFront distribution, but their DNS records were still pointing their domain traffic to CloudFront.

November 8, 2022 - Azure

Generally available: Block domain fronting behavior on newly created customer resources

Azure Front Door

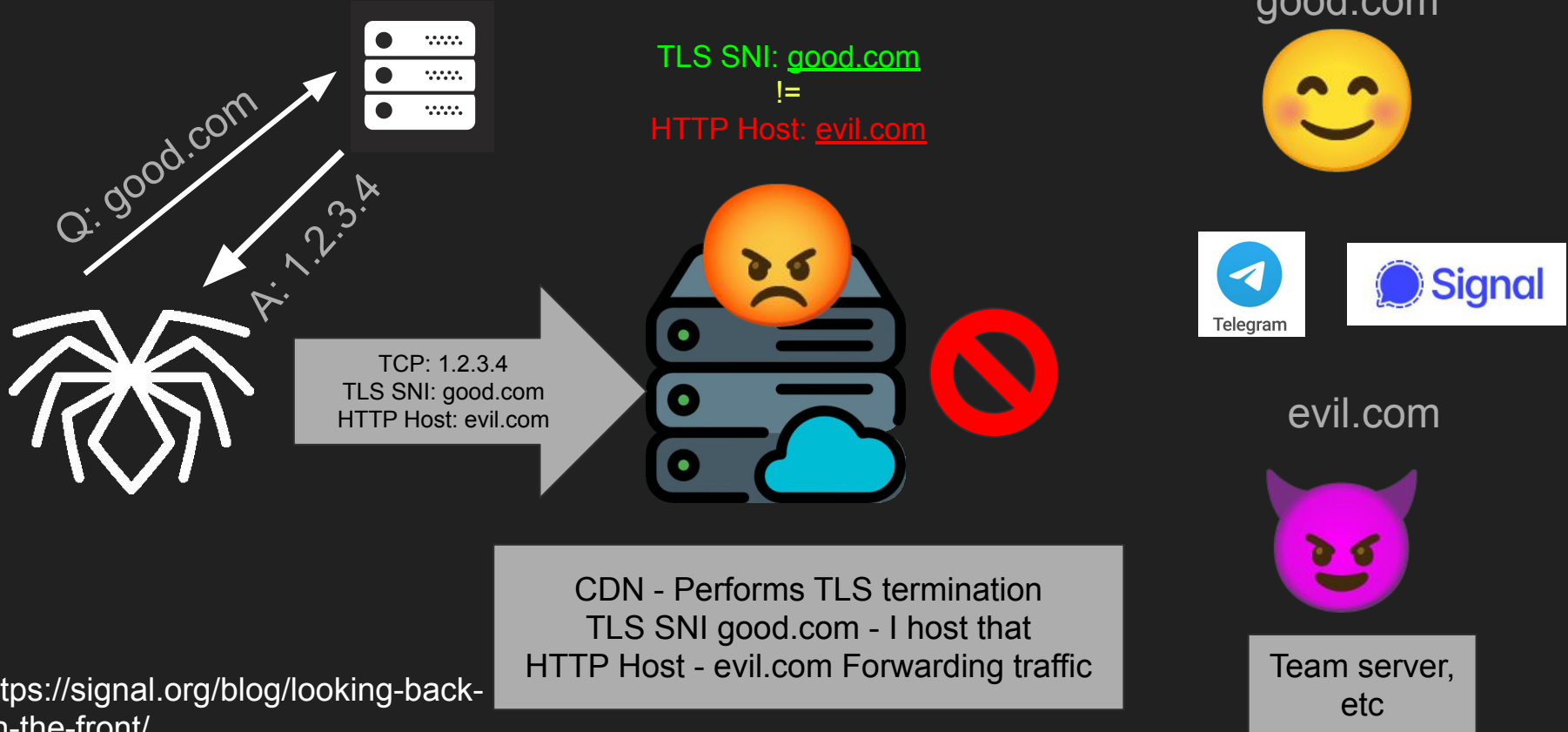
Content Delivery Network

In addition to our previous post: [Generally available: Controls to block domain fronting behavior on customer resources](#), beginning [November 8, 2022](#), all newly created Azure Front Door, Azure Front Door (classic) or Azure CDN Standard from Microsoft (classic) resources will block any HTTP request that exhibits [domain fronting behavior](#).

September 2015 - Cloudflare Changes

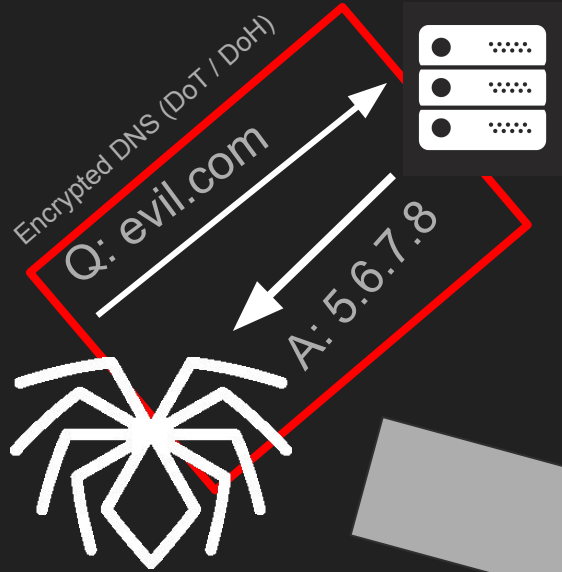
April 2018 - Google Cloud Changes

The Classic: CDN Blocking

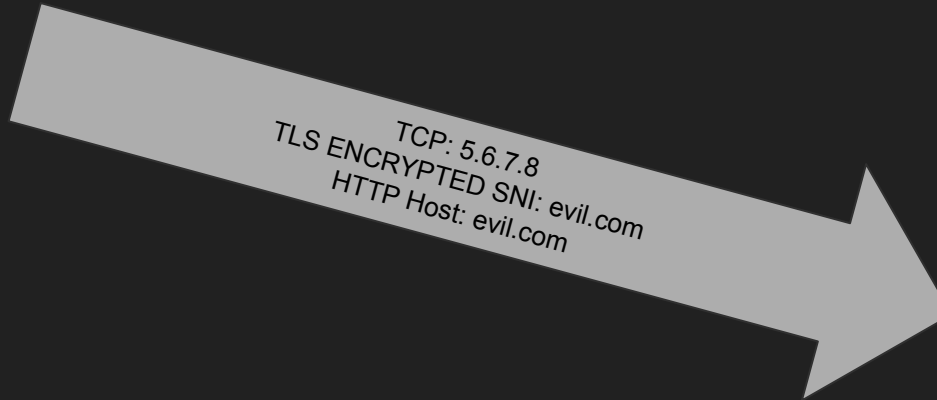


Alternatives? Yes! Many!

Alternative: Fully Encrypted



- Encrypted DNS Query (DoT / DoH)
- TLS 1.3 ESNI (Encrypted Server Name ID)
- HTTP Host (Remains Encrypted)
- No Clear text network artifacts
- IP Address of Evil server on the wire

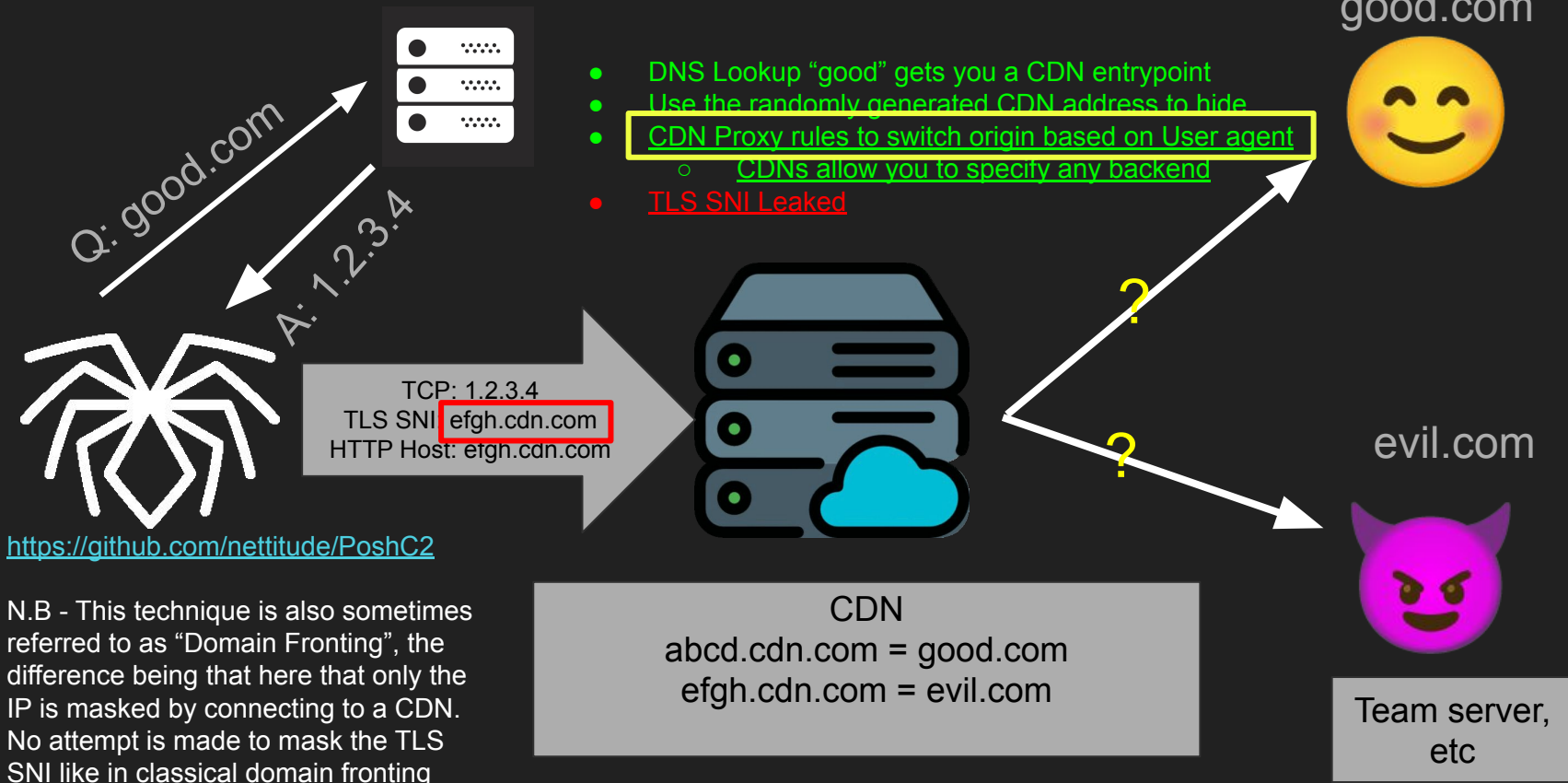


evil.com

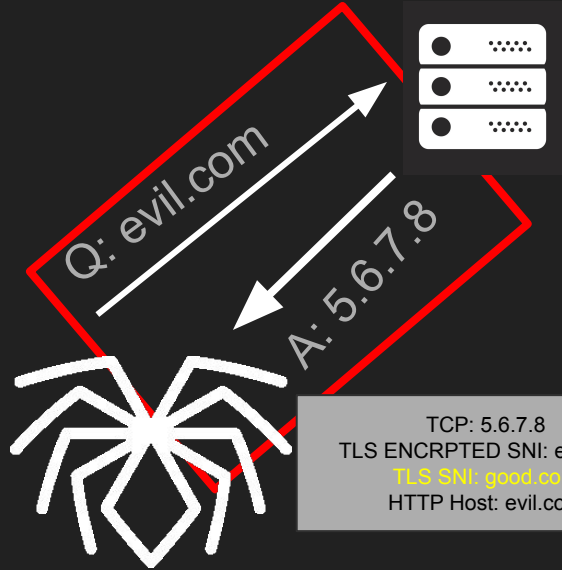


Team server,
etc

Alternative: CDN Masking (Posh C2)



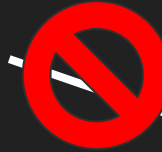
Alternative: Domain Hiding



- Encrypted DNS Query for domain (required for ESNI public key)
- TLS 1.3 ESNI (Encrypted Server Name ID)
- HTTP Host (Remains Encrypted)
- No Clear text network artifacts
- IP Address of CDN Server
- Added decoy SNI to trick defenders

- Not Widely Supported
- Blocked in Enterprises
- Blocked in Some Countries

TCP: 5.6.7.8
 TLS ENCRPTED SNI: evil.com
 TLS SNI: good.com
 HTTP Host: evil.com



evil.com




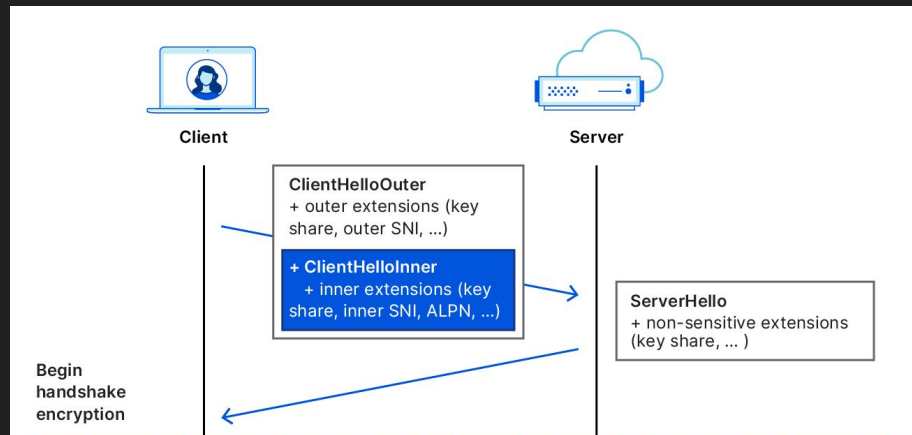
Team server,
etc

TLS SNI: good.com
 CDN
 Hosting = evil.com
 TLS Encrypted SNI: evil.com

Defcon 28
 Erik Hunstad
 "Domain Fronting is Dead,
 Long Live Domain Fronting
 Using TLS 1.3" (2020)

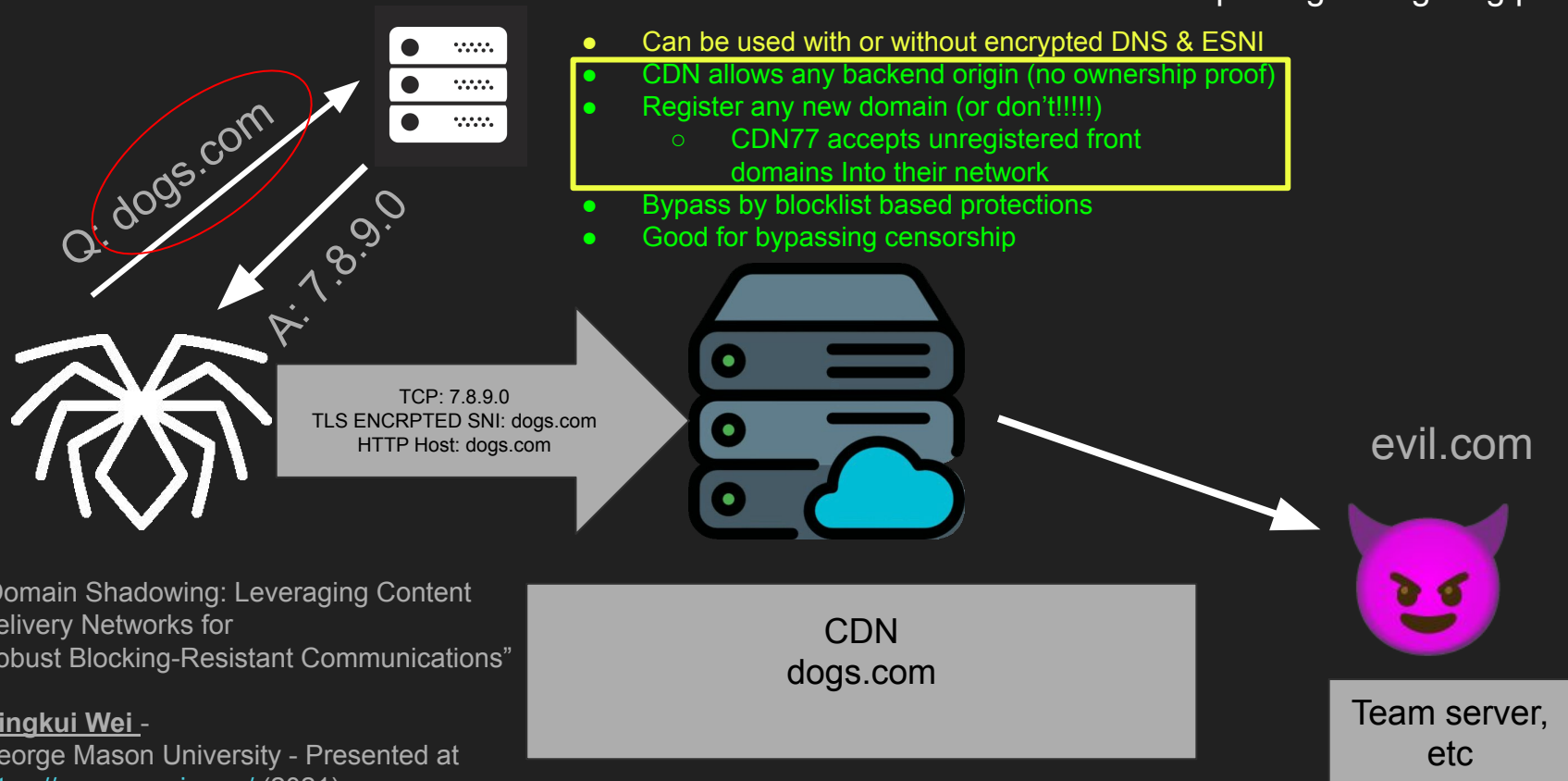
Domain Hiding - Back again?

- Decoy SNI provided with ESNI 
- “Good-bye ESNI, hello ECH!”
 - <https://blog.cloudflare.com/encrypted-client-hello/>
- TLS 1.3 standard
 - “Outer SNI” could be the new Decoy SNI!
- Possible research topic
- Jose Plascencia has beaten me to it!
- Λ - 15:00 (Red Team Village)

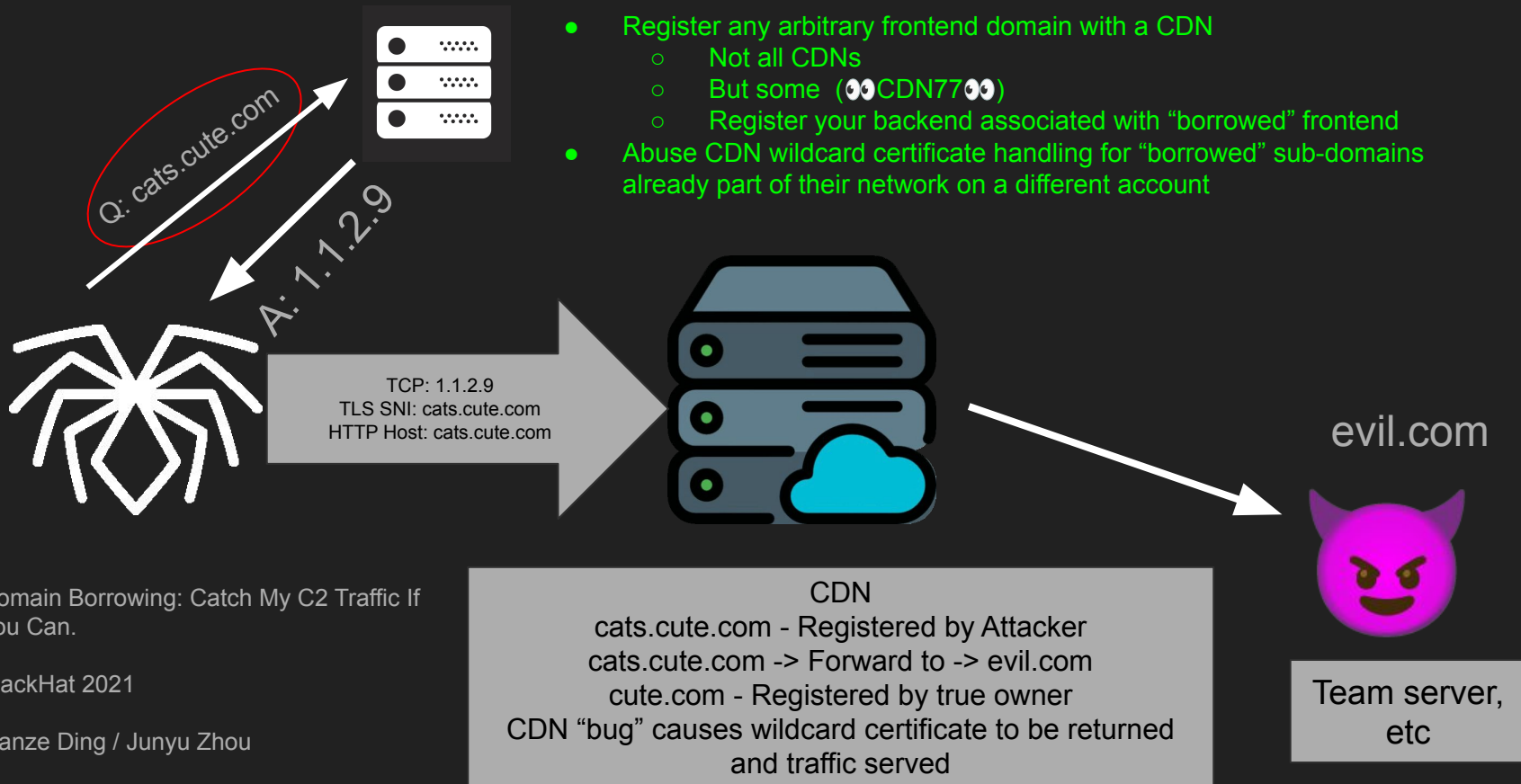


Alternative: Domain Shadowing

<https://signal.org/blog/proxy-please/>







































Last one! Alternative: Domain Borrowing



Corporate Environment & Proxies

TLS Interception Proxies - Corporate

<u>Technique</u>	<u>Passive Network Observation</u>	<u>HTTP “CONNECT” Proxy</u>	<u>Full TLS Interception</u>	<u>TLS 1.3 Required (ESNI / ECH)</u>	<u>Requires CDN “Cooperation”</u>	<u>Domain Ownership</u>
TLS SNI & Host Mismatch						
Fully Encrypted	 (ip)	 (ip)				
CDN Masking						
Domain Hiding						
Domain Shadowing						
Domain Borrowing						

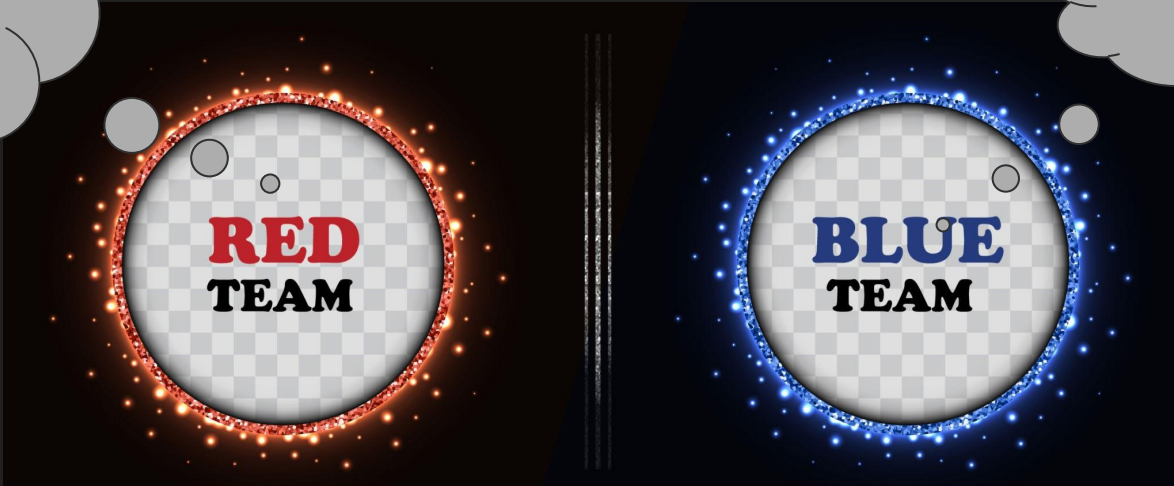
TLS Interception Proxies - Corporate

<u>Technique</u>	<u>Passive Observation</u>	<u>TLS Interception</u>	<u>TLS 1.3 Required (ESNI / ECH)</u>
TLS Miss	<div>Domain default allow list</div> <div>Eg: Palo Alto Networks Predefined Decryption Exclusions</div> <div>Excluded for Legal Reasons = Health Care / Finance</div> <div>→ Choose your domain carefully ←</div>		
Fully			
CDN			
Dom			
Dom			
Dom			<div>If forced, TLS 1.3 Only with client downgrade protections = Bypasses some proxies, Blocked by other</div>

The Shenanigans are still alive...

These techniques are still out there and still valid, have a go at them!

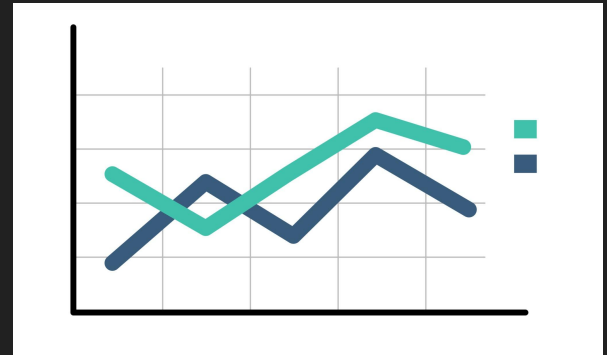
I should look
into domain
fronting for my
next
engagement



I should look
into my proxy
config

Motivation - CDN “Cooperation”

- Traditional domain fronting
- TLS SNI \neq HTTP Hosts
- How many CDN's have implemented mitigations?
- How can we take a data driven approach?
- Talk about optics, companies do not want to be seen supporting malware but also don't want to help support bypass censorship, allowed countries to control internet traffic



“Great Science is built on the Shoulders of Giants” - Sir Isaac Newton



Discovering and Measuring CDNs Prone to Domain Fronting

Karthika Subramani
Georgia Institute of Technology
Atlanta, Georgia, USA
ksubramani@gatech.edu

Roberto Perdisci
University of Georgia
Athens, Georgia, USA
Georgia Institute of Technology
Atlanta, Georgia, USA
perdisci@uga.edu

Pierros-Christos Skafidas
Georgia Institute of Technology
Atlanta, Georgia, USA
pskafidas3@gatech.edu

Manos Antonakakis
Georgia Institute of Technology
Atlanta, Georgia, USA
manos@gatech.edu

ABSTRACT

Domain fronting is a network communication technique that involves leveraging (or abusing) content delivery networks (CDNs) to disguise the final destination of network packets by presenting them as if they were intended for a different domain than their actual endpoint. This technique can be used for both benign and

1 INTRODUCTION

Domain Fronting, a technique designed to mask the true endpoints in network communications, works by leveraging (or abusing) shared hosting infrastructure provided by widespread services such as Content Delivery Networks (CDNs). By leveraging a CDN's shared infrastructure, applications may appear to connect to a de-



Methodology - 2023

- <https://github.com/karthikaS03/DomainFrontingDiscovery>
- Gather a list of CDNs and their subdomains (Manual process)
 - BunnyCDN = xyz.bunnycdn.com
- Gather Passive DNS Dataset (March 20, 2023 -> March 30, 2023)
 - University networks
 - Tranco & Active DNS Project



Methodology - 2023

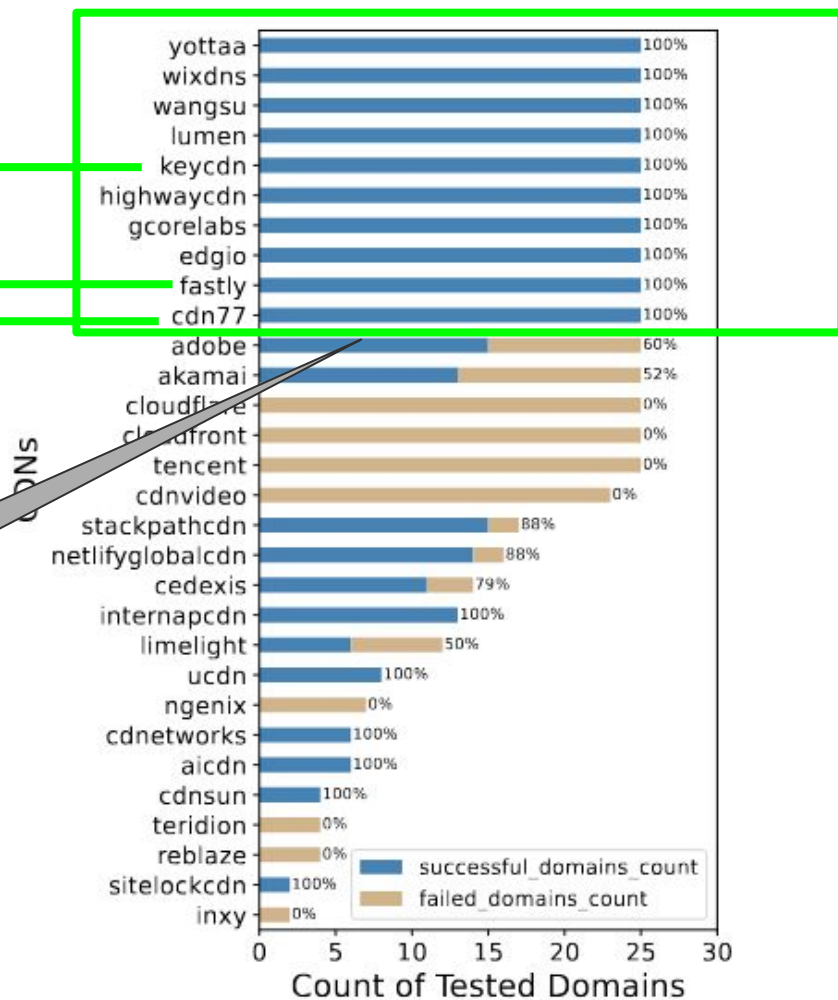
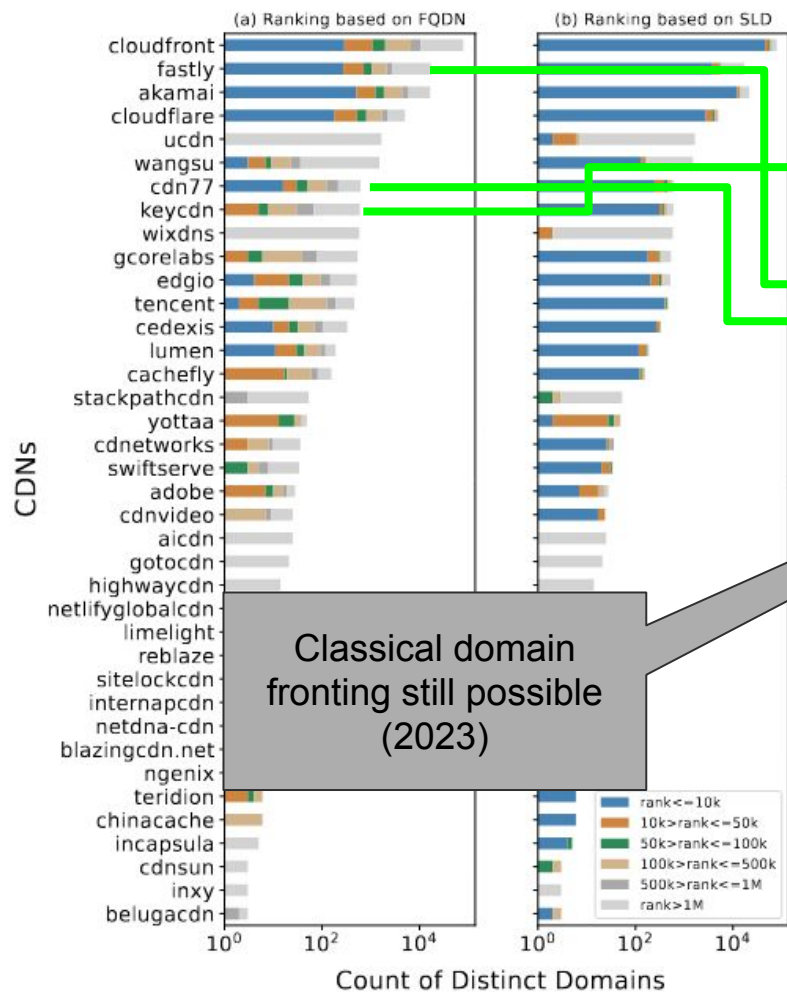


- Match Domains visited with CDN subdomains via CNAME search
 - dogs.com -> dogs.bunnycdn.com -> Asset
- Crawl websites with Puppeteer to find Static CDN served resources
 - “/” not reliable
 - Crawler_module - find static assets (*.css / *.js)

Methodology - 2023

- Create cross domain test cases
 - dogs.com - BunnyCDN
 - cats.com - BunnyCDN
 - Test host running being the same CDN
- Test Sites!
 - fronting_tester_module





Replicating the Research - 2025

- <https://github.com/copethomas/defcon-33-domain-fronting>
- <https://github.com/copethomas/DomainFrontingDiscovery>
- **Use ASN numbers instead of subdomains**
 - www.lloydsbank.com -> s5933.cdn.lloydsbanking.com -> Akamai address without using the Akamai subdomain (akamaiedge.net)
- Gather a list of CDNs and their ASN (Manual Process, bit of a pain)
 - BunnyCDN = AS200325
 - A lot can change in 2 years - CDN companies get bought and merged...

Replicating the Research - 2025

- ASN to IP (<https://iptoasn.com>)
 - `go run cmd/cdn-asn-ip-map/main.go`
- Gather Passive DNS Dataset
 - University networks ❌
 - Tranco & Active DNS Project 👻❌

Replicating the Research - 2025

- Gather a active DNS Dataset!
 - Use the <https://tranco-list.eu/> - For the top 1 million domains
 - Resolve DNS using an array of global resolvers
 - Map the IP -> ASN Range -> CDN
 - `go run cmd/resolve/main.go`
 - Now I have a nice list of Website -> CDN

```
2025-07-09T18:32:40+01:00 WRN no cdn error="no cdn found for this IP" domain=2knowmyself.com
2025-07-09T18:32:41+01:00 WRN no cdn error="no cdn found for this IP" domain=coralgames.com.au
2025-07-09T18:32:41+01:00 WRN no cdn error="no cdn found for this IP" domain=vulkan-casino-classic.xyz
2025-07-09T18:32:41+01:00 ERR Error processing domain error="DNS lookup failed for lgmv2f4ikdxkuk08qfnz.com after trying all DNS servers: all DNS servers failed to resolve lgmv2f4ikdxkuk08qfnz.com: lookup lgmv2f4ikdxkuk08qfnz.com on 8.8.8.8:53: no such host" domain=lgmv2f4ikdxkuk08qfnz.com
2025-07-09T18:32:41+01:00 ERR Error processing domain error="DNS lookup failed for kins.re.kr after trying all DNS servers: all DNS servers failed to resolve kins.re.kr: lookup kins.re.kr: i/o timeout" domain=kins.re.kr
2025-07-09T18:32:42+01:00 ERR Error processing domain error="DNS lookup failed for gonextpro.net after trying all DNS servers: all DNS servers failed to resolve gonextpro.net: lookup gonextpro.net on 8.8.8.8:53: no such host" domain=gonextpro.net
2025-07-09T18:32:42+01:00 WRN no cdn error="no cdn found for this IP" domain=lwin-application.xyz
2025-07-09T18:32:42+01:00 INF resolved cdn result={"CDN":"Cloudflare","DomainSLD":"hipmibintan.org","IP":"104.21.64.1"}
2025-07-09T18:32:42+01:00 ERR Error processing domain error="DNS lookup failed for gonextpro.net after trying all DNS servers: all DNS servers failed to resolve gonextpro.net: lookup gonextpro.net on 8.8.8.8:53: no such host" domain=gonextpro.net
```

Replicating the Research - 2025

- I'm not a web scraping expert
 - Web scraping is a pain
 - I would use <https://commoncrawl.org/> next time
- 1M domains 😬 -> 100 per CDN 😄 41 different CDNs

5. (Optional) Split the domain list into a smaller CDN selection

One million is a quite large number of domains so we can run a simple script to cut down the number we are going to test to save time on web scraping.

```
./domain_cdn_sub_selection/domain_cdn_sub_selection.sh 2>&1 | tee -a domain_cdn_sub_selection/
```

► Output

(Optional)

Combine all the subsections into a new macro-selection

```
echo "cdn, domain_sld, ip_addr" > domains_to_cdn_macro_selection.csv && cat domain_cdn_sub_selec
```

Replicating the Research - 2025

Clone Patched tooling

```
git clone git@github.com:copethomas/DomainFrontingDiscovery.git
```

Update inputs and outputs in the config file:

```
vi src/config.ini
```

```
[FILE_PATHS]
```

```
crawling_results_path = <repo_location>/data/crawler_results
```

```
cdn_domain_mapping_file_path = <defcon-2025-domain-fronting_repo_location>/domains_to_cdn_macro
```

Create python env and install packages:

```
python3 -m venv .venv
```

```
.venv/bin/pip install -r requirements.txt
```

Change working dir:

```
cd src/crawler_module
```

Run the Crawler:

```
../../.venv/bin/python3 crawl_urls.py | tee -a crawler_module.log
```

Web Scraping
time: ~1 Day

41 CDNS x
~100 each

▼ Akamai_helpspot.com

- helpspot.com_content.mhtml
- helpspot.com_headers.json
- helpspot.com_page.html
- helpspot.com_screenshot.png

Replicating the Research - 2025

```
␣"server_info":{"ip":"142.251.30.95","port":443}},{"response_url":"https://www.helpspot.com/build/assets/app-E8Grr6UP.js",␣
␣"responce_status":200,"header":{"cache-control":"max-age=31536000","cf-cache-status":"HIT","cf-ray":"963eb9364e17ef55-LHR",␣
␣"content-encoding":"gzip","content-type":"application/javascript; charset=utf-8","date":"Wed, 23 Jul 2025 22:48:07 GMT",␣
␣"etag":"W/\"680eaaa9-1faec\"",\"last-modified\":\"Sun, 27 Apr 2025 22:07:37 GMT\", \"server\":\"cloudflare\", \"vary\":\"Accept-Encoding\",␣
␣\"x-content-type-options\":\"nosniff\" \"x-frame-options\":\"SAMEORIGIN\" \"x-xss-protection\":\"1; mode=block\"}, \"server_info\":{\"ip\":\"172
␣.66.40.191\", \"port\":443}}, {"response_url":"https://www.helpspot.com/images/logos/logo-light.svg", \"responce_status\":200,␣
␣\"header\":{\"cache-control\": \"max-age=31536000\", \"cf-cache-status\": \"hit\", \"cf-ray\": \"963eb9364e17ef55-LHR\",␣
␣\"content-encoding\": \"gzip\", \"content-type\": \"image/svg+xml\", \"date\": \"Wed, 23 Jul 2025 22:48:07 GMT\", \"etag\": \"W/\"680e66b3-1fffd\" \" \"
```

All assets loaded on the page (including ones via a CDN)

Replicating the Research - 2025

- Filter URLs for only static resources
- Create tests cases for domains hosted behind the same CDN
- Perform tests to see if domain fronting is successful for each test case
- `python FrontingTester.py batch`

```
'''  
~~~~~
```

Brief explanation of test types

AHAD: Check if the Url download works when setting host name the same as the target SNI

AHFD: Check if domain fronting works when changing the SNI to front domain while the host is target domain

FHFD: Check if the URL doesn't download the same resource when SNI and Host point to the fronting domain

```
'''  
~~~~~
```

```
""" At the fronting domain level, we can't test the same content
```

```
df_success = df_front_cases[(df_front_cases['test_result']=="Success") &  
                             (df_front_cases['output_digest']==df_front_cases['original_digest']) &  
                             (df_front_cases['output_digest']!=df_front_cases['fhfd_digest']) &  
                             (df_front_cases['attack_host']!=df_front_cases['front_domain'])  
                             ]
```

Replicating the Research - 2025

```
Tue Jul 29 07:38:20 2025 Testing KeyCDN :: 2 attack domains and 6 front domains in total!!
Testing 2 URLs under domain :: ('onsen.io',) front_domain extmanagers.com
Testing 2 URLs under domain :: ('onsen.io',) front_domain extmanagers.com
Testing 2 URLs under domain :: ('onsen.io',) front_domain kxcdn.com
Testing 2 URLs under domain :: ('onsen.io',) front_domain keycdn.com
https://www.keycdn.com/images/common/d_quote.png
https://www.keycdn.com/scripts/popup.js
Testing 1 URLs under domain :: ('extmanagers.com',) front_domain onsen.io
Testing 1 URLs under domain :: ('extmanagers.com',) front_domain kxcdn.com
Testing 1 URLs under domain :: ('extmanagers.com',) front_domain onsen.io
Testing 1 URLs under domain :: ('extmanagers.com',) front_domain keycdn.com
https://www.keycdn.com/favicon.ico
Tue Jul 29 07:38:30 2025 Testing CacheFly :: 4 attack domains and 13 front domains in total!!
Testing 2 URLs under domain :: ('jspm.io',) front_domain altb.com
Testing 2 URLs under domain :: ('jspm.io',) front_domain ps0z.com
error during 'get_certificate_details' for instantcdn.net : [SSL: CERTIFICATE_VERIFY_FAILED] certificate
Testing 2 URLs under domain :: ('jspm.io',) front_domain instantcdn.net
Testing 2 URLs under domain :: ('jspm.io',) front_domain torix.ca
Testing 2 URLs under domain :: ('jspm.io',) front_domain memes.com
Testing 2 URLs under domain :: ('jspm.io',) front_domain ps0z.com
Testing 2 URLs under domain :: ('jspm.io',) front_domain porngifs.com
Testing 2 URLs under domain :: ('jspm.io',) front_domain edgeuno.com
Testing 2 URLs under domain :: ('jspm.io',) front_domain porngifs.com
Testing 2 URLs under domain :: ('jspm.io',) front_domain cachefly.com
Testing 2 URLs under domain :: ('jspm.io',) front_domain altb.com
https://www.cachefly.com/app%3Ajspm-io%40dev/src/landing.css
https://www.cachefly.com/app%3Ajspm-io%40dev/src/index.js
https://www.cachefly.com/app%3Ajspm-io%40dev/src/landing.css
https://www.cachefly.com/app%3Ajspm-io%40dev/src/index.js
https://www.cachefly.com/app%3Ajspm-io%40dev/src/landing.css
https://www.cachefly.com/app%3Ajspm-io%40dev/src/index.js
https://www.cachefly.com/app%3Ajspm-io%40dev/src/landing.css
https://www.cachefly.com/app%3Ajspm-io%40dev/src/index.js
https://www.cachefly.com/app%3Ajspm-io%40dev/src/landing.css
https://www.cachefly.com/app%3Ajspm-io%40dev/src/index.js
https://www.cachefly.com/app%3Ajspm-io%40dev/src/landing.css
https://www.cachefly.com/app%3Ajspm-io%40dev/src/index.js
```

41 different CDNs
2371 Domains X 2+ test cases

Took about ~3 Days

Testing each Site & CDN for
domain fronting support...

Full download 6GB test results
available on Git Repo

The Results!


2025 Results!


CDNs supporting domain fronting:

9/41 CDNs tested support domain fronting = 21.9%

CDN: BelugaCDN, Count: 21


CDN: Bunny.net, Count: 153


CDN: Fastly, Count: 220 

CDN: Wangsu, Count: 92 

CDN: EdgeNext, Count: 237

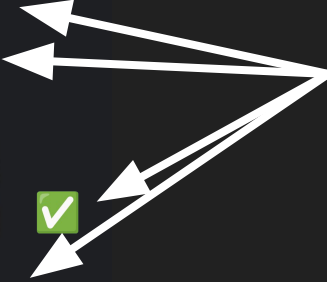
CDN: Imperva_CDN, Count: 656

CDN: G-Core_Labs, Count: 145 

CDN: CDN77, Count: 1038 

CDN: CacheFly, Count: 80

Many CDNs tested in 2023 have not
changed their support



Domain Fronting is still very possible!

2023 vs 2025

- Testing for domain fronting can be difficult
- Testing methodology is not perfect
 - Does not take into account CDN ToS and “grandfather” accounts
- **Domain fronting is still possible across a selection of CDNs**

Summary

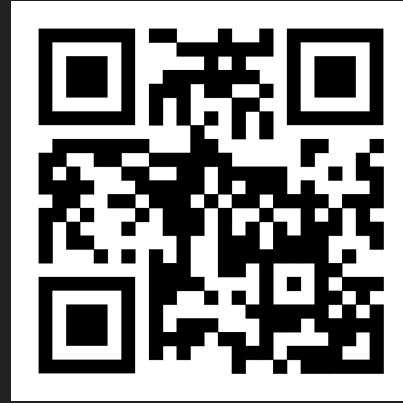
- “Classic Domain Fronting” still works!
- There is more than 1 way to hide your traffic
- Every CDN has their quirks!
- CDNs are consolidating
 - Many from 2023 no longer exist or have merged with other CDNs
- Good Luck, Have Fun!

Thank you!

Any questions?



My Website ->



<- Slides

Add me on
Linkedin ->

