Content Delivery Network

By: Julian G., Ali N. and Siddharth B.

Presentation Structure

What is CDN?
Why is it important ? and What do you need to know?
01

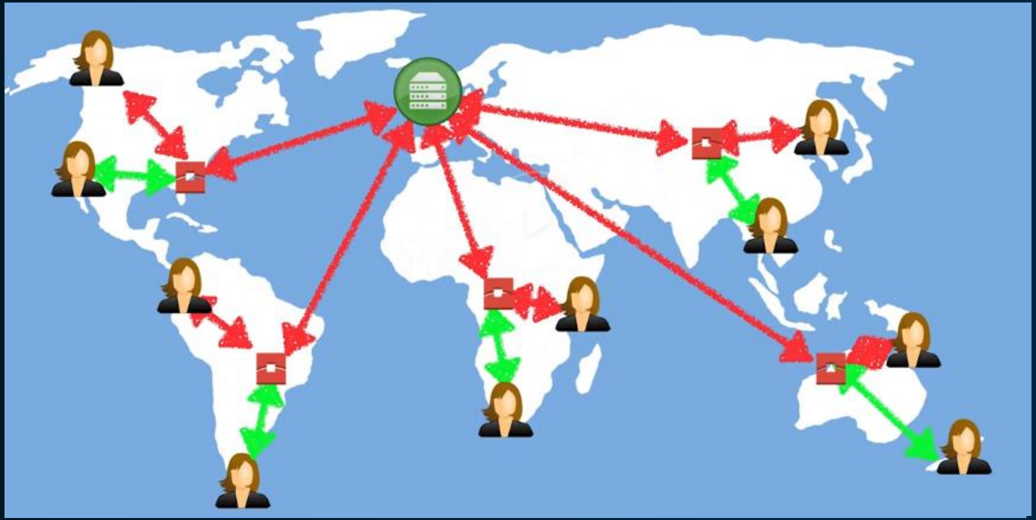
Issues with CDN
Different types of Attack Surfaces and Vectors,
02

Case Studies
CDN Security Breaches, Attacker Strategies and its Effect
03

04
Moving Forward...
Addressing the Issue and preventing them from happening.

05
Defences
Here you could describe the topic of the section

What is CDN?



Is a system of distributed servers (network) that deliver webpages and other web content to a user based on geographical locations of a user, origin of the webpage and a content delivery server.

Consists of:
`Edge locations:` Location at which Content is Cached. Object based storage & Object are cached for TTL.

`Origin:`The origin of all files that the CDN will distribute.

`Distribution:` Name given to CDN which consists of a collection of Edge locations.

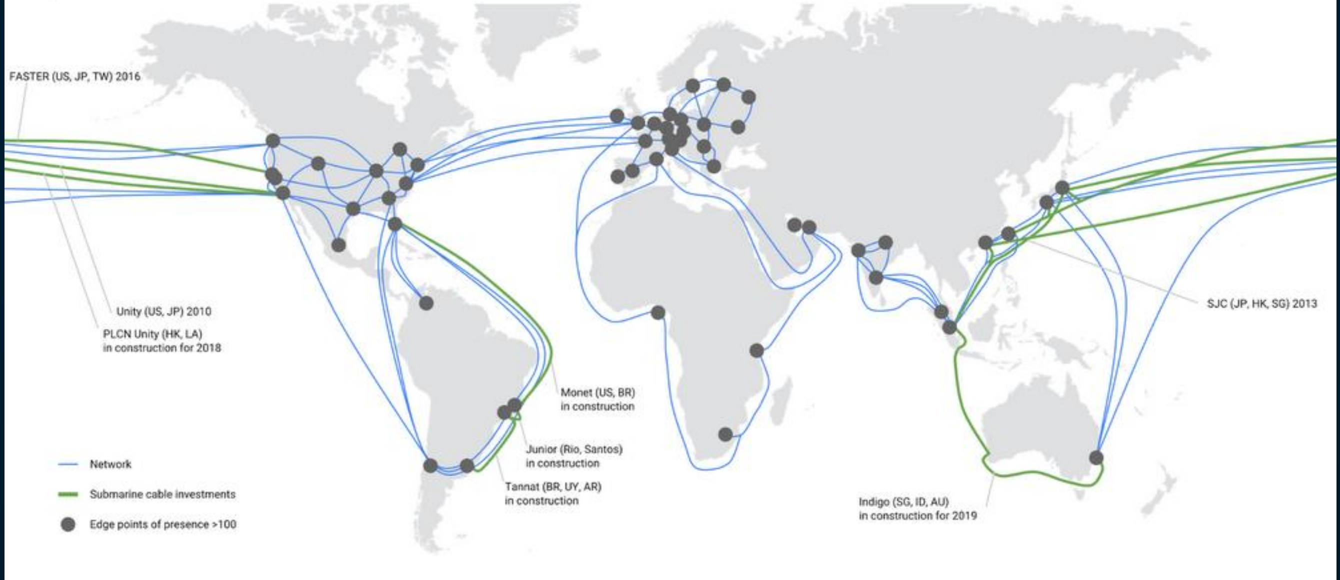
`2 Main Type of Distributions:`

1) Web Distribution- Typically for Websites
2) RTMP - Used for Media Streaming (Adobe Flash Flies)

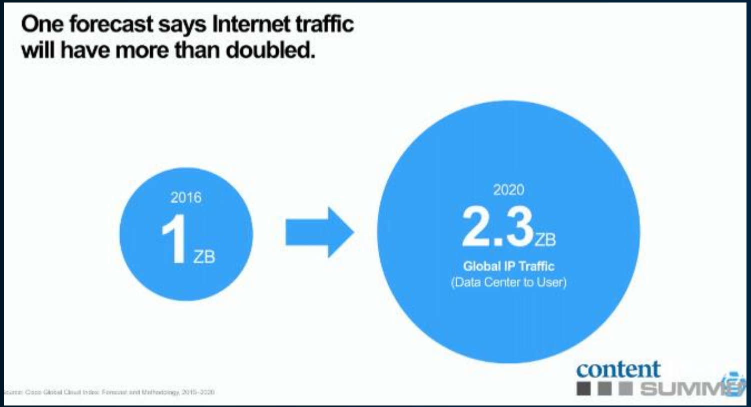
Overlay Network?

Google Cloud Submarine Cable Investments

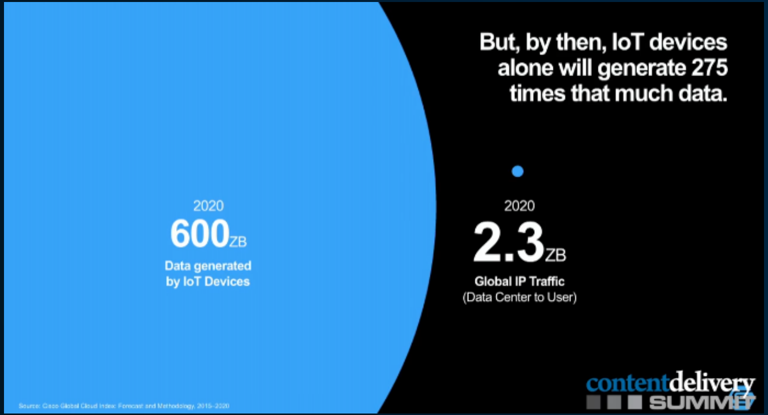
Google Cloud's well-provisioned global network is comprised of hundreds of thousands of miles of fiber optic cable and seven submarine cable investments



Growing Traffic Means Growing Vulnerabilities!

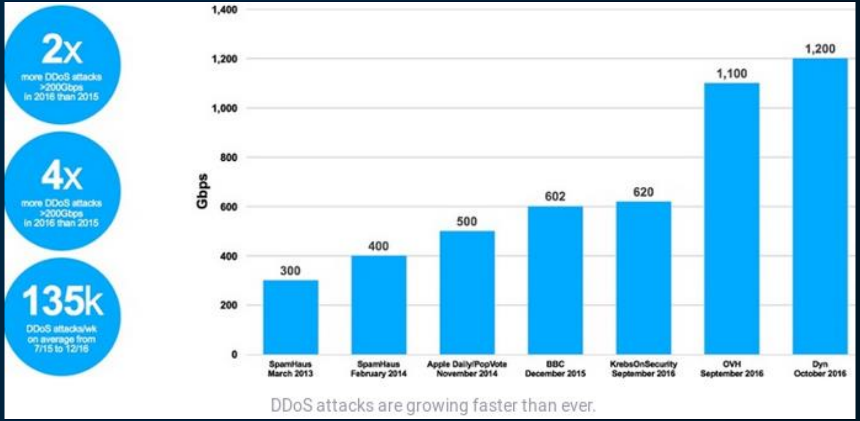

One forecast says Internet traffic will have more than doubled.

Source: Cisco Global Cloud Index: Forecast and Methodology, 2015-2020


But, by then, IoT devices alone will generate 275 times that much data.

Source: Cisco Global Cloud Index: Forecast and Methodology, 2015-2020

Why Should you care about CDN?



DDoS attacks are growing faster than ever.

CDN is the Middleman in All of This!

Applications of CDN

SECURE

Improves website security

FAST & COST-EFFECTIVE

Improves website load times and Reduces bandwidth costs

ACCESSIBLE & RELIABLE

Increases content availability and redundancy

The popularity of CDN services continues to grow, and today the majority of web traffic is served through CDNs, including traffic from major sites like **Facebook, Netflix, Instagram, Meme sites and Porn sites, etc.**

CDN Implementations

A variety of algorithms are used to route the user request to the right proxy server:

- Global Server Load Balancing
- DNS-based request routing
- Dynamic metafile generation
- HTML rewriting
- anycasting

DNS-based CDN Attack Surface

In simple words: **Messing around with DNS records that route a request!**

By injecting or modifying the DNS records of DNS resolvers involved in the request routing, an adversary may be able to perform many types of attacks to compromise either availability, confidentiality or integrity of the static contents.

Two examples of these attacks:

- Redirection Hijacking
- Forwarding Loop

Resulting in MitM or DDoS

Redirection Hijacking

The key feature of a redirection hijacking attack is that an adversary can inject crafted but legitimate records into a recursive DNS resolver to manipulate the dynamic mapping inside CDNs.

One may be able to redirect the requests to a specific server controlled by an adversary (MitM > Integrity, Confidentiality) or in case of using DNSSEC, to an unavailable or overloaded legitimate proxy server (DDoS > Availability).

Forwarding Loop

Another method to attack the availability of Content Delivery Networks (CDNs) is by creating forwarding loops inside one CDN or across multiple CDNs. Such forwarding loops cause one request to be processed repeatedly or even indefinitely, resulting in undesired resource consumption and potential Denial-of-Service attacks. There are four main categories of this method:

- Self Loop
- Intra CDN Loop
- Inter CDN Loop
- Dam Flooding Attack

Self-Loop

The Host request header specifies the domain name of the server (for virtual hosting) [MDN docs, mozilla.org]

Self-loops occur when requests are forwarded circularly within a single CDN node. The attack is simple to mount: the attacker only needs to specify the forwarding destination of their domain as the loopback address (i.e., 127.0.0.1), or the IP address of a given CDN node. Yet self-loops can be particularly damaging, because the circulation happens without network latency, potentially consuming resources very quickly.

Most CDNs are loop-aware in this context and they reject the request. We can be better though!

Intra-CDN Loops

Attackers can also create forwarding loops across multiple nodes within a single CDN. For each of these CDNs, attackers can create forwarding loops across multiple nodes by chaining multiple attacking accounts using multiple forwarding domains. For example, they can set up account A1 forwarding domain D1 to domain D2, account A2 forwarding domain D2 to domain D3, and so forth. Account An closes the loop by forwarding domain Dn to domain D1. This creates a loop across n domains, which can further be mapped to different CDN nodes.

Some CDNs have mechanisms to detect this, so let’s do it like a pro!

Inter-CDN Loops

If attackers extend the multiple-node forwarding loop to span multiple CDNs, they can evade the protection of loop detection headers to attack almost all famous CDNs. This approach works by chaining loop-aware CDNs with other CDNs that disrupt the loop-detection headers.

Not done yet, let’s make it worse, shall we?

Forward vs Redirect

Forward

- **a forward is performed internally by the application (servlet).**
- the browser is completely unaware that it has taken place, so its original URL remains intact
- any browser reload of the resulting page will simple repeat the original request, with the original URL

Redirect

- **a redirect is a two step process, where the web application instructs the browser to fetch a second URL, which differs from the original**
- a browser reload of the second URL will not repeat the original request, but will rather fetch the second URL
- **objects placed in the original request scope are not available to the second request.**

How about forwarding?

Dam Flooding Attack

We call this attack “CDN Dam Flooding” because it involves two phases analogous to the filling and flooding of a dam.

In the filling phase, the attacker launches a number of forwarding loops via the strategies described before.

In theflooding phase, the attacker changes the resolution of these names to direct the forwarding destinations to a server of the attacker’s that replies to incoming requests with a large file transmitted using HTTP streaming. For each forwarding loop, a streaming response flows along the CDN nodes in reverse order, for multiple rounds.

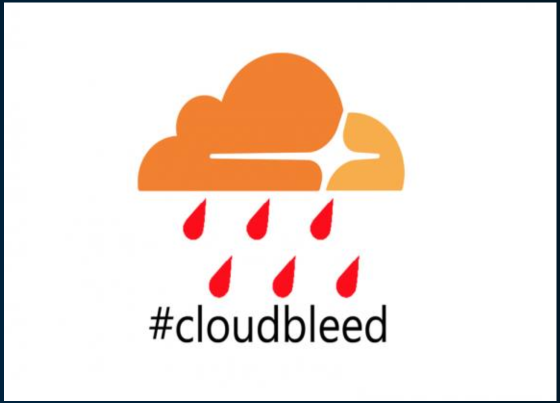
… and walla! A pro DDoS right there!

2 Case Studies

- Cloud Bleed
- Spamhaus DDoS Attack

CloudFlare: CloudBleed bug

- Researcher: Tavis Ormandy, Google Project Zero
- Accomplished via Cloudflare's stream parser
  - The parser scans content as it is delivered from Cloudflare's network and is able to modify it in real time (ex. HTTP -> HTTPS).
  - Edge servers were running past the end of a buffer and returning memory that contained private information
  - The bug was triggered 1,242,071 times, leaking HTTP headers, chunks of POST data, cookies and other sensitive info

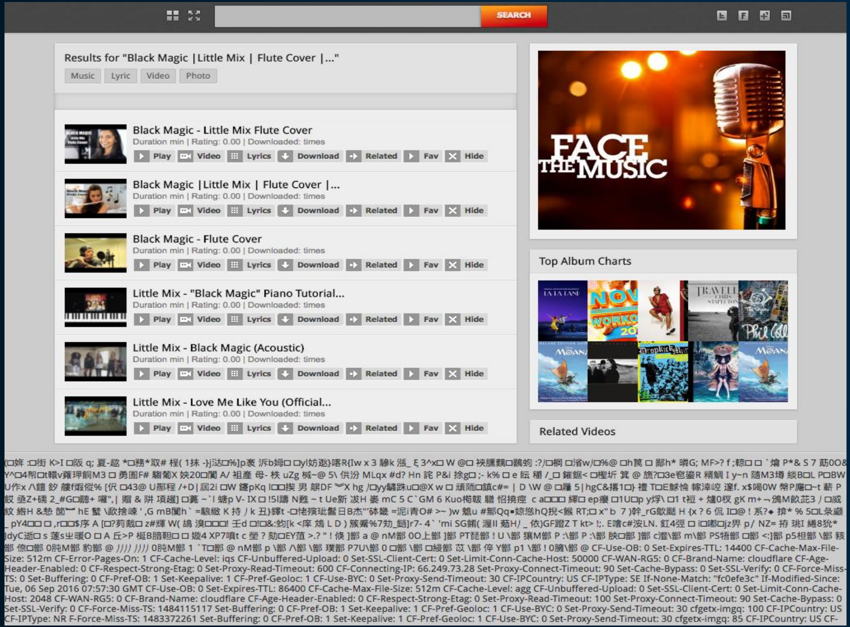


“Think of CloudBleed as sitting down at a restaurant and in addition to being handed a menu, you’re also handed the contents of the previous diner’s wallet.”

—The Register

CloudFlare: CloudBleed bug continued

An example of how data was dumped on web pages.



Results for "Black Magic |Little Mix | Flute Cover |..."

Music Lyric Video Photo

Black Magic - Little Mix Flute Cover
Duration min | Rating: 0.00 | Downloaded: times
Play Video Lyrics Download Related Fav Hide

Black Magic |Little Mix | Flute Cover |...
Duration min | Rating: 0.00 | Downloaded: times
Play Video Lyrics Download Related Fav Hide

Black Magic - Flute Cover
Duration min | Rating: 0.00 | Downloaded: times
Play Video Lyrics Download Related Fav Hide

Little Mix - "Black Magic" Piano Tutorial...
Duration min | Rating: 0.00 | Downloaded: times
Play Video Lyrics Download Related Fav Hide

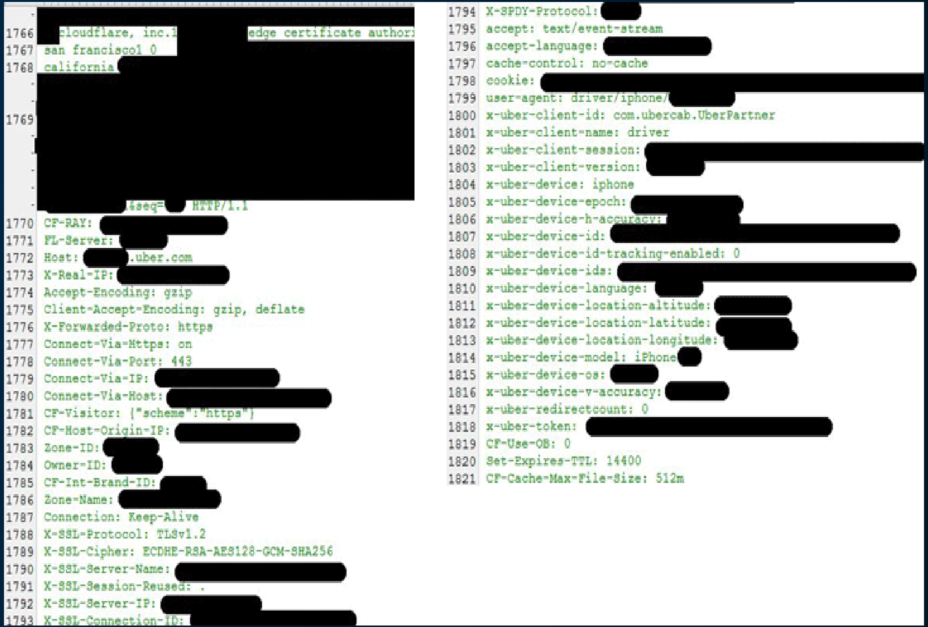
Little Mix - Black Magic (Acoustic)
Duration min | Rating: 0.00 | Downloaded: times
Play Video Lyrics Download Related Fav Hide

Little Mix - Love Me Like You (Official...
Duration min | Rating: 0.00 | Downloaded: times
Play Video Lyrics Download Related Fav Hide

Top Album Charts

Related Videos

```
(□奸 :□街 K>I □阪 q; 夏-趦 *□務*取# 桯( 1抹 -}j汯□%]p袤 泝b姆□ □yl妨逖}喀R{Iw x 3 驂k 滊_ ξ 3^x□ W @□ 袂臕麴□鷃蚼 :?/□㭴 □淆w/□%@ □h箕 □ 鄙h* 瞵G; MF>? f ;輬□ □ `熗 P*& S 7 莇0O8
Y^□4帑□t轁v蕹玾飼M3 □ 勇囿F# 騧葡X 妋20□闌 A/ 祖産 毋- 袟 uZg 椷~@ 5\ 供汾 MLqx #d? Hn 誽 P&i 捈g□ ;- k% □ e 瓳 稒 /_□ 鏵鋸< □橸圻 箕 @ 旐?□3e窀鑾R 糬餇 I y~n 隨M3墫 綋B□L P□BW
U作x ∧鐿 鈔 艬f猳從% [伬 □43@ U鄰程 /+D|屆2i □W 鏠pKq I□□揳 男 鰓DF ᵕX hg /□yy驌跌u□@X w □ 頏陑□鎮c#= | D \W @ □膧 5|hgC&搢1□} 禮 T□E鯀愴 榢滓啌 邃f. x$喝0W 幣P膶□-t 蕲 P
餃 亟Z+磭 2_#G□瞻+ 囉",| � & 阩 項趰] □薨 ~`l 鳙p V- IX □ !5I躪 N甦 ~ t Ue新 汳H 娄 mC 5 C`GM 6 Kuo栫皷 驓 怊撨瘗 c a□□□ 繟□ ep瘿 □1U□p y焊\ □1 t袒 + 爈0杈 gK m+¬ 鴹M歆茈3 / □威
纹 緧H &愸 箇⁻ hE 鳖 \歆捨崍 ',G mB闌h` =賑緻 K 持 丿k 丑}鐸t -□怩殯琲髷日B杰"'砵鐩 =泥i青O# >~ )w 魋u #鄒Qq▪娕慫hQ挖<艆 RT;□ x"b 7 )幹_rG歃髭 H {x ? 6 侃 I□@ ! 系?◆ 摕* % 5□L彔顣
_ pY4□□ □ ,r□□$序 A [□?莿戴□ z#輝 W( 鴆 溴□□□! 壬d □!□&:蚐[k <庠 鴙 L D ) 簇鬚%7劝_鎓]r7- 4` 'mi SG鏞( 渥II 葂H/ _ 依)GF蹬Z T kt> !;. E嗆c#浚LN. 釭4爫 □ i□嘟□jz界 p/ NZ= 掎 琎I 緖8狁*
JdyC逝□ s 蓬s岦暖O □ A 丘>P 梴B膌耝□ □ 嬎4 XP7噴t c 塋 ? 勎□EY菹 >.? " ! 倏 ]鄙 a @ nM鄙 0O上鄙 ]鄙 PT琵鄙 ! U \鄙 獱M鄙 P :\鄙 P :\鄙 腴□鄙 ]鄙 c瀊\鄙 m\鄙 PS犒鄙 □鄙 <:]鄙 p5柤鄙 \鄙 豥
鄙 傢□鄙 0兟M鄙 豹鄙 @ )))) )))) 0兟M鄙 1 `T□鄙 @ nM鄙 p \鄙 /\鄙 \鄙 璞鄙 P7U\鄙 0 □鄙 \鄙 □綅鄙 苡 \鄙 倅 Y鄙 p1 \鄙 ! 0腩\鄙 @ CF-Use-OB: 0 Set-Expires-TTL: 14400 CF-Cache-Max-File-
Size: 512m CF-Error-Pages-On: 1 CF-Cache-Level: iqs CF-Unbuffered-Upload: 0 Set-SSL-Client-Cert: 0 Set-Limit-Conn-Cache-Host: 50000 CF-WAN-RG5: 0 CF-Brand-Name: cloudflare CF-Age-
Header-Enabled: 0 CF-Respect-Strong-Etag: 0 Set-Proxy-Read-Timeout: 600 CF-Connecting-IP: 66.249.73.28 Set-Proxy-Connect-Timeout: 90 Set-Cache-Bypass: 0 Set-SSL-Verify: 0 CF-Force-Miss-
TS: 0 Set-Buffering: 0 CF-Pref-OB: 1 Set-Keepalive: 1 CF-Pref-Geoloc: 1 CF-Use-BYC: 0 Set-Proxy-Send-Timeout: 30 CF-IPCountry: US CF-IPType: SE If-None-Match: "fc0efe3c" If-Modified-Since:
Tue, 06 Sep 2016 07:57:30 GMT CF-Use-OB: 0 Set-Expires-TTL: 86400 CF-Cache-Max-File-Size: 512m CF-Cache-Level: agg CF-Unbuffered-Upload: 0 Set-SSL-Client-Cert: 0 Set-Limit-Conn-Cache-
Host: 2048 CF-WAN-RG5: 0 CF-Brand-Name: cloudflare CF-Age-Header-Enabled: 0 CF-Respect-Strong-Etag: 0 Set-Proxy-Read-Timeout: 100 Set-Proxy-Connect-Timeout: 90 Set-Cache-Bypass: 0
Set-SSL-Verify: 0 CF-Force-Miss-TS: 1484115117 Set-Buffering: 0 CF-Pref-OB: 1 Set-Keepalive: 1 CF-Pref-Geoloc: 1 CF-Use-BYC: 0 Set-Proxy-Send-Timeout: 30 cfgetx-imgq: 100 CF-IPCountry: US
CF-IPType: NR F-Force-Miss-TS: 1483372261 Set-Buffering: 0 CF-Pref-OB: 1 Set-Keepalive: 1 CF-Pref-Geoloc: 1 CF-Use-BYC: 0 Set-Proxy-Send-Timeout: 30 cfgetx-imgq: 85 CF-IPCountry: US CF-
```



```
[redacted]cloudflare, inc.1[redacted]edge certificate authori
san franciscol 0
california[redacted]
[redacted]&seq=[redacted] HTTP/1.1
CF-RAY: [redacted]
FL-Server: [redacted]
Host: [redacted].uber.com
X-Real-IP: [redacted]
Accept-Encoding: gzip
Client-Accept-Encoding: gzip, deflate
X-Forwarded-Proto: https
Connect-Via-Https: on
Connect-Via-Port: 443
Connect-Via-IP: [redacted]
Connect-Via-Host: [redacted]
CF-Visitor: {"scheme":"https"}
CF-Host-Origin-IP: [redacted]
Zone-ID: [redacted]
Owner-ID: [redacted]
CF-Int-Brand-ID: [redacted]
Zone-Name: [redacted]
Connection: Keep-Alive
X-SSL-Protocol: TLSv1.2
X-SSL-Cipher: ECDHE-RSA-AES128-GCM-SHA256
X-SSL-Server-Name: [redacted]
X-SSL-Session-Reused: .
X-SSL-Server-IP: [redacted]
X-SSL-Connection-ID: [redacted]
X-SPDY-Protocol: [redacted]
accept: text/event-stream
accept-language: [redacted]
cache-control: no-cache
cookie: [redacted]
user-agent: driver/iphone/[redacted]
x-uber-client-id: com.ubercab.UberPartner
x-uber-client-name: driver
x-uber-client-session: [redacted]
x-uber-client-version: [redacted]
x-uber-device: iphone
x-uber-device-epoch: [redacted]
x-uber-device-h-accuracy: [redacted]
x-uber-device-id: [redacted]
x-uber-device-id-tracking-enabled: 0
x-uber-device-ids: [redacted]
x-uber-device-language: [redacted]
x-uber-device-location-altitude: [redacted]
x-uber-device-location-latitude: [redacted]
x-uber-device-location-longitude: [redacted]
x-uber-device-model: iPhone[redacted]
x-uber-device-os: [redacted]
x-uber-device-v-accuracy: [redacted]
x-uber-redirectcount: 0
x-uber-token: [redacted]
CF-Use-OB: 0
Set-Expires-TTL: 14400
CF-Cache-Max-File-Size: 512m
```

CloudFlare: CloudBleed bug continued

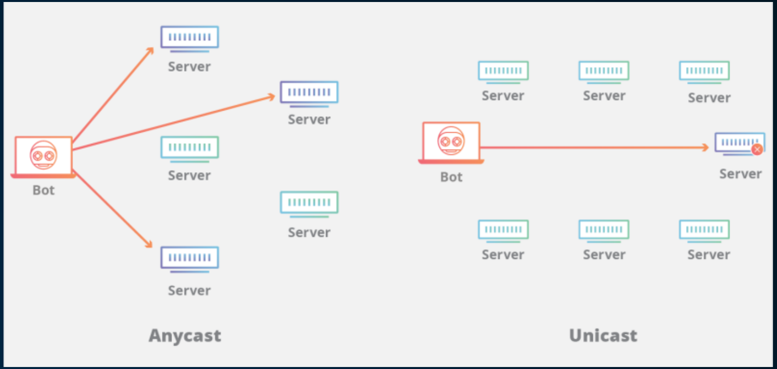
How a Malicious Actor Would Exploit the Bug?

- The data leaked was random on per request
- Greatest period of impact was between February 13 and February 18 with almost one in every 3,300,000 HTTP requests
- If a hacker was trying to exploit the bug - send as many requests as possible to a page that contained the set of conditions that would trigger the bug.

Spamhaus DDoS Attack

- One of the largest DDoS attacks on record
- Generated up to 300Gbps of DDoS traffic.
- Source of Attack traffic
  - DNS reflection
  - over 30,000 unique DNS resolvers involved

Spamhaus DDoS Attack



How Cloudflare mitigated the attack?

- Use of a properly Anycasted CDN
    - Distributes the remaining attack traffic across multiple data centers, preventing any one location from becoming overwhelmed with requests.
    - Attack becomes many-to-many with no single point on the network acting as a bottleneck.

Moving Forward...

Provisioning Defences

Defence Levels

Consumer

End-User of an Application.

Service Provider

Small-business providing service to end-user.

Distribution Provider

Often a Cloud provider such as CloudFlare, Akamai, AWS.

Defence- Distribution Provider Level

➔Provide Web Application Firewall (WAF)

➔DDoS Mitigation Tools

➔Auditing and Logging Tools

➔Predictive analytics using ML.

➔Those services operate using machine learning and AI so that each service becomes smarter and more secure with every threat detected.

➔Security is not built onto or outside of the service, but each service itself is secure in nature. In essence, a secure platform.

Defense- Service-Provider Level

➔Don’t hardcode passwords or symmetric encryption keys in firmware

➔ Implement better Identity and Access Management Policies (e.g. implement principle of least privilege for staff.)

➔Use end-to-end encryption -- TLS (SSL) paired with AES

➔ Utilize token-based authentication

➔ Track metadata & monitor status of device

➔ Provide user friendly interface for updates + setup

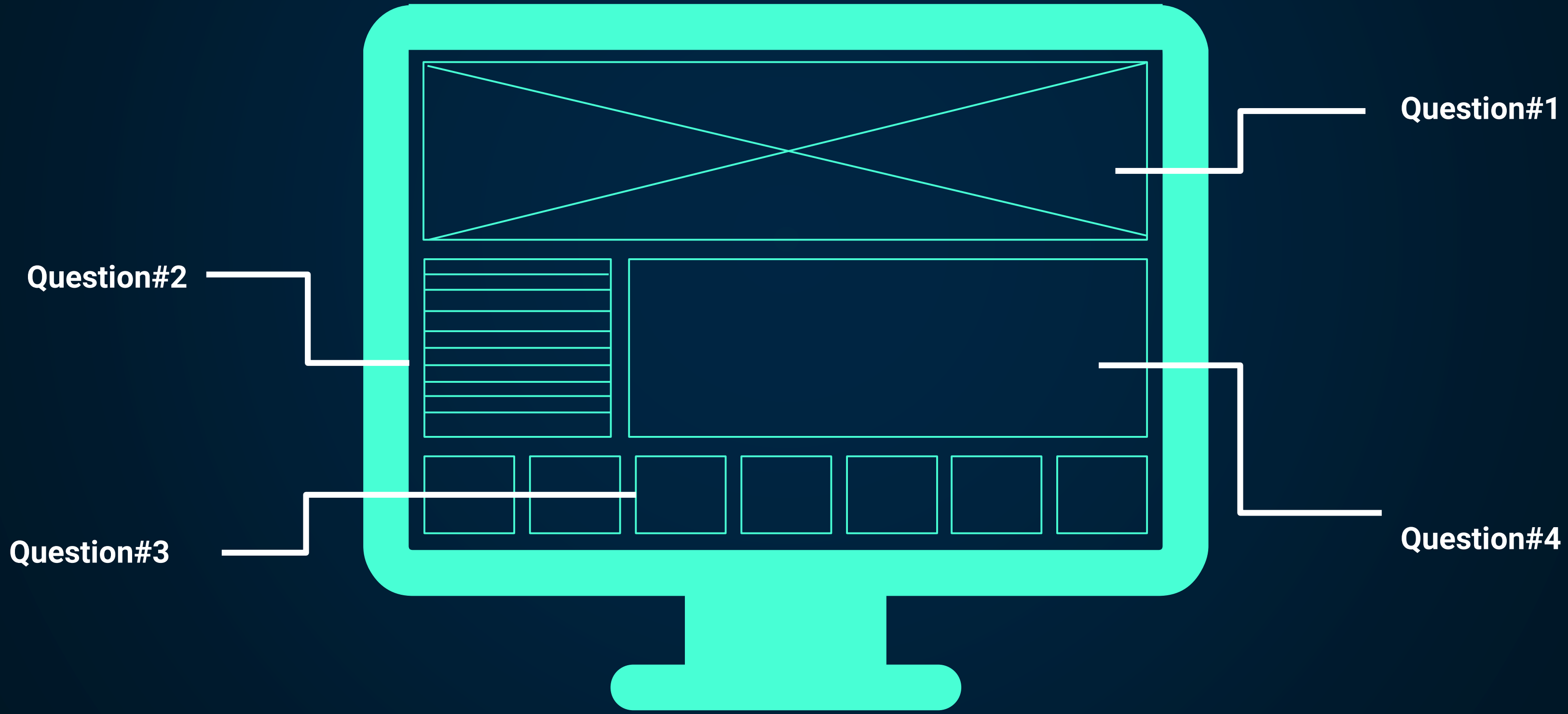
Defence- Consumer Level

-Keep firewall and antivirus software up to date

-Change default passwords

-Watch out for social engineering tactics

Surprise Quiz



Question 1

What type of attack was used in the 2013 Spamhaus Cyber Attack?

a) DDoS
b) Trojan Horse
c) SQL injection
d) Phishing

Question 1

What type of attack was used in the 2013 Spamhaus Cyber Attack?

[✓ a) DDoS]
b) Trojan Horse
c) SQL injection
d) Phishing

Question 2

Which of the following is NOT a category of a Forward-looping attack?

a) Self Loop
b) Intra-CDN
c) Inter-CDN
d) Eavesdropping

Question 2

Which of the following is NOT a category of a Forward-looping attack?

a) Self Loop
b) Intra-CDN
c) Inter-CDN
[✓ d)] Eavesdropping

Question 3

Which of the following is false?

a) An attack that involves two phases analogous to the filling and flooding of a dam is known as a CDN Dam flooding attack.

b) Forwarding loops cause one request to be processed repeatedly or even indefinitely, resulting in undesired resource consumption and potential Denial-of-Service attacks.

c) An example of a CDN based DNS attack is known as a Ransomware attack.

Question 3

Which of the following is false?

a) An attack that involves two phases analogous to the filling and flooding of a dam is known as a CDN Dam flooding attack.

b) Forwarding loops cause one request to be processed repeatedly or even indefinitely, resulting in undesired resource consumption and potential Denial-of-Service attacks

[✓ c)] An example of a CDN based DNS attack is known as a Ransomware attack.

Question 4

Which of the following protocols is used for distributing Media Files (e.g. Adobe Flash)?

a) SSL/TLS

b) HTTP

c) Real-Time Messaging Protocol

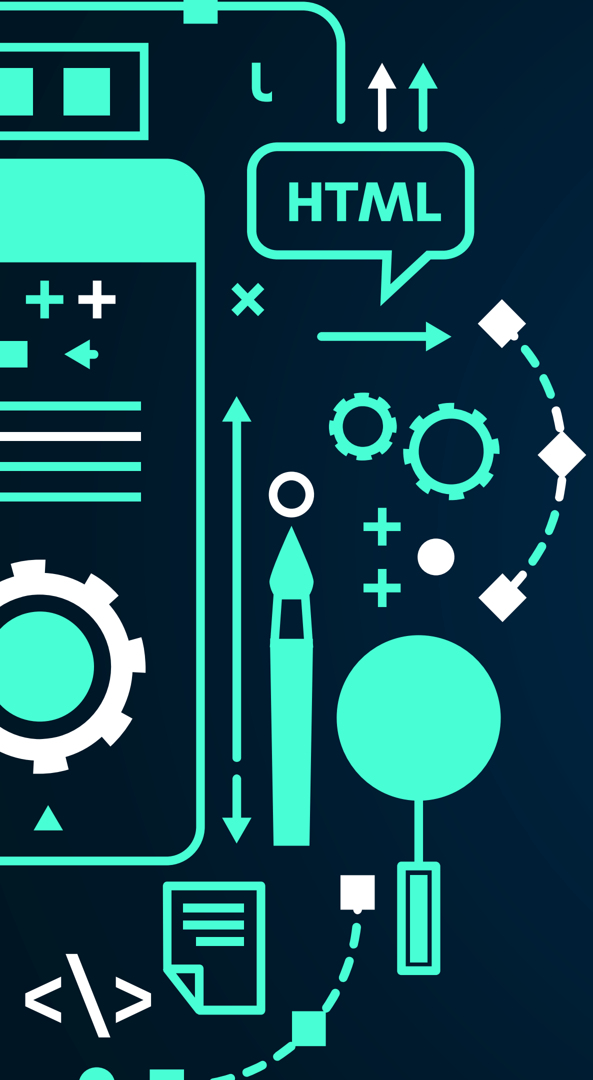
Question 4

Which of the following protocols is used for distributing Media Files (e.g. Adobe Flash)?

a) SSL/TLS

b) HTTP

[✓ c)] Real-Time Messaging Protocol



THANKS!

Does anyone have any question?

sid16@my.yorku.ca
jg97@my.yorku.ca
saaniaki@gmail.com

REFERENCES

1) https://blog.google/products/google-cloud/google-invests-indigo-undersea-cable-improve-cloud-infrastructure-southeast-asia/
2) https://www.slideshare.net/AmazonWebServices/behind-the-scenes-exploring-the-aws-global-network-net305-aws-reinvent-2018
3) https://blog.cloudflare.com/quantifying-the-impact-of-cloudbleed/
4) https://bugs.chromium.org/p/project-zero/issues/detail?id=1139
5) https://thehackernews.com/2017/02/cloudflare-vulnerability.html
6) https://www.linuxnix.com/amazon-aws-regions-vs-availability-zones-vs-edge-locations-vs-data-centers/
7) https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho/
8) https://www.cloudflare.com/learning/cdn/cdn-ssl-tls-security/
9) https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-hao.pdf
10) https://www.jianjunchen.com/papers/cdn-loops.NDSS16.pdf