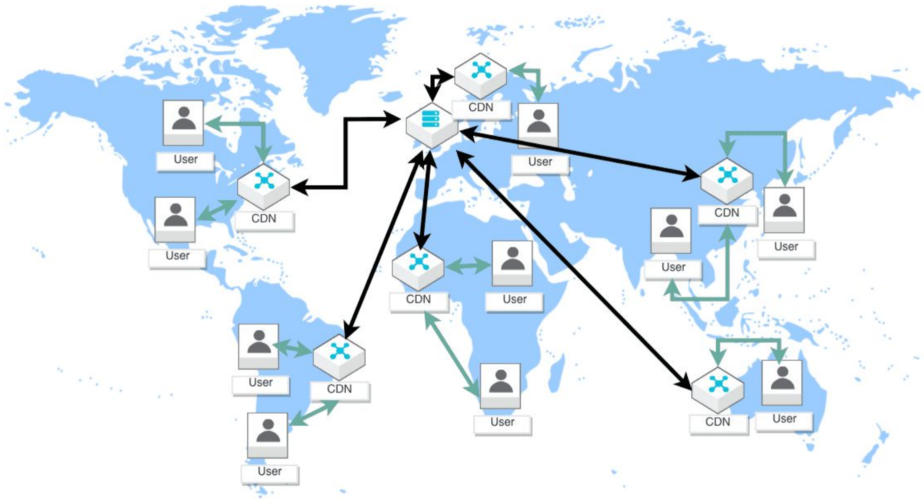
ReqsMiner: Automated Discovery of CDN Forwarding Request Inconsistencies and DoS Attacks with Grammar-based Fuzzing

Linkai Zheng, Xiang Li, Chuhan Wang, Run Guo, Haixin Duan, Jianjun Chen, Chao Zhang, Kaiwen Shen

What is a Content Delivery Network (CDN)?

- Infrastructure for performance and security
  - **Globally Distributed:** worldwide access acceleration
  - **Cache then Forward:** reduce server traffic load
  - **DDoS Protection:** off-load traffic from DDoS attack



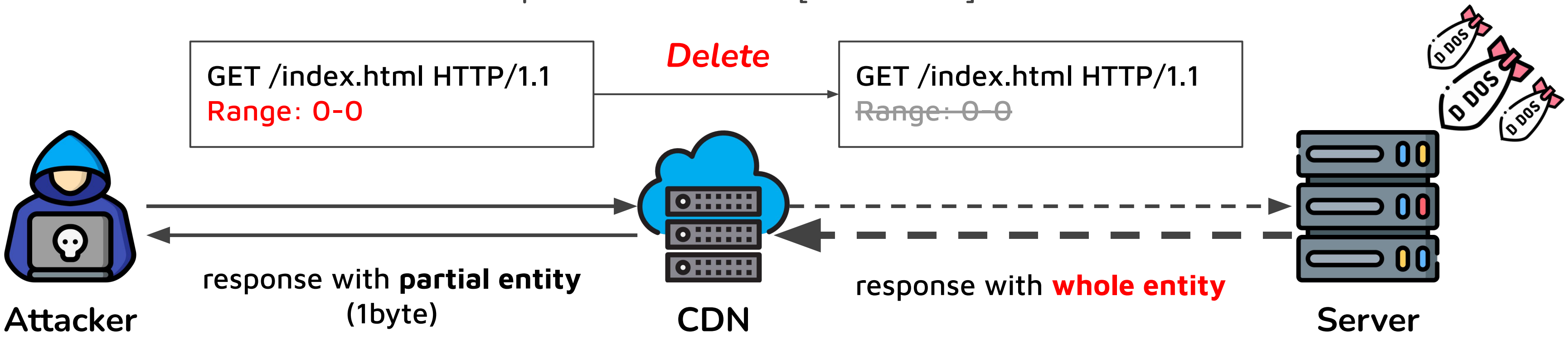
61.5%*

of the Alexa Top 10k is behind a CDN

*: BuiltWith. BuiltWith Technology Lookup. https://trends.builtwith.com/CDN/Content-Delivery-Network.

Request Inconsistencies in CDNs

- **CDNs may alter request messages**, causing request inconsistencies
- Request inconsistencies can lead to security issues
- Related works:
  - Forwarding loop attack [NDSS' 16]
  - **RangeAmp attack [DSN' 20]**
  - HTTP/2 bandwidth amplification attack [NDSS' 20]



A case study for request inconsistencies in CDNs: **RangeAmp Attack**

Our Motivation & Goals

- The majority of request inconsistencies have been discovered **manually** in prior research
- This method may result in some variations in the forwarding request being **overlooked**

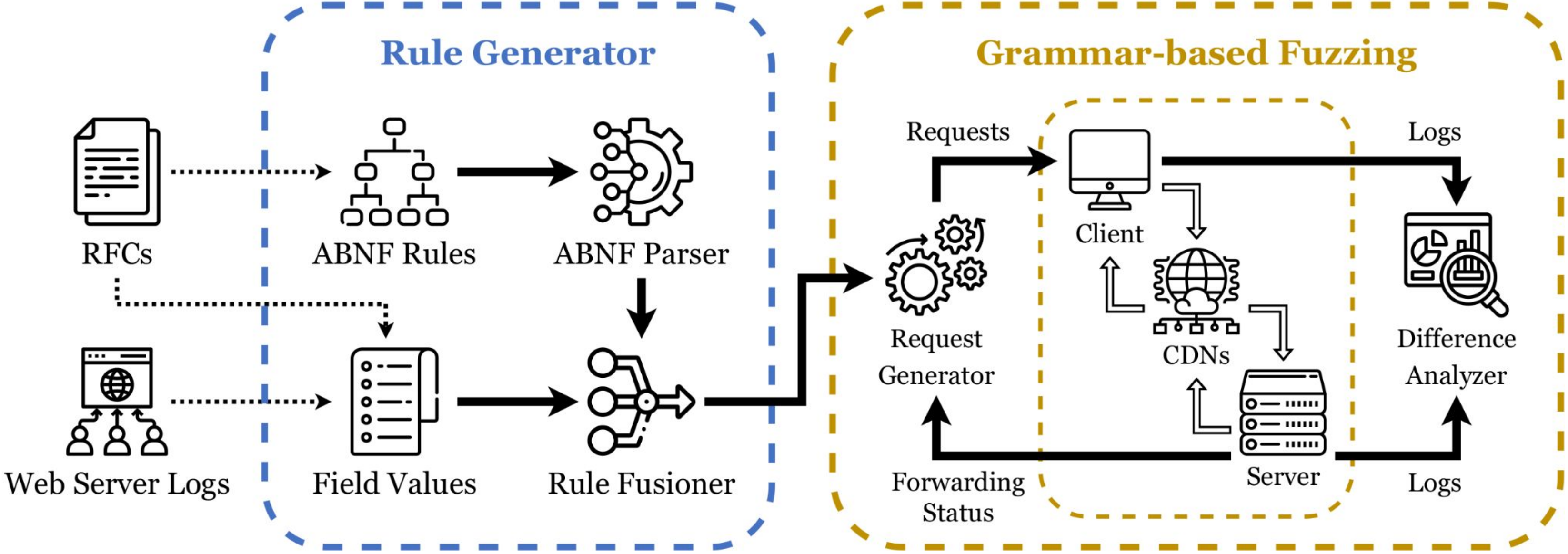
How to **systematically** and **efficiently** mining for all forwarding request inconsistencies in CDNs?

Challenges

- ❖ Techniques to evaluate HTTP implementations and CDN behaviors
  - ➢ HTTP request test case generation using ABNF rules
  - ➢ Automated testing directed towards CDNs
- ❖ But still have challenges...
  - ➢ HTTP ABNF rules are **unbounded**, test cases generated are **ineffective**
  - ➢ The cost of testing CDNs is **high**
  - ➢ CDNs, as **black-boxes**, offer **minimal** feedback concerning test requests
- ❖ These challenges impact both the **efficacy** and **efficiency** of testing

ReqsMiner: a New Detecting Framework

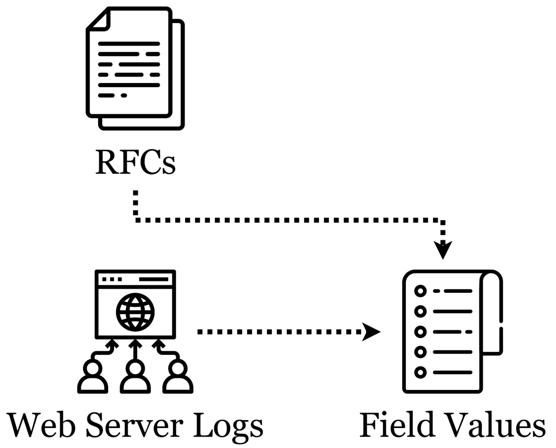
- ❖ Rule Generator
  - ➢ Combining the ABNF rules and **field values** to generate an ABNF grammar tree
- ❖ Grammar-based Fuzzing
  - ➢ Utilizing fuzzing with the **UCT-Rand** algorithm to enhance the fuzzing efficiency



https://github.com/Konano/ReqsMiner

ReqsMiner: Rule Generator

- ***Field values:*** Predefined data stored as key-value pairs
  - Extracted from **the RFCs** and **actual web server logs**
  - Merge human knowledge into the generation rules
  - Improve generation efficiency



Accept-Language:
en-US,en;q=0.9,en-GB;q=0.8,zh;q=0.7,ja

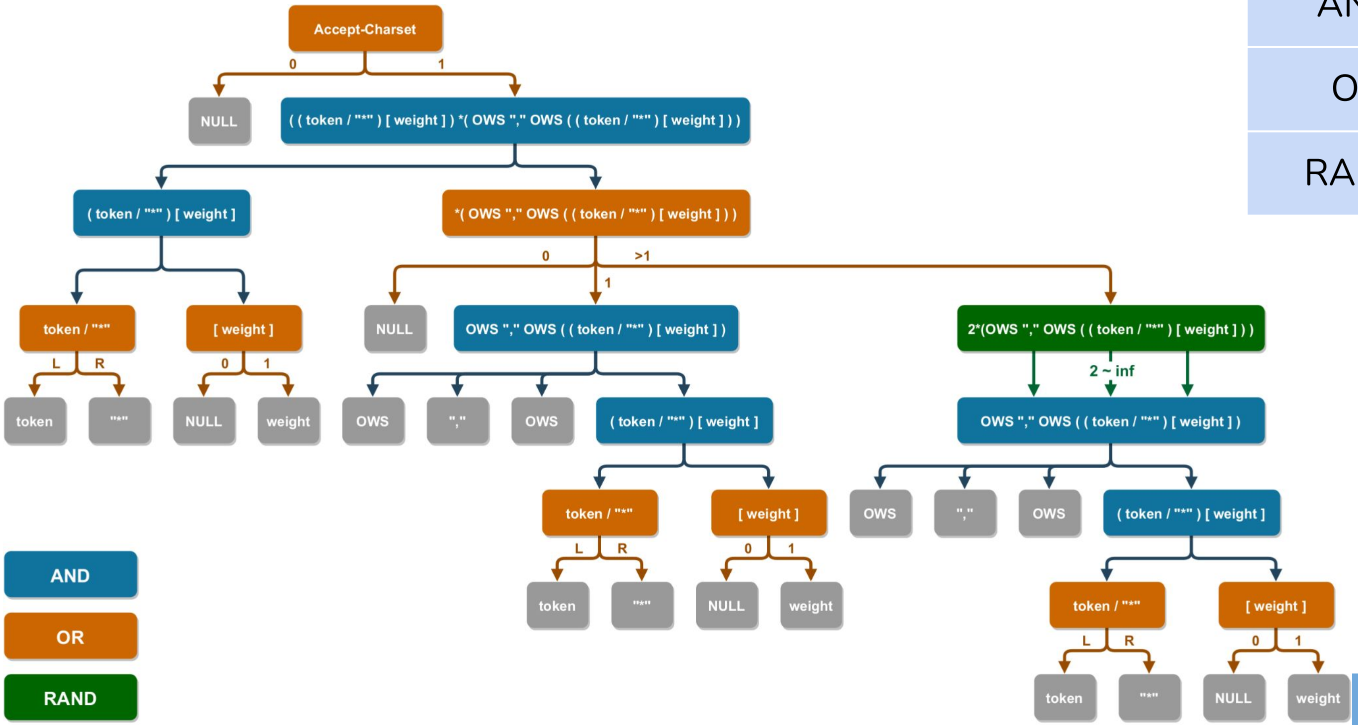
Parser Accept-Language: [language [weight] *(OWS "," OWS language [weight])]

language = en-US / en / en-GB / zh / ja
qvalue = 0.9 / 0.8 / 0.7

ReqsMiner: Rule Generator

❖ **ABNF Parser**

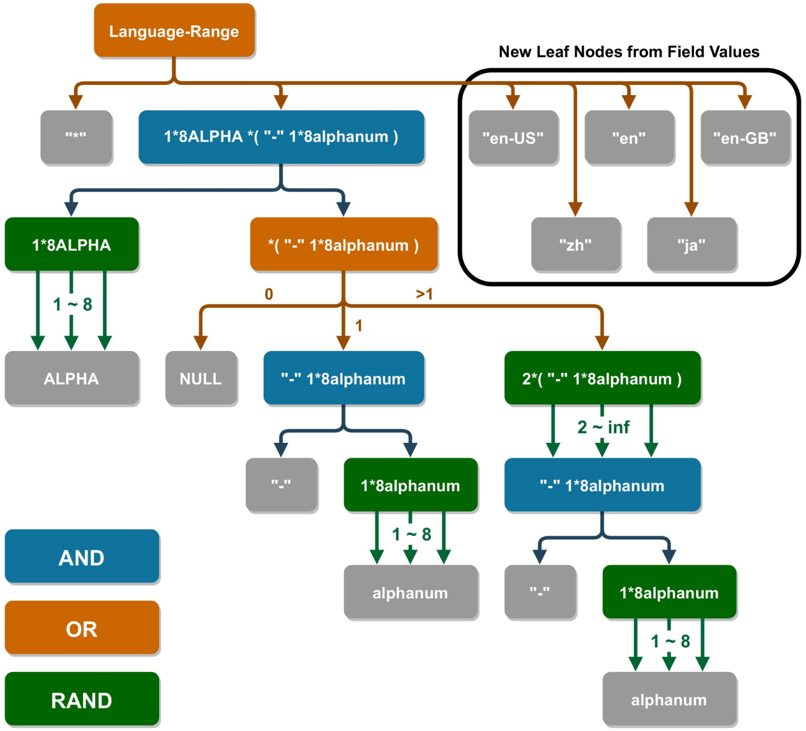
➢ Builds the **ABNF grammar tree** based on the ABNF rules



NodeType	Indication
AND	Concatenation
OR	Selection
RAND	Repetition

ReqsMiner: Rule Generator

- ## Rule Fusioner
  - Integrates field values into the **ABNF grammar tree**
  - Increase the number of subtrees of the *OR nodes* in the **ABNF grammar tree**
  - The **UCT-Rand** algorithm will have more options



Accept-Language = (1*8ALPHA *("-" 1*8alphanum)) / "*" / "en-US" / "en" / "en-GB" / "zh" / "ja"

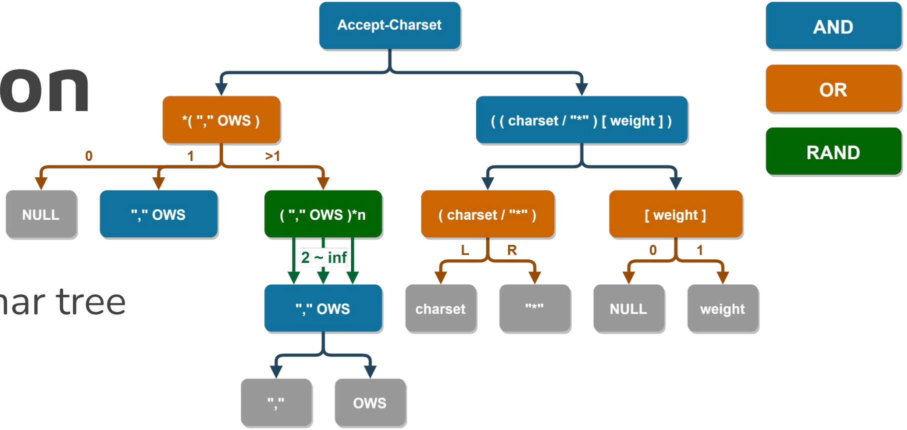
ReqsMiner: Grammar-based Fuzzing

- Challenges in Fuzzing: **Lax grammar, High costs, Black box**
- We propose a UCT-based weighted random generation algorithm **(UCT-Rand)**
- **UCT** is a variant of MCTS in game-playing AI
  - Use **the Upper Confidence Bounds (UCB)** formula to balance exploration and exploitation

$$\pi(s) := \arg\max_{a \in A(s)} \left(V_a + \sqrt{\frac{2 \ln N_s}{N_a}} \right)$$

  - **UCT-Rand** uses **weighted random selection** rather than ***the argmax function*** to choose the next child node during the selection phase
- The generation algorithm consists of 4 phases:
  - **Expansion**, **Selection**, **Simulation**, and **Backpropagation**

Expansion & Selection

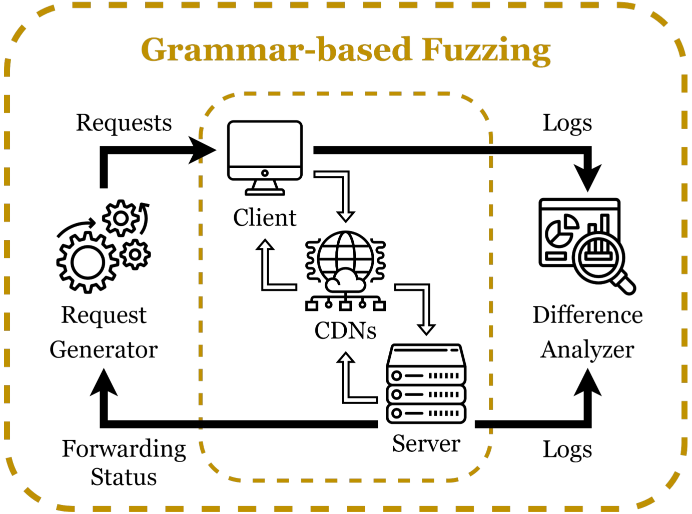


- ❖ Expansion
  - ➢ Recursively traverse the ABNF grammar tree
  - ➢ **AND:** Traverse all subtrees
  - ➢ **OR / RAND:** Go to **Selection** Phase
- ❖ Selection
  - ➢ **RAND:** Randomize the number of traversals
  - ➢ **OR:** Random unvisited sub-node is selected for traversal
    - ■ If no unvisited sub-nodes, use **the formula** to determine traversed sub-node
- ❖ Simulation
- ❖ Backpropagation

$$\pi(v) := \underset{v' \in v.\text{children}}{\text{weighted rand}} \left(Q(v, v') + \sqrt{\frac{2 \ln N(v)}{N(v, v')}} \right)$$

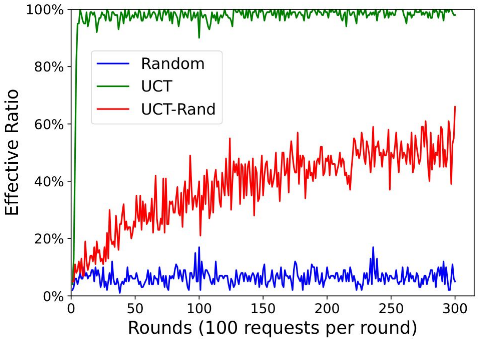
Simulation & Backpropagation

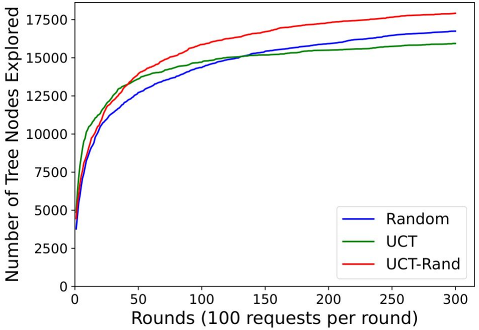
- ❖ Expansion
- ❖ Selection
- ❖ Simulation
  - ➢ Transform visited leaf nodes into HTTP requests
  - ➢ Send requests to **CDNs** via **Client**
  - ➢ Get the *forwarding status* of **CDNs** from **Server**
- ❖ Backpropagation
  - ➢ Updates the parameters of each node in the ABNF grammar tree based on the success of CDN forwarding



Evaluation of Generation Algorithms

- Metrics
  - ❌ False positive and true negative rates (Difficult vulnerability determination)
  - ✅ Effectiveness and Exploration
- Three distinct generation algorithms:
  - Random: Child nodes are randomly selected
  - UCT: Uses the argmax function to determine child nodes
  - UCT-Rand: Uses weighted random selection of child nodes



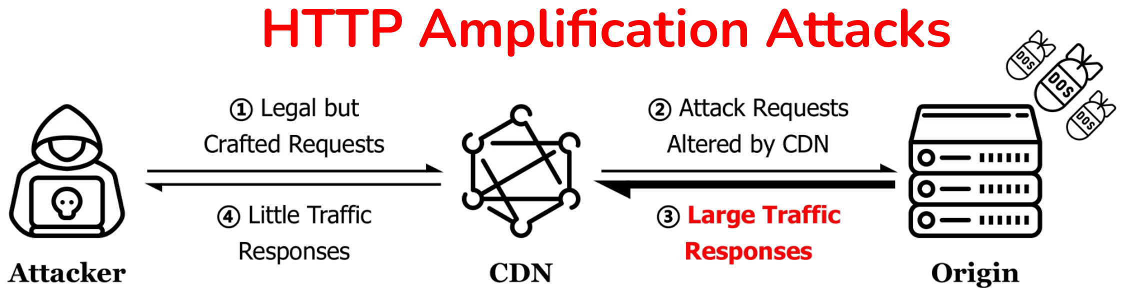


Experimental results

- Extracted 442 ABNF rules and 63 sets of field values
  - RFCs: 3986, 4647, 5234, 5646, 9110-9112
- Systematically analyzed 22 widely recognized CDN services
  - e.g. Cloudflare, Akamai, CloudFront, Fastly...
- Found numerous CDN forwarding request inconsistencies
  - Request Line
    - Request Method
    - Request URL Target
    - HTTP Version
  - Header Fields
    - due to Duplicate Headers
    - caused by Adding Headers
    - caused by Removing Headers
    - caused by Altering Headers
  - Message Body
    - caused by Removing Body
    - Transfer Encoding
- However, inconsistencies do not directly signal the existence of potential security implications

Extend: HTTP Amplification Attacks

❖ Extended and integrated into the threat model of a specified attack:



➢ Augmented the analyzer, enabling it to detect differences in **traffic size**

❖ Found 3 novel HTTP amplification attacks

➢ **HeadAmp:** HEAD Request-based HTTP Amplification Attack (max amplification: **~1.68M**)

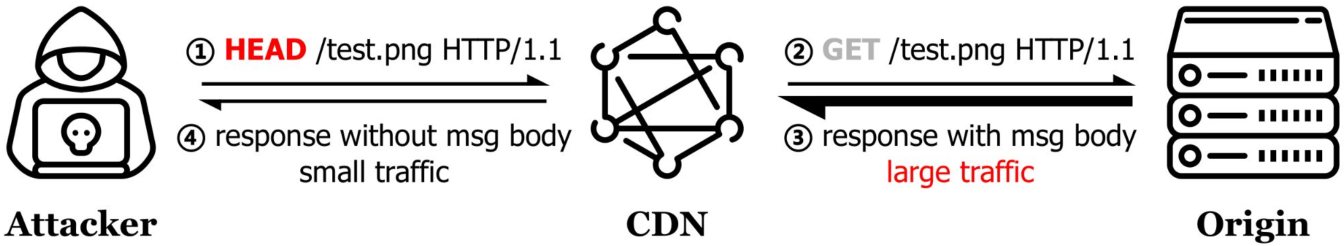
➢ **CondAmp:** Conditional Request-based HTTP Amplification Attack (max amplification: **~1.92M**)

➢ **AEAmp:** Accept-Encoding-based HTTP Amplification Attack (max amplification: **~1K**)

❖ Found **74** vuls across **19** CDN providers

Attack-1: HeadAmp Attacks

- CDN converts the request into a **GET** request when it forwards a **HEAD** request
  - *When a server receives a **HEAD request**, it should respond **with the headers** that would be returned for a GET request, but **without the actual body content**.*
- Attack conditions:
  - The attacker must successfully avoid the CDN's cache (Cache missing)
  - The target resource must be cacheable by the CDN
- Number of affected CDNs: **12**



Attack-1: HeadAmp Attacks

- The amplification factor increases with the size of the target resource
  - File Size # Amplification
  - **1MB # ~1,720**
  - **1GB # ~1,680,000**

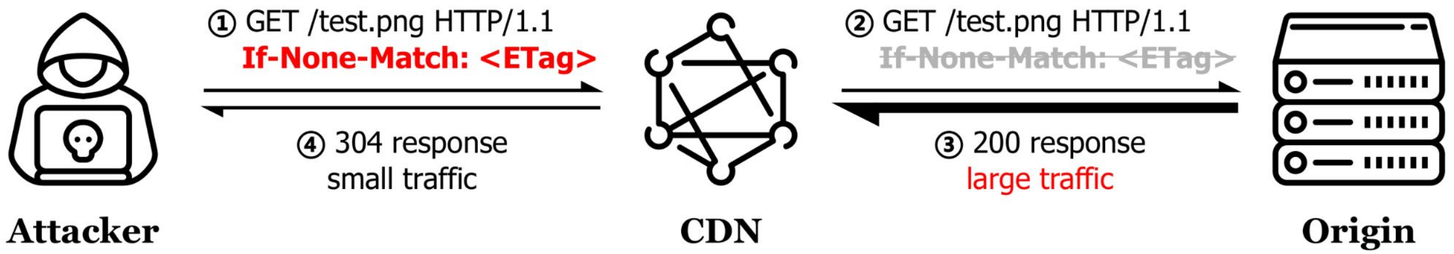
TABLE I: Amplification Factors with Different Target Resource Size of HeadAmp Attacks.

CDN	**Amplification Factor**			
	1MB	**10MB**	**25MB**	**Max (≤1GB)**
Aliyun[1]	137.52	144.05	140.49	154.20
Azure[1]	56.70	56.56	56.48	56.70
BunnyCDN	1119.00	11198.95	27575.01	1095296.82*
CDN77[1]	23.79	35.54	59.12	59.28
CDNetworks	1595.73	15599.15	39056.21	1330849.57*
ChinaNetCenter	1566.94	15667.43	38567.16	1315155.58*
Cloudflare[2]	967.15	9717.67	23827.26	483332.05
Fastly[3]	1465.48	14540.97	30.79	29243.69
Gcore	1725.39	16963.68	43094.88	1680775.18*
KeyCDN[1]	27.20	27.13	57.94	58.25
StackPath	1607.70	15853.18	40150.99	1573951.48*
Udomain[4]	1489.30	1488.06	1485.17	1491.31

* Amplification factor can be greater if the file size is larger than 1GB.
[1] Terminate the request as soon as all the headers received.
[2] Terminate the request if the file size is larger than 512MB.
[3] Refuse with "503 Service Unavailable" if the file size is larger than 20MB.
[4] First request for the first 1MB of file, then response to the client with headers.

Attack-2: CondAmp Attacks

- CDN removes the **conditional headers** when forwarding **conditional requests**
  - *When a server receives a **conditional requests**, the response should **be based on the conditions***
  - *If the conditions are **met**, the server should respond with the **requested content***
  - *If the conditions are **not met**, the server may respond with a special status code, **without content***
- There are 5 conditional headers:
  - **If-Match, If-None-Match, If-Modified-Since, If-Unmodified-Since, If-Range**
- Attack conditions:
  - The attacker must successfully avoid the CDN's cache
  - The target resource must be cacheable by the CDN
- Number of affected CDNs: **16**



Attack-2: CondAmp Attacks

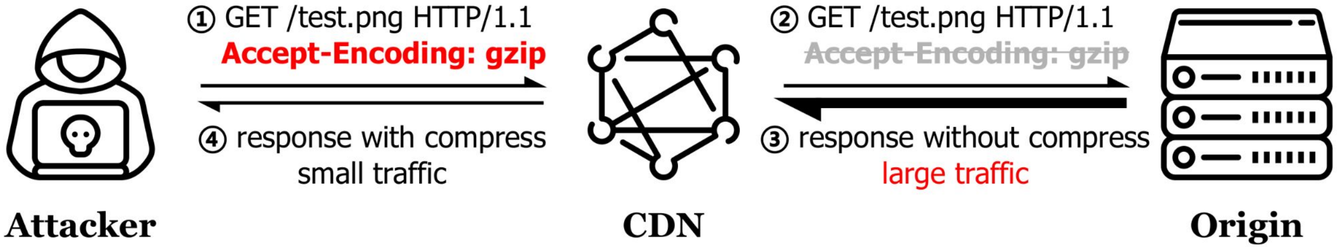
- The amplification factor increases with the size of the target resource
  - File Size # Amplification
  - **1MB # ~1,950**
  - **1GB # ~1,920,000**

(b) Attack with If-None-Match.

CDN	Amplification Factor			
	1MB	10MB	20MB	Max ($\leq$1GB)
Aliyun	1376.95	13746.82	29480.45	1143179.80*
Azure[1]	1494.58	14582.42	27178.19	27178.19
Baidu Cloud[1]	1493.35	5132.33	5147.56	7395.95
BunnyCDN	1197.92	11764.44	23553.99	1172958.12*
CDNetworks	1555.06	17321.00	30671.49	1721487.32*
CDNSun	1955.17	19475.55	39074.55	1927288.09*
ChinaNetCenter	1526.73	16045.67	30289.85	1511756.22*
Cloudflare[2]	1015.18	10154.17	20302.83	575322.41
Fastly[3]	1831.95	18274.44	32919.45	32919.45
Gcore	1917.59	18870.12	37761.45	1884424.91*
Huawei Cloud	1255.05	12579.15	24936.88	1235931.80*
Qiniu Cloud	1503.22	14855.64	29300.20	1355751.89*
Udomain[4]	1631.73	1631.83	1810.82	1810.82

Attack-3: AEAmp Attacks

- CDN adopts the **deletion policy** for handling the **Accept-Encoding** header
  - *When a server receives a request with an **"Accept-Encoding"** header, it should select an encoding from the **options available** and apply to the response body.*
- Attack conditions:
  - The attacker must successfully avoid the CDN's cache
- The amplification factor is higher for resources with greater compression rates
- Number of affected CDNs: **4**



Attack-3: AEAmp Attacks

- The amplification factor is higher for resources with greater compression rates
  - File Size # Amplification
  - **1MB # ~650**
  - **10MB # ~940**

TABLE III: Amplification Factors with Different Target Resource Size of AEAmp Attacks.

CDN	Exploited Case	Amplification Factor 1MB	10MB	25MB
Baidu Cloud	`gzip;q=1`	580.44	946.17	—
CDN77	`gzip`	571.68	929.30	963.03
CDNSun	`gzip`	650.43	972.92	984.34
Udomain	`gzip`	202.03	227.95	230.10

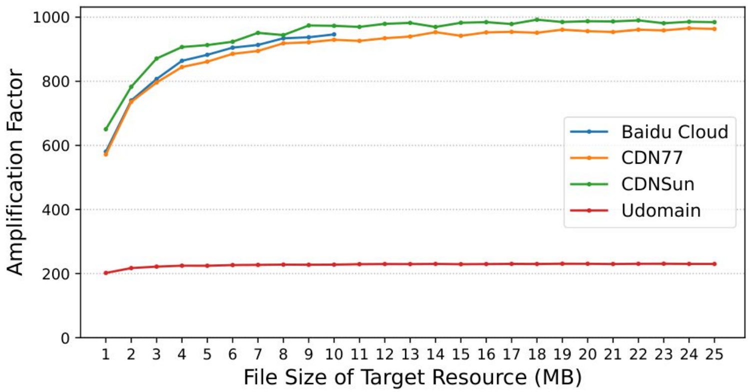


Fig. 12: Distribution of Amplification Factors for AEAmp Attacks with Different Target Resource Size and CDNs.

CDNs affected by Three HTTP Amp Attacks

TABLE IV: CDN Vendors Vulnerable to Three HTTP Amplification Attacks.

CDN	Head-Amp	CondAmp					AE-Amp
		M.[1]	N.-M.[2]	M.-S.[3]	Un.-S.[4]	R.[5]	
Akamai					✓		
Aliyun	✓		✓	✓		✓	
Azure	✓	✓	✓	✓	✓	✓	
Baidu Cloud		✓	✓	✓	✓	✓	✓
BunnyCDN	✓	✓	✓		✓	✓	
CDN77	✓						✓
CDNetworks	✓	✓	✓	✓	✓	✓	
CDNSun		✓	✓	✓		✓	✓
ChinaCache						✓	
ChinaNetCenter	✓	✓	✓	✓	✓	✓	
Cloudflare	✓	✓	✓	✓		✓	
CloudFront				✓			
Fastly	✓		✓	✓		✓	
Gcore	✓	✓	✓	✓	✓	✓	
Huawei Cloud		✓	✓	✓	✓	✓	
KeyCDN	✓						
Qiniu Cloud			✓	✓			
StackPath	✓						
Udomain	✓	✓	✓	✓	✓	✓	✓

✓: The target CDN is vulnerable. [1] `If-Match`. [2] `If-None-Match`. [3] `If-Modified-Since`. [4] `If-Unmodified-Since`. [5] `If-Range`.

Found

74

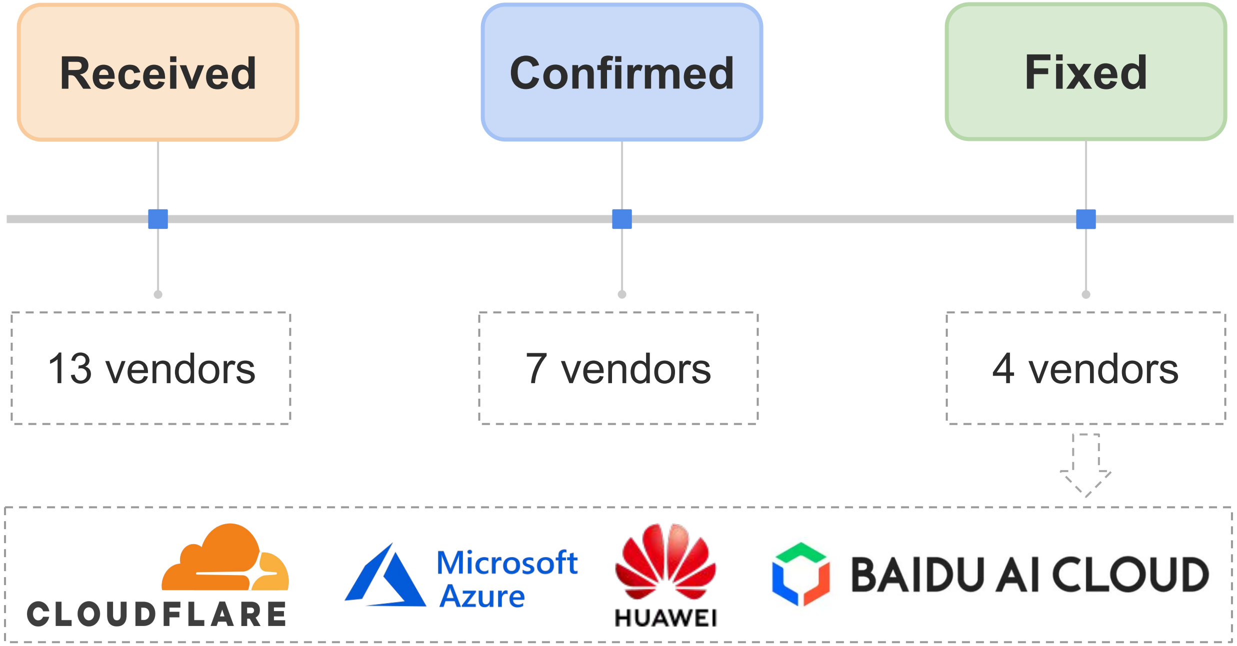
vulnerabilities across

19

CDN providers

Responsible Disclosure

- Response from affected CDN vendors.



Conclusion

- New Detecting Framework: ReqsMiner
  - For the efficient discovery of CDN forwarding request inconsistencies
  - Developed a novel UCT-based grammar-based fuzzer
- New Findings:
  - Discovered **3** novel high-impact HTTP traffic amplification attacks
  - Amplification factor can reach up to **2,000** generally, and even **1,920,000** under specific conditions.
  - Found **74** vulnerabilities on **19** popular CDN providers

Thank you for listening!

Q & A

Linkai Zheng
nanoapezlk@gmail.com
Network and Information Security Lab (NISL)
Tsinghua University