2023年全球DDoS攻击现状与趋势分析

天翼安全科技有限公司、联通数科安全、百度安全、Nexusguard、
中国移动云能力中心、中国移动卓望公司、清华大学、华为联合发布

百度安全官网

目录

01 关键信息摘要

1.1 专家观点

观点1：瞬时泛洪攻击秒级加速，挑战防御系统的响应速度。需探索更为高效的检测和清洗技术。例如设备厂商研制高效随路检测路由器，运营商研发端网协同防御技术，以有效缩短TTM（Time to Mitigation）。

观点2：高速加密攻击挑战解密防御性能，低速CC攻击绕过WAF，挑战防御系统有效性。利用行为分析算法拦截高速CC，机器学习算法精准识别低速CC，分而治之，有效应对复杂攻击。

观点3：扫段攻击成为网络基础设施面临的最大威胁，需采取多种措施增强防御。增加网段检测能力提升攻击识别精准度，端网协同防御提高多网段攻击的发现及处置效率，有效应对大规模扫段攻击。

1.2 DDoS攻击态势

超大规模攻击异常活跃。

2023年T级攻击异常活跃。超800Gbps攻击共计248次，和2022年基本持平。1月份中国电信安全团队监测到年度最大包速率攻击，峰值包速率高达972Mpps；10月份中国电信安全团队监测到年度最大带宽攻击，峰值带宽高达2.505Tbps。2023年8月，发生互联网史上最大应用层攻击，攻击峰值高达398Mrps[1]。2023年10月华为监测到最大规模的扫段攻击，230个C段先后遭受混合攻击。

攻击频次继续呈增长趋势。

2023年攻击频次是2022年的1.6倍，是2021年的1.8倍。从全球看，针对APAC的攻击最活跃，占比88.83%。

大流量攻击持续呈秒级加速态势，爬升速度再创新高，挑战防御系统响应速度。

大流量攻击持续加速，爬升至400Gbps-500Gbps区间只需2秒；爬升至800Gbps-1Tbps区间，仅需10秒。

攻击复杂度持续提升，攻击威胁加剧。

HTTP/HTTPS应用层攻击两级分化，高速攻击挑战WAF解密防御性能，低速攻击bypass WAF，高速、低速攻击混合，挑战防御系统有效性。一方面，HTTP2.0 Rapid Reset漏洞被利用，导致应用层攻击速率从2022年的千万级RPS跃迁至2023年的亿级RPS，挑战解密防御性能；另一方面，由数十万个僵尸发起的低速应用层攻击日渐常态化，绕过WAF检测，威胁加剧。

TOA漏洞被利用，引发TCP四层代理场景的互联网业务信任风险。

为减少自身网络攻击威胁，某些无良互联网企业通过修改DNS记录，将攻击流量“零成本”转移至百度。

传媒和互联网、政府和公共事业、教育、金融依然是TOP4攻击目标行业。
传媒和互联网、政府和公共事业、教育、金融攻击频次占比依次为59.88%、11.75%、2.98%、2.85%。能源行业和工业互联网受攻击频次占比连续三年增长。近三年，中国金融行业攻击频次呈持续增长趋势。

全球攻击目标地域分布排序依次为APAC、LATAM、EMEA、AMER，中国TOP3攻击目标地域分布为吉林、山东和广东。
攻击目标按大洲分布为APAC、LATAM、EMEA、AMER，占比依次为83.81%、12.51%、2.02%、1.66%；中国TOP3攻击目标地域分布为吉林、山东和广东，占比依次为32.80%、9.41%、8.68%。

1.3 DDoS僵尸网络态势

僵尸家族分布：DDoS僵尸家族以IoT和Linux为主，按活跃C2数量排名的TOP5僵尸家族分别是Mirai、Gafgyt、Mozi、XorDDoS和Dofloo，占比依次为61.73%、34.24%、2.15%、1.29%、0.58%。

僵尸网络C2地域分布：C2海外按大洲地域分布分别是EMEA、AMER、APAC、LATAM，占比依次为38.88%、34.38%、25.16%、1.59%；C2中国TOP3地域分布为广东、河南和香港，占比依次为25.81%、21.46%、15.83%。

1.4 DDoS攻击源态势

肉鸡地域分布：肉鸡海外地域分布，按AMER、EMEA、APAC、LATAM统计，占比依次为33.34%、32.17%、30.42%、4.07%；肉鸡中国TOP3地域分布为河南、浙江和广东，占比依次为14.03%、12.67%、9.88%。

肉鸡运营商/服务提供商分布：中国TOP3肉鸡运营商/服务提供商分布依次是电信、联通和阿里云，占比依次是53.18%、33.19%、6.24%；海外TOP3 DDoS肉鸡运营商/服务提供商分布依次是Amazon、KE和Google，占比依次是59.56%、10.88%、5.34%。

1.5 典型攻击分析

2023年扫段攻击频次快速增长，为躲避防御，低速扫段攻击成为主流，惯用“短平快”战术，多采用混合攻击手法。

2023下半年，扫段攻击频次激增。

73.19%的扫段攻击采用低速扫段，挑战防御系统检测算法灵敏度。

扫段攻击惯用“短平快”战术，挑战防御系统的响应速度。43.26%的C段攻击持续时间小于5分钟；当单个C段攻击持续时间小于5分钟时，93.90%的单个目标IP的攻击持续时间不超过10秒。当采用长周期的高速扫段时，攻击者会采用典型的“脉冲”攻击手法。

攻击者通过大规模扫段，挑战并发主机防御规格。86.96%的扫段攻击采用混合攻击手法，提升防御难度。

2023年针对DNS服务器的攻击无论是攻击复杂度还是攻击强度均创新高。

DNS攻击峰值QPS从百万次级别快速提升至亿次级别。11月份，百度监测到DNS攻击流量峰值速率高达572.84Mqps。

DNS攻击手法多样化。透过递归服务器攻击授权服务器，传统防御算法失效；某次针对递归服务器的NXDomain攻击，攻击报文的源IP和目的IP位于同一网段，消耗递归服务器性能的同时，产生大量ARP广播报文。

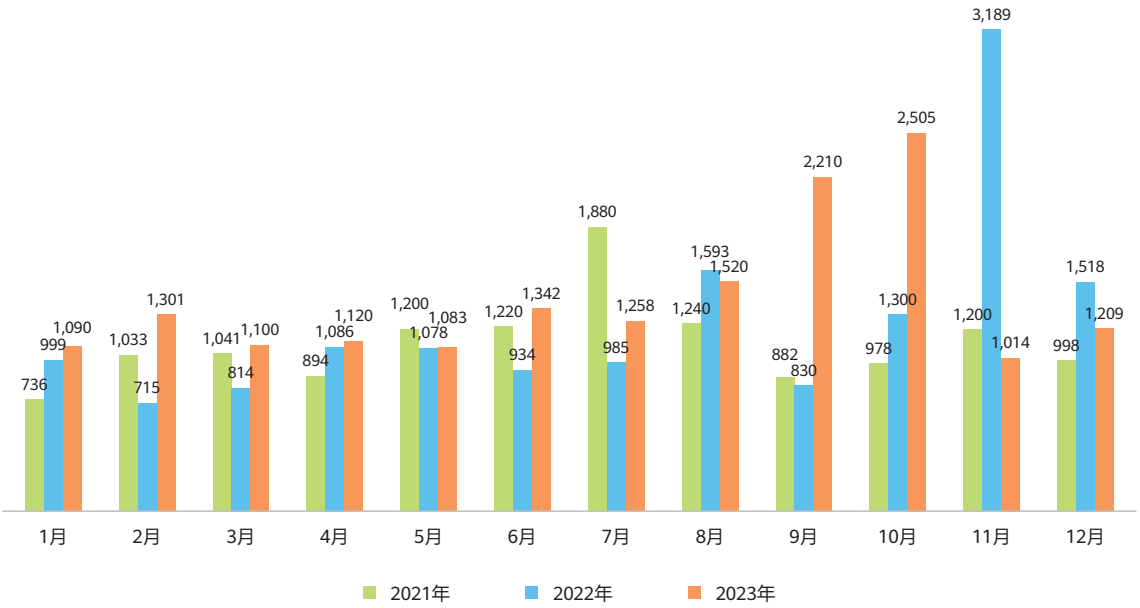
02 现状与趋势

2.1 DDoS攻击态势

2.1.1 攻击强度

2023年T级攻击主要聚集在Q1和Q4。2023年10月30日21:15:45，电信安全团队监测到年度最大带宽攻击。攻击者采用以SSDP反射为主、叠加UDP Flood和ICMP Flood的混合攻击，攻击共持续2小时31分钟，峰值带宽2.505Tbps，攻击目标IP位于江苏电信网络。

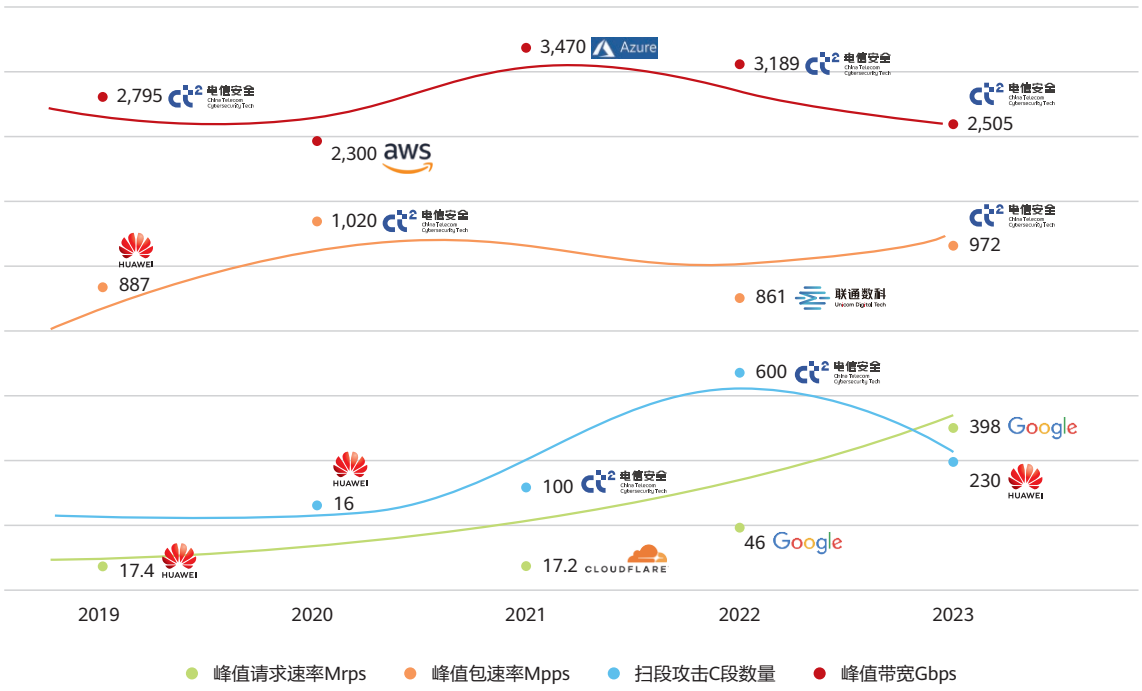
2021-2023年攻击峰值带宽月度分布（Gbps）



数据来源于电信安全&联通数科&中移云能&中移卓望&Nexusguard&华为

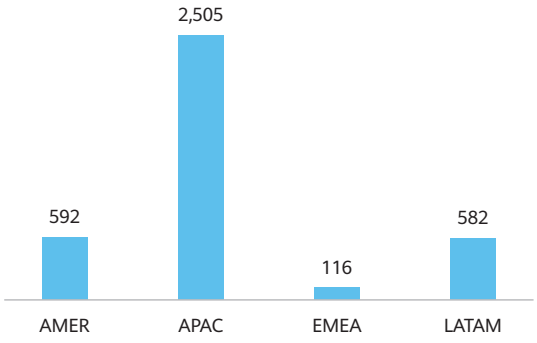
攻击成本持续降低，导致攻击规模持续增长。从近年全球记录的年度最大攻击来看，单次攻击的峰值流量带宽稳定在2.5Tbps-3.5Tbps区间[2,3]，峰值包速率则稳定在900Mpps-1000Mpps区间。单次扫段攻击峰值规模维持在200-600个C段。应用层攻击的峰值RPS则依然呈现迅速增长趋势。

全球最大DDoS攻击趋势



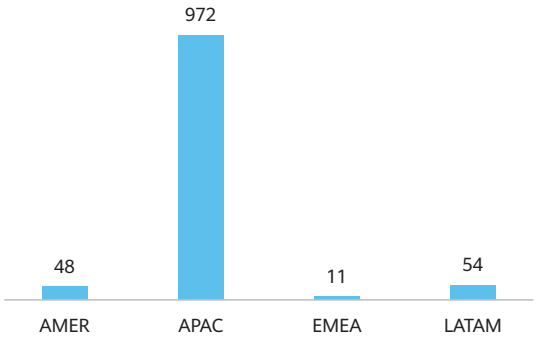
2023年8月份以前，高速应用层攻击主要利用HTTP2.0 Multiplexing特性发起，攻击峰值请求速率维持在千万级RPS[4,5]。2023年8月，HTTP2.0 Rapid Reset漏洞被发掘，应用层攻击峰值RPS直接飙升至亿次级RPS[2]（同月，Google云遭受的加密攻击峰值请求速率高达398Mrps，成为互联网史上最大应用层攻击）。超大规模加密攻击对防御成本构成严峻挑战。

2023年攻击峰值带宽地域分布（Gbps）



数据来源于电信安全&Nexusguard&华为

2023年攻击峰值包速率地域分布（Mpps）

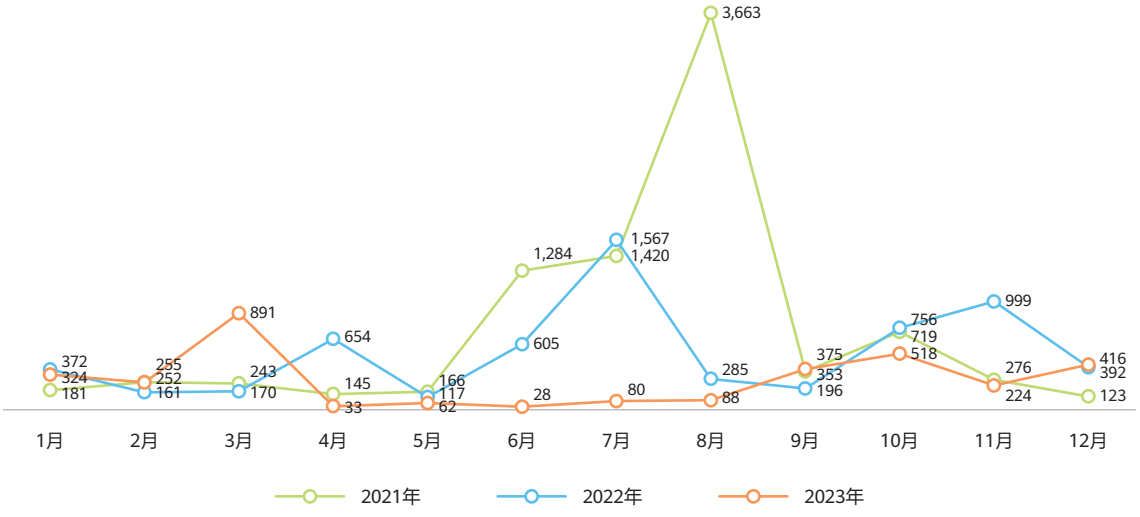


数据来源于电信安全&Nexusguard&华为

从2023年年度攻击峰值带宽和包速率地域分布来看，APAC攻击强度远超其他大洲。

2023年超500Gbps攻击共发生3291次，Q1超500Gbps攻击最活跃，共发生1467次，占全年45%。

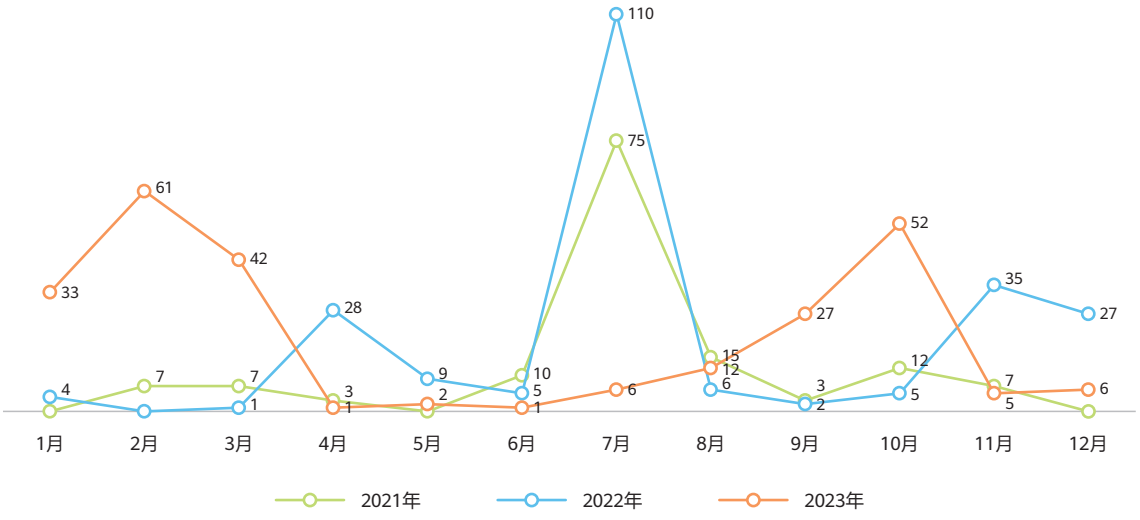
2021-2023年超500Gbps攻击频次月度分布



数据来源于电信安全&联通数科&中移云能&中移卓望&Nexusguard&华为

超800Gbps攻击全年共计发生248次，略高于2022年的232次。同样，Q1超800Gbps攻击最活跃，共计136次，占全年54%。

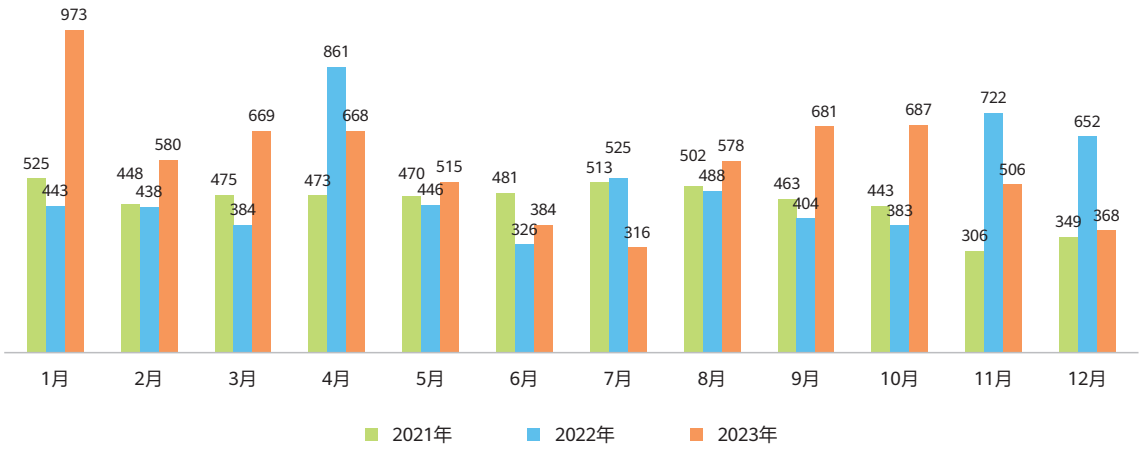
2021-2023年超800Gbps攻击频次月度分布



数据来源于电信安全&联通数科&Nexusguard&华为

2023年1月27日02:41:45，中国电信安全团队监测到年度最大包速率攻击，采用小报文UDP Flood，攻击共持续2小时44分钟，峰值包速率972Mpps，攻击目标IP位于四川电信网络。

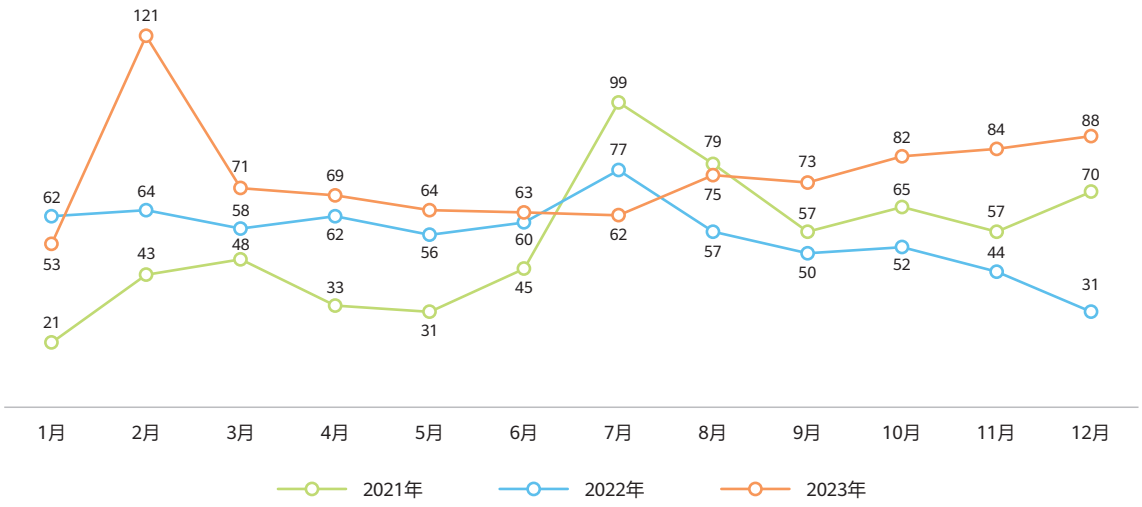
2021-2023年攻击峰值包速率月度分布（Mpps）



数据来源于电信安全&联通数科&中移云能&中移卓望&Nexusguard&华为

2023年月度平均峰值带宽最大值为121Gbps，首次突破100Gbps。对比历年年度攻击平均峰值带宽，2023年为75Gbps，2022年为56Gbps，2021年为54Gbps，说明攻击流量强度逐年提升。

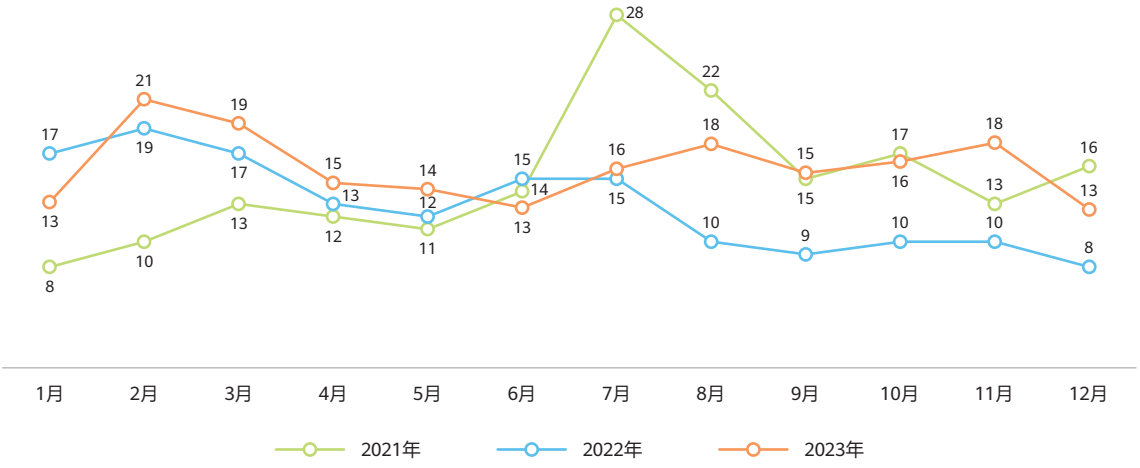
2021-2023年平均攻击峰值带宽月度分布（Gbps）



数据来源于电信安全&联通数科&Nexusguard&中移云能&中移卓望&华为

2023年月度平均攻击包速率最大值为21Mpps，出现在2月份。对比历年年度攻击平均峰值包速率，2023年为16Mpps，高于2022年的13Mpps和2021年的15Mpps。

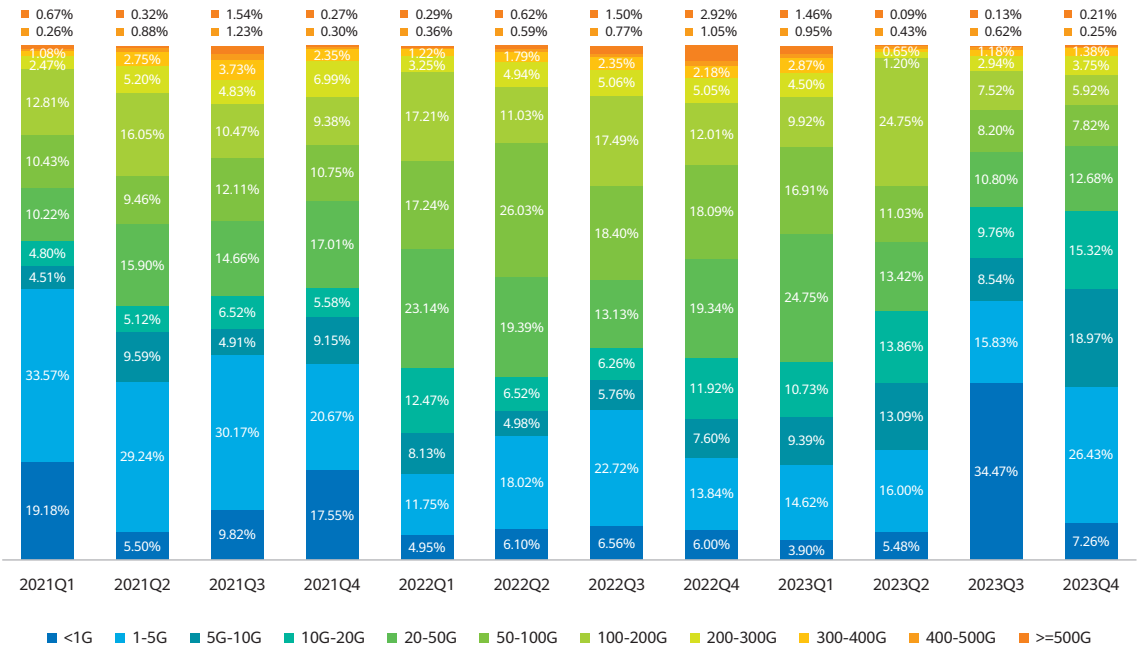
2021-2023年平均攻击峰值包速率月度分布（Mpps）



数据来源于电信安全&联通数科&Nexusguard&中移云能&中移卓望&华为

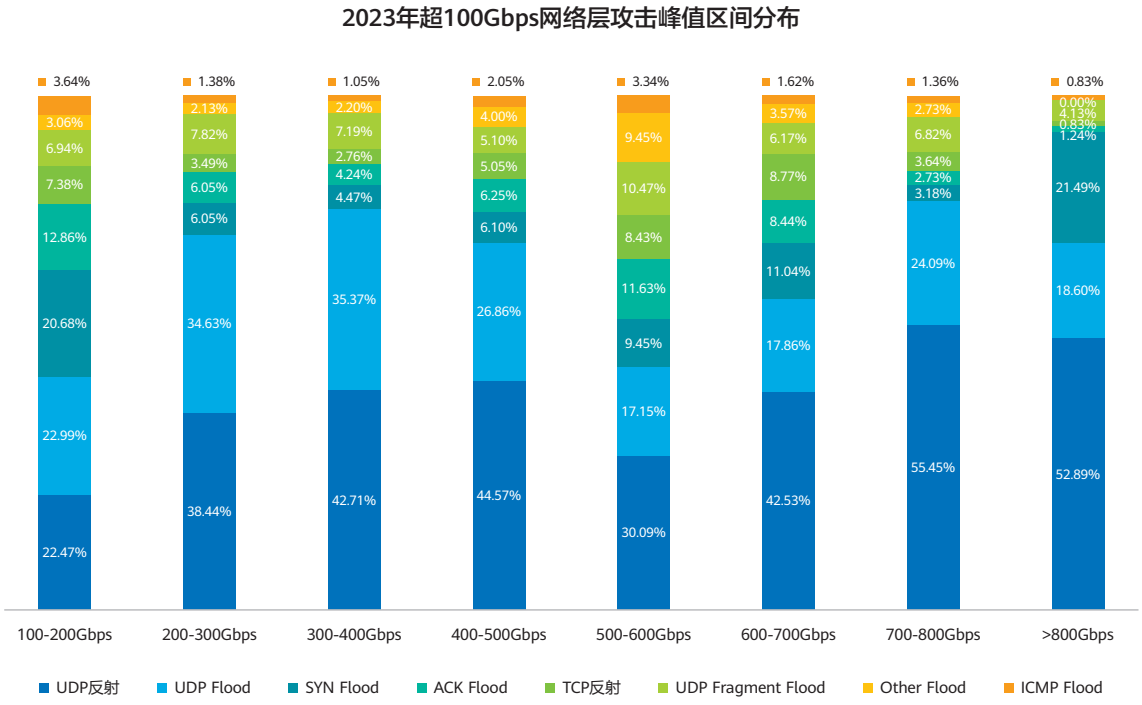
2023年的第二、第三和第四季度相比往年，攻击流量峰值区间分布发生较大变化。其中第二季度100-200Gbps攻击异常活跃，占比高达24.75%，主要原因是该季度针对UDP游戏的UDP Flood攻击活跃；第三季度，<1Gbps攻击活跃，占比34.47%，是因为该季度中国境内低速扫段攻击异常活跃；第四季度，1-5Gbps攻击占比提升至2021年水平，主要源于低速应用层攻击快速增长。

2021-2023年攻击流量峰值带宽区间分布



数据来源于华为

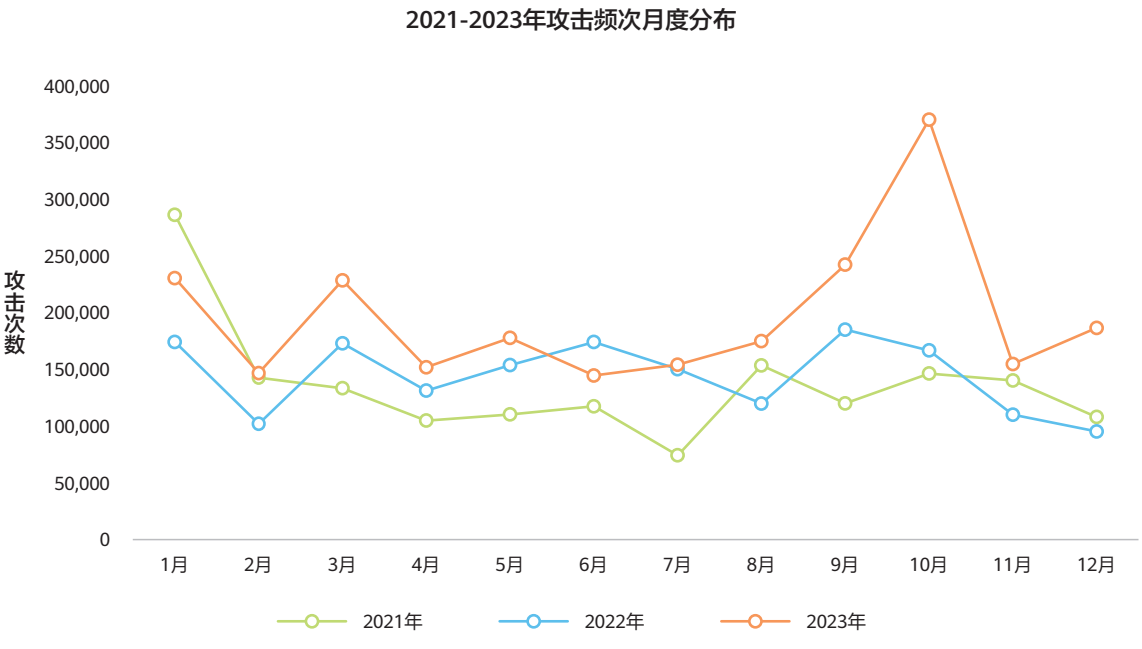
对超100Gbps攻击进行统计分析发现，UDP反射和UDP Flood是攻击者发起大流量攻击的主要手段。



数据来源于华为

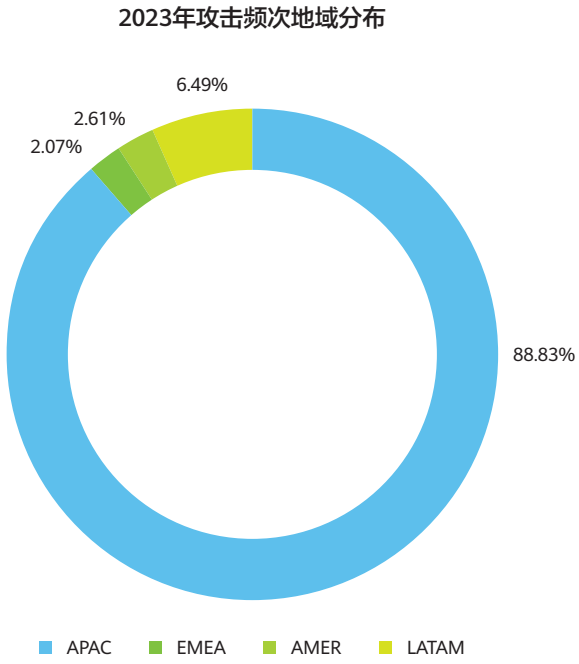
2.1.2 攻击频次

DDoS攻击频次呈持续增长趋势。2023年攻击频次是2022年的1.6倍，2021年的1.8倍。



数据来源于华为

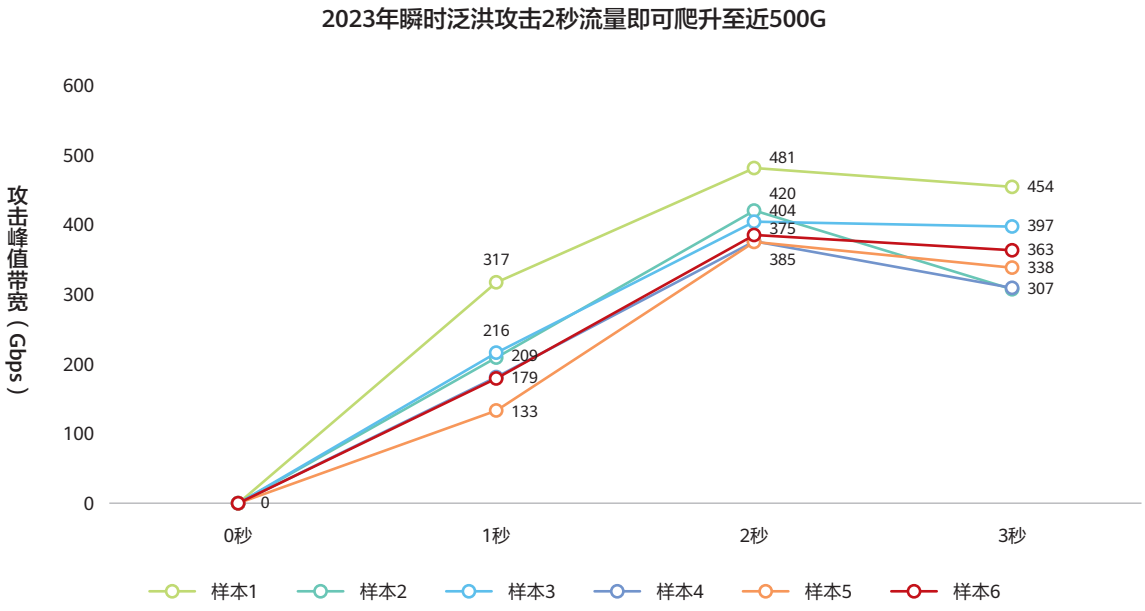
攻击频次按地域分布，APAC攻击活跃，占比88.83%。APAC扫段攻击异常活跃，拉升了APAC攻击频次占比。



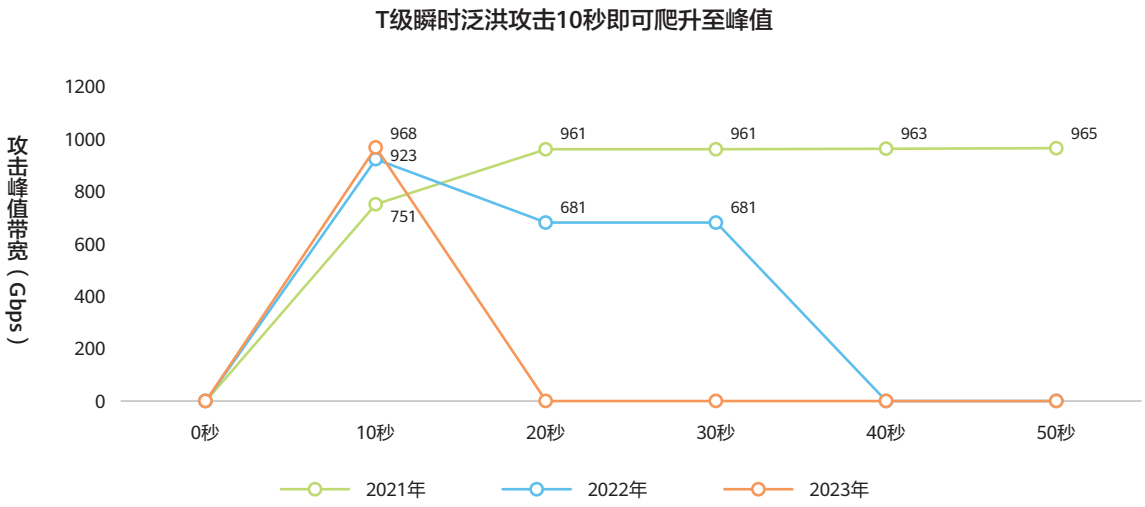
数据来源于Nexusguard

2.1.3 攻击速度

大流量攻击持续秒级加速态势，爬升速度2023年再创新高。瞬时泛洪攻击2秒流量即可爬升至近500Gbps，10秒即可爬升至900Gbps-1Tbps区间，挑战防御系统响应速度。



数据来源于华为



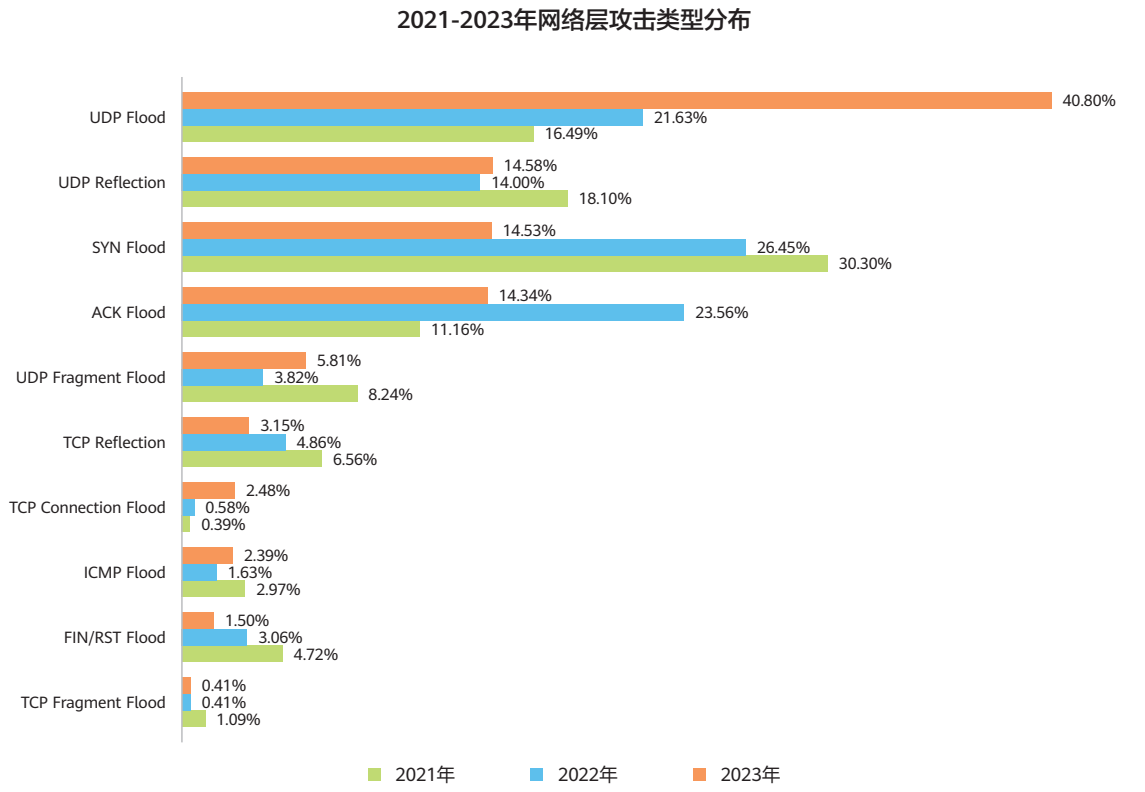
数据来源于华为

2.1.4 攻击复杂度

1. 网络攻击类型分布

近三年，UDP Flood、UDP反射、SYN Flood、ACK Flood、UDP分片均维持TOP5网络层攻击类型。

2023年UDP Flood频次明显快速增长，占比提升至40.80%。UDP Flood频次提升原因主要有两个，一方面针对TCP业务采用低成本的UDP Flood挤占带宽一直是攻击者优选；另外一方面，针对UDP游戏的UDP Flood难防御，因此备受攻击者青睐。

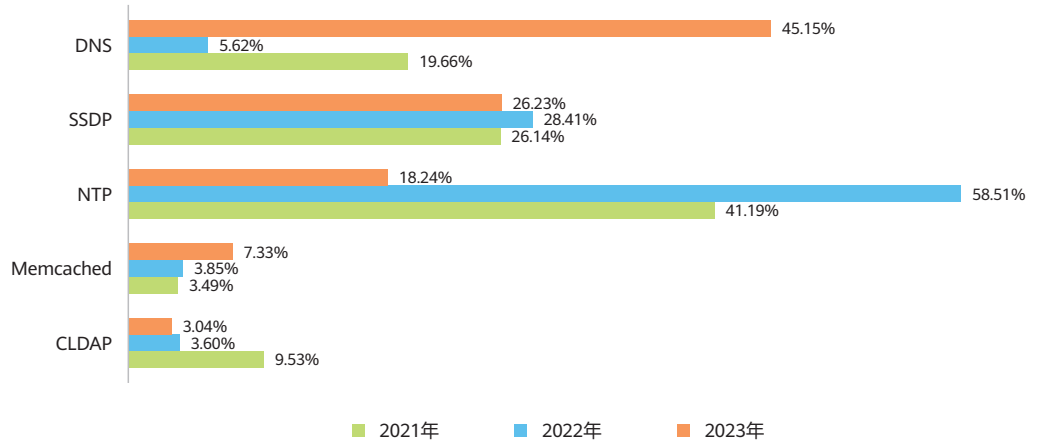


数据来源于华为

近三年ACK Flood占比持续保持高位的原因是针对TCP游戏的网络层CC攻击效果明显，防御困难，网络层CC一直被作为攻击TCP游戏服务器的杀手锏。

TOP5 UDP反射中，2023年DNS反射呈快速增长态势，占比从2022年的5.62%提升至45.15%；SSDP反射占比连续三年恒定；NTP反射占比处于减少态势，从2022年的58.51%减少至18.24%。

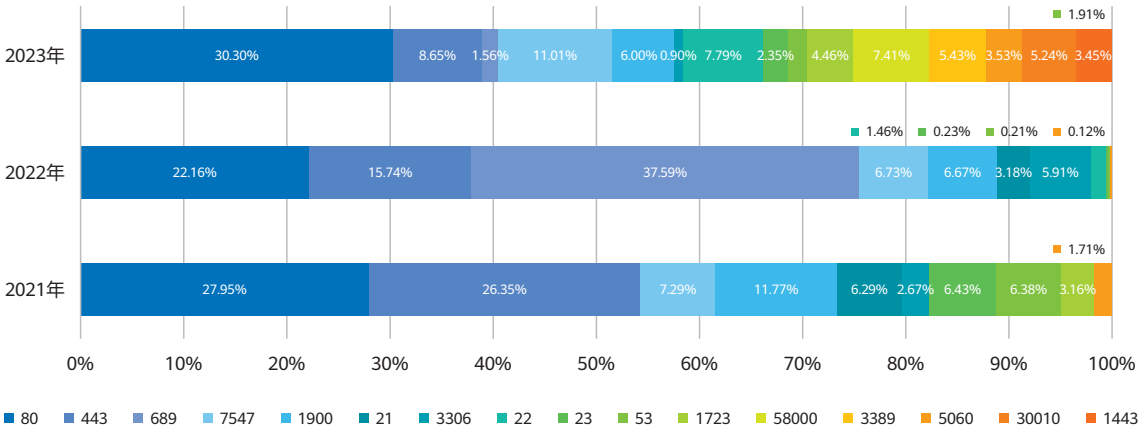
2021-2023年TOP5 UDP反射类型分布



数据来源于电信安全&联通数科&中移云能&中移卓望&Nexusguard&华为

2023年，TOP10 TCP反射端口新增58000和30010。其中58000是某品牌光猫的配置端口，30010是vsftp服务端口。利用58000和30010端口的反射攻击2022年就已经出现，但整体占比较小，2023年开始活跃。电信安全监测到的2023年11月份针对ChatGPT的DDoS攻击事件，以TCP反射为主，TOP4反射源端口就包括30010和58000。

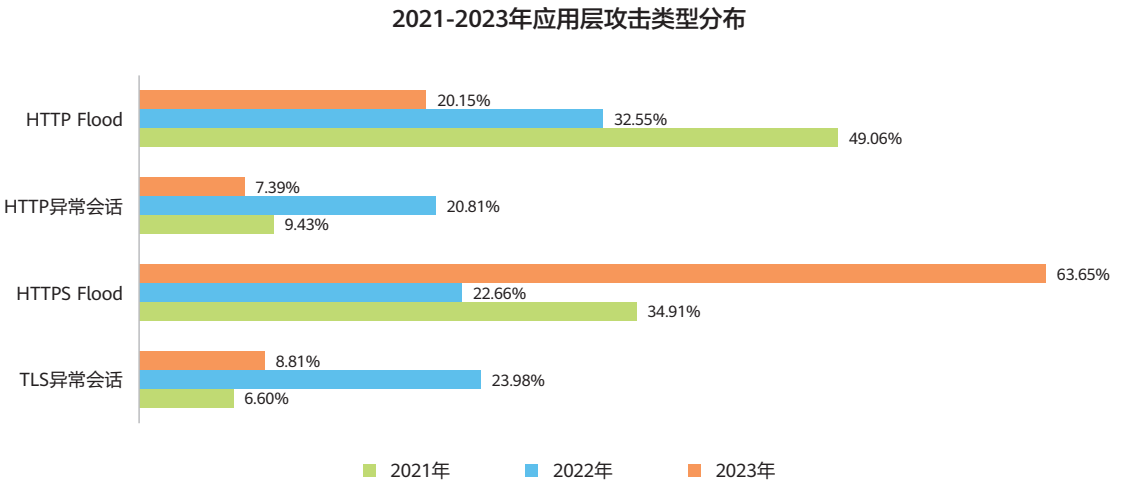
2021-2023年典型TCP反射攻击源端口分布



数据来源于华为

2. 应用层攻击类型分布

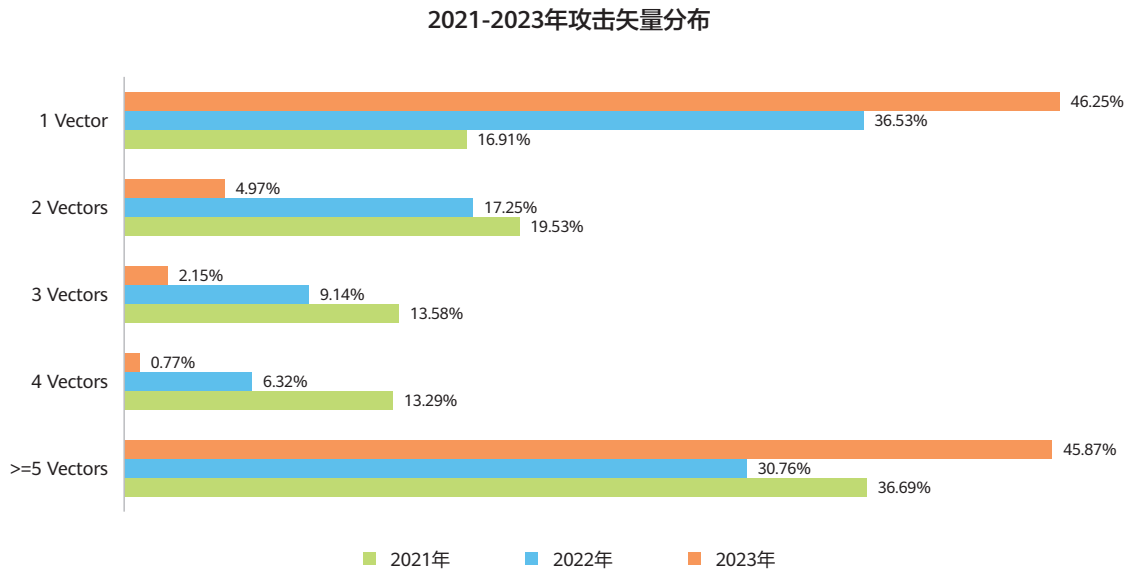
2023年，加密攻击占比快速提升至63.65%，防御难度大幅度增加。



数据来源于华为

3. 攻击矢量分布

2023年混合攻击占比较2022年有所降低，占比53.75%，但仍然是主流。单一攻击占比提升的主要原因是2023年扫段攻击频发，导致大量IP无辜“躺枪”。

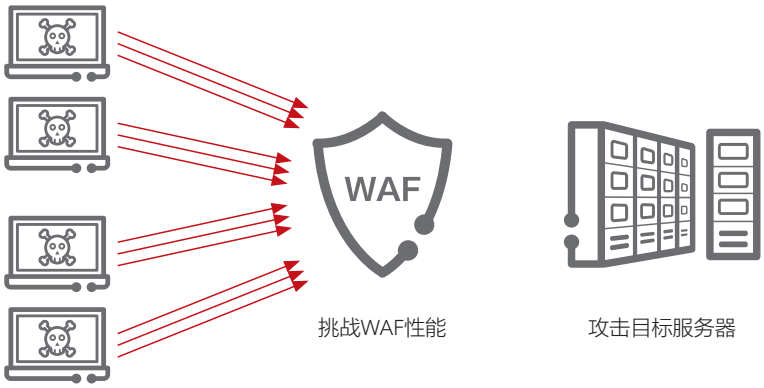


数据来源于联通数科&Nexusguard&华为

4. HTTP/HTTPS应用层攻击两级分化

HTTP/HTTPS应用层攻击两级分化，高速攻击挑战WAF解密防御性能，低速攻击bypass WAF，高速、低速攻击混合，挑战防御成功率。

高速HTTP/HTTPS应用层攻击

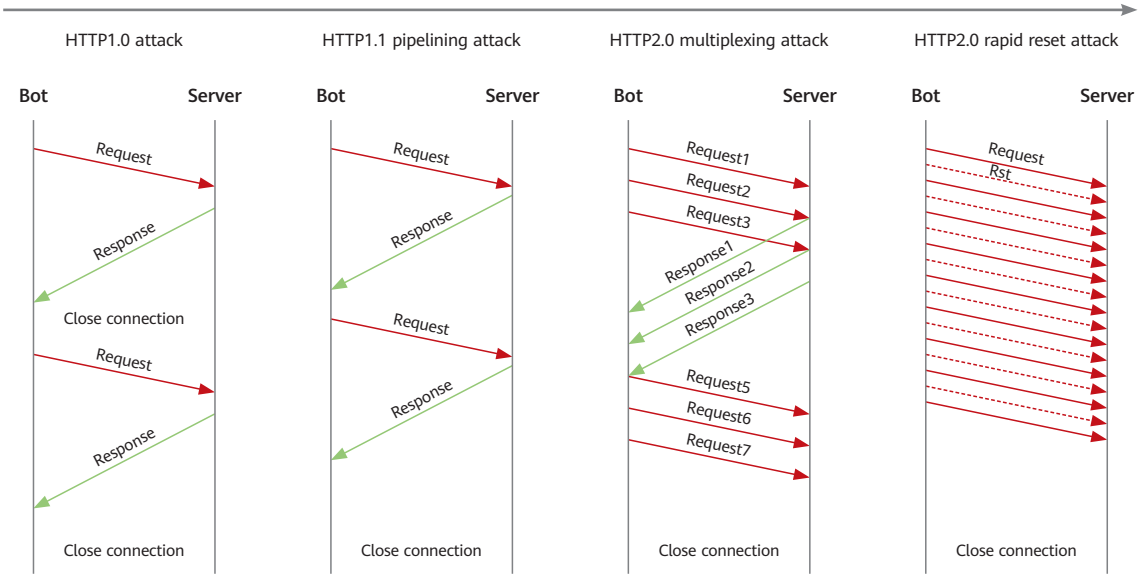


高速HTTP/HTTPS应用层攻击强度一般在千万级甚至亿次级RPS，一次攻击事件使用的肉鸡数量在5,000-30,000之间，肉鸡多为高性能的服务器或云主机，单个肉鸡的攻击速率约2,000-10,000rps，攻击多采用HTTP1.1 pipelining、HTTP2.0 multiplexing甚至HTTP2.0 Rapid Reset手法。

2023年2月，Cloudflare缓解的攻击峰值速率为71Mrps的加密攻击就属于典型的高速应用层攻击。攻击采用HTTP2.0 multiplexing攻击手法，共30,000个肉鸡参与攻击，平均每个肉鸡请求速率是2,367rps[4]。

HTTP协议从1.0开始先后演进至1.1和2.0，HTTP传输速度不断提升，应用层攻击速率亦得以快速攀升[6]。

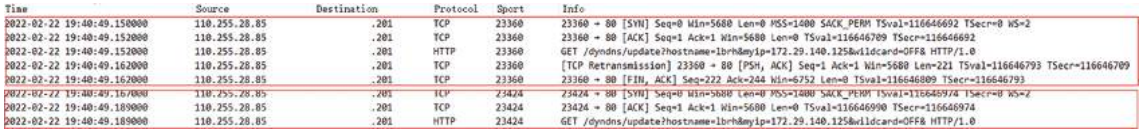
跟随HTTP协议演进，应用层攻击速率持续攀升



借助HTTP1.1 pipelining特性，2022年8月，加密攻击峰值速率达到46Mrps[5]，成为互联网史上最大应用层攻击。随着HTTP2.0普及，攻击再次提速。2023年2月，HTTP2.0 multiplexing被利用，互联网史上最大应用层攻击记录被刷新至71Mrps[4]。2023年8月，HTTP2.0 Rapid Reset漏洞被利用，互联网史上最大应用层攻击记录再次被刷新至惊人的398Mrps[1]。

» **HTTP1.0攻击分析**

HTTP1.0攻击抓包

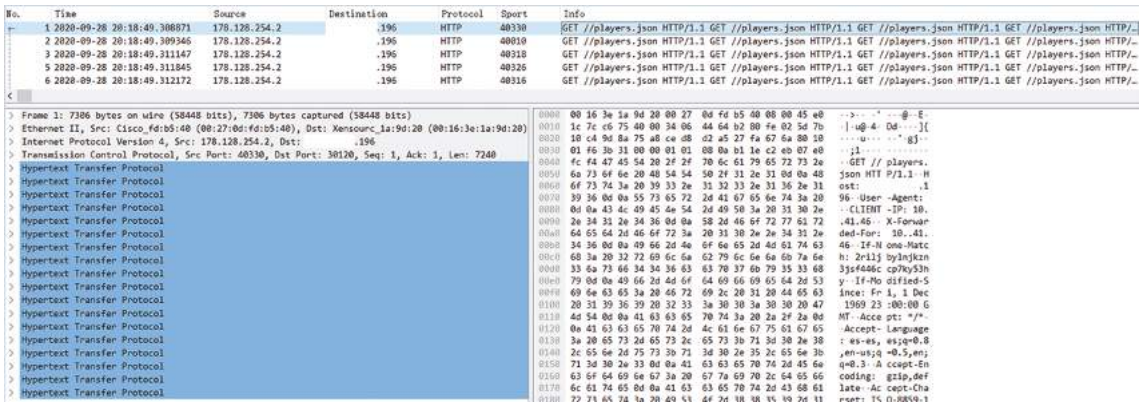


Time	Source	Destination	Protocol	Sport	Info
2022-02-22 19:40:49.150000	110.255.28.85	.201	TCP	23360	23360 → 80 [SYN] Seq=0 Win=5680 Len=0 MSS=1400 SACK_PERM TSval=116646692 TSecr=0 WS=2
2022-02-22 19:40:49.152000	110.255.28.85	.201	TCP	23360	23360 → 80 [ACK] Seq=1 Ack=1 Win=5680 Len=0 TSval=116646709 TSecr=116646692
2022-02-22 19:40:49.152000	110.255.28.85	.201	HTTP	23360	GET /dyndns/update?hostname=lbrh&myip=172.29.140.125&wildcard=OFF& HTTP/1.0
2022-02-22 19:40:49.162000	110.255.28.85	.201	TCP	23360	[TCP Retransmission] 23360 → 80 [PSH, ACK] Seq=1 Ack=1 Win=5680 Len=221 TSval=116646793 TSecr=116646709
2022-02-22 19:40:49.162000	110.255.28.85	.201	TCP	23360	23360 → 80 [FIN, ACK] Seq=222 Ack=244 Win=6752 Len=0 TSval=116646809 TSecr=116646793
2022-02-22 19:40:49.167000	110.255.28.85	.201	TCP	23424	23424 → 80 [SYN] Seq=0 Win=5680 Len=0 MSS=1400 SACK_PERM TSval=116646974 TSecr=0 WS=2
2022-02-22 19:40:49.189000	110.255.28.85	.201	TCP	23424	23424 → 80 [ACK] Seq=1 Ack=1 Win=5680 Len=0 TSval=116646990 TSecr=116646974
2022-02-22 19:40:49.189000	110.255.28.85	.201	HTTP	23424	GET /dyndns/update?hostname=lbrh&myip=172.29.140.125&wildcard=OFF& HTTP/1.0

利用HTTP1.0发起攻击时，肉鸡在一个TCP会话只能发一次HTTP请求。

» **HTTP1.1 pipelining攻击分析**

HTTP1.1 pipelining攻击抓包



No.	Time	Source	Destination	Protocol	Sport	Info
1	2020-09-28 20:18:49.308871	178.128.254.2	.196	HTTP	40330	GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/…
2	2020-09-28 20:18:49.309346	178.128.254.2	.196	HTTP	40010	GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/…
3	2020-09-28 20:18:49.311147	178.128.254.2	.196	HTTP	40318	GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/…
5	2020-09-28 20:18:49.311845	178.128.254.2	.196	HTTP	40326	GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/…
6	2020-09-28 20:18:49.312172	178.128.254.2	.196	HTTP	40316	GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/1.1 GET //players.json HTTP/…

为了提升HTTP传输效率，HTTP1.1通过pipelining特性允许客户端通过单个TCP连接发送多个HTTP请求。当HTTP1.1 pipelining被用于发起攻击时，肉鸡在一个会话上可发起近百次请求。当请求速率较快时，形成Multiple Methods攻击效果。

» **HTTP2.0 multiplexing攻击分析**

当HTTP1.1演进至HTTP2.0时，通过Multiplexing发起攻击时，肉鸡在一个TCP会话可发送100-500个HTTP请求。

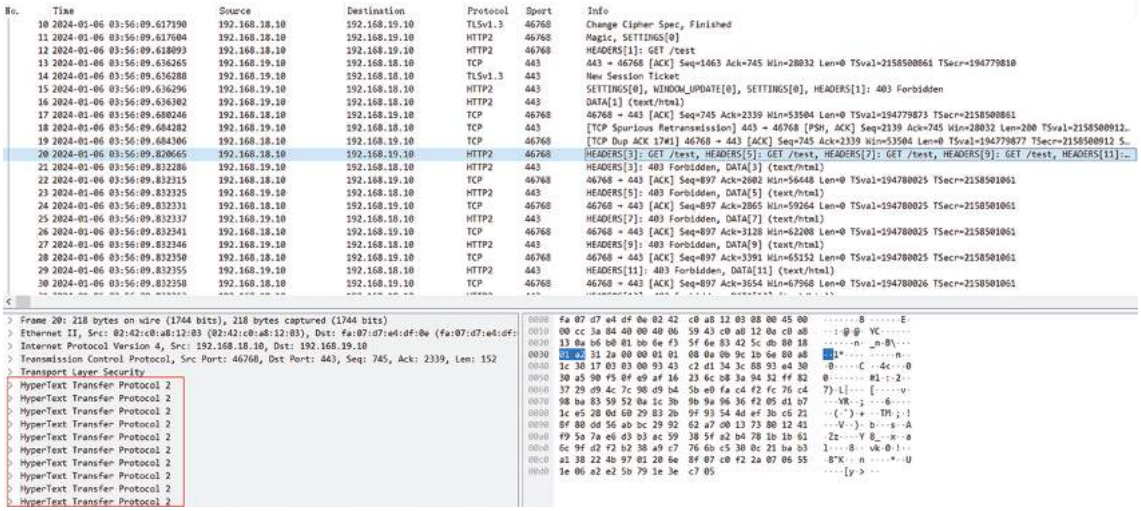
HTTP2.0攻击抓包

Time	Source	Destination	Protocol	Sport	Info
2024-01-04 09:23:01.542131	49.48.51.75	.152	TLSv1.3	46418	Client Hello
2024-01-04 09:23:01.544238	.152	49.48.51.75	TLSv1.3	443	Server Hello, Change Cipher Spec, Application Data, Application Data
2024-01-04 09:23:01.544281	.152	49.48.51.75	TLSv1.3	443	Application Data, Application Data
2024-01-04 09:23:01.963383	49.48.51.75	.152	TLSv1.3	46418	Change Cipher Spec, Application Data
2024-01-04 09:23:01.963598	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.963685	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.963719	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.971076	49.48.51.75	.152	TLSv1.3	46418	Application Data, Application Data
2024-01-04 09:23:01.972457	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.972526	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.973859	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.974939	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.975775	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.976615	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.977481	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.978421	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:01.979408	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.003808	49.48.51.75	.152	TLSv1.3	46418	Application Data, Application Data, Application Data
2024-01-04 09:23:02.004915	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.006005	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.006949	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.008124	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.009279	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.010431	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.015224	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.019201	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.023193	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.027786	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.032052	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.037059	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.038637	49.48.51.75	.152	TLSv1.3	46418	Application Data, Application Data, Application Data
2024-01-04 09:23:02.042016	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.049015	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.056250	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.063210	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.068365	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.072143	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.075726	.152	49.48.51.75	TLSv1.3	443	Application Data
2024-01-04 09:23:02.079584	.152	49.48.51.75	TLSv1.3	443	Application Data

对本次HTTP2.0攻击抓包进行分析发现，一个肉鸡最多一个会话发送105个请求报文，诱发服务器回应512个应答报文，说明肉鸡在一个会话上发送了512个HTTP2.0请求。

实验室模拟攻击抓包显示，在一个会话上快速发送HTTP2.0请求时，多个请求会直接合并到一个HTTP报文进行发送，形成Multiple Methods攻击效果。

模拟HTTP2.0攻击抓包

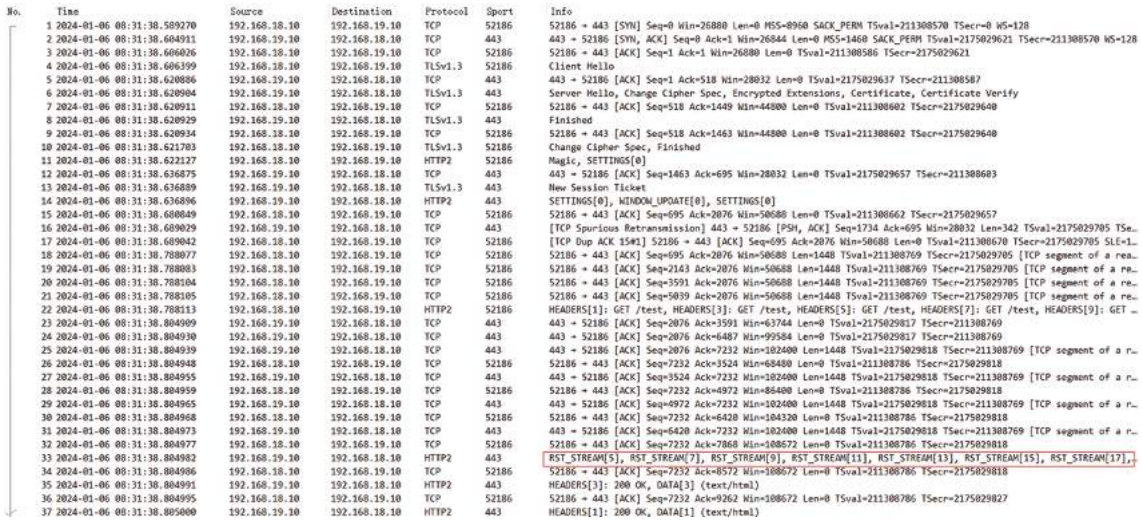


» **HTTP2.0 Rapid Reset攻击分析**

为了防止HTTP2.0被利用形成高速应用层攻击，HTTP2.0协议通过SETTINGS_MAX_CONCURRENT_STREAMS限制一个会话上最大请求数量。2023年10月10日，Rapid Reset漏洞被公布[7]：攻击者每发送一个HTTP2.0请求，随即发送一个RST_STREAM报文，可突破SETTINGS_MAX_CONCURRENT_STREAMS限制，一个会话上最多可发起上千次请求。

实验室模拟攻击，将SETTINGS_MAX_CONCURRENT_STREAMS设置为2，则一个会话上请求次数超过2时，服务器会回应RST_STREAM，同时丢弃多余请求。

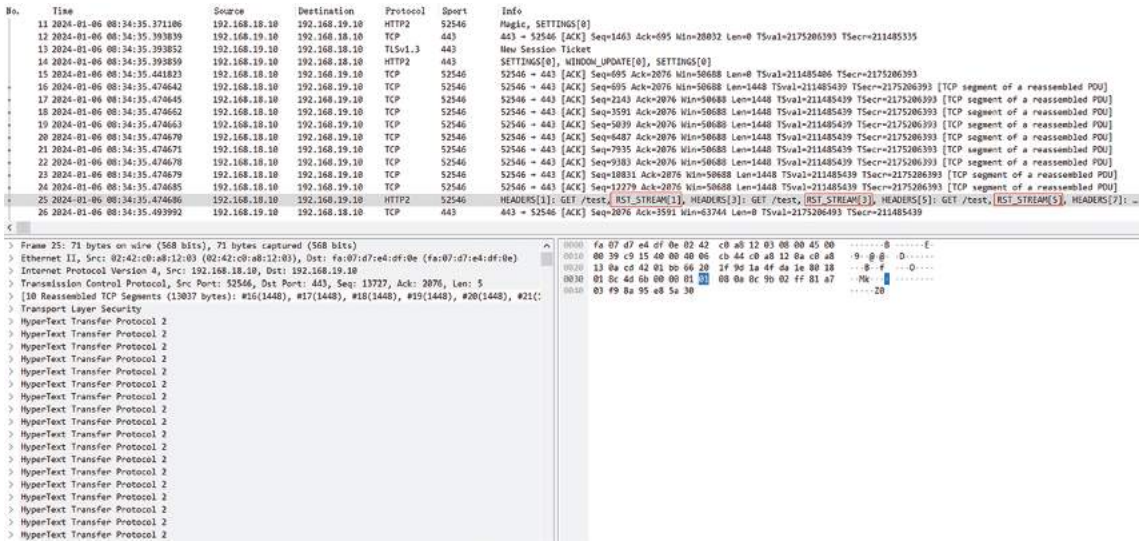
模拟HTTP2.0攻击抓包，未利用Rapid Reset漏洞时，攻击速率受SETTINGS_MAX_CONCURRENT_STREAMS限制



No.	Time	Source	Destination	Protocol	Sport	Info
1	2024-01-06 08:31:38.589270	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [SYN] Seq=0 Win=26880 Len=0 MSS=8960 SACK_PERM TSval=211308570 TSecr=0 WS=128
2	2024-01-06 08:31:38.604911	192.168.19.10	192.168.18.10	TCP	443	443 → 52186 [SYN, ACK] Seq=0 Ack=1 Win=26844 Len=0 MSS=1460 SACK_PERM TSval=2175029621 TSecr=211308570 WS=128
3	2024-01-06 08:31:38.606026	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=1 Ack=1 Win=26880 Len=0 TSval=211308586 TSecr=2175029621
4	2024-01-06 08:31:38.606399	192.168.18.10	192.168.19.10	TLSv1.3	52186	Client Hello
5	2024-01-06 08:31:38.620886	192.168.19.10	192.168.18.10	TCP	443	443 → 52186 [ACK] Seq=1 Ack=518 Win=28032 Len=0 TSval=2175029637 TSecr=211308587
6	2024-01-06 08:31:38.620904	192.168.19.10	192.168.18.10	TLSv1.3	443	Server Hello, Change Cipher Spec, Encrypted Extensions, Certificate, Certificate Verify
7	2024-01-06 08:31:38.620911	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=518 Ack=1449 Win=44800 Len=0 TSval=211308602 TSecr=2175029640
8	2024-01-06 08:31:38.620929	192.168.19.10	192.168.18.10	TLSv1.3	443	Finished
9	2024-01-06 08:31:38.620934	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=518 Ack=1463 Win=44800 Len=0 TSval=211308602 TSecr=2175029640
10	2024-01-06 08:31:38.621703	192.168.18.10	192.168.19.10	TLSv1.3	52186	Change Cipher Spec, Finished
11	2024-01-06 08:31:38.622127	192.168.18.10	192.168.19.10	HTTP2	52186	Magic, SETTINGS[0]
12	2024-01-06 08:31:38.636875	192.168.19.10	192.168.18.10	TCP	443	443 → 52186 [ACK] Seq=1463 Ack=695 Win=28032 Len=0 TSval=2175029657 TSecr=211308603
13	2024-01-06 08:31:38.636889	192.168.19.10	192.168.18.10	TLSv1.3	443	New Session Ticket
14	2024-01-06 08:31:38.636896	192.168.19.10	192.168.18.10	HTTP2	443	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
15	2024-01-06 08:31:38.680849	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=695 Ack=2076 Win=50688 Len=0 TSval=211308662 TSecr=2175029657
16	2024-01-06 08:31:38.689029	192.168.19.10	192.168.18.10	TCP	443	[TCP Spurious Retransmission] 443 → 52186 [PSH, ACK] Seq=1734 Ack=695 Win=28032 Len=342 TSval=2175029705 TSe…
17	2024-01-06 08:31:38.689042	192.168.18.10	192.168.19.10	TCP	52186	[TCP Dup ACK 15#1] 52186 → 443 [ACK] Seq=695 Ack=2076 Win=50688 Len=0 TSval=211308670 TSecr=2175029705 SLE=1…
18	2024-01-06 08:31:38.788077	192.168.19.10	192.168.18.10	TCP	52186	52186 → 443 [ACK] Seq=695 Ack=2076 Win=50688 Len=1448 TSval=211308769 TSecr=2175029705 [TCP segment of a rea…
19	2024-01-06 08:31:38.788083	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=2143 Ack=2076 Win=50688 Len=1448 TSval=211308769 TSecr=2175029705 [TCP segment of a re…
20	2024-01-06 08:31:38.788104	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=3591 Ack=2076 Win=50688 Len=1448 TSval=211308769 TSecr=2175029705 [TCP segment of a re…
21	2024-01-06 08:31:38.788105	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=5039 Ack=2076 Win=50688 Len=1448 TSval=211308769 TSecr=2175029705 [TCP segment of a re…
22	2024-01-06 08:31:38.788113	192.168.18.10	192.168.19.10	HTTP2	52186	HEADERS[1]: GET /test, HEADERS[3]: GET /test, HEADERS[5]: GET /test, HEADERS[7]: GET /test, HEADERS[9]: GET …
23	2024-01-06 08:31:38.804909	192.168.19.10	192.168.18.10	TCP	443	443 → 52186 [ACK] Seq=2076 Ack=3591 Win=63744 Len=0 TSval=2175029817 TSecr=211308769
24	2024-01-06 08:31:38.804930	192.168.19.10	192.168.18.10	TCP	443	443 → 52186 [ACK] Seq=2076 Ack=6487 Win=99584 Len=0 TSval=2175029817 TSecr=211308769
25	2024-01-06 08:31:38.804939	192.168.19.10	192.168.18.10	TCP	443	443 → 52186 [ACK] Seq=2076 Ack=7232 Win=102400 Len=1448 TSval=2175029818 TSecr=211308769 [TCP segment of a r…
26	2024-01-06 08:31:38.804948	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=7232 Ack=3524 Win=68480 Len=0 TSval=211308786 TSecr=2175029818
27	2024-01-06 08:31:38.804955	192.168.19.10	192.168.18.10	TCP	443	443 → 52186 [ACK] Seq=3524 Ack=7232 Win=102400 Len=1448 TSval=2175029818 TSecr=211308769 [TCP segment of a r…
28	2024-01-06 08:31:38.804959	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=7232 Ack=4972 Win=86400 Len=0 TSval=211308786 TSecr=2175029818
29	2024-01-06 08:31:38.804965	192.168.19.10	192.168.18.10	TCP	443	443 → 52186 [ACK] Seq=4972 Ack=7232 Win=102400 Len=1448 TSval=2175029818 TSecr=211308769 [TCP segment of a r…
30	2024-01-06 08:31:38.804968	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=7232 Ack=6420 Win=104320 Len=0 TSval=211308786 TSecr=2175029818
31	2024-01-06 08:31:38.804973	192.168.19.10	192.168.18.10	TCP	443	443 → 52186 [ACK] Seq=6420 Ack=7232 Win=102400 Len=1448 TSval=2175029818 TSecr=211308769 [TCP segment of a r…
32	2024-01-06 08:31:38.804977	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=7232 Ack=7868 Win=108672 Len=0 TSval=211308786 TSecr=2175029818
33	2024-01-06 08:31:38.804982	192.168.19.10	192.168.18.10	HTTP2	443	RST_STREAM[5], RST_STREAM[7], RST_STREAM[9], RST_STREAM[11], RST_STREAM[13], RST_STREAM[15], RST_STREAM[17],…
34	2024-01-06 08:31:38.804986	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=7232 Ack=8572 Win=108672 Len=0 TSval=211308786 TSecr=2175029818
35	2024-01-06 08:31:38.804991	192.168.19.10	192.168.18.10	HTTP2	443	HEADERS[3]: 200 OK, DATA[3] (text/html)
36	2024-01-06 08:31:38.804995	192.168.18.10	192.168.19.10	TCP	52186	52186 → 443 [ACK] Seq=7232 Ack=9262 Win=108672 Len=0 TSval=211308786 TSecr=2175029827
37	2024-01-06 08:31:38.805000	192.168.19.10	192.168.18.10	HTTP2	443	HEADERS[1]: 200 OK, DATA[1] (text/html)

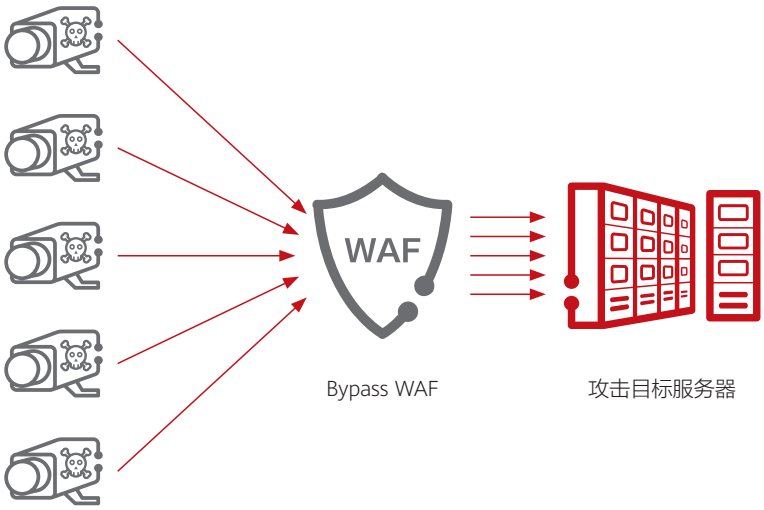
在保持SETTINGS_MAX_CONCURRENT_STREAMS设置为2不变时，当利用Rapid Reset漏洞发送攻击时，服务器不再回应RST_STREAM，攻击速率大幅度提升。

模拟HTTP2.0 Rapid Reset攻击抓包



No.	Time	Source	Destination	Protocol	Sport	Info
11	2024-01-06 08:34:35.371186	192.168.18.10	192.168.19.10	HTTP2	52546	Magic, SETTINGS[0]
12	2024-01-06 08:34:35.393839	192.168.19.10	192.168.18.10	TCP	443	443 → 52546 [ACK] Seq=1463 Ack=695 Win=28032 Len=0 TSval=2175206393 TSecr=211485335
13	2024-01-06 08:34:35.393852	192.168.19.10	192.168.18.10	TLSv1.3	443	New Session Ticket
14	2024-01-06 08:34:35.393859	192.168.19.10	192.168.18.10	HTTP2	443	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
15	2024-01-06 08:34:35.441823	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=695 Ack=2076 Win=50688 Len=0 TSval=211485406 TSecr=2175206393
16	2024-01-06 08:34:35.474642	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=695 Ack=2076 Win=50688 Len=1448 TSval=211485439 TSecr=2175206393 [TCP segment of a reassembled PDU]
17	2024-01-06 08:34:35.474645	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=2143 Ack=2076 Win=50688 Len=1448 TSval=211485439 TSecr=2175206393 [TCP segment of a reassembled PDU]
18	2024-01-06 08:34:35.474662	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=3591 Ack=2076 Win=50688 Len=1448 TSval=211485439 TSecr=2175206393 [TCP segment of a reassembled PDU]
19	2024-01-06 08:34:35.474663	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=5039 Ack=2076 Win=50688 Len=1448 TSval=211485439 TSecr=2175206393 [TCP segment of a reassembled PDU]
20	2024-01-06 08:34:35.474670	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=6487 Ack=2076 Win=50688 Len=1448 TSval=211485439 TSecr=2175206393 [TCP segment of a reassembled PDU]
21	2024-01-06 08:34:35.474671	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=7935 Ack=2076 Win=50688 Len=1448 TSval=211485439 TSecr=2175206393 [TCP segment of a reassembled PDU]
22	2024-01-06 08:34:35.474678	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=9383 Ack=2076 Win=50688 Len=1448 TSval=211485439 TSecr=2175206393 [TCP segment of a reassembled PDU]
23	2024-01-06 08:34:35.474679	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=10831 Ack=2076 Win=50688 Len=1448 TSval=211485439 TSecr=2175206393 [TCP segment of a reassembled PDU]
24	2024-01-06 08:34:35.474685	192.168.18.10	192.168.19.10	TCP	52546	52546 → 443 [ACK] Seq=12279 Ack=2076 Win=50688 Len=1448 TSval=211485439 TSecr=2175206393 [TCP segment of a reassembled PDU]
25	2024-01-06 08:34:35.474686	192.168.18.10	192.168.19.10	HTTP2	52546	HEADERS[1]: GET /test, RST_STREAM[1], HEADERS[3]: GET /test, RST_STREAM[3], HEADERS[5]: GET /test, RST_STREAM[5], HEADERS[7]: …
26	2024-01-06 08:34:35.493992	192.168.19.10	192.168.18.10	TCP	443	443 → 52546 [ACK] Seq=2076 Ack=3591 Win=63744 Len=0 TSval=2175206493 TSecr=211485439

低速HTTP/HTTPS应用层攻击



低速HTTP/HTTPS应用层攻击强度大多在百万级RPS甚至更低，一次攻击事件使用的肉鸡数量在100,000-250,000之间，肉鸡多为低性能的IoT终端，单个肉鸡的攻击速率普遍在几十RPS甚至更低。

2023年12月25日，百度某业务系统遭受HTTPS Flood攻击，攻击峰值速率480,000rps，共采用24万个肉鸡，平均每个肉鸡请求速率仅2rps。本次攻击即属于典型的低速应用层攻击。

低速应用层攻击中，肉鸡的请求速率落在业务访问速率区间，甚至远低于正常业务请求速率，导致源速率检测失效，躲避能力更强，因此这类攻击也被称为bypass WAF attacks。

为了达到更好的攻击效果，低速攻击时，攻击目标URL多是经过精心挑选的、可消耗服务器更多资源，包括消耗网络outbound带宽的大资源URL或消耗更多计算资源的查询URL。

» **利用“秒拨”动态IP技术发起的低速TLS加密攻击分析**

2023年11月份，某金融企业遭受持续性加密攻击，攻击采用“秒拨”技术，单个攻击源请求速率低，防御难度大。攻击目标是门户网站，攻击手法属于典型的大资源请求模式，以很少的请求报文获得大量的回应报文，导致企业网络链路出向带宽拥塞，进而影响到所有互联网业务访问。

利用秒拨发起的TLS加密攻击抓包

Time	Source	Destination	Protocol	Sport	Info
2023-11-15 22:00:25.909876060	1 .239.209.16	.132	TLSv1.2	14339	Client Hello
2023-11-15 22:00:25.655882140	1 .239.217.112	.132	TLSv1.2	6595	Client Hello
2023-11-15 22:00:25.232319690	1 .239.218.12	.132	TLSv1.2	10988	Client Hello
2023-11-15 22:00:25.324613840	1 .239.222.13	.132	TLSv1.2	2489	Client Hello
2023-11-15 22:00:25.112377540	1 .239.241.247	.132	TLSv1.2	7635	Client Hello
2023-11-15 22:00:25.856557020	1 .239.243.31	.132	TLSv1.2	15739	Client Hello
2023-11-15 22:00:25.107338340	1 .239.245.57	.132	TLSv1.2	7724	Client Hello
2023-11-15 22:00:25.662709300	1 .239.249.143	.132	TLSv1.2	1438	Client Hello
2023-11-15 22:00:25.180288720	1 .239.29.187	.132	TLSv1.2	11654	Client Hello
2023-11-15 22:00:25.098160290	1 .239.32.135	.132	TLSv1.2	30396	Client Hello
2023-11-15 22:00:25.825142720	1 .239.39.224	.132	TLSv1.2	11157	Client Hello
2023-11-15 22:00:25.805566540	1 .239.4.18	.132	TLSv1.2	6461	Client Hello
2023-11-15 22:00:25.064028770	1 .239.40.14	.132	TLSv1.2	4726	Client Hello
2023-11-15 22:00:25.691811580	1 .239.41.8	.132	TLSv1.2	42172	Client Hello
2023-11-15 22:00:25.767551150	1 .239.75.227	.132	TLSv1.2	5357	Client Hello
2023-11-15 22:00:25.953306000	1 .239.75.34	.132	TLSv1.2	12061	Client Hello
2023-11-15 22:00:25.794616440	1 .239.84.182	.132	TLSv1.2	3410	Client Hello
2023-11-15 22:00:25.042639590	1 .239.85.165	.132	TLSv1.2	6637	Client Hello
2023-11-15 22:00:25.160938630	1 .239.88.159	.132	TLSv1.2	1221	Client Hello
2023-11-15 22:00:25.166091770	1 .241.117.210	.132	TLSv1.2	51164	Client Hello
2023-11-15 22:00:25.709023860	1 .241.117.213	.132	TLSv1.2	43146	Client Hello
2023-11-15 22:00:25.165675580	1 .242.107.187	.132	TLSv1.2	12128	Client Hello
2023-11-15 22:00:25.199877800	1 .242.12.29	.132	TLSv1.2	33108	Client Hello
2023-11-15 22:00:25.478972000	1 .242.13.78	.132	TLSv1.2	28920	Client Hello
2023-11-15 22:00:25.017628240	1 .242.138.208	.132	TLSv1.2	15892	Client Hello
2023-11-15 22:00:25.676017530	1 .242.143.64	.132	TLSv1.2	59053	Client Hello
2023-11-15 22:00:25.082316510	1 .242.189.151	.132	TLSv1.2	43888	Client Hello
2023-11-15 22:00:25.546443150	1 .242.191.84	.132	TLSv1.2	44620	Client Hello
2023-11-15 22:00:25.791160440	1 .242.222.63	.132	TLSv1.2	37774	Client Hello
2023-11-15 22:00:25.113425390	1 .242.233.64	.132	TLSv1.2	47354	Client Hello
2023-11-15 22:00:25.480103920	1 .242.238.75	.132	TLSv1.2	6113	Client Hello
2023-11-15 22:00:25.171524070	1 .242.239.122	.132	TLSv1.2	5853	Client Hello
2023-11-15 22:00:25.768451160	1 .242.39.190	.132	TLSv1.2	54332	Client Hello
2023-11-15 22:00:25.138266590	1 .242.45.239	.132	TLSv1.2	25148	Client Hello
2023-11-15 22:00:25.952070940	1 .242.53.99	.132	TLSv1.2	32410	Client Hello
2023-11-15 22:00:25.256313600	1 .242.62.123	.132	TLSv1.2	6303	Client Hello

“秒拨”攻击技术本质上是利用运营商家庭宽带拨号上网业务允许同一个MAC地址短时间内拨号可更换不同IP地址的漏洞，攻击机每一次断线重连会获取一个新的攻击IP，攻击呈现出超低速态势，防御困难。

一次秒拨攻击抓包

Time	Source	Destination	Protocol	Sport	Info
2023-11-15 22:00:25.264844100	1 .225.19.85	.132	TLSv1.2	16322	Client Hello
2023-11-15 22:00:25.270632210	.132	1 .225.19.85	TLSv1.2	443	Server Hello
2023-11-15 22:00:25.270635470	.132	1 .225.19.85	TLSv1.2	443	Certificate, Server Key Exchange, Server Hello Done
2023-11-15 22:00:25.344398260	1 .225.19.85	.132	TLSv1.2	16322	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2023-11-15 22:00:25.346626490	.132	1 .225.19.85	TLSv1.2	443	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
2023-11-15 22:00:25.413780490	1 .225.19.85	.132	TLSv1.2	16322	Application Data
2023-11-15 22:00:25.479076860	.132	1 .225.19.85	TLSv1.2	443	Application Data
2023-11-15 22:00:25.481980060	.132	1 .225.19.85	TLSv1.2	443	Application Data
2023-11-15 22:00:25.612474050	.132	1 .225.19.85	TLSv1.2	443	Application Data

利用“秒拨”发起的攻击难防御的原因在于：运营商家庭宽带NAT地址池IP数量庞大，“秒拨”攻击机可通过不断的断线重连，轻松“轮换使用”运营商整个城域网NAT地址池中的IP，让百万量级的拨号上网IP地址沦为“肉鸡”。本次针对金融企业的攻击同时使用了9个城域网的家庭宽带NAT地址池。攻击中，秒拨机和服务器建立连接后，发起一次请求，随即断开连接，再次攻击时则换成另外一个IP。从攻击目标服务器来看，每个源IP只请求一次，比正常用户的请求速率还慢，攻击识别困难。且“秒拨”攻击机IP和正常用户IP属于同一个NAT地址池，一次“秒拨”攻击结束后，“秒拨”攻击机使用的IP资源会流转到正常用户手中，导致“秒拨”攻击机IP和正常用户IP无法区分，防御容易影响正常用户访问。

» **利用不常用Method发起的低速HTTP攻击分析**

利用不常用Method请求发起的低速攻击抓包

Time	Source	Destination	Protocol	Sport	Info	Length
2023-05-13 13:23:57.692715	36.142.138.106	.165	HTTP	9066	HEAD / HTTP/1.1	145
2023-05-13 13:23:57.732234	49.67.2.48	.165	HTTP	6311	HEAD / HTTP/1.1	145
2023-05-13 13:23:57.734209	123.127.214.252	.163	HTTP	33210	HEAD / HTTP/1.1	145
2023-05-13 13:23:57.743189	183.216.130.192	.163	HTTP	33787	HEAD / HTTP/1.1	145
2023-05-13 13:23:57.825188	36.142.137.128	.165	HTTP	61614	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.157198	14.118.55.250	.165	HTTP	15176	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.165684	183.219.3.84	.165	HTTP	9568	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.173676	111.72.55.71	.163	HTTP	6275	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.173692	111.72.55.71	.163	TCP	6275	[TCP Retransmission…	145
2023-05-13 13:23:58.184191	39.160.31.238	.165	HTTP	9046	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.196690	120.203.130.114	.165	HTTP	21417	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.205191	183.216.215.144	.163	HTTP	26843	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.230174	36.142.139.94	.163	HTTP	31312	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.280684	117.162.181.208	.165	HTTP	56921	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.289203	36.142.166.192	.163	HTTP	26309	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.307675	117.154.198.111	.163	HTTP	31564	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.307701	117.150.225.100	.165	HTTP	11262	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.330702	183.218.17.28	.163	HTTP	5395	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.368197	183.219.146.250	.165	HTTP	6027	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.382194	117.170.100.15	.163	HTTP	42359	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.467697	183.192.121.127	.163	HTTP	2656	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.500168	117.152.112.40	.163	HTTP	44356	HEAD / HTTP/1.1	145
2023-05-13 13:23:58.522176	120.231.25.45	.163	HTTP	29867	HEAD / HTTP/1.1	145

2023年5月13日，针对某运营商门户网站的攻击中，一个攻击源一秒建立一个会话，一个会话上仅发送一次145字节的HEAD请求，而服务器返回的内容是请求报文的几百倍。

» **利用服务器的“防呆”功能发起的低速HTTP攻击分析**

利用服务器的“防呆”功能发起的低速攻击抓包

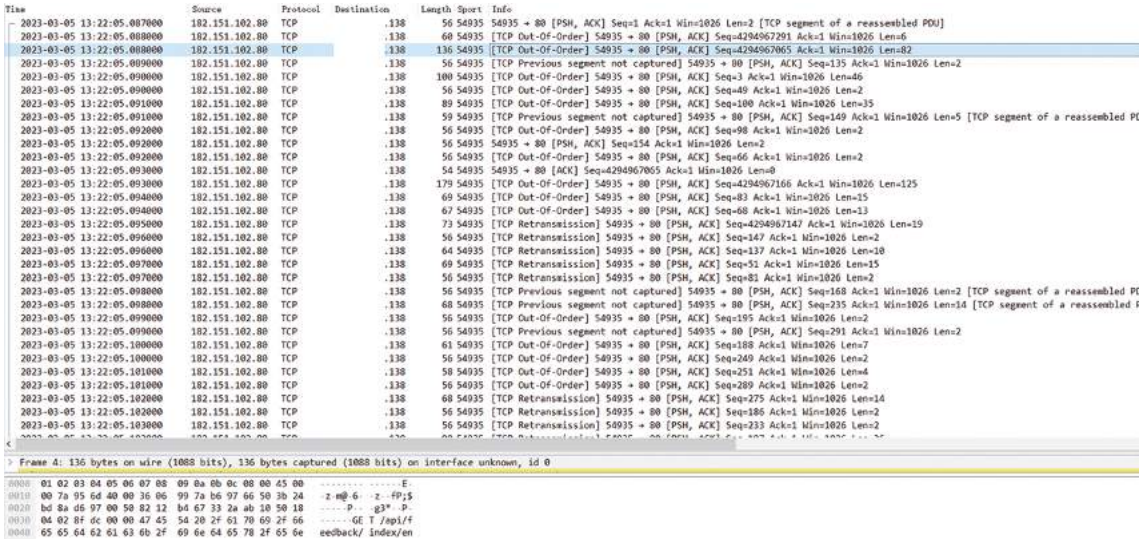
Time	Source	Destination	Protocol	Sport	Info	Length
2023-02-27 11:06:39.744971	203.95.198.84	.138	HTTP	38883	GET // HTTP/1.0	514
2023-02-27 11:06:40.262303	203.95.198.84	.138	HTTP	38574	GET // HTTP/1.0	588
2023-02-27 11:06:40.867390	203.95.198.84	.138	HTTP	38899	GET // HTTP/1.0	479
2023-02-27 11:06:41.511352	116.49.163.71	.138	HTTP	56454	GET // HTTP/1.0	498
2023-02-27 11:06:46.562794	116.49.163.71	.138	HTTP	56602	GET // HTTP/1.0	496
2023-02-27 11:06:47.992773	203.95.198.84	.138	HTTP	38941	GET // HTTP/1.0	462
2023-02-27 11:06:48.085122	218.252.206.89	.138	HTTP	50970	GET // HTTP/1.0	462
2023-02-27 11:06:49.389249	106.105.218.244	.138	HTTP	38072	GET // HTTP/1.0	601
2023-02-27 11:06:49.602072	116.49.163.71	.138	HTTP	56668	GET // HTTP/1.0	588
2023-02-27 11:06:49.615099	81.34.221.110	.138	HTTP	35954	GET // HTTP/1.0	496
2023-02-27 11:06:51.236113	116.49.163.71	.138	HTTP	56686	GET // HTTP/1.0	588
2023-02-27 11:06:50.983864	203.95.198.84	.138	HTTP	38969	GET // HTTP/1.0	498
2023-02-27 11:06:51.503769	116.49.163.71	.138	HTTP	56698	GET // HTTP/1.0	601
2023-02-27 11:06:52.359114	203.95.198.84	.138	HTTP	38976	GET // HTTP/1.0	519
2023-02-27 11:06:52.615125	39.175.85.98	.138	HTTP	19143	GET // HTTP/1.0	498
2023-02-27 12:47:13.298904	203.95.198.84	.138	HTTP	10198	GET // HTTP/1.0	514
2023-02-27 12:47:16.976489	123.205.34.233	.138	HTTP	34150	GET // HTTP/1.0	588
2023-02-27 12:47:18.892002	203.95.198.84	.138	HTTP	10097	GET // HTTP/1.0	601

门户网站一般具备一定的“防呆”功能，以容错互联网用户的输入性错误。2023年2月27日，某金融企业门户网站遭受低速HTTP Flood攻击。从攻击抓包截图可看到，攻击者利用门户网站对根目录的“防呆”功能，发起低速攻击，攻击者期望以此躲避安全系统的内容过滤。

» **报文分段以躲避安全系统的内容过滤**

2023年3月5日，某金融企业门户网站遭受低速HTTP Flood攻击。一个完整的HTTP请求报文被拆分成几十个报文进行传输，如果安全系统支持对报文进行重组，则消耗安全系统性能；如果不重组，则该种攻击模式可躲过安全系统的内容过滤。

报文分段以躲避安全系统的内容过滤的攻击抓包



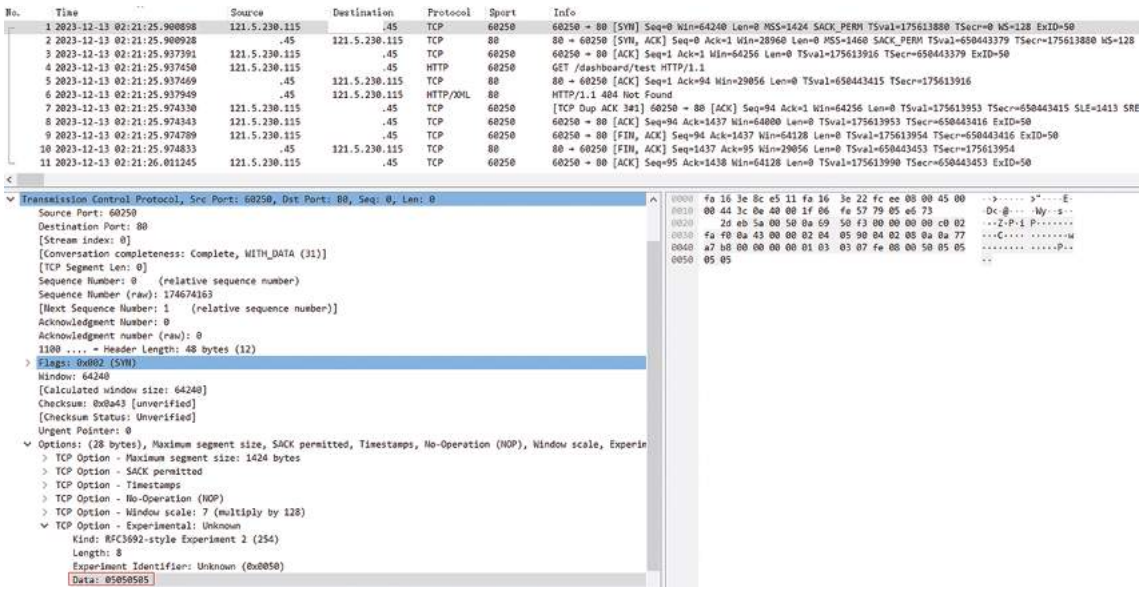
如果把如上报文进行重组，则完整的HTTP报文头内容如下图所示：

```
GET /api/feedback/index/environment/variable?random=0.40249836870292444 HTTP/1.1
Host: www.xxxx.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.0.0 Safari/537.36
Accept: application/json, text/javascript, */*; q=0.01
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close
Content-Type: application/json
Referer: http://www.xxxx.cn/market/index.html
X-Request-Type: ajax
X-Requested-With: XMLHttpRequest
```

5. TOA漏洞被利用，引发TCP四层代理场景互联网业务信任风险

四层代理场景，业务需要通过TOA（TCP Option Address）获取用户原始IP，但TOA由于缺乏严格的校验机制，便于伪造，从而引发TCP四层代理业务场景的互联网业务信任风险。2023年12月初，TCP四层代理机制TOA（TCP Option Address）漏洞被披露[8]，12月13日，华为云监测到利用TOA漏洞的攻击报文。

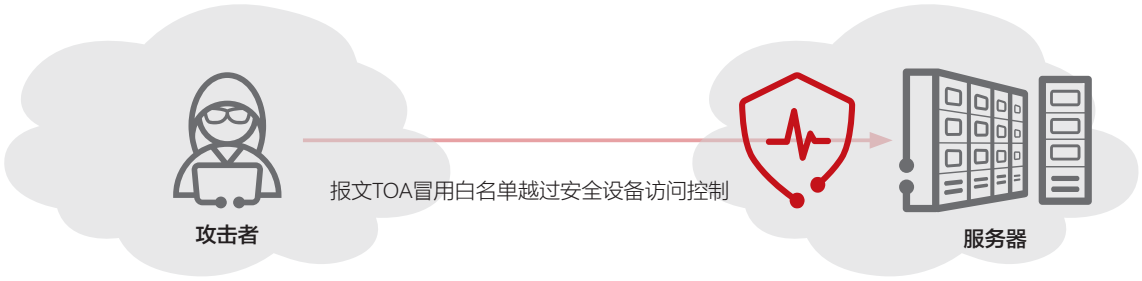
利用TOA漏洞的攻击抓包



在该利用TOA漏洞的攻击报文中，伪造的原始IP地址是5.5.5.5。

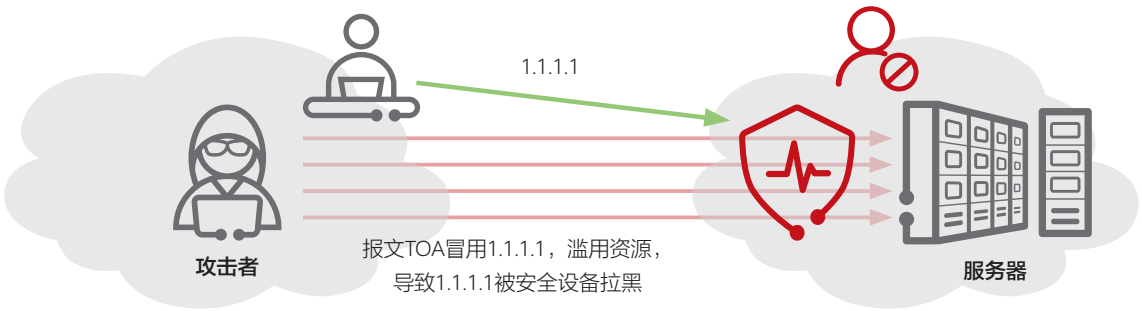
TOA漏洞被利用形成攻击威胁的场景主要有两种。场景1是攻击者把TOA伪造成数据中心经常使用的白名单，实现越权访问的目的。

场景1：TOA伪装成白名单，越权访问



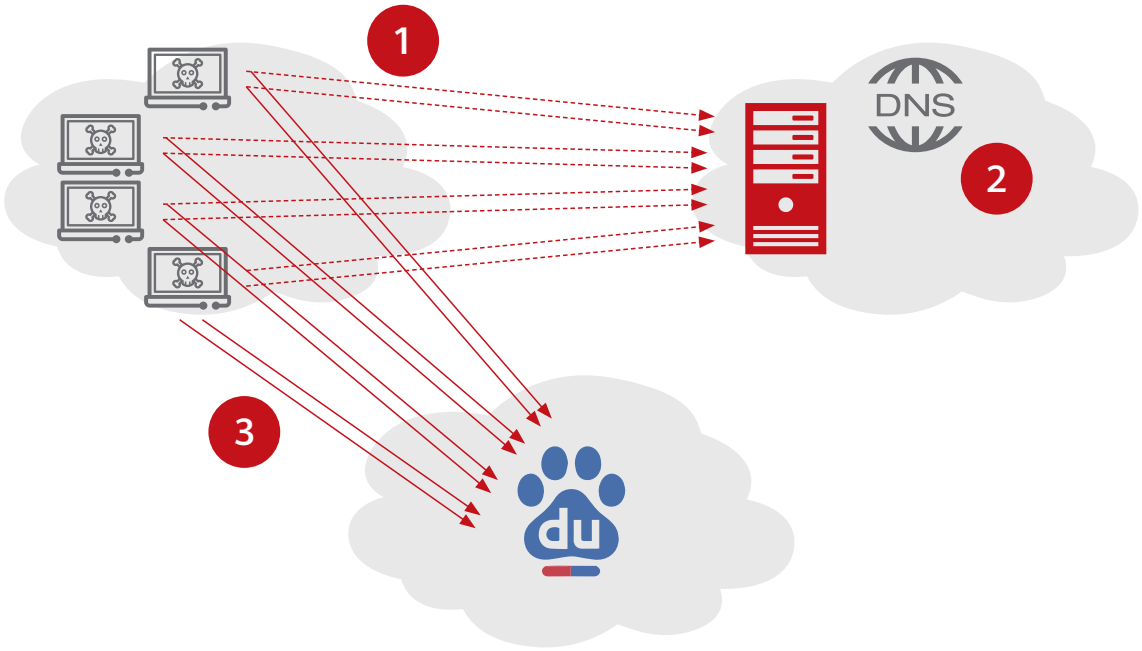
场景2是攻击者利用TOA伪装成“仇家”IP，对数据中心服务器资源不停滥用，比如打CC攻击，导致该IP被安全设备加入到黑名单，当“仇家”访问数据中心资源时被安全设备直接拦截。

场景2：TOA伪装成“仇家”IP，实现攻击栽赃



6. 通过修改DNS记录，零成本 “转移” 攻击危害

通过修改DNS记录，零成本 “转移” 攻击危害

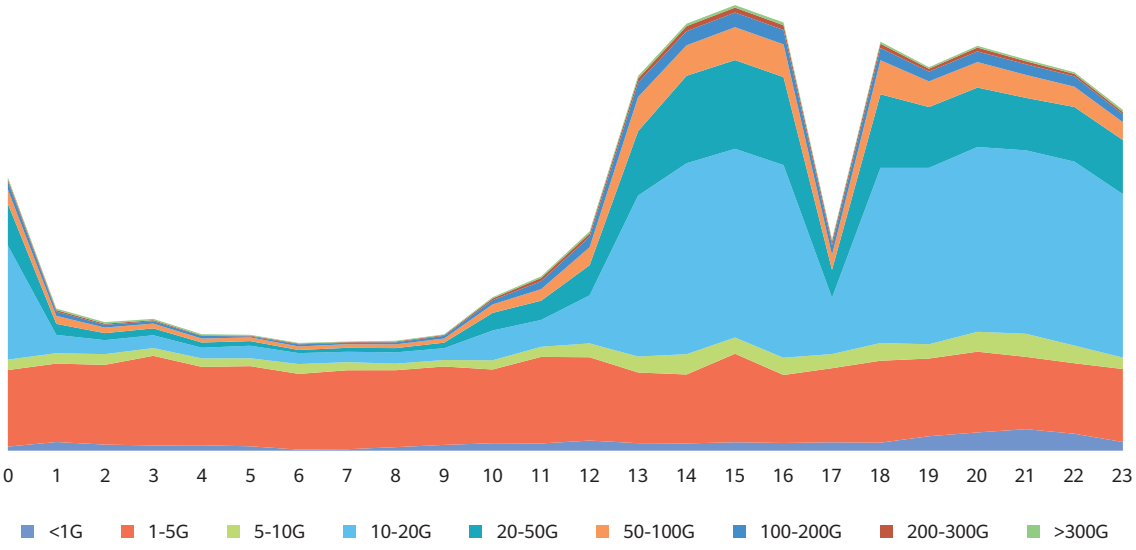


2023年，百度数据中心多次被迫成为“公共清洗池 ”。当某些互联网业务成为持续性攻击目标时，被攻击的互联网企业为减少损失，通过更改DNS记录将被攻击域名解析到百度，攻击流量被“转移”至 百度。

2.1.5 攻击发生时段

2023年攻击发生时间段和往年一样，为达到以最低的攻击成本实现最大化的攻击效果，攻击发生时间和互联网用户的作息时间保持一致。

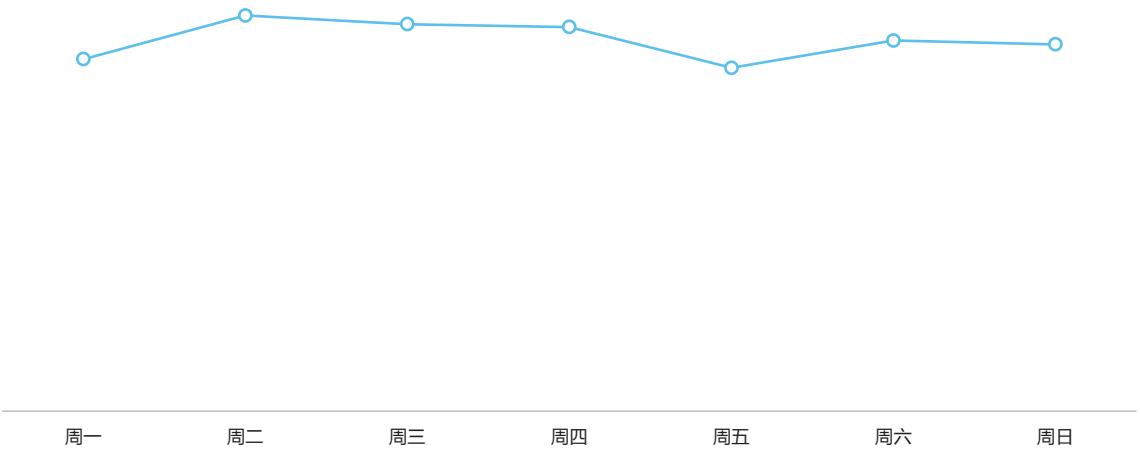
2023年攻击发生时段分布



数据来源于电信安全

攻击频次周天分布显示，攻击一周内分布较均匀。

2023年攻击频次周天分布

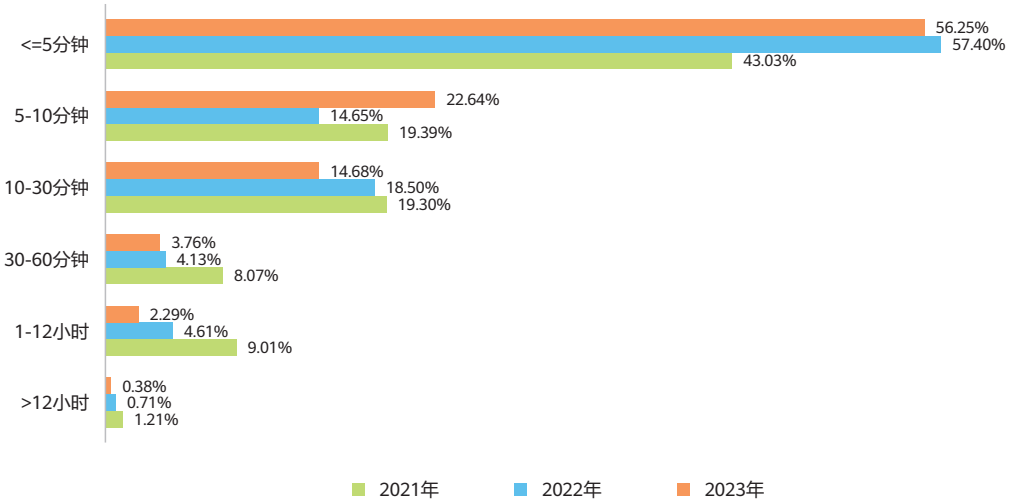


数据来源于电信安全&中移云能&中移卓望&华为

2.1.6 攻击持续时间

2023年，56.25%的网络层攻击持续时间不超过5分钟，可见“瞬时泛洪”依然是网络层攻击的一大特点，瞬时泛洪攻击挑战防御系统的自动化程度和运维团队的响应速度。

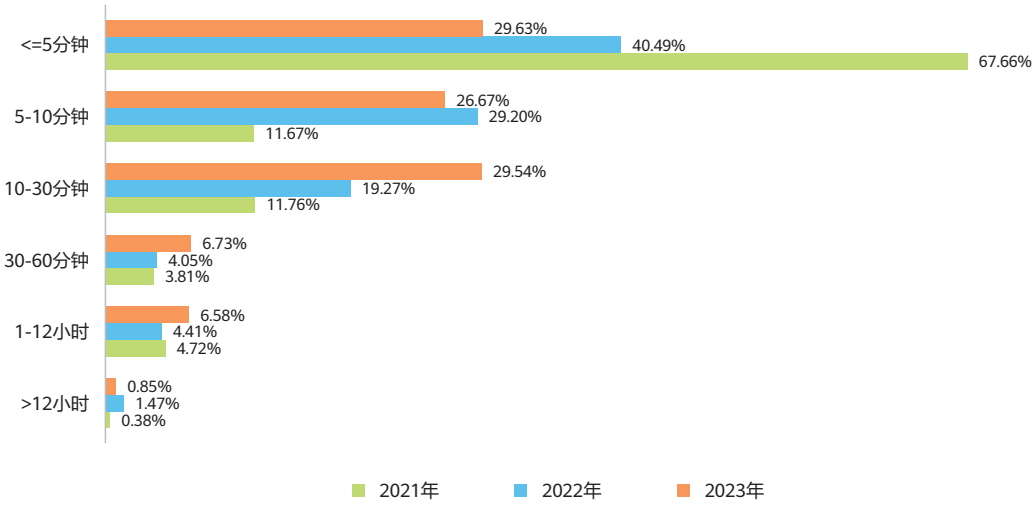
2021-2023年网络层攻击持续时间分布



数据来源于电信安全&Nexusguard&华为

对比近三年应用层攻击持续时间可发现，持续10-30分钟的应用层攻击占比持续提升。说明应用层攻击成本持续降低，为了提升攻击目标的攻击损失或提升防御成本，攻击者普遍采取延长攻击时长的做法。

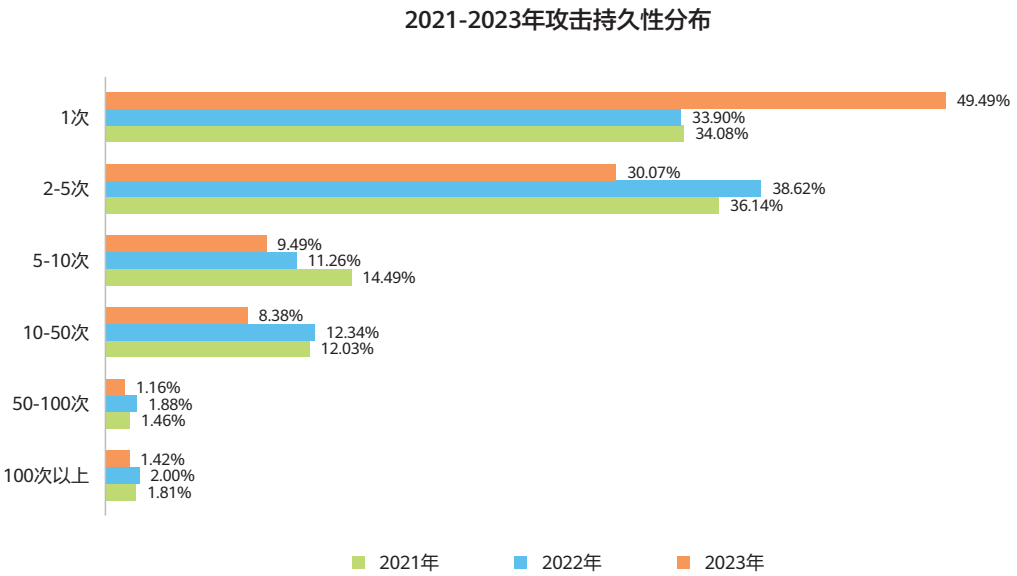
2021-2023年应用层攻击持续时间分布



数据来源于电信安全&Nexusguard&华为

2.1.7 攻击持久性

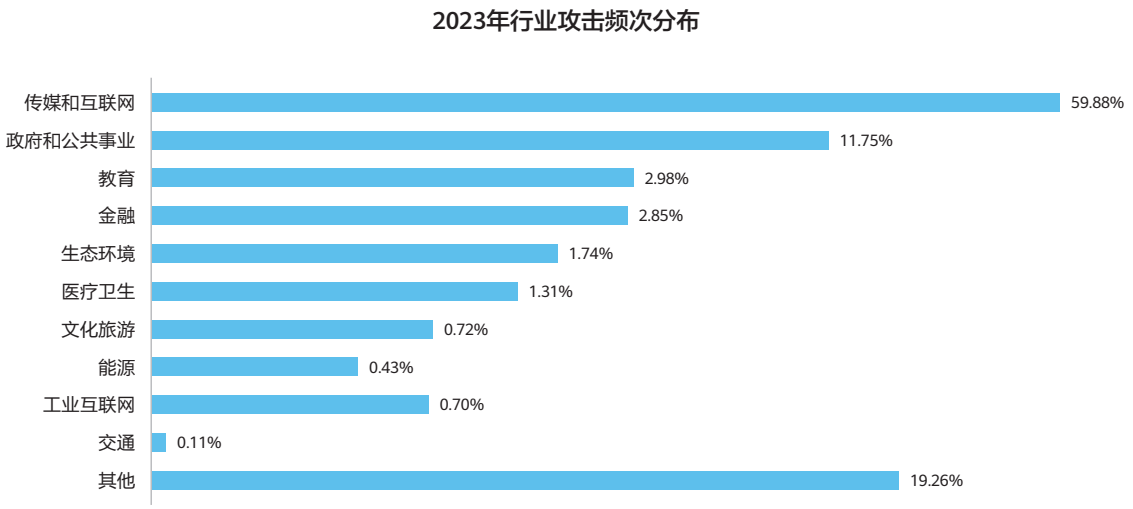
2023年，遭受过一次攻击的IP数量占比提升至49.49%，主要源于扫段攻击活跃，很多IP不幸“躺枪”。



数据来源于电信安全&联通数科&华为

2.1.8 攻击目标行业分布

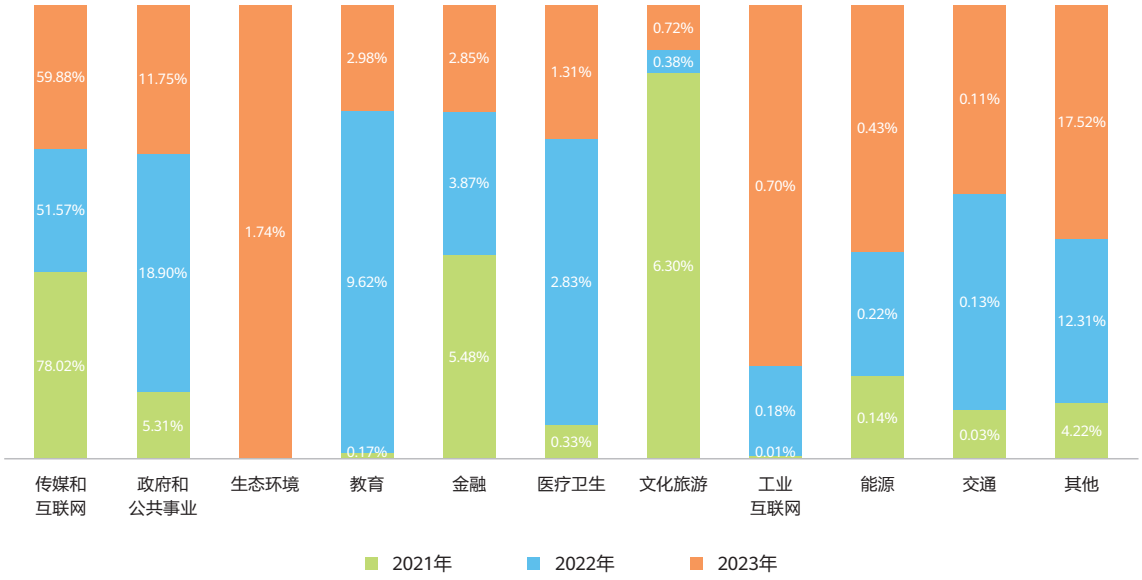
2023年，传媒和互联网、政府和公共事业、教育、金融依然是TOP4攻击目标行业，攻击频次占比依次为59.88%、11.75%、2.98%、2.85%。



数据来源于联通数科

2021-2023年行业攻击频次分布显示，能源行业和工业互联网受攻击频次占比连续三年增长；其他企业受攻击频次占比上升明显，说明DDoS攻击已遍布各个行业。

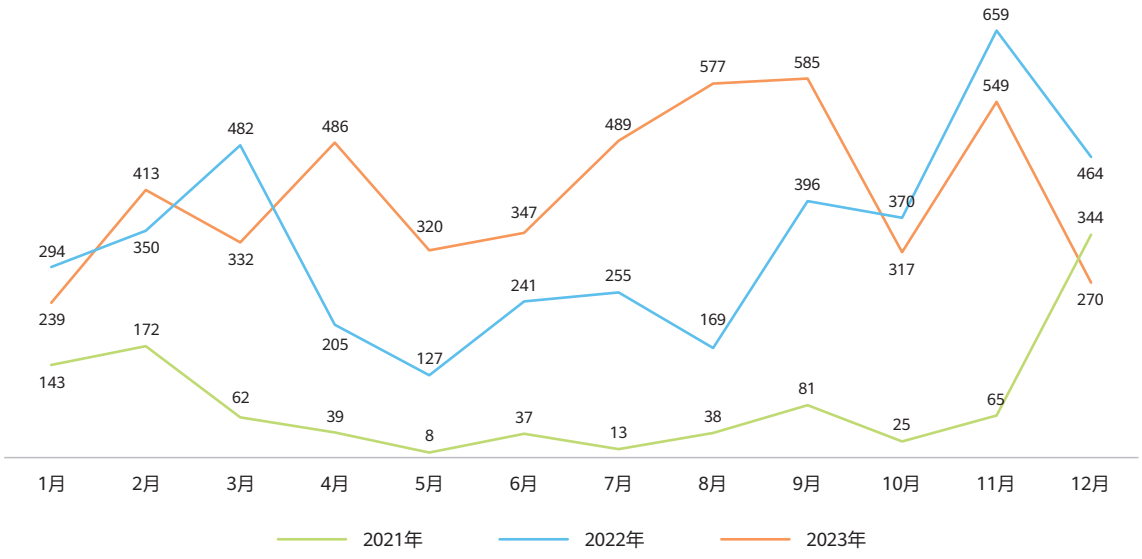
2021-2023年行业攻击频次趋势分布



数据来源于联通数科

2021-2023年中国金融行业攻击频次月度分布显示，中国金融行业攻击频次呈持续增长趋势，2023年全年共发生4,924次攻击，是2022年的1.23倍，2021年的4.79倍。

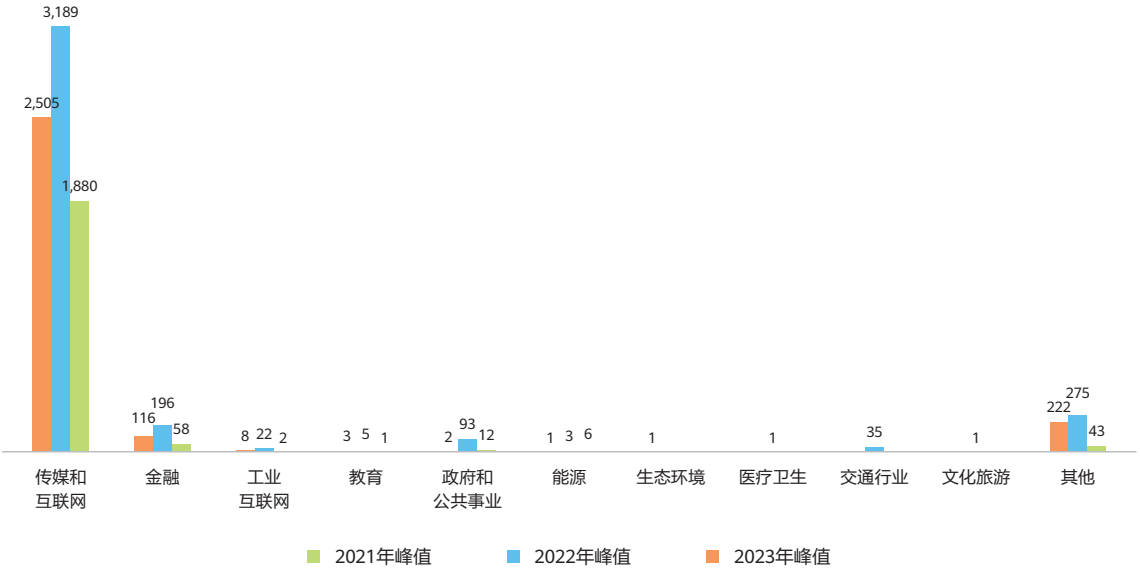
2021-2023年中国金融行业攻击频次月度分布



数据来源于联通数科

2021-2023年行业最大攻击峰值带宽分布显示，竞争激烈的传媒与互联网行业受攻击强度远大于其他行业，最大攻击峰值带宽维持在T级；金融行业受攻击强度一直居高不下，近两年攻击峰值带宽均超过100Gbps，远超金融数据中心实际网络链路带宽。

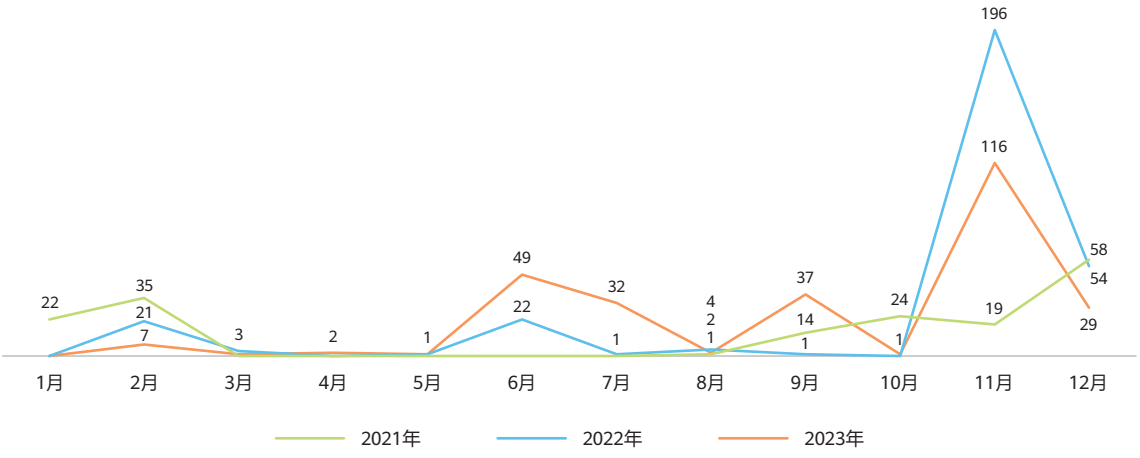
2021-2023年行业攻击强度分布（Gbps）



数据来源于电信安全&联通数科&华为

2023年中国金融行业遭受的最大攻击峰值带宽为116Gbps，发生在11月，采用混合攻击，主要包括NTP反射、DNS反射和UDP Flood，攻击持续19分钟，攻击目标是门户网站。

2021-2023年中国金融行业攻击峰值带宽月度分布（Gbps）

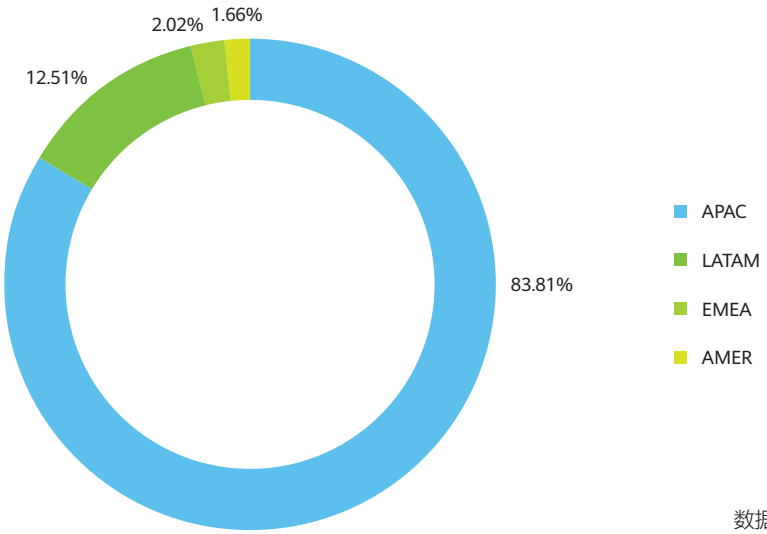


数据来源于联通数科

2.1.9 攻击目标地域分布

2023年攻击目标大洲分布显示，攻击目标主要集中在APAC，占比83.81%。主要源于APAC大部分地域带宽资源较昂贵，扫段攻击威胁聚集，直接拉升了APAC攻击目标地域占比。

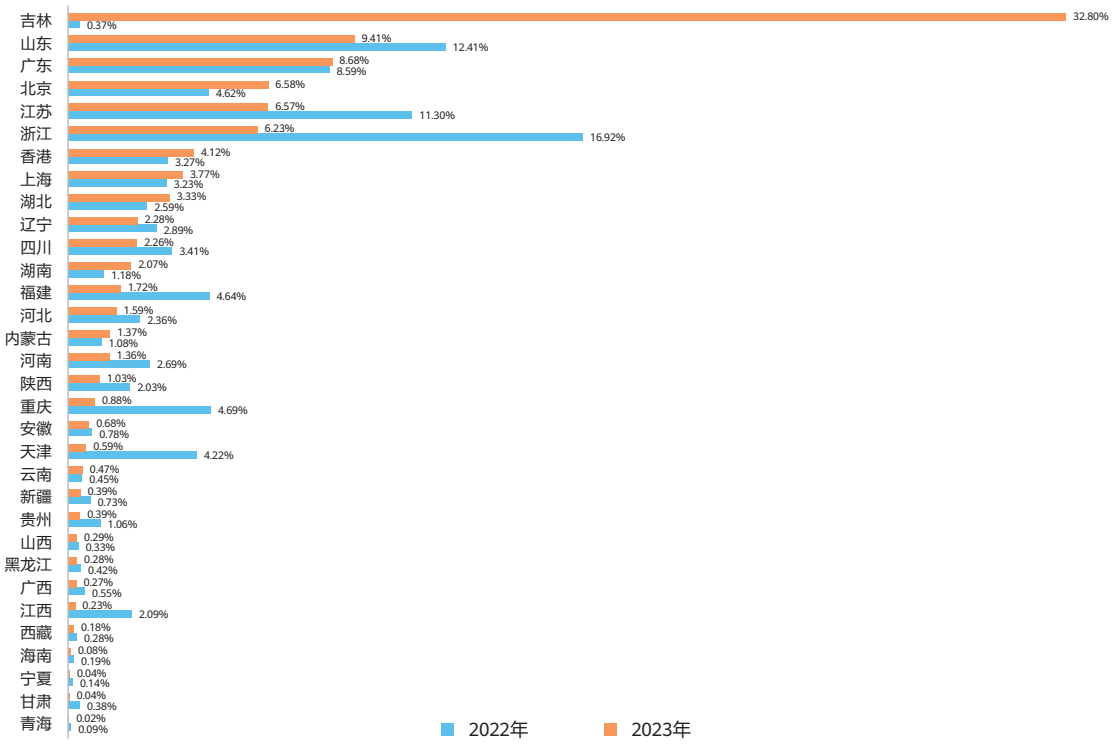
2023年攻击目标大洲分布



数据来源于Nexusguard

2023年，中国TOP3攻击目标地域分布为吉林、山东和广东。和2022年相比，攻击目标中国地域分布突变源于扫段攻击在某些地域较聚集。

2022-2023年攻击目标中国地域分布



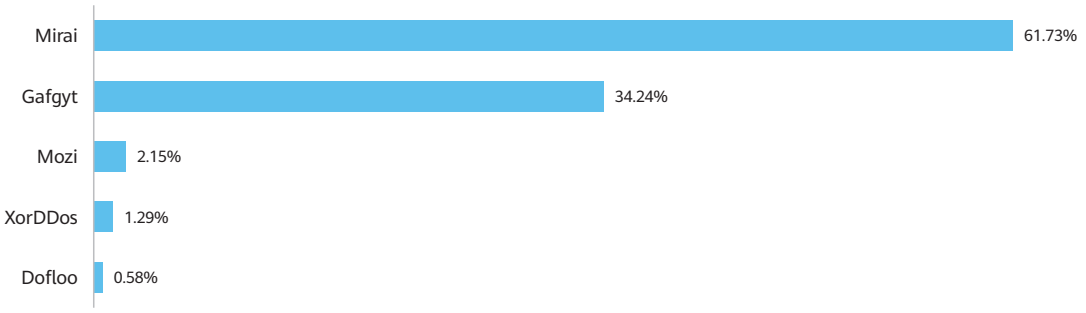
数据来源于电信安全&联通数科&华为

2.2 DDoS僵尸网络态势

2.2.1 僵尸家族分布

DDoS僵尸家族以IoT和Linux为主。2023年，按活跃C2数量排名的TOP5僵尸家族分别是Mirai、Gafgyt、Mozi、XorDDoS和Dofloo，其中Mirai、Gafgyt和Mozi是典型的IoT僵尸网络，而XorDDoS和Dofloo是典型的Linux僵尸网络。

2023年DDoS僵尸家族TOP5分布



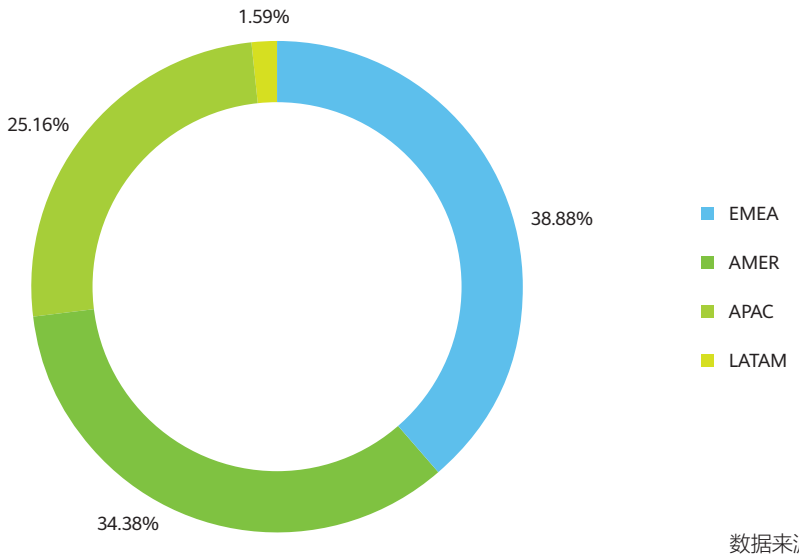
数据来源于百度安全&华为

需要说明的是Mozi僵尸网络从2023年8月份开始，活跃度陡然下降，疑似Mozi作者在执法部门的要求下发布了“自杀开关”[9]。

2.2.2 C2地域分布

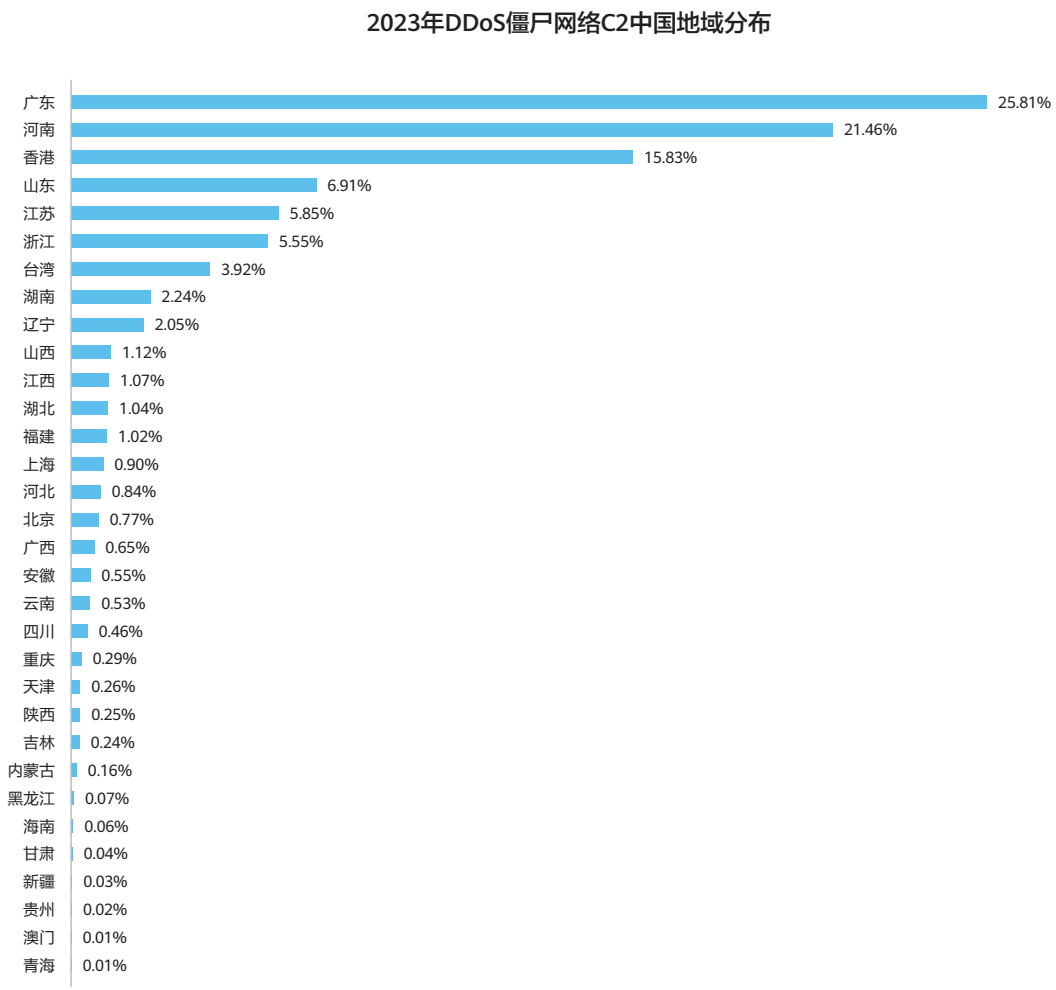
2023年DDoS僵尸网络C2海外地域分布显示，除LATAM外，C2在EMEA、AMER和APAC分布较均匀。

2023年DDoS僵尸网络C2海外地域分布



数据来源于百度安全&华为

2023年，DDoS僵尸网络C2中国TOP3地域分布依次为广东、河南和香港。

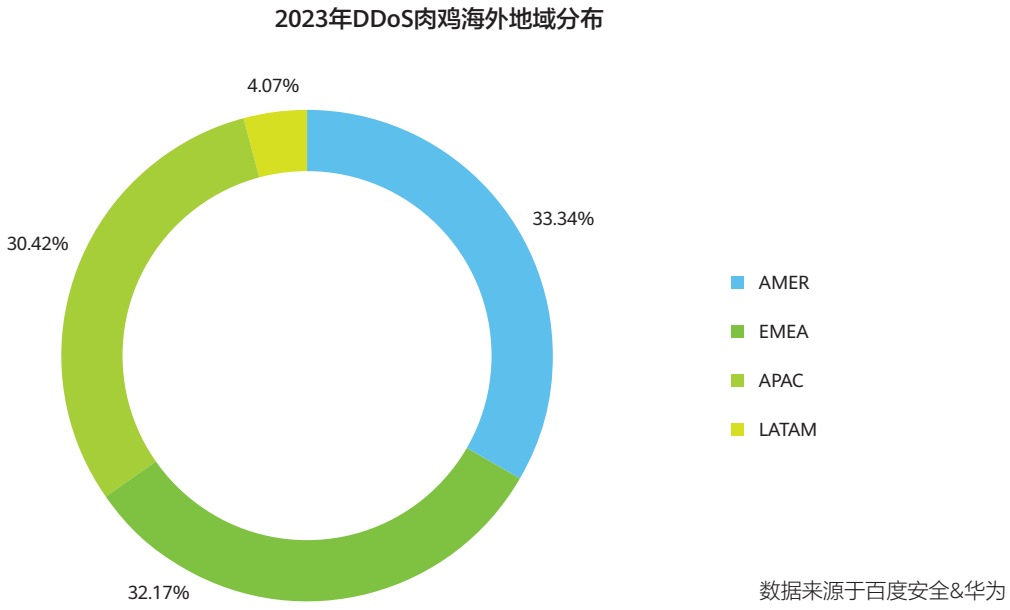


数据来源于百度安全&华为

2.3 DDoS攻击源态势

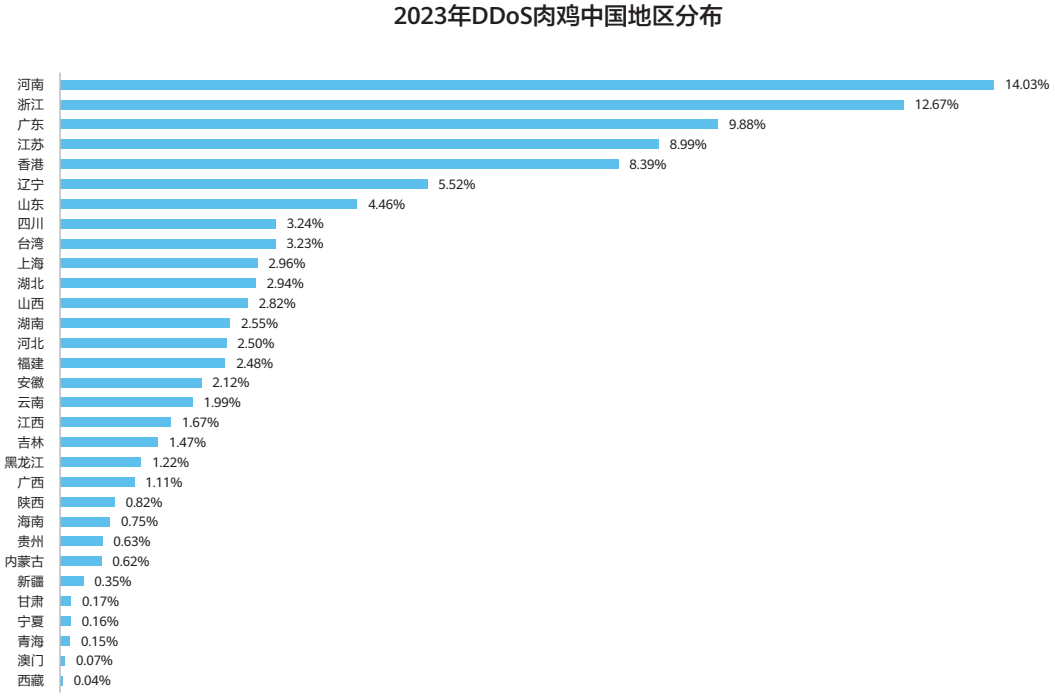
2.3.1 肉鸡地域分布

2023年，DDoS肉鸡海外地域分布显示，除LATAM外，肉鸡在AMER、EMEA和APAC分布较均匀。



数据来源于百度安全&华为

肉鸡中国TOP3地域分布为河南、浙江和广东。

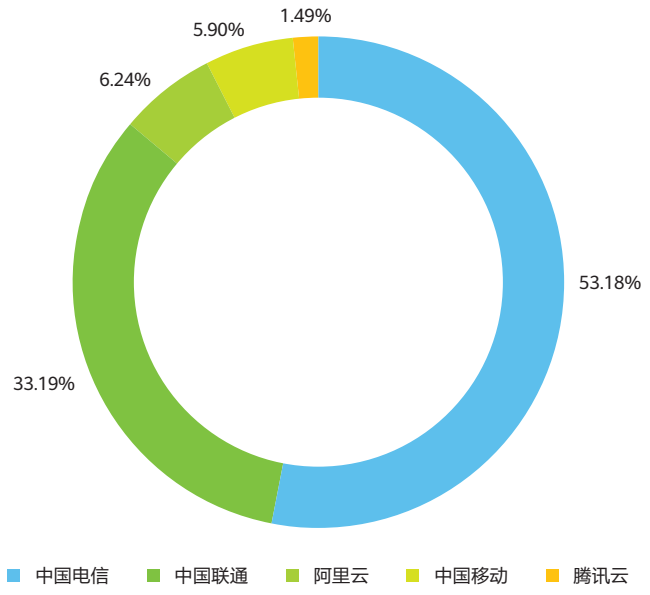


数据来源于百度安全&华为

2.3.2 肉鸡运营商/服务提供商分布

DDoS肉鸡中国TOP5运营商/服务提供商分布依次为中国电信、中国联通、阿里云、中国移动和腾讯云。

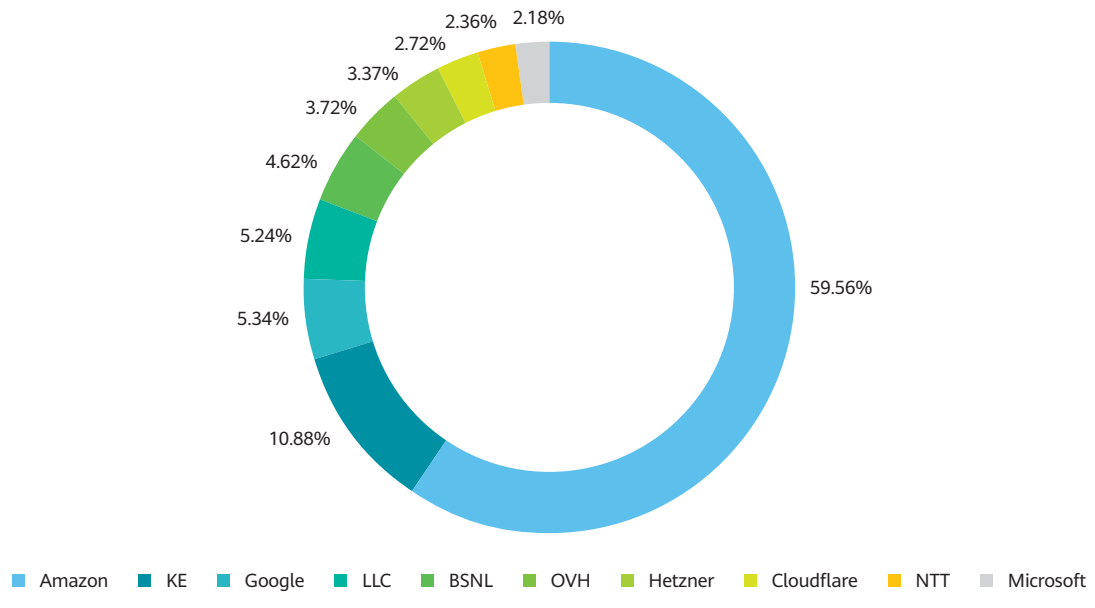
2023年DDoS肉鸡中国TOP5运营商分布



数据来源于百度安全&华为

DDoS肉鸡海外TOP3运营商分布依次为Amazon、KE和Google。公有云肉鸡分布占比较高，说明公有云提供商聚焦云基础设施安全，公有云和租户对云主机自身安全性投入不足。

2023年DDoS肉鸡海外TOP10运营商分布



数据来源于百度安全&华为

03 典型DDoS攻击分析

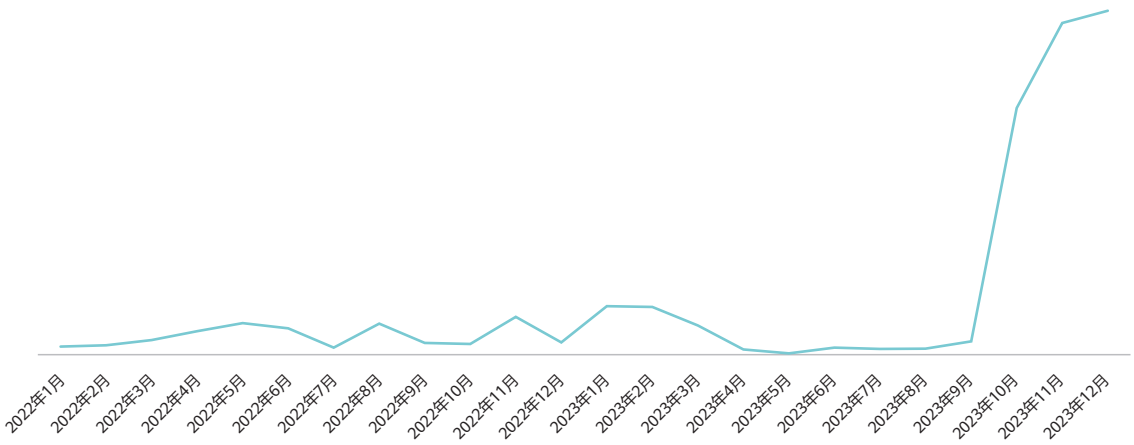
3.1 扫段攻击

扫段攻击威胁范围广，成为攻击网络基础设施的惯用手段。

3.1.1 扫段攻击频次快速增长

2023年H2，扫段攻击频次激增。

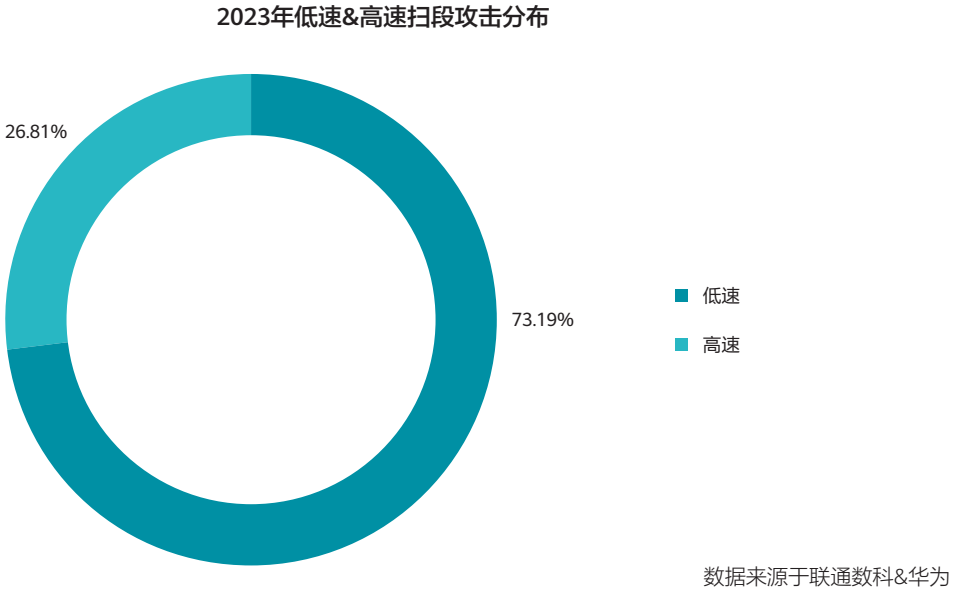
扫段攻击频次快速增长



数据来源于电信安全

3.1.2 低速扫段攻击难检测

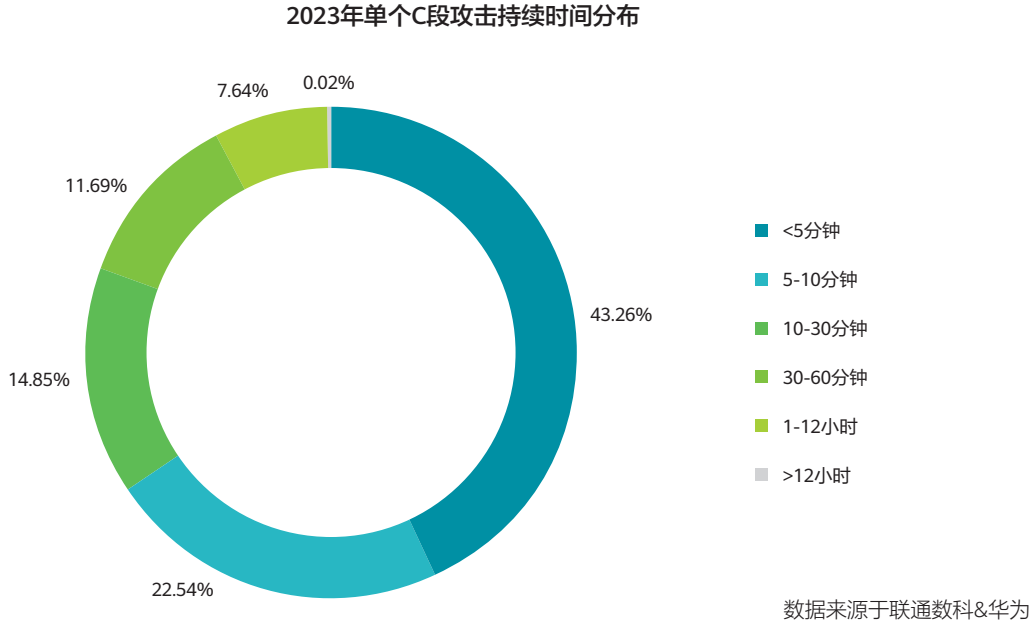
低速扫段攻击单IP流量低，无法触发主机检测阈值，躲避能力强。



数据来源于联通数科&华为

2023年，73.19%的扫段攻击呈现低速态势，挑战传统检测算法有效性。

3.1.3 惯用“短平快”战术，挑战防御系统响应速度

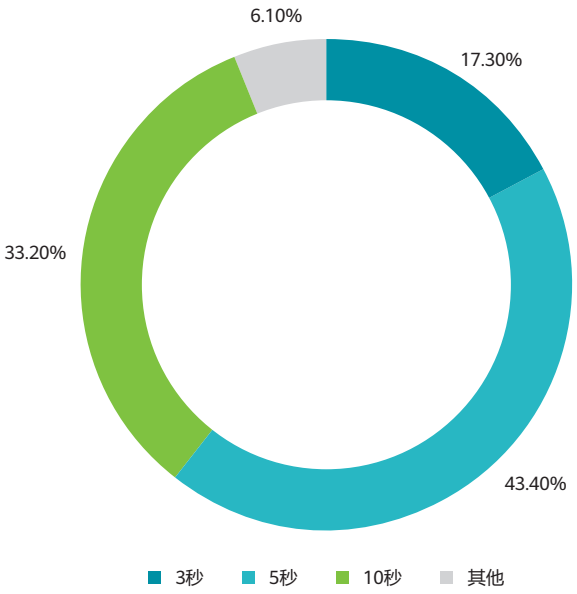


数据来源于联通数科&华为

扫段攻击中，43.26%的C段攻击持续时间小于5分钟，挑战防御系统的响应速度。

典型 DDoS 攻击分析

2023年扫段攻击单个目标IP攻击持续时间分布

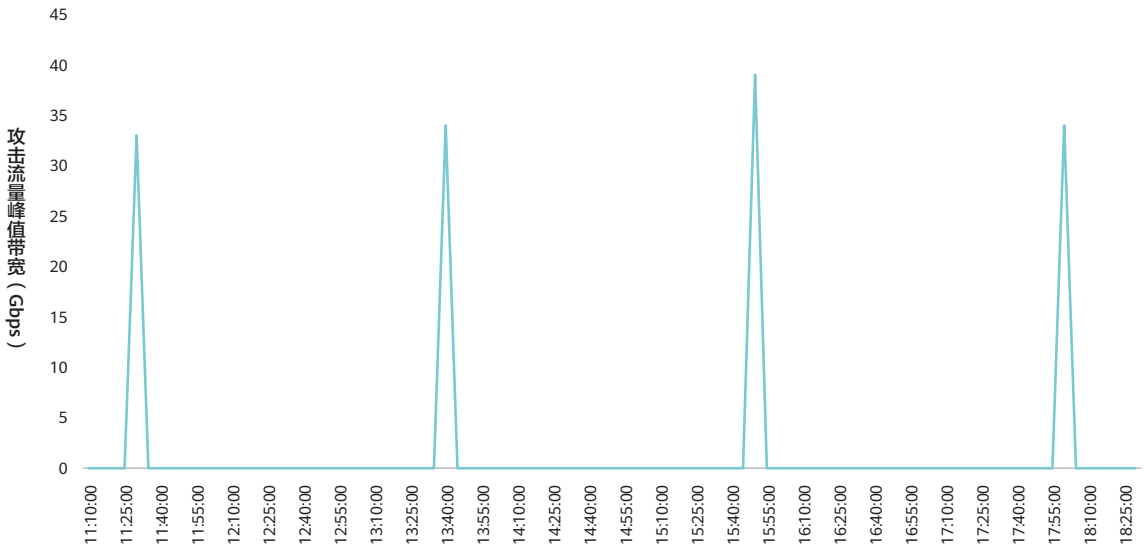


数据来源于百度

当一个C段攻击持续时间小于5分钟时，93.90%的单个目标IP的攻击持续时间不超过10秒，即使基于主机防御的抗D系统能检测到攻击，检测时延、引流时延也会造成大量攻击报文漏防。

当针对单个C段的攻击持续时间较长且采用高速扫段时，攻击者会结合“脉冲”攻击手法，就单个目标IP而言，攻击呈现出典型的“脉冲”特点，挑战传统主机防御系统的响应速度。

典型高速扫段攻击单个攻击目标IP流量呈现“脉冲”波形

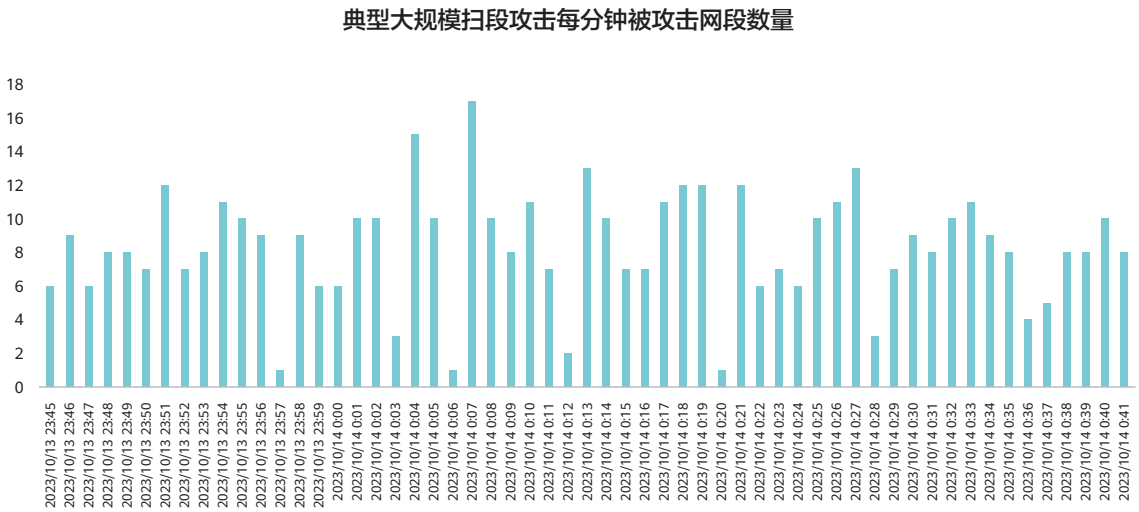


数据来源于华为

3.1.4 攻击手法复杂，难防御

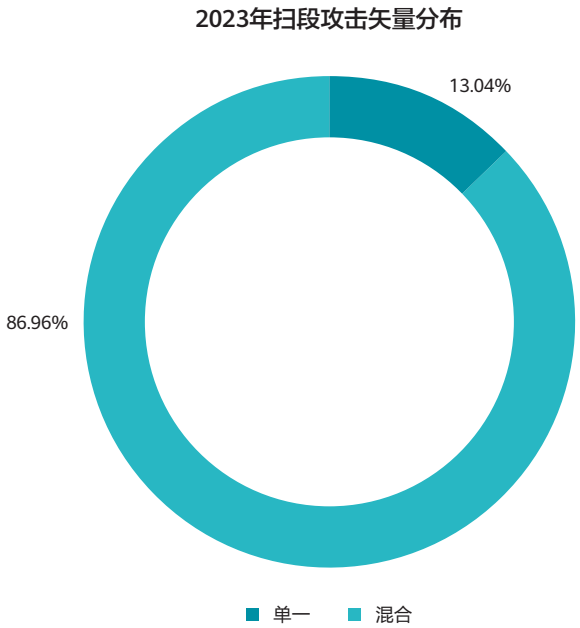
1. 大规模扫段，挑战并发防御规格

大规模扫段攻击发生时，每分钟多个C段被同时攻击，当采用传统的主机防御时，并发防御规格不足。



数据来源于联通数科&华为

2. 采用混合攻击手法，提升防御难度



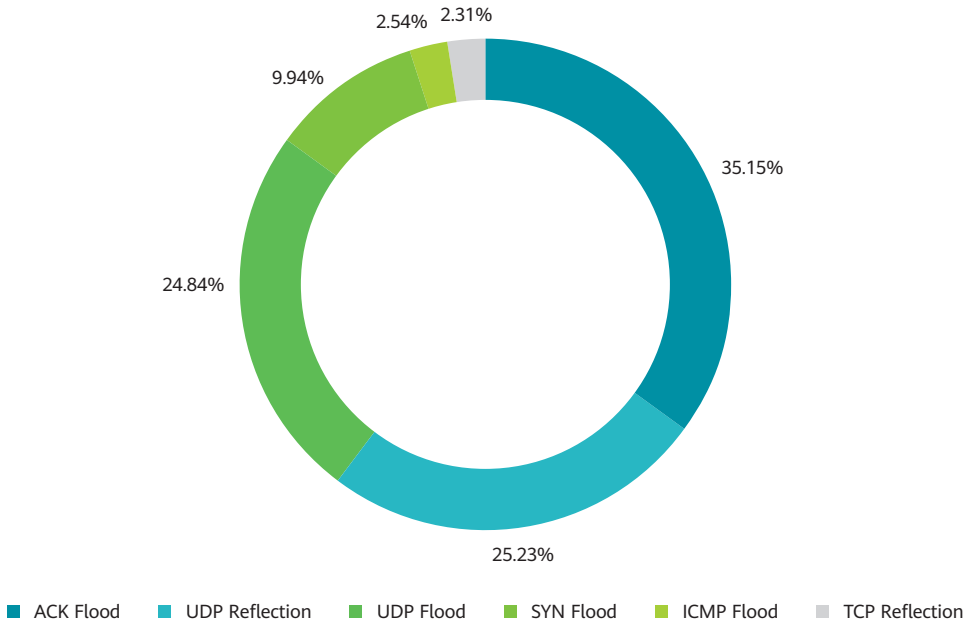
数据来源于联通数科&华为

86.96%的扫段攻击采用混合攻击手法，即一次攻击事件采用多种攻击类型，提升防御难度。

典型 DDoS 攻击分析

攻击者通过使用反射攻击提升扫段攻击对被攻击企业的带宽拥塞威胁，通过使用虚假源泛洪攻击加大防御系统的防御难度。

2023年扫段攻击类型分布



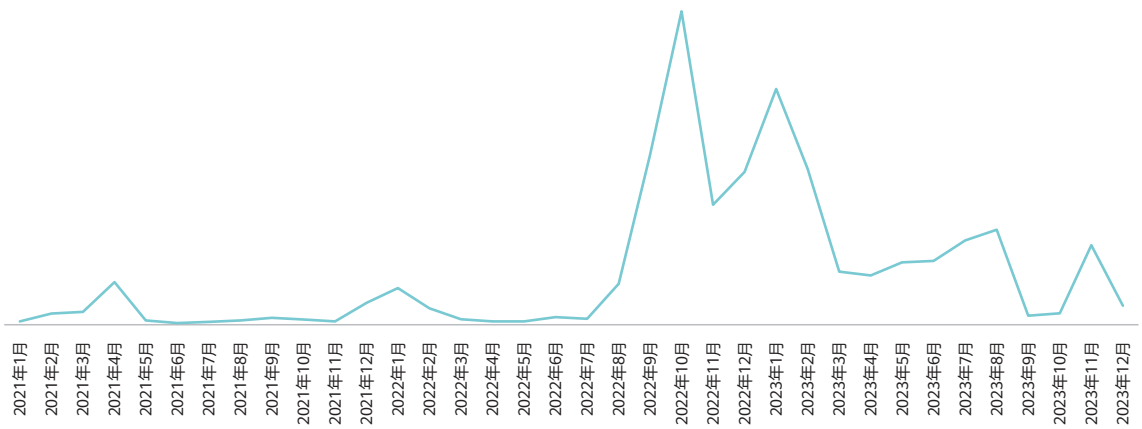
数据来源于联通数科&华为

3.2 DNS攻击

3.2.1 DNS攻击频次快速增长

DNS作为重要的网络基础设施，是DDoS攻击重要目标。对比近三年DNS攻击频次统计数据发现，2022年8月开始，针对DNS的攻击开始活跃，2022年10月达到顶峰，2023年3月份开始攻击活跃度有所下降，但相比2021年，2023全年依然处于活跃态势。

2021-2023年DNS攻击频次趋势

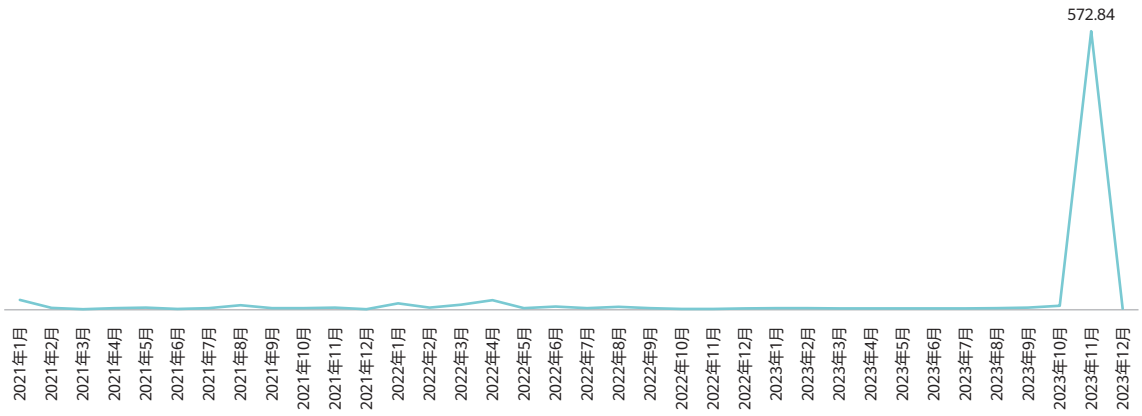


数据来源于电信安全&百度安全

3.2.2 DNS攻击强度迅猛攀升至亿次QPS级别

2023年DNS攻击峰值QPS从百万次级别快速提升至亿次级别。2023年11月份，百度监测到攻击流量峰值速率是572.84Mqps，攻击威胁攀升。

2021-2023年DNS攻击峰值QPS趋势（Mqps）

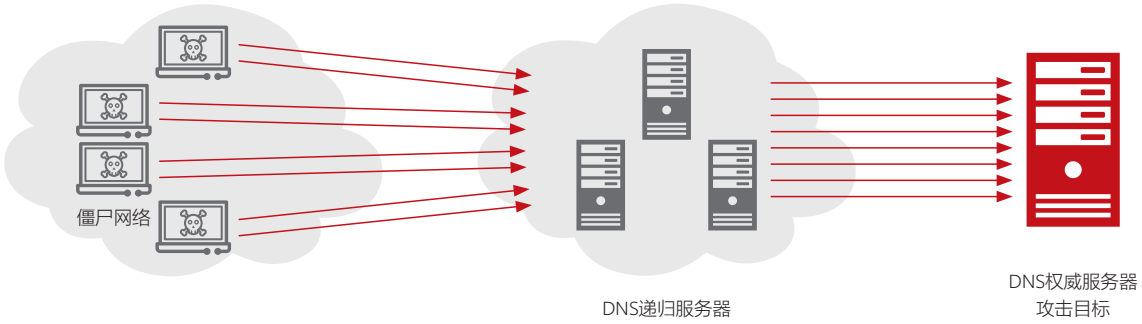


数据来源于电信安全&百度安全

3.2.3 DNS攻击复杂度再创新高

1. 攻击透过递归服务器攻击权威服务器

透过递归服务器攻击权威服务器



2023年11月3日，华为监测到东欧某国DNS权威服务器遭受大规模NXDomain攻击，超80%的攻击流量通过DNS递归服务器发起，导致传统防御算法失效。

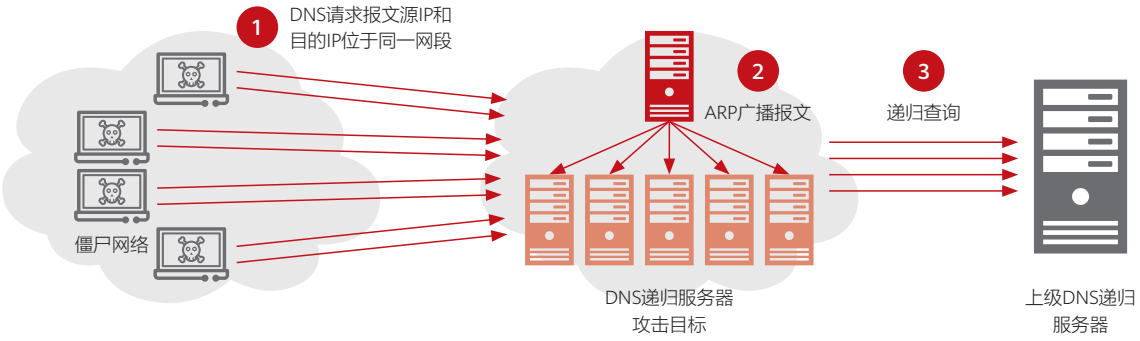
攻击抓包如下图所示：

透过递归服务器攻击权威服务器的攻击抓包

Time	Source	Destination	Protocol	Sport	Info
2023-11-03 07:04:51.101000	.28.50.238	.201	DNS	58723	Standard query 0x9744 A nt-mobile.new. OPT
2023-11-03 07:04:51.101000	.28.50.238	.201	DNS	57016	Standard query 0x1585 A mailermobil.belarusfacts. OPT
2023-11-03 07:04:51.102000	.28.50.238	.201	DNS	18805	Standard query 0xf67b A cctvold.new. OPT
2023-11-03 07:04:51.103000	.28.50.238	.201	DNS	4386	Standard query 0xe835 A im-nat.new. OPT
2023-11-03 07:04:51.118000	.28.50.238	.201	DNS	15862	Standard query 0xdb2e A rtmp.mobile.new. OPT
2023-11-03 07:04:51.119000	.28.50.238	.201	DNS	34292	Standard query 0x4c35 A css.lp.new. OPT
2023-11-03 07:04:51.215000	.28.50.238	.201	DNS	17409	Standard query 0x5eb4 A idpvle.new. OPT
2023-11-03 07:04:51.249000	.232.160.51	.201	DNS	22847	Standard query 0xe059 A movil-sa.belarusfacts. OPT
2023-11-03 07:04:51.249000	.232.160.51	.201	DNS	53626	Standard query 0xc1af A switzerland.media.belarusfacts. OPT
2023-11-03 07:04:51.249000	.232.160.51	.201	DNS	49781	Standard query 0x971d A netscalerfrontpage.new. OPT
2023-11-03 07:04:51.254000	.232.160.51	.201	DNS	26407	Standard query 0xe66f A mineRvaORg.belArUSfaCTS. OPT
2023-11-03 07:04:51.255000	.232.160.51	.201	DNS	46443	Standard query 0xc2d8 A NaGiOs.mercury.bElaruSfACTS. OPT
2023-11-03 07:04:51.255000	.232.160.51	.201	DNS	7328	Standard query 0x3436 A lax2-testing2.new. OPT
2023-11-03 07:04:51.255000	.232.160.51	.201	DNS	47242	Standard query 0x7eac A pl-jupiter.new. OPT
2023-11-03 07:04:51.256000	.232.160.51	.201	DNS	56190	Standard query 0x1229 A neon-addon.belarusfacts. OPT
2023-11-03 07:04:51.256000	.232.160.51	.201	DNS	46505	Standard query 0xa0d7 A idpcanada.new. OPT
2023-11-03 07:04:51.257000	.232.160.51	.201	DNS	58677	Standard query 0xeaeb A calendar-nt.new. OPT
2023-11-03 07:04:51.257000	.232.160.51	.201	DNS	8263	Standard query 0x7079 A neTwoRK-IpSp.neW. OPT
2023-11-03 07:04:51.258000	.232.160.51	.201	DNS	39504	Standard query 0xf5ec A nEtWoRk-IPsp.neW. OPT
2023-11-03 07:04:51.258000	.232.160.51	.201	DNS	42489	Standard query 0xb216 A MInERvaORG.BelaRUsfACtS. OPT
2023-11-03 07:04:51.263000	.232.160.51	.201	DNS	26780	Standard query 0x8a95 A dev4.lab.belarusfacts. OPT
2023-11-03 07:04:51.263000	.232.160.51	.201	DNS	53969	Standard query 0x41b5 A merchcloudflare-resolve-to.belarusfacts.
2023-11-03 07:04:51.264000	.232.160.51	.201	DNS	23080	Standard query 0xd12f A archive-on.new. OPT
2023-11-03 07:04:51.264000	.232.160.51	.201	DNS	62535	Standard query 0x3162 A statskorea.belarusfacts. OPT
2023-11-03 07:04:51.265000	.232.160.51	.201	DNS	47618	Standard query 0x799c A SLovakIanet.new. OPT
2023-11-03 07:04:51.265000	.232.160.51	.201	DNS	49940	Standard query 0xbbca A fmb.belarusfacts. OPT
2023-11-03 07:04:51.266000	.232.160.51	.201	DNS	20305	Standard query 0x738c A linuxgh.new. OPT

2. 用源IP和目的IP位于同一网段的请求报文攻击递归服务器

源IP和目的IP位于同一网段的NXDomain攻击



2023年7月初，华为监测到国内某运营商DNS递归服务器连续遭受DNS Query Flood攻击，攻击报文源IP和目的IP位于同一网段。递归服务器短时间内收到大量固定域名请求及NXDomain请求，消耗服务器性能；DNS请求报文的源IP和目的IP位于同一网段，引发大量ARP广播报文；NXDomain攻击导致递归服务器短时间内产生大量递归查询请求。

攻击抓包如下图所示：

源IP和目的IP位于同一网段的NXDomain攻击抓包

Time	Source	Destination	Protocol	Sport	Info
2023-07-25 17:12:39.777000	.238.58	.238.58	DNS	29575	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:39.778000	.238.146	.238.58	DNS	26796	Standard query 0xaa02 NS crf819.com
2023-07-25 17:12:39.779000	.238.92	.238.58	DNS	24238	Standard query 0xaa02 NS crf819.com
2023-07-25 17:12:39.779000	.238.12	.238.58	DNS	23155	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:39.780000	.238.110	.238.58	DNS	16877	Standard query 0xaa02 NS crf819.com
2023-07-25 17:12:39.780000	.238.228	.238.58	DNS	33896	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:39.781000	.238.12	.238.58	DNS	33635	Standard query 0x57f4 NS hziy6d.fuc697.com
2023-07-25 17:12:40.182000	.238.229	.238.58	DNS	26900	Standard query 0x57f4 NS cfusfa.fuc697.com
2023-07-25 17:12:40.183000	.238.199	.238.58	DNS	7931	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:40.183000	.238.149	.238.58	DNS	10920	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:40.184000	.238.223	.238.58	DNS	22236	Standard query 0x57f4 NS 8299vm.fuc697.com
2023-07-25 17:12:40.184000	.238.31	.238.58	DNS	25965	Standard query 0x57f4 NS 2jfj0v.fuc697.com
2023-07-25 17:12:40.185000	.238.37	.238.58	DNS	3624	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:40.185000	.238.171	.238.58	DNS	7693	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:40.186000	.238.227	.238.58	DNS	9596	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:40.187000	.238.17	.238.58	DNS	22193	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:40.187000	.238.111	.238.58	DNS	10180	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:40.188000	.238.117	.238.58	DNS	6888	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:40.188000	.238.129	.238.58	DNS	9287	Standard query 0xaa02 NS crf819.com
2023-07-25 17:12:45.099000	.238.84	.238.58	DNS	3797	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:45.100000	.238.26	.238.58	DNS	3788	Standard query 0x57f4 NS y19zov.fuc697.com
2023-07-25 17:12:45.100000	.238.220	.238.58	DNS	12515	Standard query 0x57f4 NS od01f7.fuc697.com
2023-07-25 17:12:45.101000	.238.188	.238.58	DNS	31723	Standard query 0x01c0 NS hwk835.com
2023-07-25 17:12:45.102000	.238.4	.238.58	DNS	8175	Standard query 0x57f4 NS fnx1ms.fuc697.com

04 专家观点

观点1：瞬时泛洪攻击秒级加速，挑战防御系统的响应速度。需探索更为高效的检测和清洗技术。例如设备厂商研制高效随路检测路由器，运营商研发端网协同防御技术，以有效缩短TTM（Time to Mitigation）。

瞬时泛洪攻击的规模和复杂性逐年增加，其秒级加速的特性对基于传统Flow检测的防御系统构成了重大挑战，Flow检测延迟直接导致TTM大于1分钟，不仅增加了网络的脆弱性，而且在攻击达到峰值前，防御系统很难采取有效措施，从而导致服务中断等安全风险。

为解决这类攻击问题，一种可行的解决方案是引入路由器随路检测技术。通过在路由器上基于包检测机制，可秒级发现异常流量。在算力及存储资源有限的情况下，设计高效的流量统计分析算法，以应对大规模攻击流量的检测和清洗时效性需求。

为有效缩短TTM，除了在骨干网中部署随路检测路由器外，还可以采用端网协同技术。结合企业网络端点设备的细粒度数据反馈、网络设备的实时监控和处置能力，为瞬时泛洪攻击提供全面的处置视角。通过多源数据集成和多方协同，提高对大规模突发流量的识别精度和速度，实现瞬时泛洪攻击的秒级响应。

观点2：高速加密攻击挑战解密防御性能，低速CC攻击绕过WAF，挑战防御系统有效性。利用行为分析算法拦截高速CC，机器学习算法精准识别低速CC，分而治之，有效应对复杂攻击。

日趋复杂的应用层CC攻击已经成为防御系统面临的一项严峻挑战，目前CC攻击分化为两大方向：高速攻击对解密防御性能提出了巨大挑战，而低速攻击则躲避检测能力强对防御系统的有效性构成了严重威胁。高速攻击通常利用高性能服务器或云主机构建的僵尸网络，基于HTTP协议的高速传输实现千万级甚至亿次级RPS攻击速率；而低速攻击则利用现有防御系统的算法漏洞，绕过源速率检测或内容过滤，对目标系统发动持续性攻击，耗尽目标资源。

在防御资源有限的情况下，传统的防御方法难以有效识别这两类攻击。学术界和工业界的广泛尝试已经表明，行为分析算法能够在不解密的情况下识别高速攻击，而机器学习算法则显示出对低速攻击的精准识别能力。综合应用这两种算法有望有效地应对复杂CC攻击。强对抗型CC攻击威胁不断升级，迫切需要研究软硬件结合的新型防御设备，以及如何将新的防御方法应用到现有的防御系统中，确保与其他防御机制高效协同工作。

观点3：扫段攻击成为网络基础设施面临的最大威胁，需采取多种措施增强防御。增加网段检测能力提升攻击识别精准度，端网协同防御提高多网段攻击的发现及处置效率，有效应对大规模扫段攻击。

2023年扫段攻击规模、频次和复杂程度进一步攀升，成为网络基础设施最大威胁。扫段攻击通过将攻击流量分散在受害者的大量地址中，旨在绕过检测，挑战防御系统响应速度和处置规模。为了有效应对扫段攻击，必须采取多种措施增强防御系统的攻击清洗能力。

首先，创新扫段攻击检测技术，以提升攻击检测灵敏度。目前，DDoS攻击检测算法主要基于单个攻击目标IP的流量统计进行攻击判定。然而，低速扫段攻击通过尽可能分散流量，使单个攻击目标IP受到的攻击流量较低，无法触发检测阈值，从而降低检测算法灵敏度。因此，需创新检测算法，在现有方案的基础上增加网段检测能力，以快速识别扫段攻击。其次，采用端云协同防御架构可以提高防御系统响应速度和并发防御规模。扫段攻击并发攻击的C段数量不断攀升，同时采用“短平快”战术，对防御系统响应速度和并发防御规模提出更高要求。然而运营商网络普遍采用的Flow检测存在分钟级延迟，导致扫段攻击发现慢。通过企业网络边界防御系统提供的秒级发现能力，利用端云协同防御，主动请求上游运营商清洗服务，最大限度提升防御系统响应速度，并最大限度利用运营商网络路由器自身过滤能力，结合运营商网络清洗资源池的清洗能力，实现大规模扫段攻击有效防御。

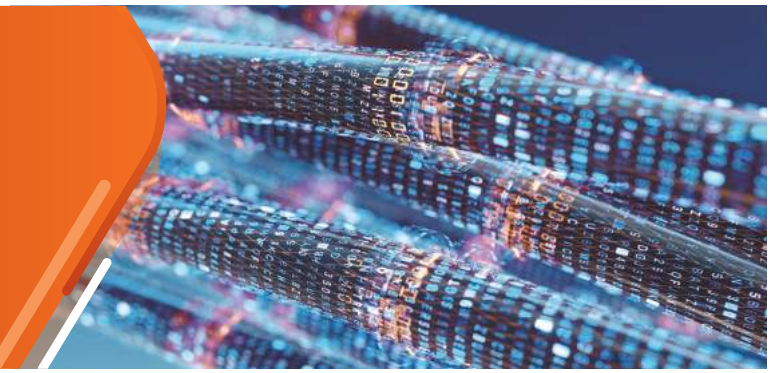
名词解释：

1. **瞬时泛洪攻击：**也叫 Fast Flooding，形容大流量攻击发生时如决堤的洪水一样倾泻而下，攻击流量断崖式上升，在几秒内即可达到几百 Gbps。
2. **TTM：**是 Time to Mitigation 的缩写，指从攻击开始到启动清洗需要的时长，用于描述防御系统对攻击响应的速度。
3. **“脉冲”攻击：**也叫 Pulse-wave，在攻击持续时间段内，攻击流量以相似的时间间隔反复冲高又迅速降落，且每次攻击流量冲高后形成的流量峰值相似，形成一个个规律的“脉冲”波形。
3. **网络层 CC：**主要包括真实源 SYN Flood、真实源 ACK Flood，多数情况下，网络层 CC 攻击指的是真实源 ACK Flood，即攻击者利用僵尸网络和被攻击目标服务器建立大量 TCP 连接后，不断发送垃圾 ACK Flood，以消耗服务器连接资源或带宽资源。
4. **应用层 CC：**难防御的 HTTP、HTTPS 应用层攻击被统称为应用层 CC，即攻击者利用僵尸网络和被攻击目标服务器建立 TCP 连接，对目标服务器发起应用层请求，以消耗服务器资源甚至是网络链路带宽资源，按应用协议不同又可分为 HTTP CC、HTTPS CC 等。
5. **低速 CC：**一般采用低速 CC 攻击时，僵尸多为低性能的 IoT 终端，僵尸数量较多，单个僵尸的攻击速率较低，企图绕过安全系统基于源请求速率的检测，挑战防御系统的检测灵敏度。
6. **高速 CC：**一般采用高速 CC 攻击时，僵尸多为高性能的服务器或云主机，僵尸数量较少，单个僵尸的攻击速率较高，挑战防御系统的响应速度。
7. **加密 CC：**属于应用层 CC 范畴，即加密的应用层 CC。
8. **扫段攻击：**也叫 Carpet bombing attack，属于一种新型攻击，攻击目标不再是单个服务器 IP 地址，同时针对 1 个或者多个 C 类 IP 地址段内多个 IP 地址进行攻击。
8. **低速扫段：**到攻击目标 C 段内的单个 IP 的攻击流量较小，挑战防御系统检测灵敏度。但因被攻击的 C 段数量多，同时被攻击的 IP 地址数量庞大，导致整体的攻击流量较大，挤占被攻击网络带宽或消耗网络会话资源，达到 DDoS 攻击效果。
9. **高速扫段：**相对低速扫段，到攻击目标 C 段内的单个 IP 的攻击流量较大，基于目的 IP 的攻击可以发现攻击，但因同时被攻击的 IP 数量多，挑战防御系统并发主机防御规格，同样能达到 DDoS 攻击效果。
10. **攻击矢量：**用于描述一次攻击事件采用几种攻击类型，采用一种攻击类型的攻击也叫 single-vector attack，采用多种攻击类型的攻击也叫 Multi-vector attack。
11. **混合攻击：**也叫 Multi-vector attack，即一次攻击事件采用多种攻击类型。

引用：

1. https://cloud.google.com/blog/products/identity-security/google-cloud-mitigated-largest-ddos-attack-peaking-above-398-million-rps
2. https://www.zdnet.com/article/aws-said-it-mitigated-a-2-3-tbps-ddos-attack-the-largest-ever/
3. https://securityaffairs.com/127279/cyber-crime/record-ddos-attack-azure.html
4. https://www.bleepingcomputer.com/news/security/cloudflare-blocks-record-breaking-71-million-rps-ddos-attack/
5. https://cloud.google.com/blog/products/identity-security/how-google-cloud-blocked-largest-layer-7-ddos-attack-at-46-million-rps
6. https://cloud.google.com/blog/products/identity-security/how-it-works-the-novel-http2-rapid-reset-ddos-attack
7. https://www.cve.org/CVERecord?id=CVE-2023-44487
8. https://github.com/BeichenDream/FakeToa
9. https://www.welivesecurity.com/en/eset-research/who-killed-mozi-finally-putting-the-iot-zombie-botnet-in-its-grave/

05 数据来源



5.1 数据来源

本报告中所涉及的数据来源于电信安全、联通数科安全、百度安全、Nexusguard、中国移动云能力中心、中国移动卓望公司、华为云及华为客户授权的DDoS攻击数据。在此特别鸣谢北京海御科技有限公司、杭州速联信息科技有限公司、金华唯安信息科技有限公司、宿迁蒲公英网络有限公司、杭州优云科技股份有限公司给与报告的数据支持。