

DDoS Test Results Analysis and Recommendations (#01 / Financial Corp)

Privacy Notice and Proprietary Information

This document contains confidential and proprietary information intended solely for Financial Corp (the "Customer"). Disclosing this information to any person or entity other than the Customer is strictly prohibited. Additionally, the content of this document shall not be transmitted, duplicated, or used for any purpose other than its intended use.

Any use or disclosure, in whole or in part, of the information contained in this document without explicit written permission from Red Button is strictly prohibited. Red Button does not provide any warranty that the information in this document is complete or free from errors.

This report is exclusively for the information of the Customer and the Customer's management. It should not be circulated, quoted, or referred to for any other purpose, nor should it be included or referenced, in whole or in part, in any document without prior written consent from Red Button.

Version History

Date	Version	Change	By
01/09/2023	1.00	Document creation	Red Button

Table of Contents

Executive Summary	4
Motivation	4
DDoS Resiliency Score	4
Industry Threat Level and Target DRS Score	4
Protection Status Analysis	6
Financial Corp Environment / Architecture	6
Current Protection State	6
Test Summary	8
Test Plan	9
Test Results	9
Attack Vectors Analysis Diagrams (AVADs)	10
Findings & Recommendations	11
Recommendations in Details	11
Testing Work Logs	13
Attack Vector #1	13
Attack Vector #2	15
Attack Vector #3.1	16
Attack Vector #3.2	17
Attack Vector #4	18
Attack Vector #5	20
Attack Vector #6.1	21
Attack Vector #6.2	22
APPENDIX A - GLOSSARY	23
APPENDIX B – Assets’ Testing Priority	24

Executive Summary

Red Button conducted a comprehensive Distributed Denial of Service (DDoS) attack testing on Financial Corp's infrastructure and services. The primary purpose of this testing was to assess and validate the organization's ability to withstand and mitigate potential DDoS attacks on its online assets. The evaluation focused on various aspects, including protective measures, configurations, detection mechanisms, procedures, and protocols.

The testing comprised eight attack scenarios, with varying degrees of severity. Out of these, six scenarios were successfully detected and mitigated by Financial Corp's defense systems, demonstrating the effectiveness of their security measures. However, one scenario was only partially mitigated, while another had no mitigation and significantly impacted the organization's services accessible over the internet.

Based on the test results, Red Button proposes the following recommendations to enhance Financial Corp's resilience against DDoS attacks:

1. **Investigate Optimizely's Failures:** A basic attack vector at a low rate caused a complete denial of service due to a sizing issue in the Optimizely system. To prevent such failures in the future, it is crucial to understand the source of this failure and address it accordingly.
2. **Investigate Imperva's Volumetric Attack Mitigation Failure:** Imperva's DDoS protection measures failed to block an advanced attack vector. It is highly recommended to investigate the reasons behind this failure in collaboration with Imperva and implement necessary improvements.
3. **Perform Additional DDoS Attack Simulations:** The testing focused on only two out of Financial Corp's five main assets vulnerable to DDoS attacks. To ensure comprehensive protection, additional testing should be carried out to test DDoS protection measures for the remaining assets.

Implementing these recommendations will align Financial Corp's DDoS attack resilience with the current threat level and substantially strengthen their security posture.

Motivation

The simulation was meticulously designed to assess the efficiency and effectiveness of Financial Corp's existing infrastructure, protection measures, personnel training, and protocols in detecting, mitigating, and recovering from DDoS attacks. The scenarios were carefully tailored to match the organization's existing infrastructure scope and known threat level and risks.

DDoS Resiliency Score

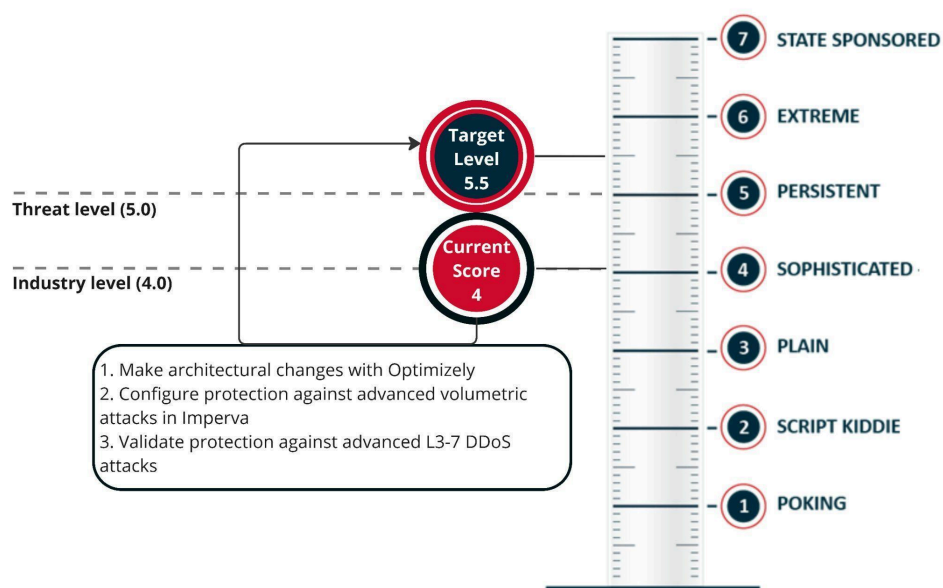
Red Button has developed the [DRS](#), an open standard, to quantitatively measure an organization's ability to withstand DDoS attacks. This standard considers the sophistication and volumes of potential attacks.

Industry Threat Level and Target DRS Score

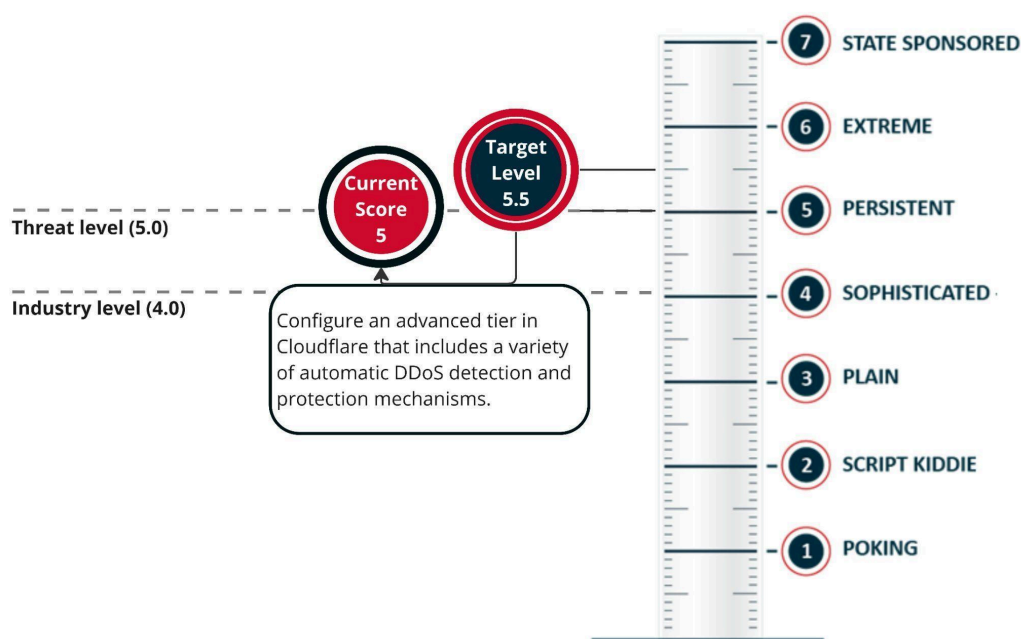
As a retail and commercial bank operating in Europe, Financial Corp is one of the largest banks in the region. European banks of this scale are generally protected at an average level of 4 (the 'industry level'). Simultaneously, the threat of DDoS attacks in the industry is ranked at 5 (the 'threat level').

DDoS simulation results for Financial Corp revealed DRS scores of 4 and 5, whereas the target DRS score is 5.5. To reach the target DRS score and bolster their security, Red Button strongly advises the organization to invest in fortifying its security measures.

Financial Corp main site (protected by Imperva)



Financial Corp members area (protected by Cloudflare)

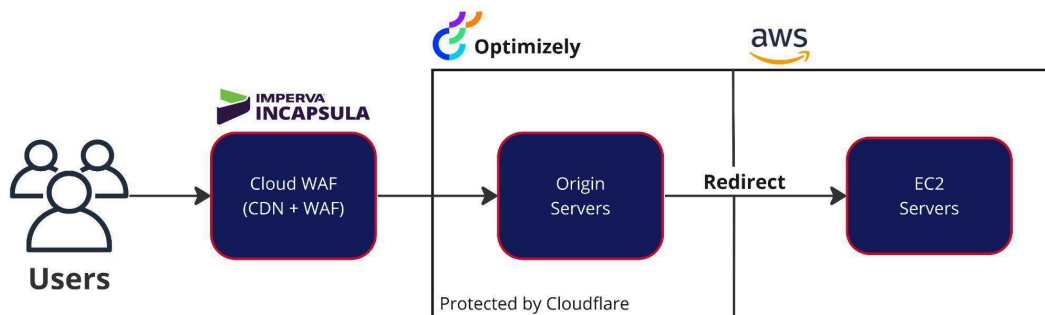


In conclusion, the DDoS attack testing conducted by Red Button has provided valuable insights into Financial Corp's security preparedness. By addressing the identified weaknesses and embracing the recommended measures, the organization can significantly enhance its ability to withstand DDoS attacks and ensure the uninterrupted delivery of services to its customers.

Protection Status Analysis

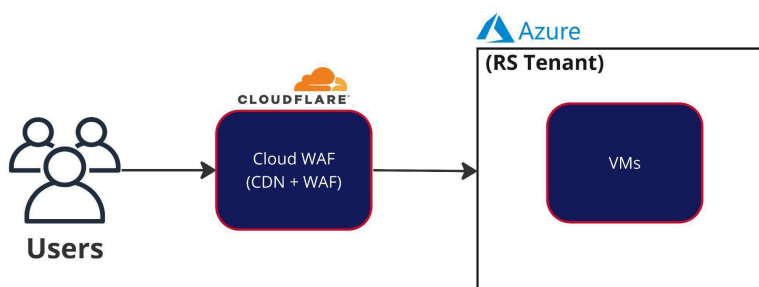
Financial Corp Environment / Architecture

<https://www.financialcorp.eu/>



The bank's main website is hosted on servers managed by Optimizely PaaS, with an added layer of Cloudflare protection, which operates as a black box to the customer, ensuring an extra level of security. Additionally, certain services on the website are stored on servers managed by AWS. To safeguard all incoming traffic directed towards the bank's website, it undergoes filtering and inspection through Imperva's cloud WAF protection, enhancing the overall security posture and defending against potential web-based threats.

<https://members.financialcorp.eu/>



The members area service is used by customers to manage loans and make payments. The service is hosted on servers located in Azure environment. All traffic directed to this service is filtered by Cloudflare's cloud WAF protection.

Current Protection State

Imperva DDoS protection:

Imperva provides Financial Corp a cloud web-application DDoS protection service to protect against network layer and application layer attacks, which includes two main components:

Imperva CDN (content delivery network)

The main purpose of CDN is to enhance the delivery of web content to end-users by reducing latency, optimizing performance, and offloading the load from content servers. CDNs store cached copies of content on many globally distributed servers. End-users who access a service are directed to a CDN Point of Presence that acts of behalf of the

service and serve the requests either from its servers' cache or by forwarding the request to the origin content servers. CDNs reduce the exposure of content servers on the internet and improve their overall security.

CDNs can act as a security layer, protecting the origin server from various online threats, including Distributed Denial of Service (DDoS) attacks. By filtering and absorbing malicious traffic at the CDN's edge servers, only legitimate traffic is forwarded to the origin server. This protects the origin server from being overwhelmed and helps maintain its availability during DDoS attacks.

Imperva cloud WAF (Web Application Firewall)

The main purpose of a cloud WAF is to provide robust security for web applications by identifying and mitigating various types of cyber threats. These threats can include SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and other application-layer attacks. The WAF uses a set of predefined and custom rules and policies to detect and block malicious traffic before it reaches the web application.

Cloud WAF solutions include Distributed Denial of Service (DDoS) protection features. They can detect and mitigate large-scale DDoS attacks that attempt to overwhelm a web application with a flood of traffic. By filtering out malicious traffic and allowing only legitimate requests, a Cloud WAF helps maintain the availability of web applications during DDoS attacks.

Imperva cloud WAF includes the following functionalities:

- Creating security rules and policies - Control over how traffic reaches the web applications by enabling the security team to configure rules that control bot traffic and block common attack patterns.
- Ability to create rate-based rules per IP address for actions like block or challenge.
- DDoS settings - Imperva cloud WAF includes automated reactions to DDoS attacks, which can be configured by the security team (e.g., what threshold of traffic will trigger a response and what action will be followed by all the website's clients - cookie or JS challenge, Captcha, etc.).
- Client classification - Imperva can filter bot activity, determine if it is benign or malicious, and only restrict hostile activity. This mechanism can be determined while configuring WAF rules.

Financial Corp current configuration is set to the defaults recommended by Imperva.

Cloudflare DDoS protection:

Financial Corp uses a cloud web application DDoS protection service, provided by Cloudflare, to protect against network and application layer attacks. The used protection measures consist of the following:

Cloudflare CDN

Automatic DDoS protection

Cloudflare automatically detects and mitigates DDoS attacks using its Autonomous Edge. The Autonomous Edge includes multiple dynamic mitigation rules known as Cloudflare DDoS protection managed rulesets. Mitigation rules can be customized to optimize and tailor protection to user needs.

Cloudflare protection provides an additional cost protection that may incur while under an attack.








Financial Corp current configuration is set to the defaults recommended by Cloudflare.

WAF (web application firewall)


Financial Corp current configuration is set to the defaults recommended by Cloudflare.

Test Summary

The table below details the attack vectors used in the DDoS testing, following the initial test plan. For each vector, the table indicates whether the attack was mitigated. For further details, please refer to the sections below.

Attack Vector	Destination	Max Rate	Expected Protection	Observed Protection	Test Results
https://www.financialcorp.eu/					
HTTPS GET / flood	https://www.financialcorp.eu/	50K RPS	DDoS settings (global rate limit)	DDoS settings (global rate limit)	
HTTPS POST flood	https://www.financialcorp.eu/banking/	50K RPS	DDoS settings (global rate limit)	DDoS settings (global rate limit)	
Large file download	https://www.financialcorp.eu/static/legacyJs/index.js	2 Gbps	CDN caching	CDN caching	
Large file download with randomized parameters	https://www.financialcorp.eu/static/legacyJs/index.js/?clientid=\$rand	30 Gbps	DDoS settings (Automatic DDoS rules)	None (Absorbed by Optimizely infrastructure)	
https://members.financialcorp.eu/					
HTTPS GET / flood on login	https://members.financialcorp.eu/login.aspx	50K RPS	Automatic HTTP DDoS	Automatic HTTP DDoS (JS challenge)	
HTTPS GET flood with randomized paths	https://members.financialcorp.eu/login.aspx/clientid=\$rand	50K RPS	Automatic HTTP DDoS	Automatic HTTP DDoS (302 redirect)	
Large file download	https://members.financialcorp.eu/bundles/miscmasterjs?v=k5C2xid7TOh2x1i87l5tDO0rcq1jEKjVDzd77b5Kurl1	6.5 Gbps	CDN caching	Automatic HTTP DDoS (403 error)	
Large file download with randomized parameters	https://members.financialcorp.eu/bundles/miscmasterjs?v=k5C2xid7TOh2x1i87l5tDO0rcq1jEKjVDzd77b5Kurl1/?clientid=\$rand	20 Gbps	Automatic HTTP DDoS	Automatic HTTP DDoS (302 redirect)	

Legend

	No detection No mitigation Severe impact		Partial detection Partial mitigation Severe impact		Detection Partial mitigation Limited impact		Detected & mitigated No impact on services
---	--	---	--	---	---	---	---

Test Plan

Red Button conducted a carefully tailored tiered simulation to evaluate the effectiveness of Financial Corp's defense plan, which relies on Imperva and Cloudflare DDoS protections for detection and mitigation. The test plan encompassed six application layer attack scenarios, with three targeting the protections provided by Imperva and the other three targeting those provided by Cloudflare. The attacks were launched against two critical areas: the Financial Corp main site, safeguarded by Imperva, and the Financial Corp member area, protected by Cloudflare.

Test Results

AV1 - HTTPS GET / flood on 'https://www.financialcorp.eu/'

The attack, launched at a rate of 800 RPS, caused the web servers managed by Optimizely cloud provider to reach 100% CPU utilization and become unresponsive. Within 17 minutes, the attack rate was increased to 17K RPS, breaching Imperva's global rate limit rule. Imperva detected and mitigated the attack, after the threshold was breached, leading to service recovery within 4 minutes. The service remained available until the test's end, even when the attack reached 50K RPS. An alert was generated when the attack rate breached Imperva's global rate limit threshold.

AV2 - HTTPS POST flood on 'https://www.financialcorp.eu/banking/'

The attack, launched at a rate of 500 RPS and gradually increased up to 50K RPS, had no observable impact, even below Imperva's global rate limit threshold. Upon breaching the threshold, Imperva effectively detected and mitigated the attack in accordance with its DDoS settings. An alert was issued when the attack rate breached Imperva's global rate limit threshold.

AV3.1 - Large file download on 'https://www.financialcorp.eu/static/legacyJs/index.js'

The attack, conducted with 265 bots at 1 Gbps, gradually increased to 2 Gbps. Imperva's CDN caching delivered the file to the bots, without affecting the origin server. No alerts were generated during this attack.

AV3.2 - Large file download with randomized parameters on 'https://www.financialcorp.eu/static/legacyJs/index.js?clientid=\$rand'

The attack, launched at 1 Gbps and increasing up to 30 Gbps, employed a randomized parameter, bypassing the CDN cache, and reaching the origin server. The website experienced some latency increase (maximum 5 seconds at 30 Gbps), but the attack was absorbed by the Optimizely network (on AWS infrastructure), and the service remained available. No alerts were generated during this attack.

AV4 - HTTPS GET / flood on 'https://members.financialcorp.eu/login.aspx'

The attack, initiated at 800 RPS and gradually increased to 50K RPS, remained undetectable on Cloudflare dashboard, and the service remained unaffected. However, after redirecting the attack target to 'https://members.financialcorp.eu/' at a rate of 33K RPS, Cloudflare detected and mitigated the attack using an HTTP automatic DDoS mitigation measure (JS challenge). An alert was sent after the attack target was changed.

AV5 - HTTPS GET flood with rand. paths on 'https://members.financialcorp.eu/login.aspx/clientid=\$rand'

The attack, launched at 10K RPS and increasing to 50K RPS, was successfully detected and mitigated by Cloudflare using an HTTP automatic DDoS mitigation measure (302 redirect to the login page served by the cache). An alert was issued regarding the attack.

AV6.1 - Large file download on 'https://members.financialcorp.eu/bundles/miscmasterjs?v=k5C2xid7TOh2x1i87l5tDO0rcq1jEKjVDzd77b5Kurl1'

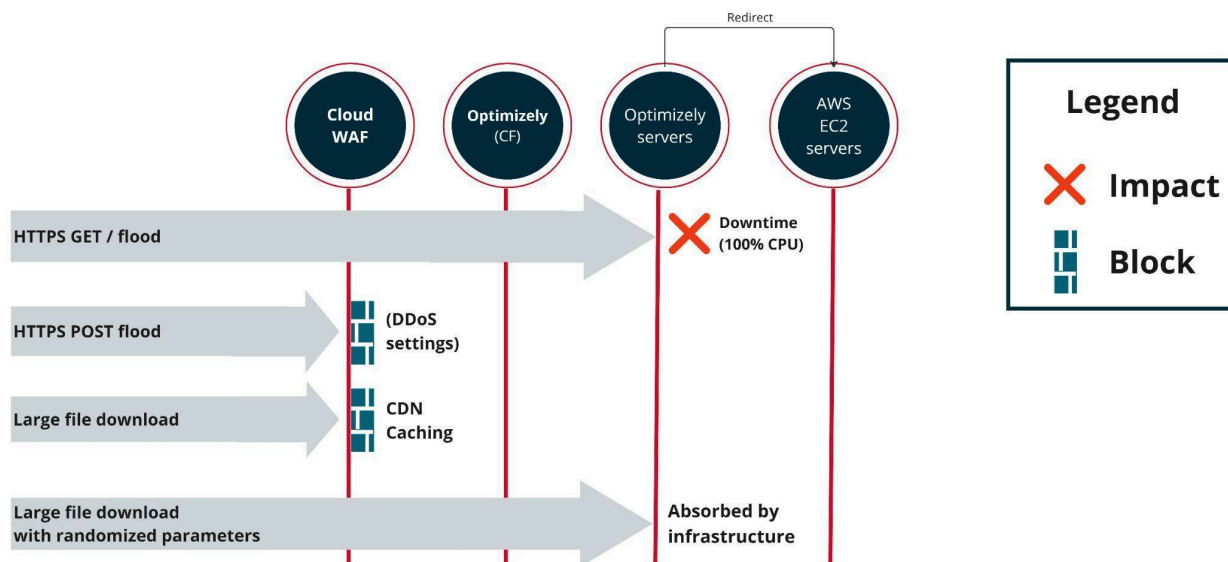
The attack, launched at 6.5 Gbps, with each bot sending 6 RPS to remain below the rate limit threshold, was detected and mitigated by Cloudflare using an HTTP automatic DDoS mitigation measure (403 error, behavioral DoS). An alert was sent concerning the attack.

AV6.2 - Large file download with randomized parameters on 'https://members.financialcorp.eu/bundles/miscmasterjs?v=k5C2xid7TOh2x1i87l5tDO0rcq1jEKjVDzd77b5Kurl1/?clientid=\$rand'

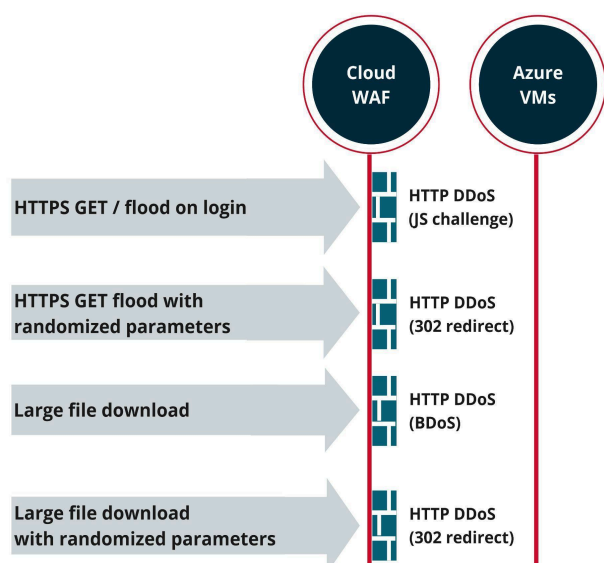
The attack, launched at 2 Gbps and gradually increased to 20 Gbps, was successfully detected and mitigated by Cloudflare using an HTTP automatic DDoS mitigation measure (302 redirect to the login page served by the cache). An alert was generated regarding the attack.

Attack Vectors Analysis Diagrams (AVADs)

<https://www.financialcorp.eu/> protected by **Imperva**.



<https://members.financialcorp.eu/> protected by **Cloudflare**.



Findings & Recommendations

Red Button's comprehensive testing and analysis have revealed certain gaps in the organization's DDoS readiness. To attain the desired levels of protection and readiness, Red Button strongly recommends addressing these identified gaps. For clarity and ease of implementation, this report presents actionable recommendations that may simultaneously resolve multiple identified issues. These recommendations encompass essential protection components, enhancements to existing protection measures, monitoring and detection procedures, as well as mitigation and recovery protocols.

The table below outlines all recommendations in descending order of priority:

Priority	Recommendation
High	Investigate the failures with Optimizely and address the sizing issue that led to a complete denial of service.
High	Investigate the failure of mitigating 'Large File Download' with Imperva. Although the attack did not lead to a denial of service, the failure to detect and apply mitigation measures by Imperva should be investigated, as a higher rate attack may lead for DoS.
High	Conduct additional DDoS attack simulations on the bank's key assets to thoroughly test and fortify protection measures for all critical areas
Medium	The testing focused on application layer attacks. Red Button recommends covering, in future testing, more advanced application layer attack vectors.

The subsequent sections will delve into each recommendation, providing comprehensive insights and details on implementation strategies. By embracing these recommendations, Financial Corp can significantly enhance its DDoS preparedness and ensure the uninterrupted delivery of services to its customers.

Recommendations in Details

High - Investigate the failures with Optimizely.

An HTTPS GET / flood vector, operating below Imperva's rate limit threshold, was not detected or mitigated, resulting in a denial of service. Considering Financial Corp's scale, an attack rate below 800 RPS should be manageable by the web servers. If the servers are not designed to handle such a rate, activation of an autoscaling policy should minimize the impact on service availability. Hence, it is crucial to investigate with Optimizely the underlying cause of the sizing issue. Additionally, during the POST flood attack, no denial of service on the main site was detected at rates lower than the rate limit rule's threshold. However, it is essential to consider that a POST request might access another server, such as a database, potentially impacting other services without directly affecting the main website's availability. To gain clarity, discussions with Optimizely are necessary to understand the architectural differences between the services.

High – Investigate the failures with Imperva.

'Large file download with randomized parameters' attack vector was not detected or blocked by Imperva - This attack vector prevents relying on the CDN caching mechanism for absorption, due to the use of randomized

parameters. Even though the attack was absorbed by the infrastructure and did not impact the service, the risk level is high. Higher attack rates may saturate infrastructure resources and increase costs of cloud usage if Imperva's mitigation is not triggered.

High - Perform additional DDoS testing on the bank's key assets.

The testing covered two services, out of five main services. Preliminary discussions with Financial Corp personnel revealed that the remaining services lack essential settings for optimal protection. Red Button recommends conducting further testing to address these services and cover both network and application layer attack vectors.

Note: The recommended order of tests appears in [Appendix B](#).

Medium - Conduct advanced DDoS testing at the application layer.

The testing focused on application layer attacks. Red Button recommends covering, in future testing, more advanced application layer attack vectors.

Testing Work Logs

Attack Vector #1

Name	HTTPS GET / flood
Type	Application
Target / Destination	https://www.financialcorp.eu/
Max attack rate	50K RPS
Expected protection	DDoS settings (Global rate limit policy)
Attack log	<p>[01:30] - Attack started with 800 RPS from 264 bots.</p> <p>The site experienced a very high latency of 90 seconds.</p> <p>Status Cake monitoring tool indicated the website is down.</p> <p>[01:42] - The attack rate was increased to 1K RPS from 364 bots,</p> <p>The website was still down and not responding.</p> <p>[01:47] - The attack rate was increased to 17K RPS.</p> <p>[01:51] - The website fully recovered.</p> <p>[02:00] - The attack rate was increased to 50K RPS.</p> <p>[02:04] - The attack terminated.</p>
Results	Full downtime at a rate lower than Imperva's global rate limit rule threshold due to 100% CPU utilization on Optimizely's cloud servers. At a rate higher than the configured threshold, Imperva detected and entirely blocked the attack.
Alerts	DDoS alert received from Imperva.
Analysis / Recommendations	Investigate with Optimizely the failure of the servers to meet a very low rate of requests.

A timeout occurred

Error code 524

Visit cloudflare.com for more information.

2023-04-13 00:33:12 UTC

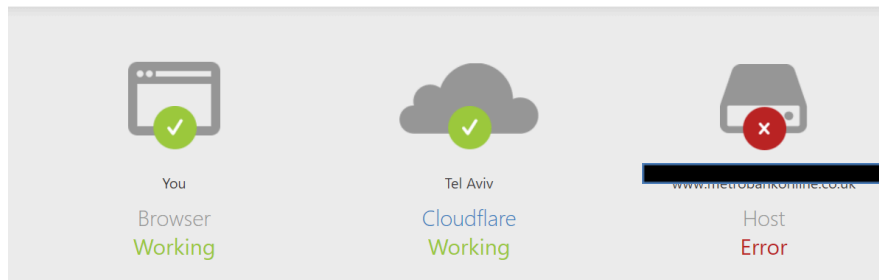


Figure 1 - '524 Timeout' error message - According to the error message, Cloudflare, which protects Optimizely's infrastructure, is working. However, the attack rate is lower than the defined threshold, so the denial of service is due to a failure of Optimizely's servers to keep up with the low rate of requests.

Requests over time

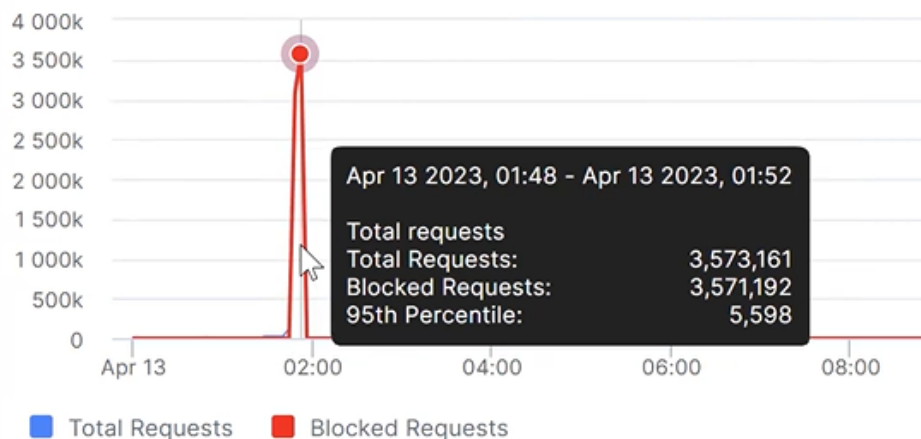


Figure 2 - Imperva security dashboard [01:53] - After increasing the attack rate above the global rate limit threshold (1,000 RPS), Imperva detected and fully blocked the attack by challenging each request with a JS challenge.

Client Type	Hacking Tool	Time	13 Apr 2023, 01:43:58	Hits	126143
Client App	Ruby HTTP library	Session ID	768000300455894005	Page	126143
Entry Page	www.metrobankonline.co.uk/	Country	Netherlands	Views	
Method	GET	Source IP	157.140.138.25	HTTP	1.0
User Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109...			Cookies	Not supported
DDoS (124041)		CAPTCHA (Fail)		More Details	

Figure 3 - Imperva WAF security log example [02:09] - Imperva logged the attack's traffic about 20 minutes after it was detected and blocked. According to the logs, Imperva defined the traffic as 'DDoS' and replied to the requests with a JS challenge.

Attack Vector #2

Name	HTTPS POST flood
Type	Application
Target / Destination	https://www.financialcorp.eu/banking/
Max attack rate	50K RPS
Expected protection	DDoS settings (Global rate limit policy)
Attack log	<p>[2023/13/04, 02:18] - The attack started with 500 RPS from 264 bots. No latency.</p> <p>[02:20] - The attack rate increased to 1.2K RPS from 264 bots. No latency.</p> <p>[02:22] - The bots are getting blocked; the attack was terminated.</p> <p>[02:23] - The attack was re-launched at 800 RPS from 100 bots.</p> <p>[02:27] - The attack was terminated.</p>
Results	Imperva mitigated the attack by the DDoS settings.
Alerts	DDoS alert received from Imperva.
Analysis / Recommendations	Investigate with Optimizely the infrastructure architecture and what are the consequences of POST flood on the backend, if any.

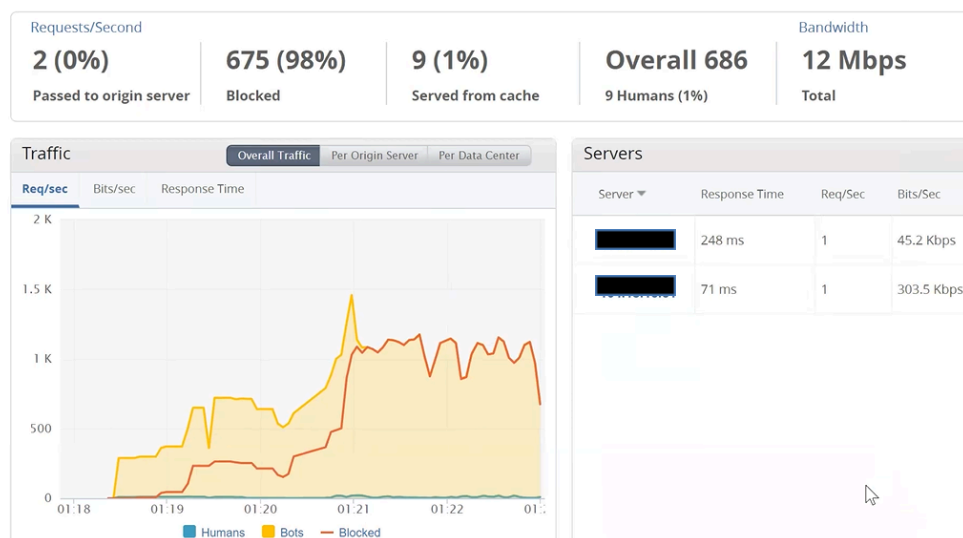


Figure 4 - Imperva security dashboard [02:09] - Imperva blocked 99% of the traffic in accordance with the DDoS settings (global rate limit rules) immediately after the threshold was hit.

Attack Vector #3.1

Name	Large file download
Type	Volumetric
Target / Destination	https://www.financialcorp.eu/static/legacyJs/index.js
Max attack rate	2 Gbps
Expected protection	Imperva CDN caching
Attack log	<p>[2023/13/04, 02:48] - The attack started with 1 Gbps from 164 bots, each sending 2 RPS. No impact on the website.</p> <p>[02:51] - The attack rate increased to 2 Gbps from the other 100 bots. No impact on the website.</p> <p>[02:56] - The attack was terminated.</p>
Results	Imperva caching mitigated the attack.
Alerts	None
Analysis / Recommendations	None

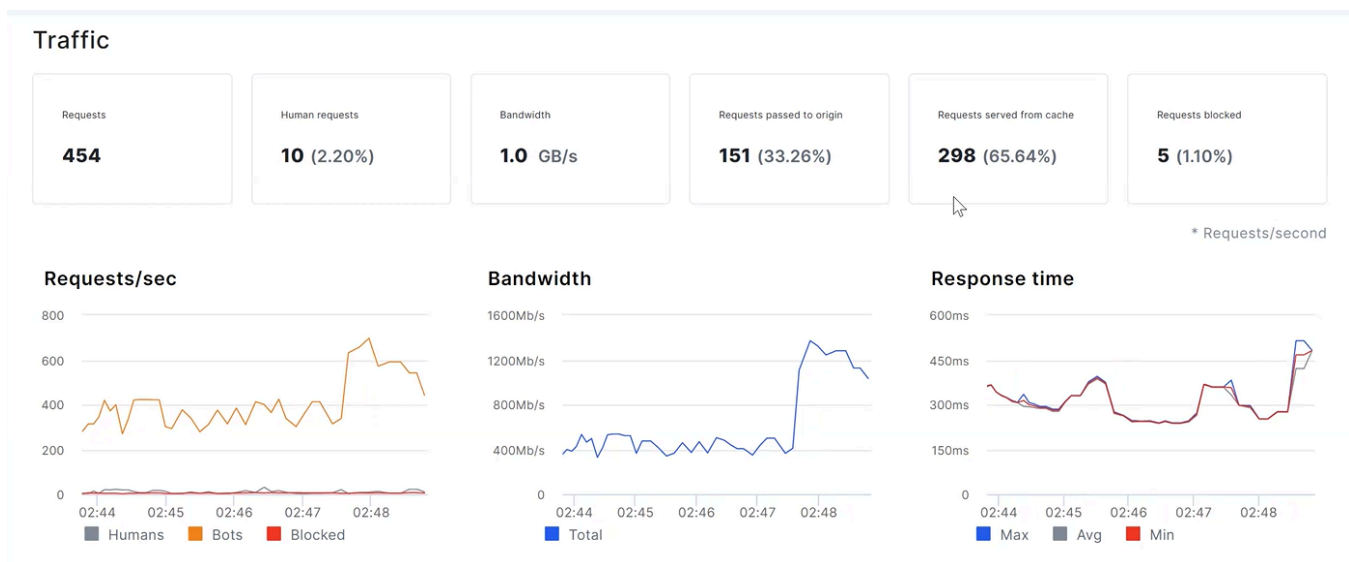


Figure 5 - Imperva real-time traffic dashboard [02:50] - The requests were served from the cache (65.64%), and the total bandwidth increased to 1 Gbps, without affecting the service.

Attack Vector #3.2

Name	Large file download with randomized parameters
Type	Volumetric
Target / Destination	https://www.financialcorp.eu/static/legacyJs/index.js?clientid=\$rand
Max attack rate	30 Gbps
Expected protection	DDoS settings
Attack log	<p>[02:57] - The attack started with 1 Gbps from 164 bots, each sending 2 RPS. A low latency from the website.</p> <p>[03:00] - The attack rate increased to 2 Gbps from the other 100 bots. Each sends 2 RPS. The website experienced a latency of 1 second.</p> <p>[03:05] - The attack rate increased to 2.5 Gbps. No meaningful impact on the website.</p> <p>[03:09] - The attack rate increased to 23 Gbps. The website experienced a latency of 2-3 seconds.</p> <p>[03:13] - The attack rate increased to 30 Gbps. The website experienced a latency of 5 seconds.</p>
Results	Imperva didn't detect nor mitigated the attack. Optimizely network (upon AWS infrastructure) absorbed the attack.
Alerts	None
Analysis / Recommendations	Imperva didn't detect the attack and the bots' traffic wasn't blocked, even though the threshold was exceeded.

Attack Vector #4

Name	HTTPS GET / flood on login
Type	Application
Target / Destination	https://members.financialcorp.eu/login.aspx
Max attack rate	50K RPS
Expected protection	Cloudflare HTTP automatic DDoS
Attack log	<p>[03:22] - The attack started with 800 RPS from 264 bots. No impact on the website.</p> <p>[03:33] - The attack rate increased to 5K RPS. No impact on the website.</p> <p>[03:43] - The attack rate increased to 25K RPS. No impact on the website.</p> <p>[03:48] - The attack rate increased to 50K RPS. No impact on the website.</p> <p>[03:52] - The attack terminated. No traffic was visible on CF dashboard, so we changed the attack script (added "/" after the domain name).</p> <p>[03:54] - We changed the script target from 'https://members.financialcorp.eu/login.aspx' to 'https://members.financialcorp.eu/'. The traffic became visible on CF dashboard.</p> <p>[04:00] - The attack was relaunched at 33K RPS. The website experienced high latency for 1 minute.</p> <p>[04:01] - Cloudflare fully detected and mitigated the attack.</p> <p>[04:07] - The attack was terminated.</p>
Results	Cloudflare HTTP automatic DDoS fully detected and mitigated the attack (managed JS challenge).
Alerts	DDoS alert received from Cloudflare.
Analysis / Recommendations	None

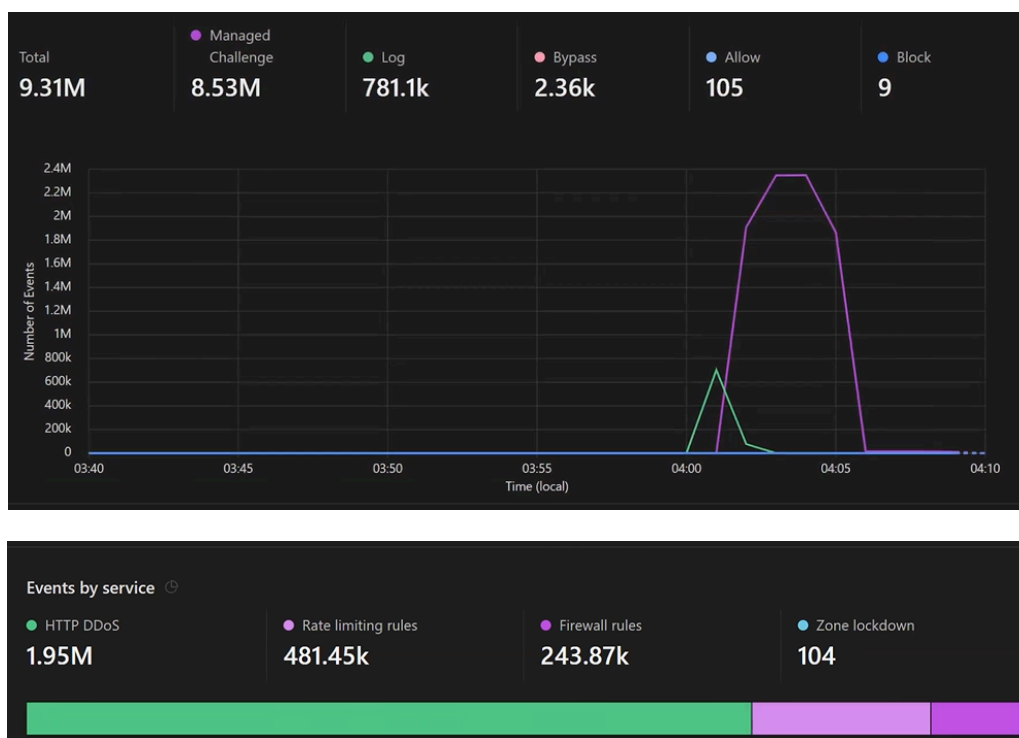


Figure 6 - Cloudflare security event dashboard [04:07] - After relaunching the attack, the traffic was logged for about a minute (green curve), until the HTTP DDoS mechanism was activated and blocked the attack with a managed JS challenge (purple curve). Note: There is no color compatibility between the top and bottom graphs (a known issue in Cloudflare).

Attack Vector #5

Name	HTTPS GET flood with randomized paths
Type	Application
Target / Destination	https://members.financialcorp.eu/login.aspx/clientid=\$rand
Max attack rate	50K RPS
Expected protection	Cloudflare HTTP automatic DDoS
Attack log	[04:13] - The attack started with 10K RPS from 264 bots. [04:14] - The attack rate increased to 50K RPS. [04:15] - The attack was terminated.
Results	Cloudflare mitigated the attack by HTTP DDoS (302 redirect to the login page that served by the cache).
Alerts	DDoS alert received from Cloudflare.
Analysis / Recommendations	None

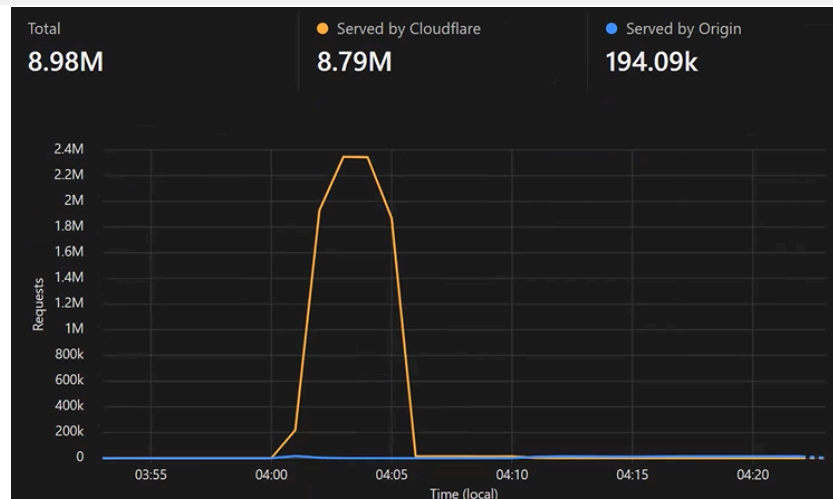


Figure 7 - Cloudflare caching overview dashboard [04:15] - Immediately as the attack was launched, the HTTP DDoS mechanism blocked the traffic by '302 redirect' to the login page served from the caching server, and therefore did not affect the origin server.

Attack Vector #6.1

Name	Large file download
Type	Volumetric
Target / Destination	https://members.financialcorp.eu/bundles/miscmasterjs?v=k5C2xid7TOh2x1i87I5tDO0rcq1jEKjVDzd77b5Kurl1
Max attack rate	6.5 Gbps
Expected protection	Cloudflare CDN caching
Attack log	[04:19] - The attack started with 6.5 Gbps from 264 bots. [04:28] - The attack was terminated.
Results	Cloudflare mitigated the attack by HTTP DDoS (403 error, behavioural DoS).
Alerts	DDoS alert received from Cloudflare.
Analysis / Recommendations	None

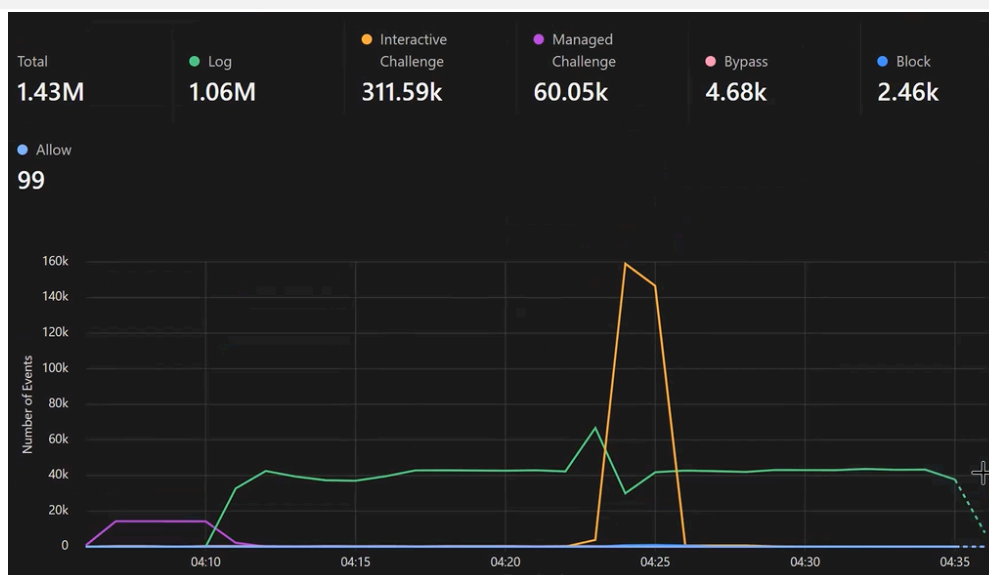


Figure 8 - Cloudflare security event dashboard [04:30] - Immediately as the attack was launched, the HTTP DDoS mechanism blocked the traffic by interactive challenges (orange curve).

Attack Vector #6.2

Name	Large file download with randomized parameters
Type	Volumetric
Target / Destination	https://members.financialcorp.eu/bundles/miscmasterjs?v=k5C2xid7TOh2x1i87I5tDO0rcq1jEKjVDzd77b5Kurl1/?clientid=\$rand
Max attack rate	20 Gbps
Expected protection	Cloudflare HTTP automatic DDoS
Attack log	<p>[04:35] - The attack started with 2 Gbps from 264 bots.</p> <p>[04:41] - The attack rate increased to 4 Gbps from 264 bots.</p> <p>[04:44] - The attack rate increased to 10 Gbps from 264 bots.</p> <p>[04:53] - The attack rate increased to 20 Gbps from 264 bots.</p> <p>[04:57] - The attack terminated</p>
Results	Cloudflare mitigated the attack by HTTP DDoS (302 redirect to the login page that served by the cache).
Alerts	DDoS alert received from Cloudflare.
Analysis / Recommendations	None

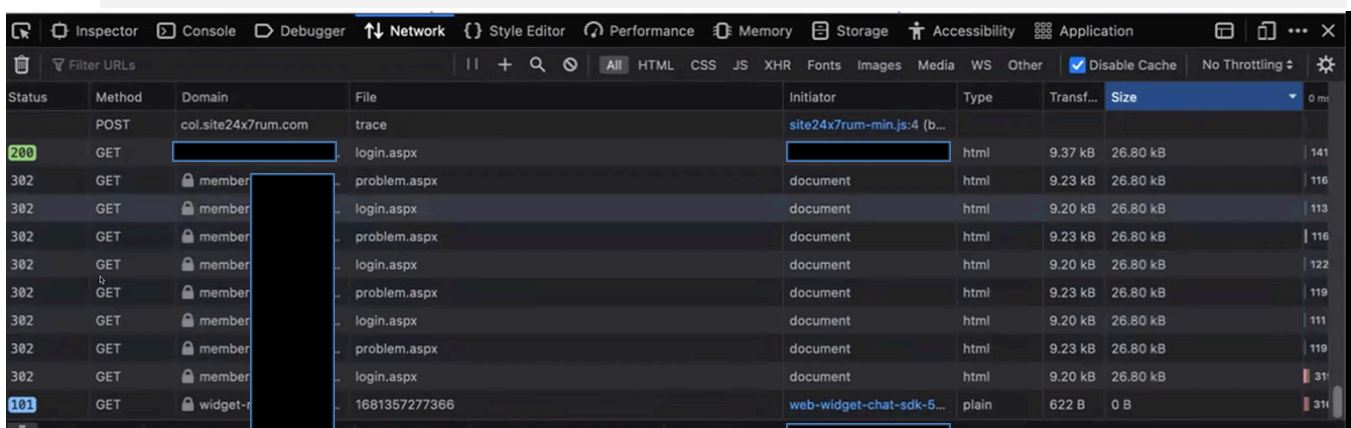


Figure 9 - Browser screenshot [04:54] - Cloudflare redirected (status 302) all the website resources with the HTTP automatic DDoS protection.

Appendix A - Glossary

Term	Definition
Application-Level Attack	Also called Layer 7 (L7) attacks. These attacks target a specific service, such as Web or DNS. Attack traffic attempts to overwhelm the service by sending service requests, e.g., HTTP GET SLASH flood.
Network-Level Attack	Also called Layer 3/4 attacks. These attacks target a network resource, e.g., a router, firewall, load balancer, or even a server's IP stack. Attack traffic attempts to overwhelm the network resource, or even the link capacity, with excessive traffic. Attacks can be volumetric (e.g., UDP Flood, ICMP Flood) or protocol based (e.g., SYN Flood, ACK Flood).
Rate-Limit Protection	<p>Rate Limit is a general name for a simple protection measure against DDoS attacks. The mechanism maintains the rate of resource requests below a defined (or adaptive) threshold, preventing resource exhaustion.</p> <p>Note: Rate limit mechanisms cannot distinguish between legitimate and hostile requests; therefore, they risk blocking some legitimate traffic.</p>
CDN (Content Delivery Network)	<p>A Content Delivery Network (CDN) is a geographically distributed network of proxy servers. CDN service improves users' accessibility to content, at lower costs to the content distributor, by serving them from a node that is geographically closer to them, relying on caching and other technologies.</p> <p>In the context of DDoS, CDN can act as a powerful layer of defense, by absorbing attack requests at the PoPs, and preventing the requests to reach the limited static resources, the origin of the content. Many CDN services offer additional web protection services, e.g., WAF, that mitigate DDoS and intrusion attacks, beyond simple absorption.</p>

Appendix B – Assets' Testing Priority

Priority	Term	Definition
High	<ul style="list-style-type: none"> The VPN service of banks is one of the central services attacked by DDoS attacks. We recommend testing the L3-4 protection provided by Arbor. It is currently unclear if there is L7 protection on the property. 	<p>The diagram illustrates the VPN Access architecture. It shows a group of 'Users' (represented by three person icons) connecting to a box labeled 'NETSCOUT Arbor'. This box is connected to a box labeled 'Origin Servers' (represented by a server icon). The entire system is branded with the 'rackspace' logo in the top right corner.</p>
Medium	<ul style="list-style-type: none"> Imperva's basic settings are identical to the tested service. However, protections may behave differently per web applications. 	<p>The diagram illustrates the Secondary Service architecture. It shows a group of 'Users' (represented by three person icons) connecting to a box labeled 'IMPERVA INCAPSULA'. This box is connected to a box labeled 'Origin Servers' (represented by a server icon). The entire system is branded with the 'rackspace' logo in the top right corner.</p>
Low	<ul style="list-style-type: none"> The asset isn't external facing and has an IP restriction policy. 	<p>The diagram illustrates the Internal Users architecture. It shows a group of 'Internal Users' (represented by three person icons) connecting to a box labeled 'IP Restriction'. This box is connected to a box labeled 'Application gateway' (which also contains 'No DDoS Protections'). This box is connected to a box labeled 'Azure VMs (internal)' (which also contains 'Internal URL'). The entire system is branded with the 'Azure' logo in the top right corner.</p>

