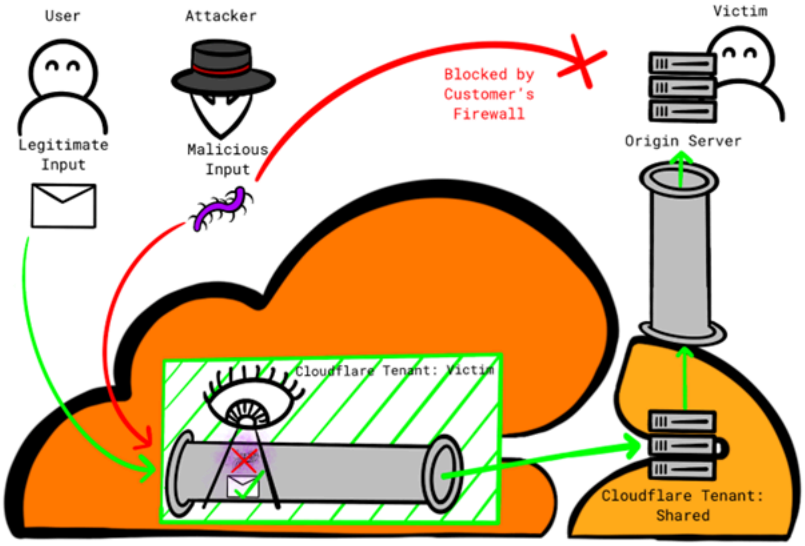
USING CLOUDFLARE TO BYPASS CLOUDFLARE

Written by Stefan Proksch on 28.09.2023

Cloudflare customer-configured protection mechanisms (e.g., Firewall, DDoS prevention) for websites can be bypassed due to gaps in cross-tenant security controls, potentially exposing customers to attacks Cloudflare is supposed to prevent. Attackers can utilize their own Cloudflare accounts to abuse the per-design trust-relationship between Cloudflare and the customers websites, rendering the protection mechanism ineffective. Cloudflare customers should review their origin-server protection strategy to ensure their configured protections are reliably enforced.

Introduction

Cloudflare is a major cybersecurity vendor, that is offering hosted website protection services [1] (e.g., Web Application Firewall (WAF), DDoS protection, Bot management). This is achieved by hosting a network of reverse-proxy servers, which sit between a customer’s webserver (further referred to as “origin server”) and its users, enabling traffic analysis for malicious activity. As customers of Cloudflare potentially rely upon this service offering, it is important to shield the origin server from any access that did not pass through the configured reverse-proxy servers. Upon following the official Cloudflare documentation [2], customers may inadvertently use mechanisms susceptible to exploitation **through the Cloudflare platform itself**. This vulnerability arises from the shared infrastructure available to all tenants within Cloudflare, whether legitimate or malicious, allowing them to circumvent configured security measures and target customer systems.


User
Attacker
Victim
Blocked by Customer's Firewall
Legitimate Input
Malicious Input
Origin Server
Cloudflare Tenant: Victim
Cloudflare Tenant: Shared

Cloudflare Origin Server Protection Services Visualized

This gap in protection is not clearly documented, which is why we have decided to responsibly disclose this to Cloudflare via their bug-bounty program. Cloudflare has categorized this report as "Informative" and closed it; we are therefore publicly disclosing these details. This is to allow customers to evaluate their configurations for the vulnerabilities discussed below.

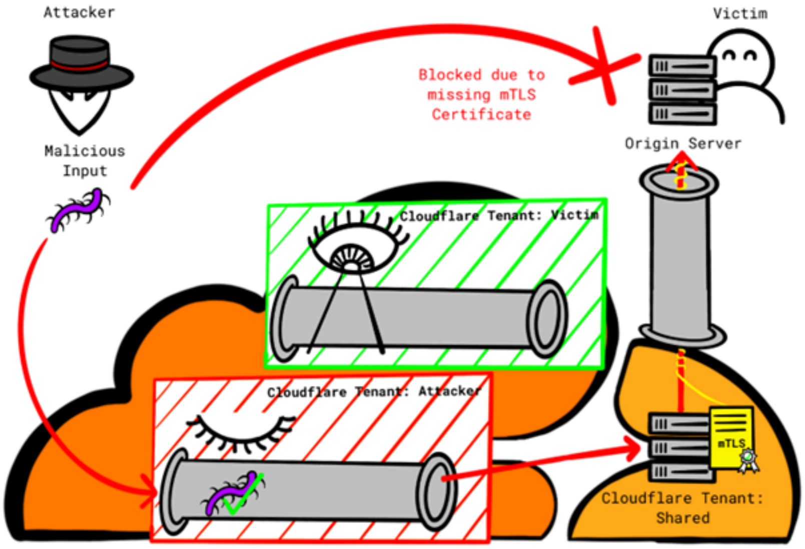
Vulnerability Overview

Cloudflare outlines various mechanisms to "*prevent attackers from discovering and overloading your origin server with requests*" in their official documentation [2] on various layers of the OSI Model (Application Layer, Transport Layer, and Network Layer) . The mechanisms are annotated with varying degrees of security levels, either "moderately secure" or "very secure", and the associated technical challenges. During our analysis, we found that two proposed mechanisms are based on the premise, that all traffic to the origin server originating from Cloudflare is to be trusted, while traffic from other parties is to be rejected. **We show that attackers can abuse this trust in Cloudflare by sending their malicious payload via the Cloudflare platform**, bypassing various protection mechanism (e.g., the Web Application Firewall) that a customer might have configured for their environment. The effective impact of this bypass is dependent on the customers origin server configuration.

Authenticated Origin Pulls

When using the mechanism "Authenticated Origin Pulls" on Transport Layer, stated as "very secure" in the Cloudflare documentation, the Cloudflare reverse-proxy servers authenticate to the origin server using a client SSL certificate [3]. The zone setup documentation [4] presents two options for authenticating connections from clients, which are being routed through Cloudflare's reverse-proxy server, to the origin server. Customers can choose either a "Cloudflare certificate" or a custom certificate. However, the documentation does not discuss the security implications associated with these options. It is also to be noted, that the custom certificate can only be configured using an API. Without additional information, it is reasonable to assume, that customers will opt for the more convenient choice of using the Cloudflare certificate.

The unstated serious implication of using a shared “Cloudflare certificate [5] over a tenant-specific custom CA is that all connections originating from Cloudflare are permitted, regardless of which Cloudflare tenant is initiating the connection. **An attacker can setup a custom domain with Cloudflare and point the DNS A record to victims IP address. The attacker then disables all protection features for that custom domain in their tenant and tunnel their attack(s) through the Cloudflare infrastructure. This approach allows attackers to bypass the protection features by the victim.**


Attacker
Malicious Input
Blocked due to missing mTLS Certificate
Victim
Origin Server
Cloudflare Tenant: Victim
Cloudflare Tenant: Attacker
mTLS
Cloudflare Tenant: Shared

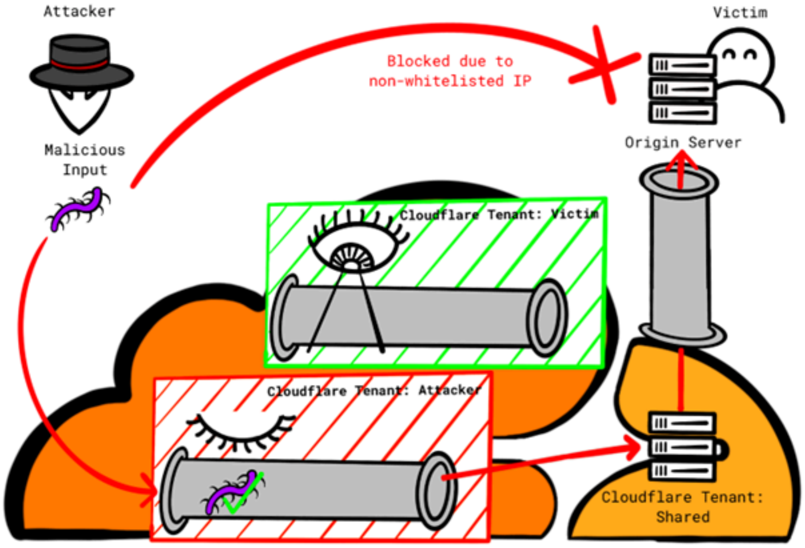
Example Exploitation of Shared Cloudflare Certificates

This currently can only be mitigated by using custom certificates, which requires the customer to create and maintain their own origin pull certificates.

Allowlist Cloudflare IP addresses

When using the mechanism “Allowlist Cloudflare IP addresses” on Network Layer, stated as “moderately secure”, the origin server rejects any connection not originating from within the Cloudflare IP address ranges. The setup documentation [6] documents how to setup those ranges using an .htaccess file or iptables.

As with authenticated origin pulls, the unstated serious implication of this mechanism is that all connections originating from Cloudflare regardless of tenant are permitted. **An attacker can establish a custom domain with Cloudflare, direct the DNS A record to the victims IP address. Next, they disable all protection features for that custom domain and route their attack(s) through Cloudflare’s infrastructure, effectively bypassing the protection features that the victim has configured.**


Attacker
Malicious Input
Blocked due to non-whitelisted IP
Victim
Origin Server
Cloudflare Tenant: Victim
Cloudflare Tenant: Attacker
Cloudflare Tenant: Shared

Example Exploitation of Allowlist Cloudflare IPs

This currently can only be mitigated by using the Cloudflare Aegis [7], which offers dedicated egress IP addresses rather than using the shared IP address range. This service might not be available to all customers.

Proof of Concept

The following outlines the setup for a successful bypass of a WAF protected domain *victim.test* that attempts to shield the origin server *203.0.113.42* by using "Authenticated Origin Pulls" with the Cloudflare Origin Certificate as-well as "Allowlist Cloudflare IP addresses" as outlined by the official documentation. The attacker simply configures the domain *attacker.test* without any WAF protection and sets the same origin IP address as *victim.test*. This allows the attacker to successfully send requests to *203.0.113.42* via *attacker.test* that would be blocked when attempting to do so via *victim.test*.

Configuration of the victim's Cloudflare account

Domain: *victim.test*
DNS A record points to: *203.0.113.42*
Cloudflare settings:

- SSL/TLS encryption mode: "Full (strict)"
- Authenticated Origin Pulls enabled
- Cloudflare Origin Certificate created
- WAF Cloudflare Managed Ruleset enabled
- WAF Cloudflare OWASP Core Ruleset enabled
- Security Level: "I'm under attack" – Always Use HTTPS enabled

Configuration of the victim's origin server

- Cloudflare Origin Certificate installed (SSL/TLS enabled)
- Authenticated Origin Pulls CA installed (SSLVerifyClient enabled)
- iptables only allows ingress traffic from Cloudflare IPs on port 443

Configuration of the attackers Cloudflare account

Domain: *attacker.test*
DNS A record points to: *203.0.113.42*
Cloudflare settings:

- SSL/TLS encryption mode: “Full”
- Authenticated Origin Pulls enabled
- WAF Cloudflare Managed Ruleset disabled
- WAF Cloudflare OWASP Core Ruleset disabled
- Security Level: “Essentially off”

Protection bypass

Cloudflare WAF as protecting the victim’s origin server from a potentially malicious input.

```
> GET https://victim.test/?test=cat%20/etc/passwd HTTP/2
< HTTP/2 403 Forbidden
```

Cloudflare forwarding the potentially malicious input to the victim’s origin server, bypassing the victims WAF configuration.

```
> GET https://attacker.test/?test=cat%20/etc/passwd HTTP/2
< HTTP/2 200 OK
```

Recommendation for Cloudflare customers

The “Allowlist Cloudflare IP addresses” mechanism should be regarded as defence-in-depth, and not be the sole mechanism to protect origin servers. The “Authenticated Origin Pulls” mechanism should be configured with custom certificates rather than the Cloudflare certificate. Other mechanisms for authenticating the Cloudflare tenant (rather than Cloudflare itself) outlined in the documentation [2] along with their various trade-offs (e.g., running third-party code on sensitive webservers) should be considered when protecting origin servers.

We would recommend Cloudflare to implement protection mechanisms against such attacks and warn customers with weak configurations.

UPDATE October 4th 2023

On Oct 4th, Cloudflare changed the severity of our vulnerability report from Informative to High (7,5) and announced that they will take steps to enhance the documentation and update the dashboard to encourage Per-Hostname and Per-Zone Authenticated Origin Pulls for protecting origin servers and to encourage users to protect their origin servers by implementing Host header validation and protecting the confidentiality of their origin server IP address.

Disclosure timeline

2023-03-16	Issue reported to Cloudflare via HackerOne (report 1909867)
2023-03-16	Issue acknowledged by Cloudflare and closed (bug informative)
2023-09-28	Public disclosure (>180 days)
2023-10-04	Cloudflare changed the severity from informative to high (7,5)
2023-10-04	Cloudflare announced enhancements to documentation and dashboard

References

[1] https://www.cloudflare.com/application-services/products/
[2] https://developers.cloudflare.com/fundamentals/get-started/task-guides/origin-health/enterprise/
[3] https://developers.cloudflare.com/ssl/origin-configuration/authenticated-origin-pull/explanation/
[4] https://developers.cloudflare.com/ssl/origin-configuration/authenticated-origin-pull/set-up/zone-level/
[5] https://developers.cloudflare.com/ssl/static/authenticated_origin_pull_ca.pem
[6] https://developers.cloudflare.com/fundamentals/get-started/setup/allow-cloudflare-ip-addresses/
[7] https://blog.cloudflare.com/cloudflare-aegis/

This security issues was identified by Florian Schweitzer and Stefan Proksch.

« Newer Older »

CERTITUDE

Certitude Consulting GmbH
Barichgasse 40-42
1030 Vienna

Services
About Us
Career

Privacy statement
Legal Notice
Imprint

Contact
LinkedIn