HTTP 403, 401 Bypass Techniques

Eyad El-Taher

January 22, 2026

How I Bypassed 403 Forbidden & Accessed Restricted Pages

Introduction

A 403 Forbidden error is often seen as a dead end, but in security testing, it is frequently a sign of a misconfigured gateway or Web Application Firewall (WAF). By using specific bypass techniques, ethical hackers can often access restricted administrative panels and sensitive directories.

What is a 403 Forbidden Error?

A 403 error occurs when the server understands the request but refuses to fulfill it.
Common Causes:

- **IP Restrictions:** Access is limited to specific internal IP ranges.
- **WAF Rules:** A firewall identifies the request as "suspicious" and blocks it.
- **Permission Misconfiguration:** Incorrectly set file or directory permissions.

Successful Bypass Techniques

1. Alternative HTTP Methods

Security filters often only restrict the `GET` method. Changing the request method can bypass the filter while still triggering the desired backend logic.

```
curl -X POST https://target.com/admin
curl -X HEAD https://target.com/admin
```

Tip: Always test with `OPTIONS`, `TRACE`, `PUT`, or `PATCH`.

2. Modifying Request Headers

Many WAFs look at headers to determine if a request is coming from a trusted local source or a specific proxy.

- **Spoofing Localhost:** `curl -H "X-Forwarded-For: 127.0.0.1" https://target.com/admin`
- **Spoofing Referer:** `curl -H "Referer: https://target.com/allowed-page" https://target.com/admin`
- **Mimicking Crawlers:** `curl -H "User-Agent: Googlebot" https://target.com/admin`

3. Path Obfuscation Tricks

Normalizers on the backend may treat different paths as identical, while the WAF may only look for a specific string match like `/admin`.

- **Suffixing:** `https://target.com/admin/.`
- **Path Traversal:** `https://target.com/admin/%2e%2e/`
- **Whitespace/Encoding:** `https://target.com/admin%20/`

4. Direct IP Access

If the 403 restriction is implemented at the Virtual Host level, accessing the server via its direct IP address instead of its domain name can sometimes bypass the restriction entirely.

```
curl http://192.168.1.1/admin
```

5. HTTP vs. HTTPS Switching

Security rules are sometimes inconsistently applied across protocols. A resource forbidden on HTTPS might be left accessible (or less strictly filtered) over plain HTTP.

```
curl -k http://target.com/admin
curl -k https://target.com/admin
```

Advanced 403 Forbidden Bypass Techniques

1. Applying Special Characters to the URL

Adding special characters to the URL path can trick the server into interpreting the request differently, often bypassing string-matching security filters.

- **Dot Segments:** Use `/admin/./` or `/admin%2e/` to represent the current directory.
- **Trailing Slashes:** Appending a slash (e.g., `/admin/`) can bypass rules that look for an exact string match without the slash.
- **String terminators** use (%00, 0x00, //, ;, %, !, ?, [] etc.) — adding those to the end of the path and inside the path
- **Encoding:** Use encoded representations like `%20` (space) to alter the signature of the path.

Note: Always test payloads on isolated systems to avoid unintended consequences.

2. Headers Manipulation

Altering or injecting specific HTTP headers can trick the server into believing the request originated from a trusted internal source.

- **Internal IP Injection:** Use headers like `X-Forwarded-For`, `X-Originating-IP`, or `X-Remote-IP`.
- **Trusted Values:** Experiment with `127.0.0.1`, `localhost`, or internal cloud IP ranges (e.g., `10.0.0.1`).

3. Changing IP Address or Using a VPN

Restrictions are often tied to the reputation of an IP address. Web Application Firewalls (WAFs) may block IPs flagged for suspicious activity.

- **VPNs:** Use services like **NordVPN** to quickly rotate your exit IP.
- **ProxyChains:** Useful for chaining multiple proxies to evade IP-based rate limiting.

4. Fuzzing HTTP Methods

Fuzzing involves systematically testing an endpoint with various HTTP methods to identify unexpected server behavior.

Testing with FFUF:

```
ffuf -u https://example.com/endpoint -X FUZZ -w http_methods_wordlist.txt
```

Replace `FUZZ` with methods such as `OPTIONS, PATCH, TRACE`, etc.

5. Fuzzing the URL Path

Automated fuzzing can uncover misconfigurations in path validation logic by testing hundreds of URL variations.

Example Payload Test:

```
curl https://example.com/admin/./
```

This is highly effective against logic that fails to properly normalize the path before applying security checks.

Supplementary 403/401 Bypass Methodology

Distinguishing 401 Unauthorized vs. 403 Forbidden

Understanding the specific error code is critical for selecting the correct bypass vector.

- **401 Unauthorized:** The request lacks valid authentication credentials. This is an "Identity" issue. The server is asking for a login or a valid token.
- **403 Forbidden:** The server understands the identity (or lack thereof) but refuses access based on permissions or security policies. This is an "Authorization" or "Policy" issue.

1. Advanced URL Manipulation & Case Sensitivity

Servers running on Windows (IIS) or specific configurations of Apache/Nginx may handle case sensitivity and path segments inconsistently.
Unique Path Variations:

- **Case Flipping:** `/Admin`, `/AdMin`, or `/ADMIN` (Effective if the WAF is case-sensitive but the backend is not).
- **Semicolon Injection:** `/;/admin` or `//;//admin`.
- **Path Normalization:** `/.;/admin` or `/./admin/..`.
- **Format Appending:** `/admin.json` (Tricks the server into thinking it is a public API/static file).
- **Double Slashes:** `//admin//`.

2. URL Rewriting & Routing Headers

Some applications use internal routing headers to determine the destination of a request. If the WAF only inspects the main URL but allows these headers, a bypass is possible.

X-Original-URL / X-Rewrite-URL Technique: Instead of requesting the forbidden page directly, request a known public page and tell the application to rewrite the destination internally.

```
GET /anything HTTP/1.1
Host: target.com
X-Original-URL: /admin

OR

GET /anything HTTP/1.1
Host: target.com
X-Rewrite-URL: /admin
```

3. Parameter Tampering

Access controls are sometimes implemented via client-side parameters that can be manipulated in the request.

- **Boolean Logic:** Change `isAdmin=false` to `isAdmin=true`.
- **View State:** Change `?view=restricted` to `?view=public`.
- **Parameter Removal:** Try removing the parameter entirely to see if the server defaults to an "Allow" state when the restriction variable is missing.

4. Method Overriding via Headers

If the server blocks specific HTTP methods but allows header-based overrides, you can trick the backend into executing a forbidden method while the WAF only sees a permitted one.

```
POST /admin HTTP/1.1
Host: target.com
X-HTTP-Method-Override: GET

OR

X-HTTP-Override: GET
```

5. Strategic Fuzzing Targets

While `/admin` might be blocked, the developers may have forgotten to protect sub-directories. Always fuzz for children of restricted paths:

- `/admin/users`
- `/admin/config`
- `/admin/settings`

  Technique Insight: Using the **Param Miner** extension in Burp Suite is highly effective for discovering "hidden" headers like `X-Original-URL` that the application might be listening for.

Protocol Downgrade: Bypassing 403 via HTTP/1.0

The Vulnerability Concept

In modern web traffic, HTTP/1.1 is the standard, requiring a `Host` header for correct request routing. However, many legacy systems maintain backward compatibility with **HTTP/1.0**. This can be exploited if a Web Application Firewall (WAF) or security gateway is only configured to validate and filter HTTP/1.1 traffic.

Real-World Comparison: Microsoft Lyncdiscover Case

The following example demonstrates how stripping headers and downgrading the protocol version can trick a server into a "local trust" state, bypassing a 403 Forbidden error.

Initial Request (403 Forbidden): Standard HTTP/1.1 request including all identifying headers.

```
GET /reach%2fsip.svc HTTP/1.1
Host: lyncdiscover.microsoft.com
User-Agent: curl/7.79.1
Accept: */*
Connection: close
```

Bypass Request (200 OK): The protocol is downgraded to HTTP/1.0 and all header values are cleared.

```
GET /reach%2fsip.svc HTTP/1.0
```

Why This Works

- **Defaulting to Local:** When the `Host` header is absent in an HTTP/1.0 request, some misconfigured security mechanisms default the destination address to the server itself.
- **Identity Masking:** This causes the request to be identified as "local" rather than external, effectively bypassing IP-based or external-access restrictions.
- **Information Leakage:** This method can also bypass CDNs to obtain original server IPs by observing the `Location` header in a resulting redirect.

  Note: This specific bypass was validated against `lyncdiscover.microsoft.com` and resulted in access to restricted `.svc` files and WSDL documentation.

Vulnerability Report: 403 Forbidden Bypass (DoD VDP)

Vulnerability Summary

- **Type:** 403 Forbidden Bypass
- **Target:** `www.[redacted].mil`
- **Severity:** Medium (Assessed by DoD)
- **Status:** Resolved
- **Reported via:** DoD Vulnerability Disclosure Program (VDP)

Vulnerability Description

A vulnerability was identified where certain restricted pages returning a `403 Forbidden` status could be accessed by manipulating the HTTP request structure. The access control mechanism failed to properly validate HTTP methods, allowing unauthorized access when the following conditions were met:

1. Changing the HTTP method from `GET` to `POST`.
2. Appending the HTTP header: `Content-Length: 0`.

Steps to Reproduce

1. Attempt to access a restricted page via a standard browser or `GET` request → **403 Forbidden**.
2. Intercept the request and modify the method to `POST`.
3. Add the header `Content-Length: 0`.
4. Send the request → The server responds with the protected content.

Proof of Concept (PoC)

Denied Request:

```
curl -X GET https://www.[redacted].mil/[path]
```

Successful Bypass:

```
curl -H "Content-Length: 0" -X POST https://www.[redacted].mil/[path]
```

> **Technique Insight:** This bypass exploits misconfigured security filters that only apply restrictions to `GET` requests or fail to authorize `POST` requests that do not carry a data payload (indicated by `Content-Length: 0`).

Vulnerability Report: 403 Bypass via Trusted Proxy Header

Vulnerability Summary

- **Type:** Access control bypass via trusted proxy header
- **Technique:** Header spoofing (`X-Forwarded-For`)
- **Affected Platform:** Yelp Business Owner App
- **Severity:** Medium (CVSS ∼6.1)

Vulnerability Description

The web application incorrectly trusted the client-supplied `X-Forwarded-For` HTTP header. By supplying a loopback address (`127.0.0.1`), the backend treated the request as originating from an internal service.

Accessed Endpoints:

- Health and system status endpoints
- Swagger API documentation
- Business Owner App backend APIs

Root Cause

The backend logic relied solely on the `X-Forwarded-For` header to differentiate between internal and external traffic. Because this header was not sanitized or stripped at the load balancer layer, external attackers could spoof an internal identity.

Proof of Concept (PoC)

PoC 1 – Internal Status Endpoint

A standard request to the `/status` endpoint is denied by the server:

```
curl -k https://biz-app.yelp.com/status
-- Response: {"error":{"id":"PredicateMismatch"}}
```

Adding the spoofed header grants full access to system metadata:

```
curl -k https://biz-app.yelp.com/status -H "X-Forwarded-For: 127.0.0.1"
```

PoC 2 – Swagger API Documentation

The Swagger specification, normally restricted, is returned when the internal IP is spoofed:

```
curl -k https://biz-app.yelp.com/swagger.json -H "X-Forwarded-For: 127.0.0.1"
```

Evidence of Internal Trust

The backend's response headers explicitly confirmed that the spoofed IP was accepted as internal:

- `x-is-internal-ip-address: true`
- `x-routing-service: routing-main--useast1`
- `x-proxied: 10-65-64-83-useast1aprod`

Impact

This vulnerability represents a significant trust boundary violation. Attackers could enumerate internal-only APIs, view sensitive service metadata, and potentially interact with administrative backend functionality.

> **Remediation:** Edge proxies and load balancers should be configured to strip or overwrite `X-Forwarded-For` headers coming from untrusted external sources.