

# Rekonesans / zbieranie informacji o systemie

Jak działają aplikacje WWW ?



# Statystyki z ostatniego roku

Obraz autorstwa Freepik.com

### Attacks Bar - Dynamic

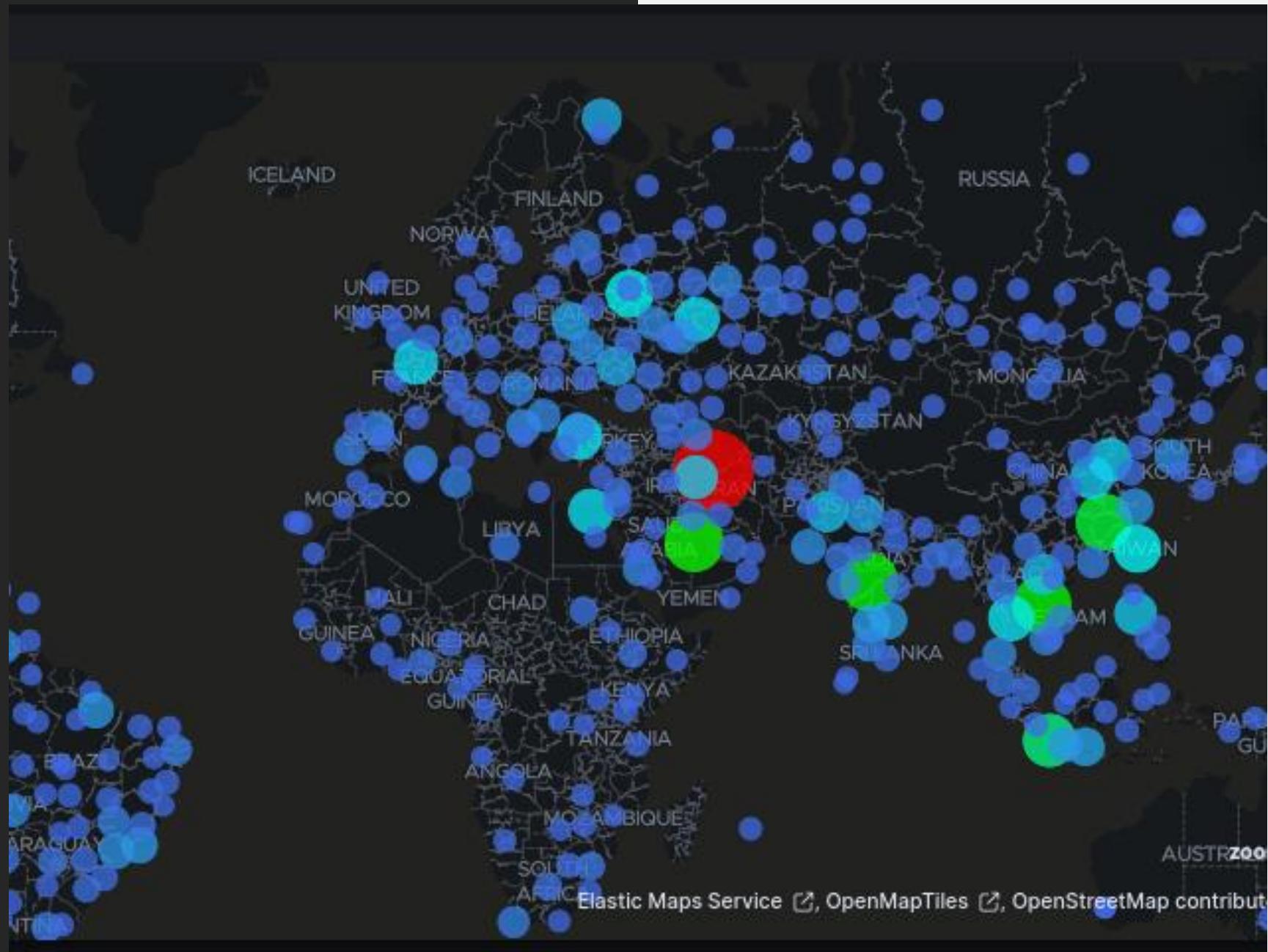


Attacks

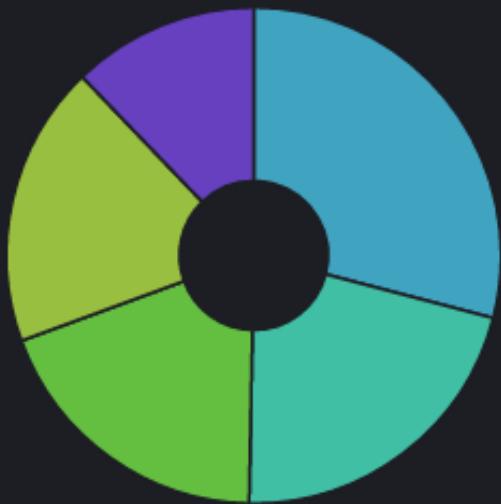
### Attacks - Dynamic

**756,274**  
Attacks

**31,365**  
Unique Src IPs



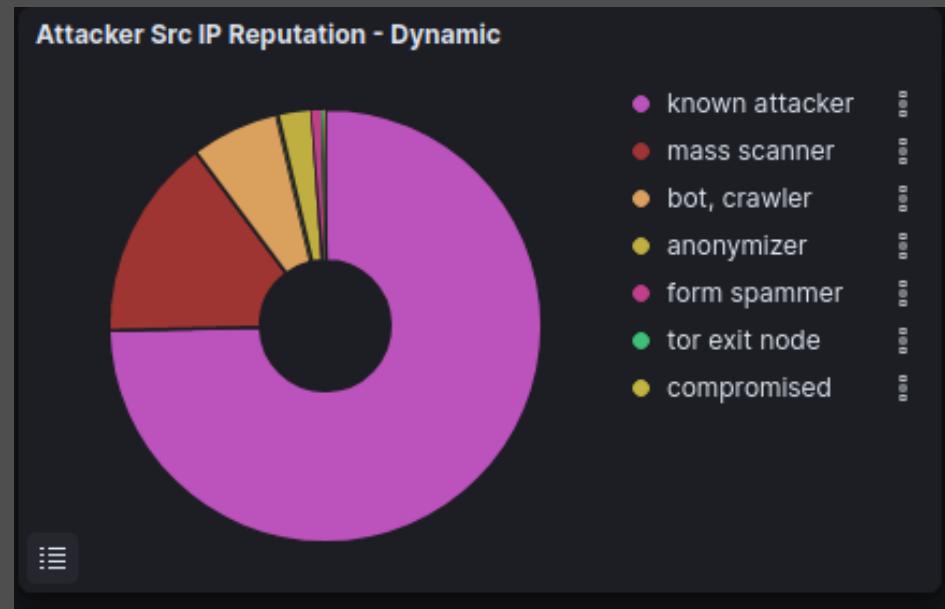
### Attacks by Country - Dynamic



- Iran
- Russia
- India
- China
- Vietnam

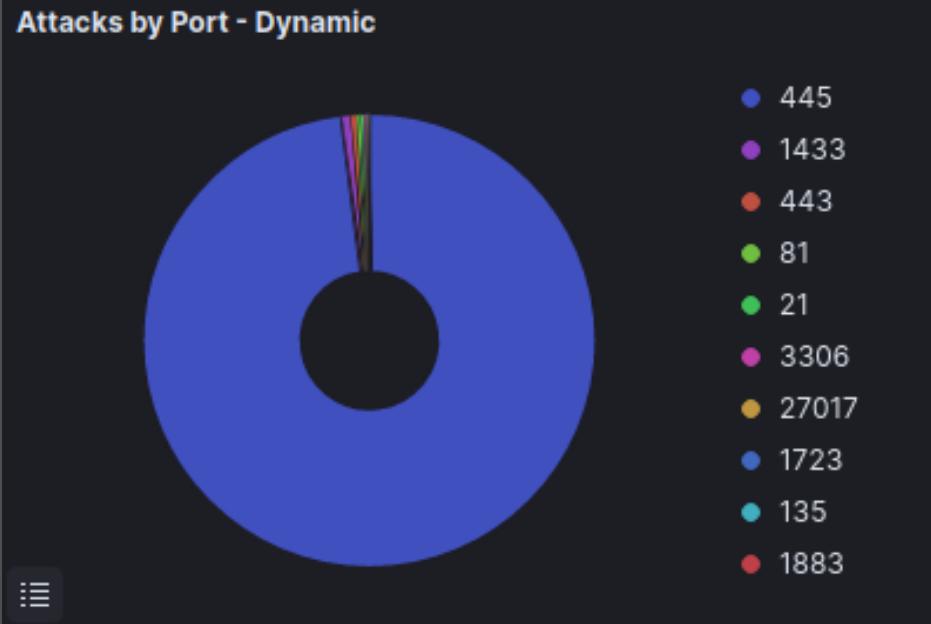
geoip.country_name.keyword: Descending	Count
Iran	91,418
Russia	66,835
India	60,120
China	57,833
Vietnam	38,380

ip_rep.keyword: Descending	Count
known attacker	5,220
mass scanner	1,058
bot, crawler	460
anonymizer	178
form spammer	55
tor exit node	13
compromised Ⓛ Ⓜ	9 Ⓛ Ⓜ

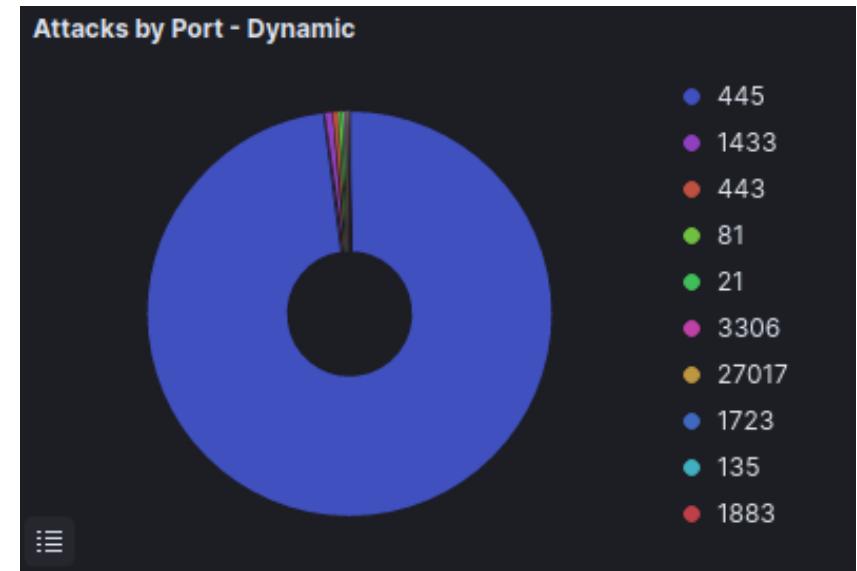


DestPort: Descending	Count
445	735,640
1433	4,813
443	3,173
81	2,208
21	1,983
3306	1,020
27017	786
1723	373
135	369
1883	246

Attacks by Port - Dynamic



Numer portu	Nazwa usługi
445	SMB
1433	SQL Server
443	HTTPS
81	TOR ?
21	FTP
3306	MySQL
27017	MongoDB
1723	PPTP
135	RPC
1883	MQTT



Numer portu	Nazwa usługi
445	SMB
1433	SQL Server
443	HTTPS
81	TOR ?
21	FTP
3306	MySQL
27017	MongoDB
1723	PPTP
135	RPC
1883	MQTT

- Microsoft CVE-2022-24500: Windows SMB Remote Code Execution Vulnerability
- Microsoft CVE-2022-29143: Microsoft SQL Server Remote Code Execution Vulnerability

sa	3,134
anonymous	465
root	306
admin	251
www	132
(empty)	99
ftp	63
administrator	61
data	61
user	61
web	61
db	60
wwwroot	59
Admin	58
user123	4

personip system  
 wwwroot (empty) www null  
 bob db admin Sa root administrator  
 Admin anonymous ftp useraccess  
 OPM1 user web user123  
 trash cron login

(empty)	457
anonymous@	268
1qaz2wsx	90
password	80
12345678	73
123456	72
!QAZ2wsx	70
1234	68
abc123	66
000000	61
admin	60
123	57
12345	57
666666	53
123123	50
112233 ⊕ ⊖	49 Ⓜ

Aa12345678 baseball password1 888888 monkey homelesspa  
 123456789a 123123123 anonymous admin 1 123 admin123 1qaz!QAZ  
 388888 qwerty abc123 1qaz2wsx 1234 112233 654321  
 111111 12345 password (empty) 123456 saadmin  
 11111111 12345 1111 anonymous@ 12345678 123456a  
 football 123321 666666 abc !QAZ2wsx 123123 Aa123456  
 1q2w3e4r5t 5201314 1234567890 123qwe !@#\$%^&\*  
 admin@123  
 yqbm4,m` ~!@~#\$/%^&(\*(),.;;



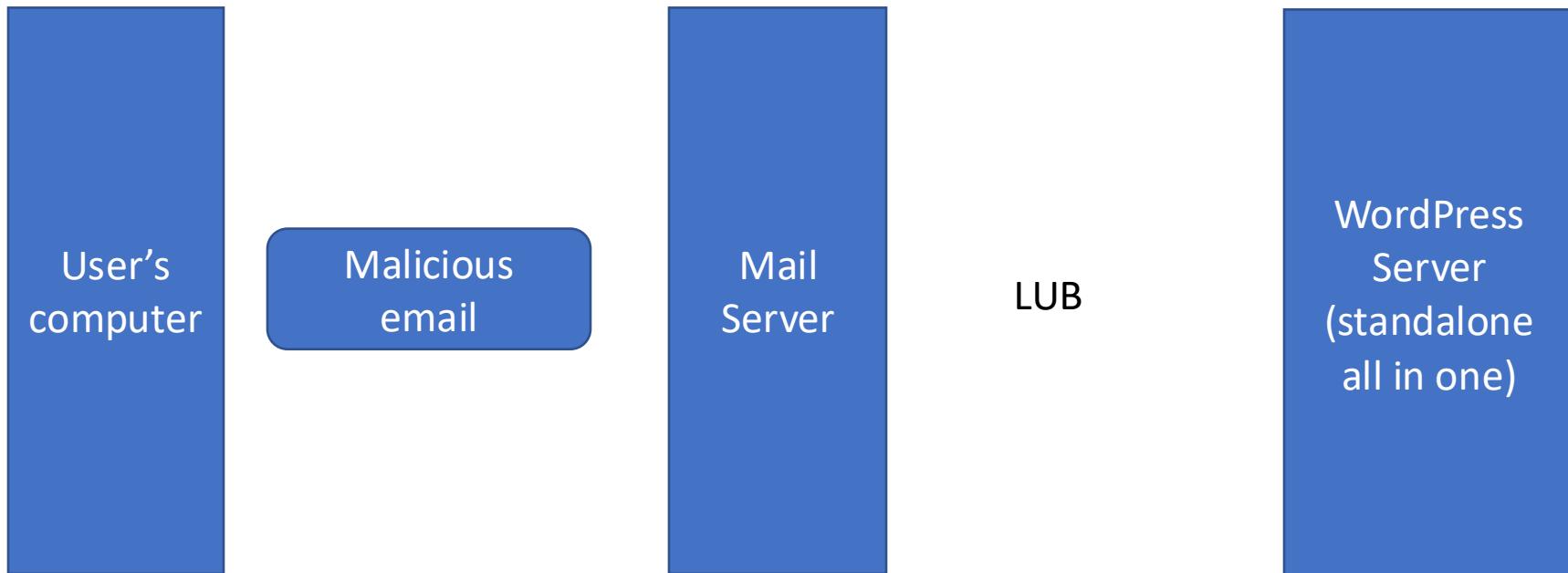
Top 100 values of alert.cve_id.keyword	Count of records
CVE-2020-11899	6,144
CVE-2019-12263 CVE-2019-12261 CVE-2019-12260 CVE-2019-12255	23
CVE-2020-11900	2
CVE-2020-11897	1

# Security is Teamwork



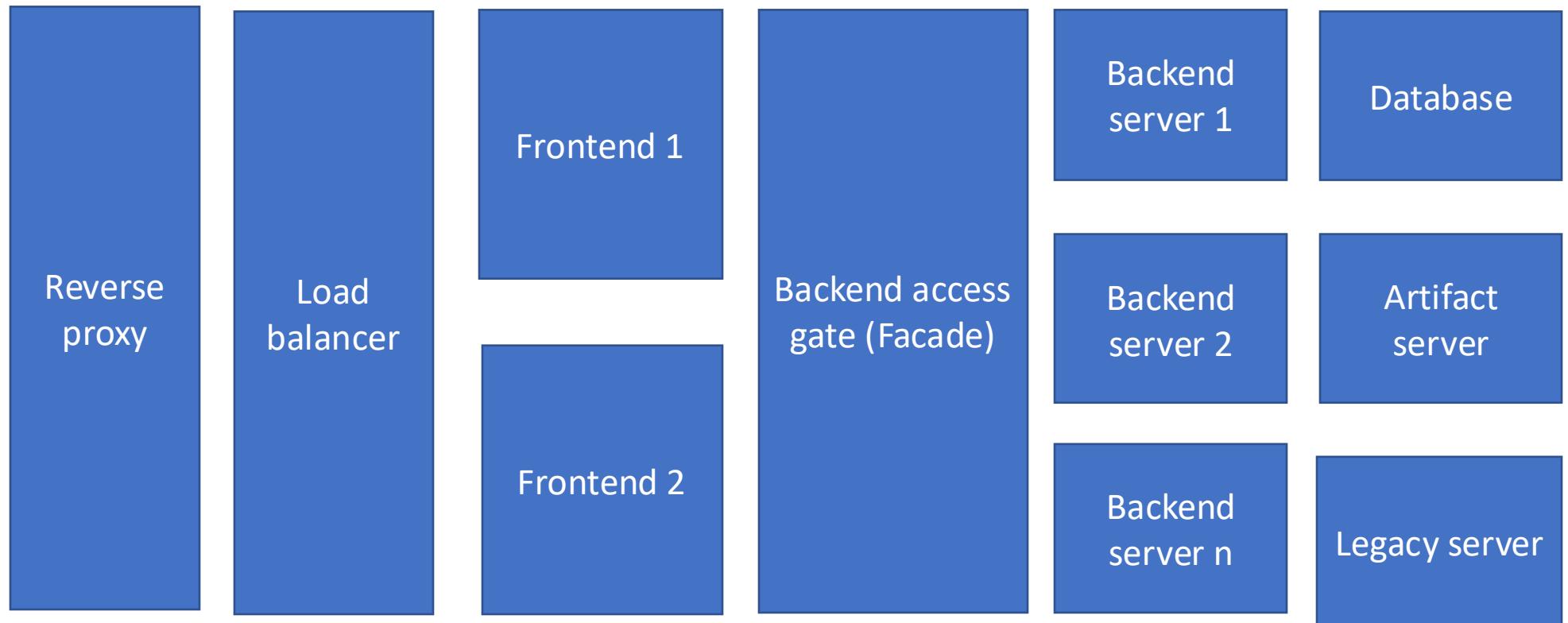
Tomasz Janczewski

# Uproszczona architektura



Prosta infrastruktura

# Aplikacje WWW typowa infrastruktura

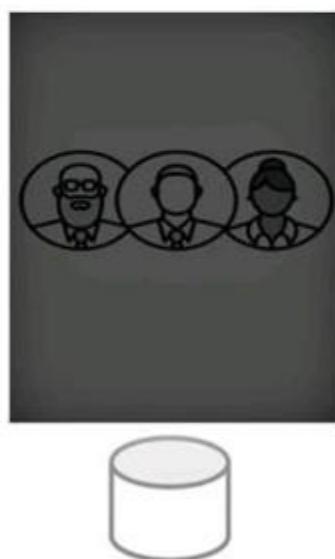


N-layers architecture często połączona z DevOps

# Microservices

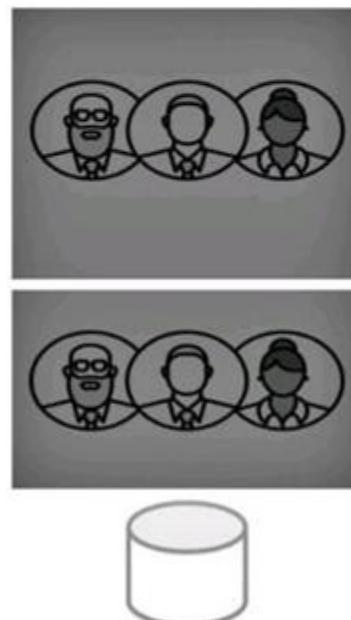
---

Past



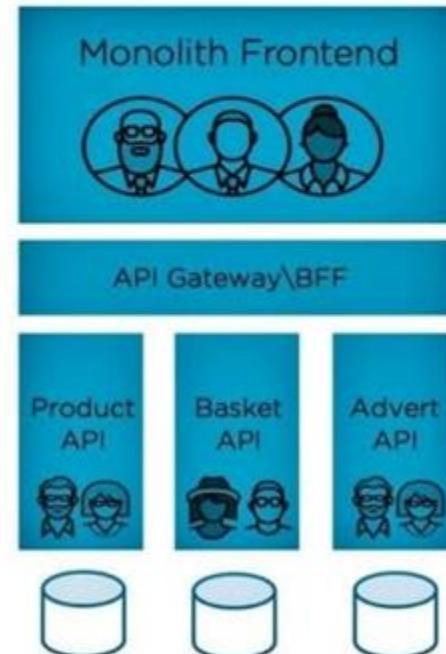
Yesterday

Frontend  
+  
Backend



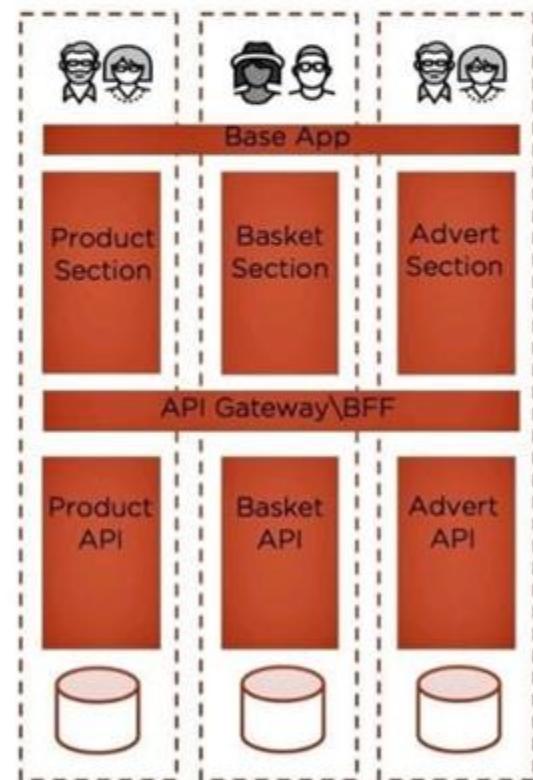
Present

Microservices



Future

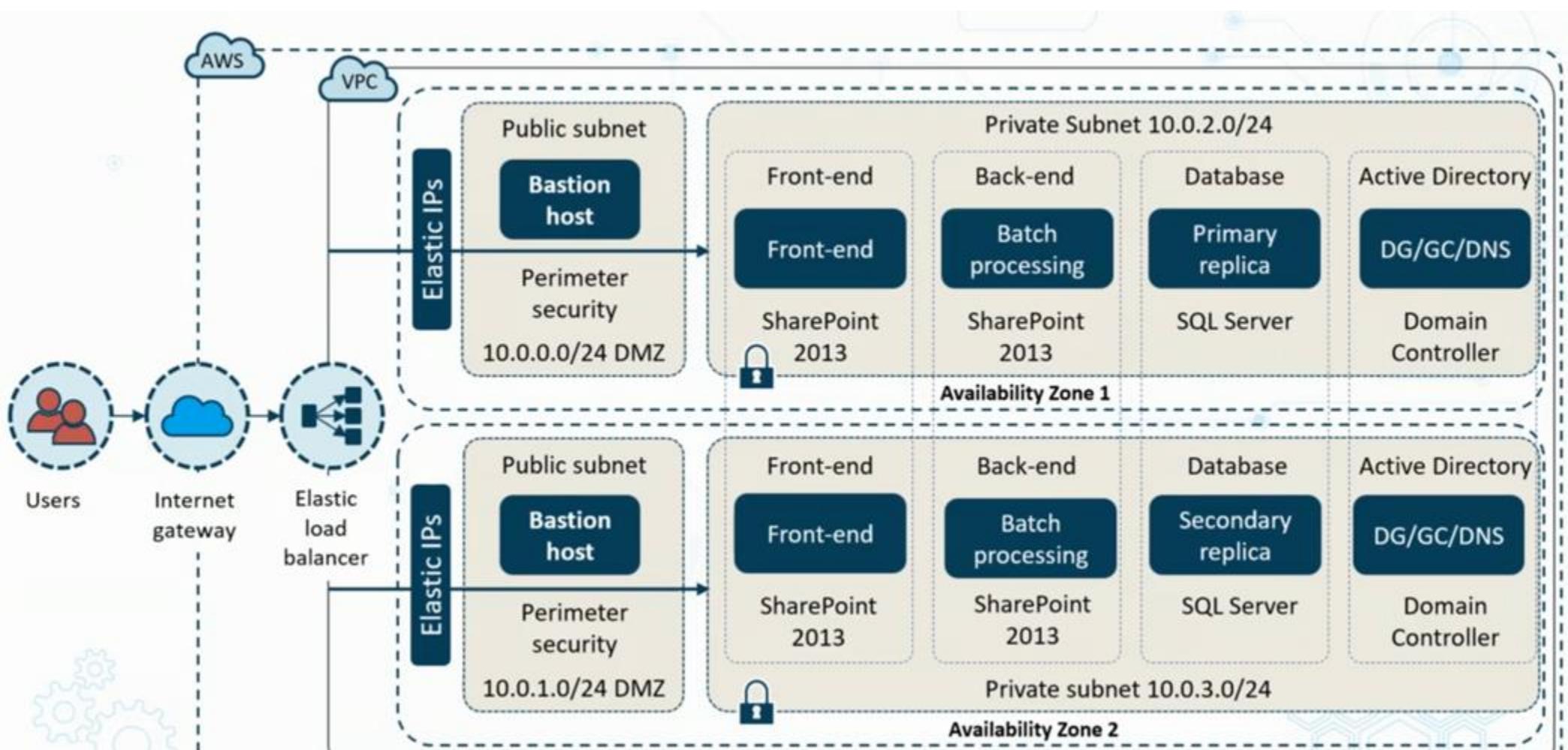
Micro Frontends  
+  
Microservices



---

# Architecture

# Infrastruktura chmurowa



# Podejście testowe

Rodzaje testów:

BLACK BOX

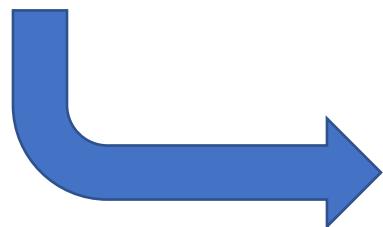
GREY BOX

WHITE BOX

- *Aplikacja WWW jako BOX*
- *Nowe podatności są widoczne dopiero po exploitacji poprzednich*
- *Brak czasu! Projekt musi być dowieziony na deadline*

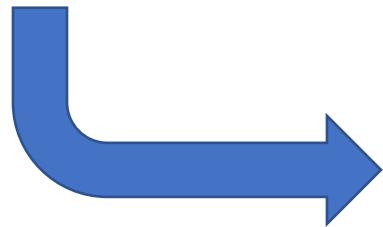
# Od czego zacząć ?

Jak zacząć?



DNS... ?

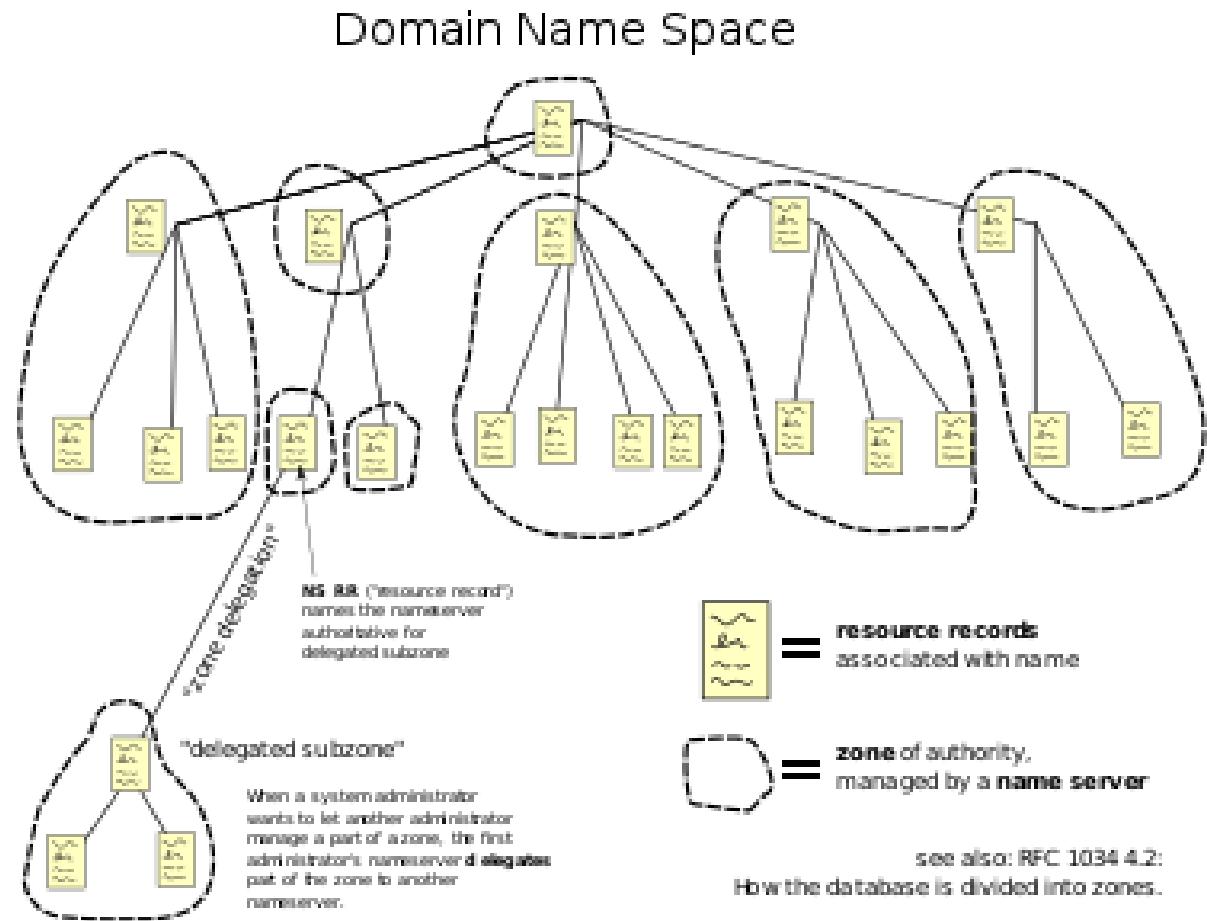
Czemu ?



*Trudno ukrywalne info o infrastrukturze...*

# DNS

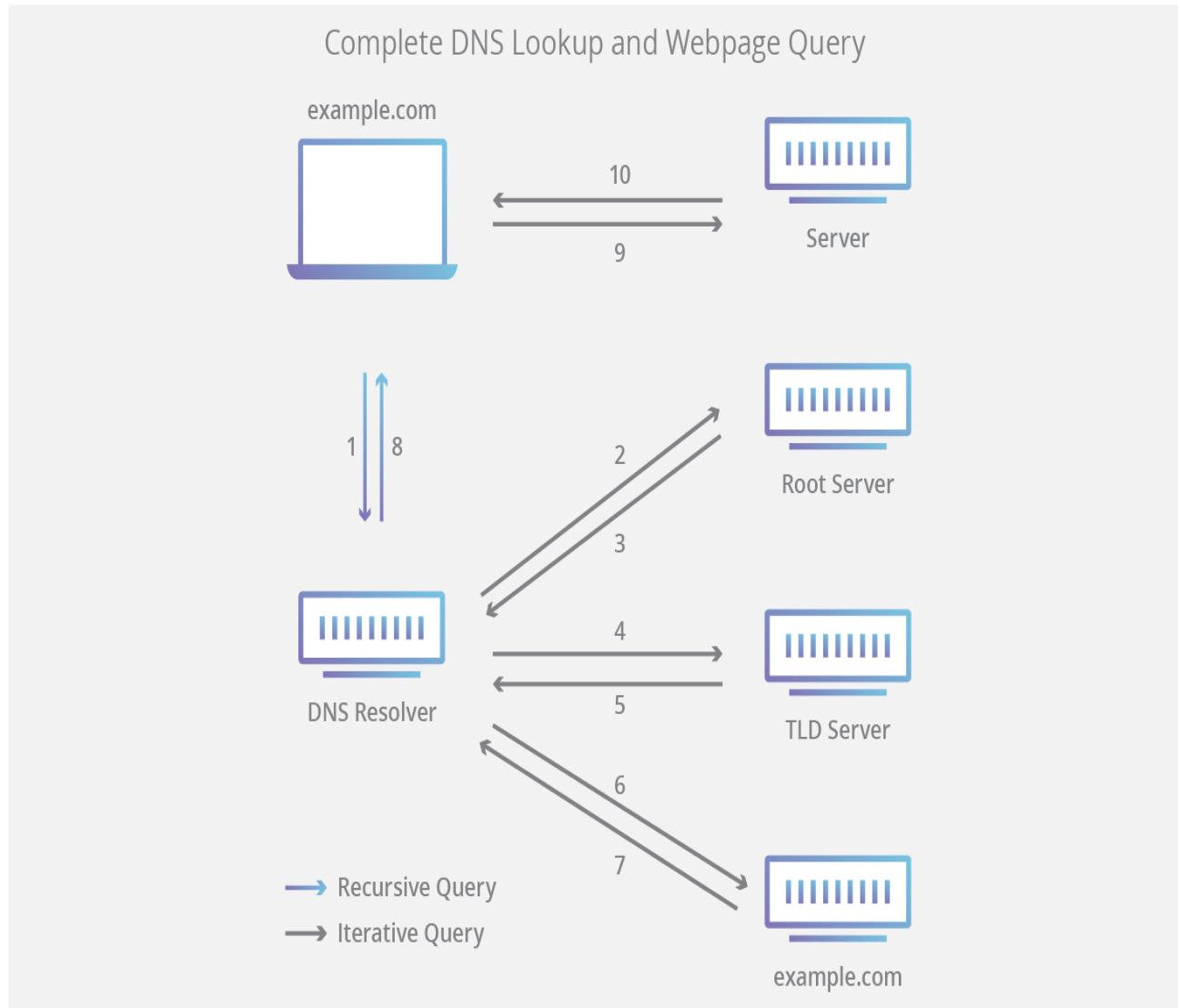
The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network.



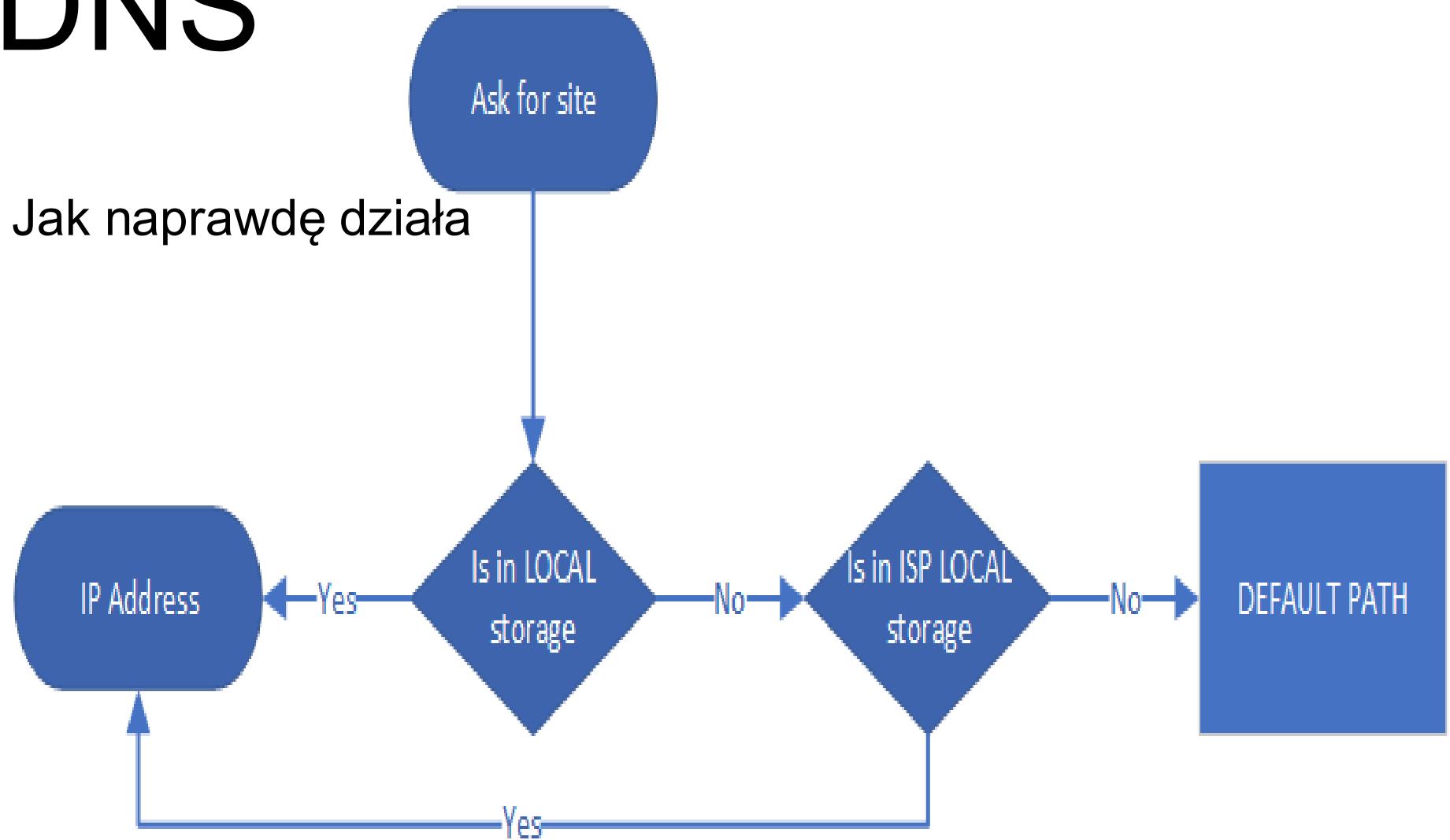
# DNS

Jak działa?

Poziom ogólny

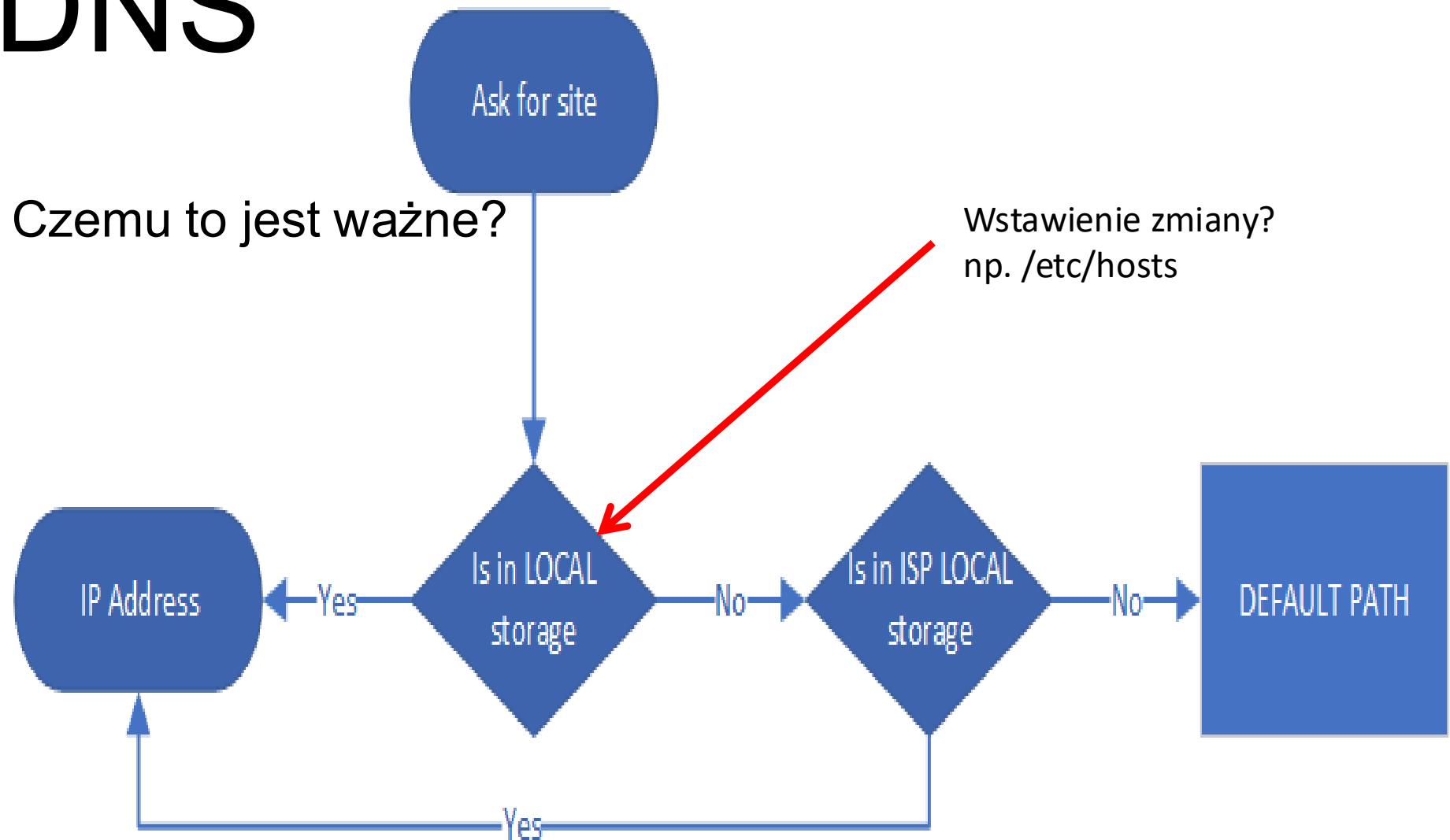


# DNS



# DNS

Czemu to jest ważne?



# DNS

Czemu to jest ważne?

/etc/hosts

127.0.0.1 your.bank.com

# DNS

## Ćwiczenie

Zapytaj o dowolną domenę, która posiada aplikację WWW.

`nslookup -type=ANY janczewski.it`

- `ANY` jest **niepewny** — nie polegaj na nim do inwentaryzacji.
- `TXT` często ujawnia SPF, DMARC, czasem info o usługach (np. Google, Microsoft).
- `SOA` i `NS` pokażą autorytywny kontekst domeny.
- `CNAME` dla `www` pokaże, czy ruch jest przekierowywany do CDN / innego hosta.
- Możesz użyć `nslookup -debug` albo `set debug` w trybie interaktywnym, aby zobaczyć dodatkowe nagłówki/TTL.

# DNS

Istotne informacje  
zwracane przez DNS ?

<https://www.ultratools.com/tools/dnsLookup>

A – IP

MX – e-mail

TXT – dowolny tekst

# DNS

## Jak bronić DNS?

DNSSEC

Użycie 8.8.8.8 / Google lub 1.1.1.1 / Claudflare

Własne skrypty w CI / CD odpytujące DNS

PI-HOLE i inne blokery / filtry DNS

[http://hole.cert.pl/domains/domains\\_hosts.txt](http://hole.cert.pl/domains/domains_hosts.txt)

# DNS

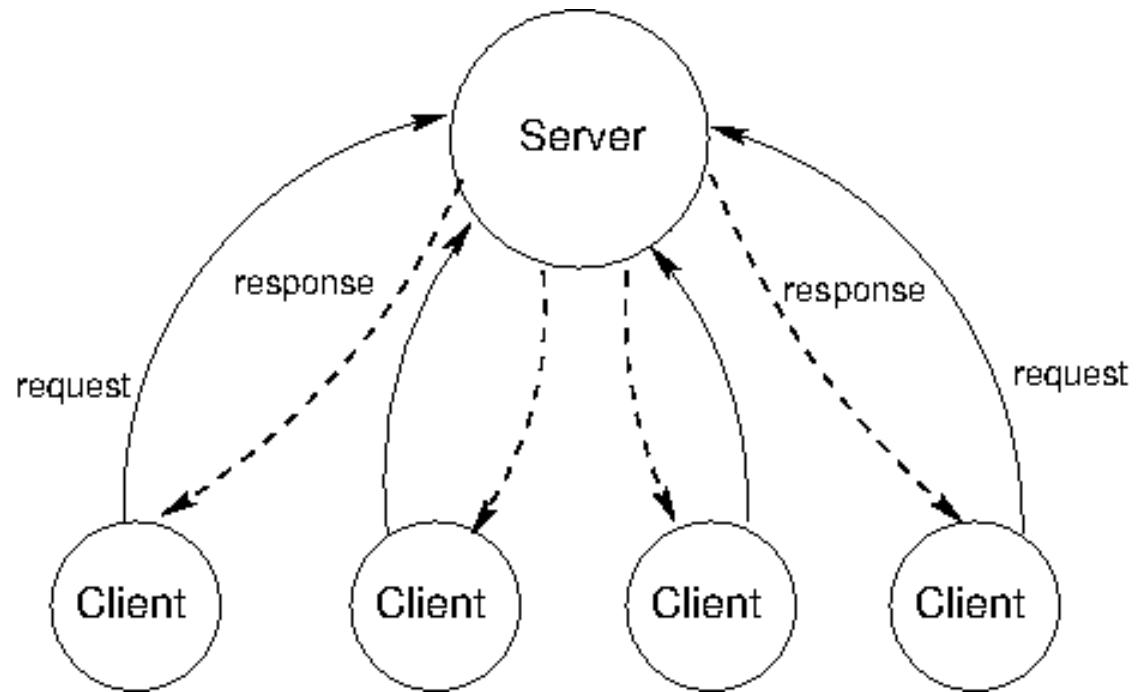
Mamy rezultaty,

Co dalej ?



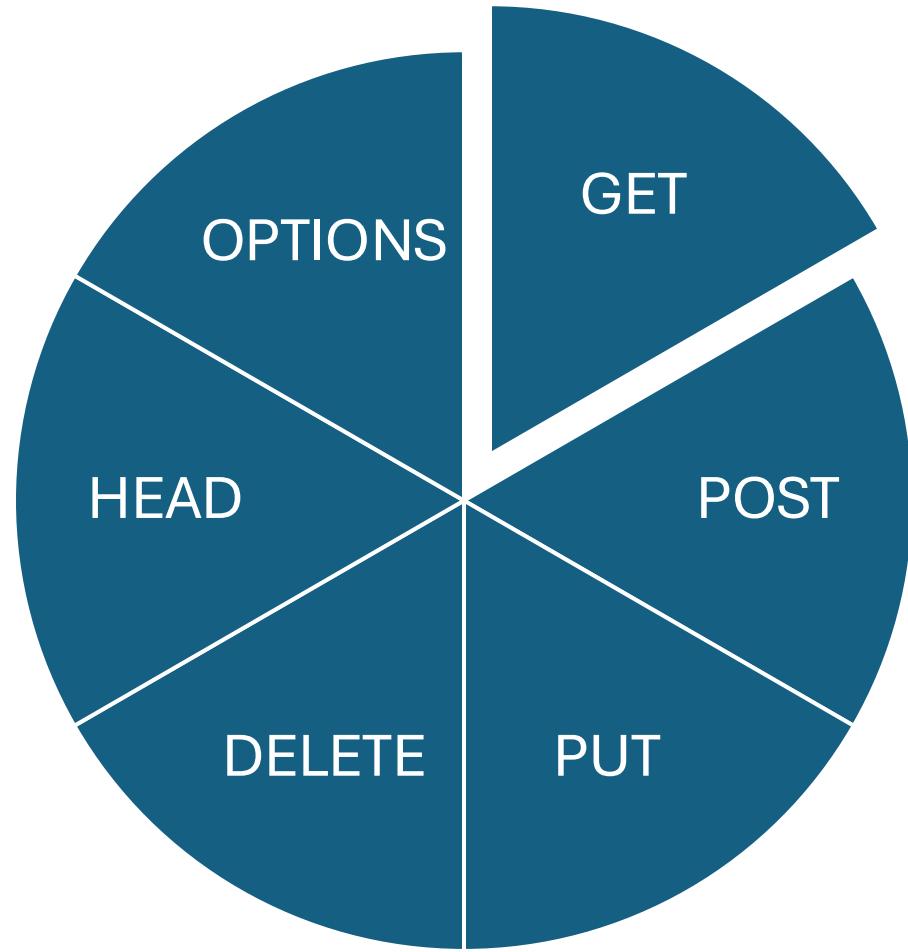
# Client Server Architecture

Jak działa WWW ?



# REST

Szybkie info jak działa  
HTTP i co to są metody  
HTTP ?



# CURL

Co to jest ?

Open-source  
command line  
klient do transferu  
zasobów web.

- Command line
- Łatwe skrypt'owanie
- Uniwersalny

# CURL

Jak pobrać ?

<https://github.com/curl/curl>

sprawdź Distribution tab!

Last version 7.67.0

# CURL

<https://httpie.org/>

Przyjazny zamiennik ?

Instalacja :

```
pip install --upgrade pip setuptools
```

```
pip install --upgrade httpie
```

# CURL / HTTPie

---

Podstawowe funkcje...

---

Expressive and intuitive syntax

---

Formatted and colorized terminal output

---

Built-in JSON support

---

Forms and file uploads

---

HTTPS, proxies, and authentication

---

Arbitrary request data

---

Custom headers

---

Persistent sessions

---

...

# CURL / HTTPie

Zapytania o WWW :

1. http -v lazarski.pl
2. http lazarski.pl > lazarski.html
3. http -- download lazarski.pl
4. http -f POST lazarski.pl user=student

Przykłady: <https://httpie.org/doc#examples>

# CURL / HTTPie

Sprawdź jakie nagłówki zwraca dowolnie wybrana strona.

Ćwiczenie:

curl for headers

```
curl -I https://lazarski.pl
```

```
curl -I https://www.lazarski.pl | findstr /R /C:"Strict-Transport-Security"
```

```
for %m in (GET HEAD OPTIONS POST) do @echo ----- %m ----- & curl -s -D - -o NUL -X %m https://www.lazarski.pl
```

```
for m in GET HEAD OPTIONS POST; do printf '----- %s -----\\n' "$m"; curl -s -D - -o /dev/null -X "$m"  
https://www.lazarski.pl; done
```

# CURL / HTTPie

```
for m in GET HEAD OPTIONS POST; do printf '----- %s -----\\n' "$m"; curl -s -D - -o /dev/null -X "$m"  
https://www.lazarski.pl; done
```

```
----- OPTIONS -----  
HTTP/2 405  
server: nginx/1.20.1  
date: Fri, 10 Oct 2025 19:15:04 GMT  
content-type: text/html  
content-length: 157  
  
----- POST -----  
HTTP/2 200  
server: nginx/1.20.1  
content-type: text/html; charset=UTF-8  
content-length: 460211  
x-powered-by: PHP/8.3.25  
cache-control: must-revalidate, no-cache, private  
date: Fri, 10 Oct 2025 19:15:05 GMT  
content-language: pl  
x-content-type-options: nosniff  
x-frame-options: SAMEORIGIN  
expires: Sun, 19 Nov 1978 05:00:00 GMT  
x-generator: Drupal 10 (https://www.drupal.org)
```

Drupal 8+ (czyli też 10, jak w tym przypadku) korzysta z frameworka Symfony.

Symfony rozpoznaje metody HTTP i jeśli zapytanie trafia do routingu, który nie ma przypisanej obsługi

`OPTIONS`, to framework sam zwraca `405`.

Co więcej, jeśli włączony jest „Route Normalizer” (co widać po nagłówku `x-drupal-route-normalizer: 1`), to znaczy, że Drupal **naprawdę uczestniczył** w przetwarzaniu tego żądania.

Zatem:

- ➡ `OPTIONS` trafiło do Drupala,
- ➡ framework rozpoznał metodę,
- ➡ ale nie znalazł pasującej ścieżki dla tej metody,
- ➡ i zwrócił `405`.

Czyli nie Nginx, tylko **Drupal (Symfony)** wygenerował tę odpowiedź.

```
----- HEAD -----  
HTTP/2 301  
server: nginx/1.20.1  
content-type: text/html; charset=utf-8  
content-length: 350  
location: https://www.lazarski.pl/pl  
x-powered-by: PHP/8.3.25  
date: Fri, 10 Oct 2025 09:34:53 GMT  
x-drupal-route-normalizer: 1  
content-language: pl  
x-content-type-options: nosniff  
x-frame-options: SAMEORIGIN  
expires: Sun, 19 Nov 1978 05:00:00 GMT  
cache-control: max-age=16588800, public  
last-modified: Fri, 10 Oct 2025 09:34:53 GMT  
etag: "1760088893"  
vary: Cookie  
x-generator: Drupal 10 (https://www.drupal.org)  
x-drupal-cache: HIT
```

**żaden WAF (Web Application Firewall), żaden reverse proxy typu Cloudflare, ModSecurity, ani żadne security gateway — nie przechwycił go wcześniej.**

# CURL / HTTPie

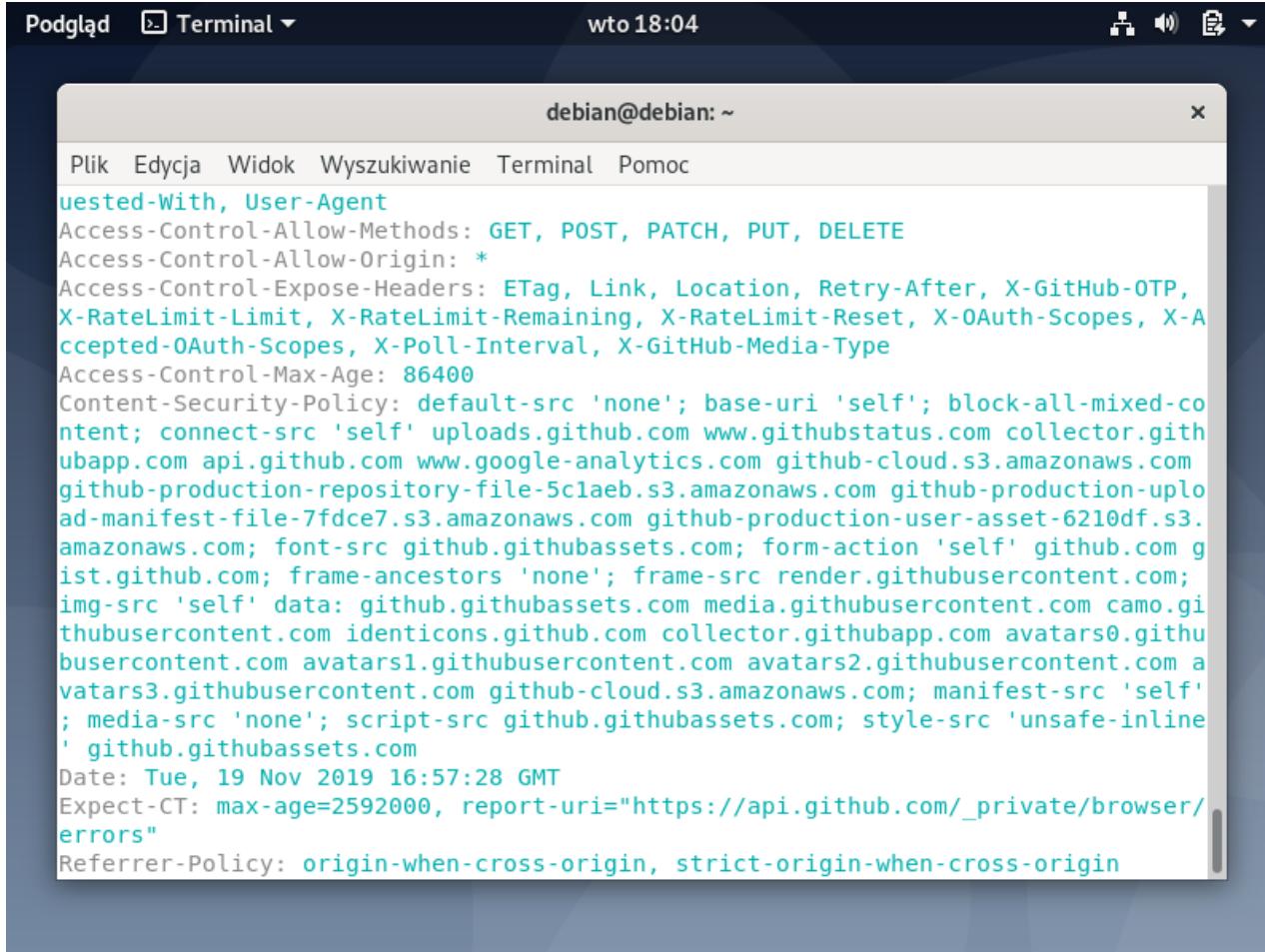
OK ale po co to robimy ?

- Automatyczne sprawdzanie vulnerabilities przez skrypty
- OPTIONS oraz HEAD zapytania

# CURL / HTTPie

http OPTIONS https://api.github.com/users

## Przykład



```
Podgląd Terminal wto 18:04
debian@debian: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
uested-With, User-Agent
Access-Control-Allow-Methods: GET, POST, PATCH, PUT, DELETE
Access-Control-Allow-Origin: *
Access-Control-Expose-Headers: ETag, Link, Location, Retry-After, X-GitHub-OTP,
X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Reset, X-OAuth-Scopes, X-A
ccepted-OAuth-Scopes, X-Poll-Interval, X-GitHub-Media-Type
Access-Control-Max-Age: 86400
Content-Security-Policy: default-src 'none'; base-uri 'self'; block-all-mixed-co
ntent; connect-src 'self' uploads.github.com www.githubstatus.com collector.gith
ubapp.com api.github.com www.google-analytics.com github-cloud.s3.amazonaws.com
github-production-repository-file-5c1aeb.s3.amazonaws.com github-production-uplo
ad-manifest-file-7fdce7.s3.amazonaws.com github-production-user-asset-6210df.s3.
amazonaws.com; font-src github.githubassets.com; form-action 'self' github.com g
ist.github.com; frame-ancestors 'none'; frame-src render.githubusercontent.com;
img-src 'self' data: github.githubassets.com media.githubusercontent.com camo.gi
thubusercontent.com identicons.github.com collector.githubapp.com avatars0.gith
ubusercontent.com avatars1.githubusercontent.com avatars2.githubusercontent.com a
vatars3.githubusercontent.com github-cloud.s3.amazonaws.com; manifest-src 'self'
; media-src 'none'; script-src github.githubassets.com; style-src 'unsafe-inline'
' github.githubassets.com
Date: Tue, 19 Nov 2019 16:57:28 GMT
Expect-CT: max-age=2592000, report-uri="https://api.github.com/_private/browser/
errors"
Referrer-Policy: origin-when-cross-origin, strict-origin-when-cross-origin
```

# CURL / HTTPie

## Przykład

Przechwycenie i filtrowanie za pomocą  
grep wersji PHP !

http -h --pretty=none OPTIONS <https://www.lazarski.pl> | grep PHP

```
root@kali:~#[https -h - pretty=none] OPTIONS https://www.lazarski.pl | grep PHP
X-Powered-By: PHP/7.1.533[:PASS] [-auth-type {basic,digest}] [--ignore-netrc]
root@kali:~#[offline] [-proxy PROTOCOL:PROXY_URL] [--follow]
```

# CURL / HTTPie

## Przykład

WordPress header.

```
--path-as-is
root@kali:~# http --headers https://www.3pio.pl
HTTP/1.1 200 OK
Connection: keep-alive
Content-Encoding: gzip
Content-Length: 4845
Content-Type: text/html; charset=UTF-8
Date: Wed, 14 Oct 2020 09:34:39 GMT
Link: <https://www.3pio.pl/wp-json/>; rel="https://api.w.org/"
Server: nginx
Strict-Transport-Security: max-age=31536000
Vary: Accept-Encoding
```

# Fuzz Faster U Fool

Fuzzing to znajdowanie błędów lub luk w zabezpieczeniach poprzez automatyczne generowanie i wysyłanie dużych ilości różnorodnych danych do aplikacji.

Na przykładzie:

***Wykrywania katalogów i plików na serwerach webowych***

<https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/common.txt>

## Słownik:

## Polecenie:

```
ffuf -w ./common.txt -u  
https://lazarski.pl/FUZZ | tee logs.txt
```

## Weryfikacja:

```
cat logs.txt | grep 200
```

```
└─(kali㉿kali)-[~]
└─$ ffuf -w SecLists/Discovery/Web-Content/common.txt -u https://lazarski.pl/FUZZ | tee l
```



v2.1.0-dev

---

```
:: Method          : GET
:: URL             : https://lazarski.pl/FUZZ
:: Wordlist        : FUZZ: /home/kali/SecLists/Discovery/Web-Content/common.txt
:: Follow redirects: false
:: Calibration     : false
:: Timeout         : 10
:: Threads         : 40
:: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
```

---

```
.ssh                  [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 55ms]
.htpasswd              [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 36ms]
.sh_history            [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 47ms]
.gitkeep               [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 53ms]
.rhosts                [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 54ms]
.bash_history          [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 54ms]
.config                [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 55ms]
.subversion             [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 59ms]
.cvsignore              [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 54ms]
.gitk                  [Status: 403, Size: 10112, Words: 839, Lines: 67, Duration: 60ms]
```

00  
Status: 200, Size: 74267, Words: 20625, Lines: 1661,  
Status: 200, Size: 90846, Words: 24236, Lines: 2027,  
Status: 200, Size: 69829, Words: 20354, Lines: 1619,  
Status: 200, Size: 74323, Words: 20629, Lines: 1661,  
Status: 200, Size: 182967, Words: 30526, Lines: 3092,  
Status: 200, Size: 15086, Words: 28, Lines: 20, Durat  
Status: 200, Size: 206537, Words: 32516, Lines: 3194,  
Status: 200, Size: 100896, Words: 22948, Lines: 1843,  
Status: 200, Size: 70739, Words: 20446, Lines: 1640,  
Status: 200, Size: 204697, Words: 32516, Lines: 3194,  
Status: 200, Size: 70722, Words: 20446, Lines: 1640,  
Status: 200, Size: 90846, Words: 24236, Lines: 2027,  
Status: 200, Size: 204697, Words: 32516, Lines: 3194,  
Status: 200, Size: 1706, Words: 133, Lines: 66, Durat  
Status: 200, Size: 85477, Words: 22970, Lines: 1950,  
Status: 200, Size: 81325, Words: 21745, Lines: 1843,  
Status: 200, Size: 70708, Words: 20446, Lines: 1640,

# Automatyczne skanery WWW

**Auto skanery są OK – jednak robią  
bałagan!**

Najczęściej używane  
skanery:

- Nikto 2
- OWASP ZAP

# NIKTO 2

Co to jest  
(według producenta) ?

Nikto is a web server assessment tool. It is designed to find various default and insecure files, configurations and programs on any type of web server.

# NIKTO 2

---

Gdzie go pobrać ?

---

Main page:

---

<https://github.com/sullo/nikto>

---

Docs:

---

<https://cirt.net/nikto2-docs/>

---

# NIKTO 2

Jak używać?

Jest wiele sposobów uruchamiania jednak najłatwiej użyć docker ...

```
git clone https://github.com/sullo/nikto.git
```

```
cd nikto
```

```
docker build -t sullo/nikto .
```

```
# Call it without arguments to display the full help
```

```
docker run --rm sullo/nikto
```

```
# Basic usage
```

```
docker run --rm sullo/nikto -h http://www.example.com
```

# NIKTO 2

## Przykładowy rezultat

```
debian@debian:~/nikto$ sudo docker run --rm sullo/nikto -h https://www.3pio.pl
- Nikto v2.1.6
-----
+ Target IP:      85.128.179.252
+ Target Hostname: www.3pio.pl
+ Target Port:    443
-----
+ SSL Info:      Subject: /CN=3pio.pl/C=PL
                  AltNames: 3pio.pl, www.3pio.pl
                  Ciphers: TLS_AES_256_GCM_SHA384
                  Issuer: /C=PL/O=nazwa.pl sp. z o.o./OU=http://nazwa.pl/CN=nazwaSSL
+ Start Time:    2019-11-19 21:20:25 (GMT0)
-----
+ Server: Apache/2
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The site uses SSL and Expect-CT header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ The Content-Encoding header is set to "deflate" this may mean that the server is vulnerable to the BREACH attack.
+ Apache/2 appears to be outdated (current is at least Apache/2.4.39). Apache 2.2.34 is the EOL for the 2.x branch.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
```

# NIKTO 2

Podpowiedzi

Skanery tworzą bałagan w logach

Skanery tworzą bałagan w aplikacji

**False positives!**

Sprawdzają tylko proste scenariusze!

# NIKTO 2

## Ćwiczenie

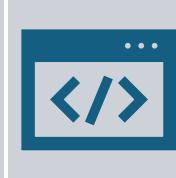
Uruchom skaner względem  
strony szkolnej i sprawdź wyniki  
względem False Positive.

# OWASP ZAP

Co to jest ?



The OWASP Zed Attack Proxy (ZAP) is one of the world's most popular free security tools.



Bardzo podobny do nikto – jednak zawiera GUI.

# OWASP ZAP

Gdzie znaleźć ?

Main page:

[https://www.owasp.org/index.php/OWASP\\_Zed\\_Attack\\_Proxy\\_Project](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)

Docs:

<https://github.com/zaproxy/zap-core-help/wiki>

# OWASP ZAP

Jak zainstalować ?

Ponieważ ZAP posiada GUI – posiada przyjemny w obsłudze interfejs okienkowy!

Sprawdź:

<https://github.com/zaproxy/zaproxy/wiki/Downloads>

**Try by Your own!**

# OWASP ZAP

Jak używać ?

Enjoy!

Start - GUI

Click Attack  
button

Wybierz -  
Automated  
Scan

Wpisz nazwę  
domeny w  
pole „URL to  
attack”

# OWASP ZAP

Jak się skończy ?

The screenshot shows the OWASP ZAP interface. At the top, there is a menu bar with tabs: History, Search, Alerts, Output, Spider, Active Scan, and a plus sign icon. Below the menu is a toolbar with icons for New Scan, Progress (showing 0: https://www.3pio.pl), and various scan controls. A progress bar is displayed, with the text '100%' highlighted by a red circle. To the right of the progress bar, it says 'Current Scans: 0 : Num requests: 16 : New Alerts: 0 : Export'. The main area is a table with the following columns: Id, Req. Timestamp, Resp. Timestamp, Method, URL, Code, Reason, RTT, Size Resp. Header, and Size Resp. Body. Below the table, there are sections for 'Alerts' (0 alerts, 1 info, 5 warnings, 0 errors) and 'Current Scans' (0 for each category).

ID	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
18	11/19/19 10:53:40 PM	11/19/19 10:53:40 PM	GET	https://www.3pio.pl/WEB-INF/classes/schema/or...	404	Not Found	409 ms	231 bytes	3,890 bytes
19	11/19/19 10:53:40 PM	11/19/19 10:53:40 PM	GET	https://www.3pio.pl/WEB-INF/classes/search_in...	404	Not Found	113 ms	231 bytes	3,890 bytes
20	11/19/19 10:53:40 PM	11/19/19 10:53:40 PM	GET	https://www.3pio.pl	200	OK	202 ms	224 bytes	28,912 bytes
21	11/19/19 10:53:40 PM	11/19/19 10:53:40 PM	GET	https://www.3pio.pl/robots.txt	404	Not Found	285 ms	231 bytes	3,890 bytes
22	11/19/19 10:53:40 PM	11/19/19 10:53:41 PM	GET	https://www.3pio.pl/sitemap.xml	404	Not Found	110 ms	231 bytes	3,890 bytes
23	11/19/19 10:53:41 PM	11/19/19 10:53:41 PM	GET	https://www.3pio.pl/robots.txt/	404	Not Found	117 ms	231 bytes	3,890 bytes
24	11/19/19 10:53:41 PM	11/19/19 10:53:41 PM	GET	https://www.3pio.pl/	200	OK	234 ms	224 bytes	28,912 bytes
25	11/19/19 10:53:41 PM	11/19/19 10:53:41 PM	GET	https://www.3pio.pl/sitemap.xml/	404	Not Found	110 ms	231 bytes	3,890 bytes

Alerts: 0 P 1 I 5 W 0 E Current Scans: 0 🚨 0 📈 0 ⚙️ 0 🔥 0 🎯 0 🕸️ 0 🖌️ 0

# OWASP ZAP

Przykładowy rezultat ...

The screenshot shows the OWASP ZAP interface with the following details:

**Toolbar:** History, Search, Alerts, Output, Spider, Active Scan, New.

**Left Panel (Alerts):**

- Folder icon: Alerts (6)
  - Flag icon: X-Frame-Options Header Not Set
    - Document icon: GET: https://www.3pio.pl (highlighted with a blue bar)
    - Flag icon: Absence of Anti-CSRF Tokens (3)
    - Flag icon: Cross-Domain JavaScript Source File Inclusion
    - Flag icon: Incomplete or No Cache-control and Pragma HTTP Header Set
    - Flag icon: Web Browser XSS Protection Not Enabled (3)
    - Flag icon: X-Content-Type-Options Header Missing

**Right Panel (Selected Alert Details):**

**X-Frame-Options Header Not Set**

URL: https://www.3pio.pl  
Risk: Medium  
Confidence: Medium  
Parameter: X-Frame-Options  
Attack:  
Evidence:  
CWE ID: 16  
WASC ID: 15  
Source: Passive (10020 - X-Frame-Options Header Scanner)  
Description:

Alerts: 0 1 5 0

# OWASP ZAP

Podpowiedzi

Skanery tworzą bałagan w logach

Skanery tworzą bałagan w aplikacji

**False positives!**

Sprawdzają tylko proste scenariusze!

# OWASP ZAP

The screenshot shows the OWASP ZAP interface with the following details:

- Alerts (11) - SQL Injection:** The main alert is for a POST request to `http://blog.thm/wp-comments-post.php`.
  - URL:** `http://blog.thm/wp-comments-post.php`
  - Risk:** High
  - Confidence:** Medium
  - Parameter:** `comment_post_ID`
  - Attack:** 16/2
  - Evidence:** None
  - CWE ID:** 89
  - WASC ID:** 19
  - Source:** Active (40018 - SQL Injection)
  - Description:** None
- Request Headers:**

```
POST http://blog.thm/wp-comments-post.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Content-Length: 140
Referer: http://blog.thm/2020/05/26/note-from-mom/?replaytocom=2
Cookie: wordpress_test_cookie=WP+Cookie+check
Host: blog.thm
```
- Request Body:**

```
comment=&author=ZAP&email=foo-bar%40example.com&url=http%3A%2F%2Fwww.example.com&submit=Post+Comment&comment_post_ID=16%2F2&comment_parent=2
```

Quick Start Request Response +

Header: Text Body: Text

```
HTTP/1.1 200 OK
Date: Wed, 15 Dec 2021 18:20:29 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Vary: Accept-Encoding
Content-Length: 2832
Content-Type: text/html; charset=utf-8
```

```

border-color: #999999;
-webkit-box-shadow: 0 0 3px rgba( 0, 115, 170, .8 );
box-shadow: 0 0 3px rgba( 0, 115, 170, .8 );
outline: none;
}

.button:active {
background: #eee;
border-color: #999;
-webkit-box-shadow: inset 0 2px 5px -3px rgba( 0, 0, 0, 0.5 );
box-shadow: inset 0 2px 5px -3px rgba( 0, 0, 0, 0.5 );
-webkit-transform: translateY(1px);
-ms-transform: translateY(1px);
transform: translateY(1px);
}

</style>
</head>
<body id="error-page">
<p><p><strong>ERROR</strong>: please type a comment.</p></p>
<p><a href='javascript:history.back()'>&laquo; Back</a></p>
</body>
</html>
```

# OWASP ZAP

## Ćwiczenie

Uruchom skaner względem  
strony szkolnej i sprawdź  
wyniki względem False  
Positive.

# Site teleportation

Co to jest ?

Uruchomienie wget  
względem strony i  
sprawdzenie kodu

LUB

Wciśnięcie F12 w  
przeglądarce i sprawdzenie  
kodu.

# Site teleportation



Czego szukamy ?

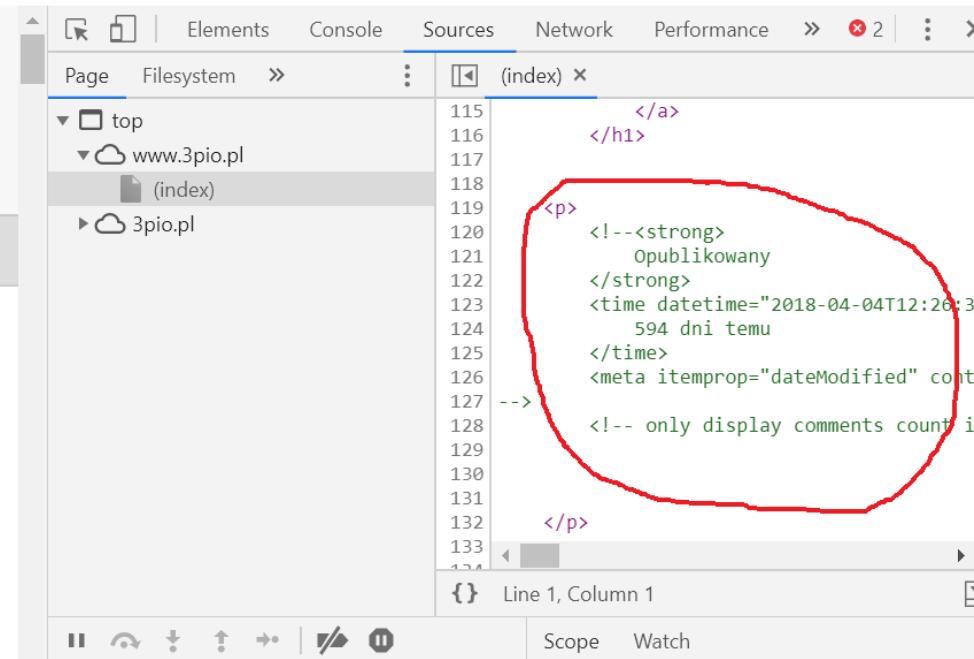
## Tomasz TJ Janczewski Blog

All You need is Java!

Strona główna Articles 

## Configuring Spring in Stand-Alone Apps

Spring is a powerful framework — and not only for dependency injection. It can strongly benefit applications as a whole. Sometimes, you need to create your own highly customized stand-alone application instead using an out-of-the-box Spring Boot



```
Elements Console Sources Network Performance > x 2 | :>
Page Filesystem >> :: (index) x
  □ top
  ▾ ▄ www.3pio.pl
    ▾ ▄ (index)
  ▾ ▄ 3pio.pl
115   </a>
116   </h1>
117
118   <p>
119     <!--<strong>
120       Opublikowany
121       </strong>
122       <time datetime="2018-04-04T12:26:3
123         594 dni temu
124       </time>
125       <meta itemprop="dateModified" cont
126       -->
127     <!-- only display comments count i
128   </p>
129
130
131
132
133
{ } Line 1, Column 1
Scope Watch
```

# Site teleportation



Czego szukamy ?

← → ⌂ ⌂ view-source:https://1lo.ostroleka.edu.pl/old/

```
1 <br />
2 <b>Fatal error</b>: Uncaught Error: Call to undefined function mysql_connect() in /old/maincore.php:302
3 Stack trace:
4 #0 /old/maincore.php(91): dbconnect('localhost', '32671262_000041', 'ANmX421fANmX421...', '32671262_000041')
5 #1 /old/index.php(18): require_once('/old/maincore.p...')
6 #2 {main}
7 thrown in <b>/old/maincore.php</b> on line <b>302</b><br />
8
```

# Site teleportation

Co można znaleźć w komentarzach ?

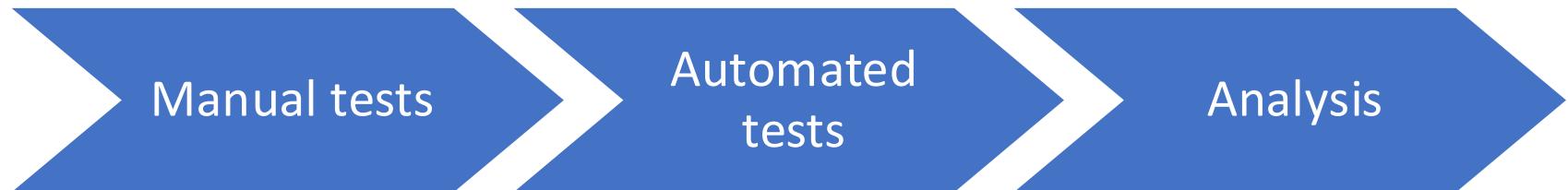
- Domyślne hasła
- Informacje o stosie technologicznym
- Informacje o błędach – komunikaty błędów
- ...

# Filtering

Nie wszystko jest prawdziwe!

- Sprawdzenie FALSE POSITIVES
- HONEY POT
- Developers / DevOps / Admins automatyczne alarmy
- Alarm triggering
- ...

# Idealny proces

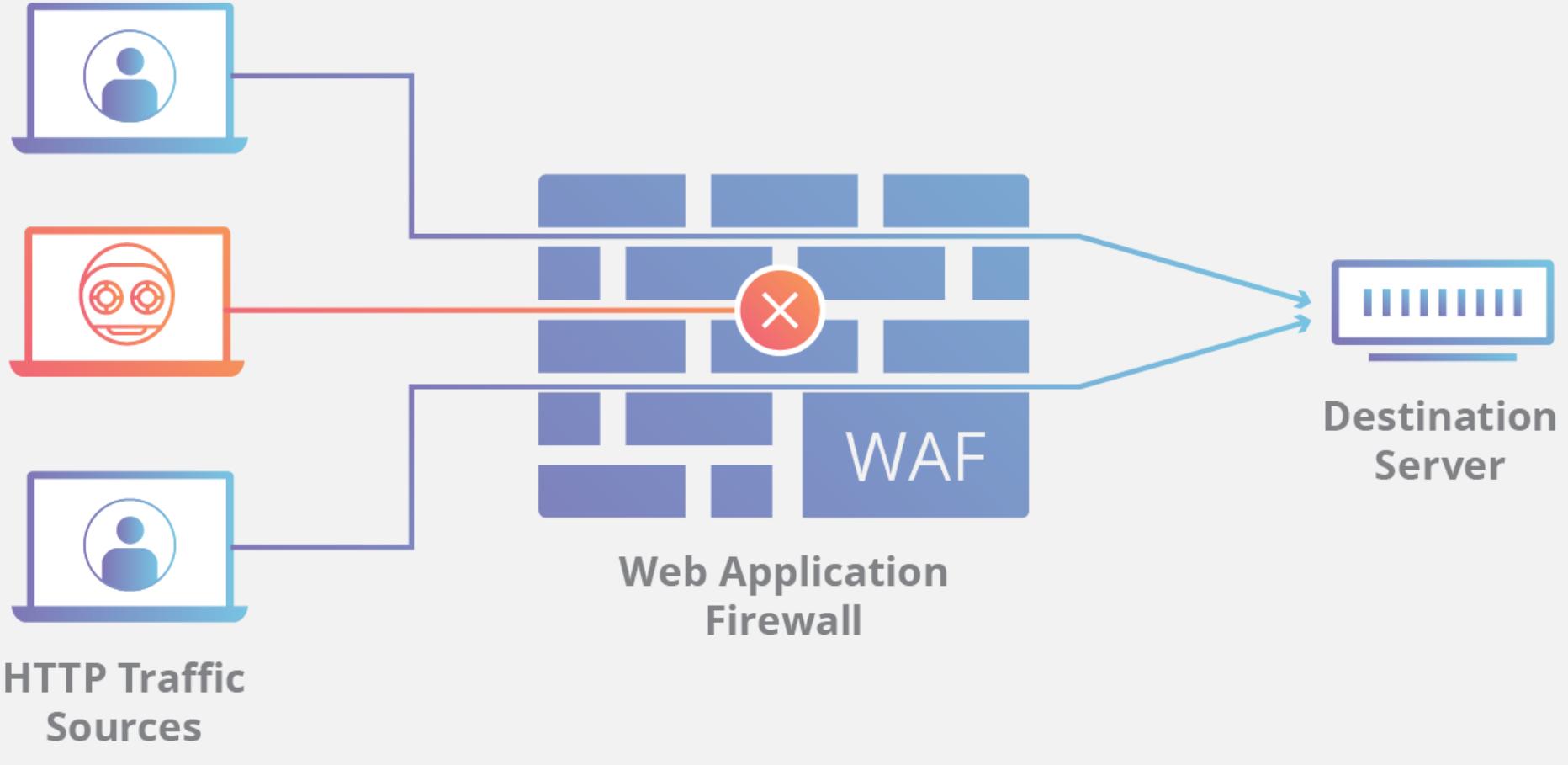


# WAF

Co to jest WAF ?

A WAF or Web Application Firewall helps protect web applications by **filtering and monitoring HTTP traffic between a web application and the Internet**. It typically protects web applications from attacks such as cross-site forgery, cross-site-scripting (XSS), file inclusion, and SQL injection, among others.

# WAF



# WAF

Typy WAF

Blacklisted WAF

Whitelisted WAF

# WAF

## Blacklisted WAF



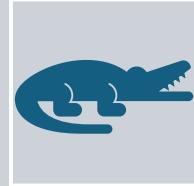
Protects against known attacks.



Check request payload and match with known attack vector.

# WAF

## Whitelisted WAF



Protects against known attacks, by accepting only pre-approved traffic.



Like accepting only letters from text input at the www page.

# WAF

## Przykłady WAF



Cloudflare WAF



Akamai Kona Site Defender



Amazon Web Services WAF



Incapsula Web Application Firewall

# WAF

## Pokonywanie WAF

- .Zmiana kodowania znaków
- .File upload
- .Zmiana notacji sql na scientific notation
- .<svg onload=alert(1)/>
- ....

[https://app.grammarly.com/docs/new?config=%22account%22:{%22subscription%22:%22javascript:alert\(document.domain\)//%22},%22api%22:{%22redirect%22:%22javascript:alert\(document.domain\)//%22}}](https://app.grammarly.com/docs/new?config=%22account%22:{%22subscription%22:%22javascript:alert(document.domain)//%22},%22api%22:{%22redirect%22:%22javascript:alert(document.domain)//%22}})

# Dziękuję.