SQL инъекции

да, опять

SQL

- Язык запросов к базам данных
- Много диалектов (MySQL, PostgreSQL, MSSQL, …)
- Можно найти везде
- Полезные фичи для инъекций

SQL

- **Язык запросов к базам данных**
- Много диалектов (**MySQL**, PostgreSQL, MSSQL, …)
- Можно найти везде
- Полезные фичи для инъекций

Чуть-чуть про синтаксис

- Типы запросов

  SELECT, INSERT, DELETE, CREATE, UPDATE, ALTER TABLE…

Чуть-чуть про синтаксис

- Типы запросов

  SELECT, INSERT, DELETE, CREATE, **UPDATE**, ALTER TABLE…

Чуть-чуть про синтаксис

- Типы запросов

  SELECT, INSERT, DELETE, CREATE, **UPDATE**, ALTER TABLE…
- Модификаторы

  WHERE (есть булевы модификаторы), ORDER BY, LIKE, LIMIT,

  INTO OUTFILE…

Чуть-чуть про синтаксис

- Типы запросов

  SELECT, INSERT, DELETE, CREATE, **UPDATE**, ALTER TABLE…
- Модификаторы

  WHERE (есть булевы модификаторы), ORDER BY, **LIKE**, LIMIT,

  INTO OUTFILE…
- Функции

  COUNT, **LOAD_FIL**E, **CONCAT**, **SUBSTR**, **SLEEP, …**

Чуть-чуть про синтаксис

- Типы запросов

  SELECT, INSERT, DELETE, CREATE, **UPDATE**, ALTER TABLE…
- Модификаторы

  WHERE (есть булевы модификаторы), ORDER BY, **LIKE**, LIMIT,

  INTO OUTFILE…
- Функции

  COUNT, **LOAD_FIL**E, **CONCAT**, **SUBSTR**, **SLEEP, …**
- **UNION**

Чуть-чуть про синтаксис

https://www.w3schools.com/sql/

- Типы запросов

  SELECT, INSERT, DELETE, CREATE, **UPDATE**, ALTER TABLE…
- Модификаторы

  WHERE (есть булевы модификаторы), ORDER BY, **LIKE**, LIMIT,

  INTO OUTFILE…
- Функции

  COUNT, **LOAD_FIL**E, **CONCAT**, **SUBSTR**, **SLEEP, …**
- **UNION**

SQL

- Язык запросов к базам данных
- Много диалектов (MySQL, PostgreSQL, MSSQL, …)
- Можно найти везде
- **Полезные фичи для инъекций**

Information_schema

- Служебная БД в MySQL
- Есть аналоги в других СУБД (гуглите)
- Можно восстановить структуру таблиц

Information_schema

- `SELECT schema_name FROM information_schema.schemata;`

  ^ достать все БД в текущей базе

- `SELECT table_schema, table_name FROM information_schema.tables WHERE table_schema != 'mysql' AND table_schema != 'information_schema'`

  ^ достать все таблицы из базы

- `SELECT table_schema, table_name, column_name FROM information_schema.columns WHERE table_schema != 'mysql' AND table_schema != 'information_schema'`

  ^ достать все колонки из базы

(условия можно менять)

Filter bypass

- Комменты (-- # /**/)

Filter bypass

- Комменты (-- # /**/)
- Другие представления строк

  unhex(), char(), …

Filter bypass

- Комменты (-- # /**/)
- Другие представления строк

  unhex(), char(), …
- Замены пробелу – многострочные комменты, вызов конструкций как функций (`union(select(1),2,3)`)

Filter bypass

- Комменты (-- # /**/)
- Другие представления строк

  unhex(), char(), …
- Замены пробелу – многострочные комменты, вызов конструкций как функций (`union(select(1),2,3)`)
- Другие регистры (SELECT != SeLeCt)

Filter bypass

- Комменты (-- # /**/)
- Другие представления строк

  unhex(), char(), …
- Замены пробелу – многострочные комменты, вызов конструкций как функций (`union(select(1),2,3)`)
- Другие регистры (SELECT != SeLeCt)
- …
- https://websec.wordpress.com/2010/12/04/sqli-filter-evasion-cheat-sheet-mysql/

«Слепые» инъекции

- Получаем данные не напрямую
  sleep(), benchmark

«Слепые» инъекции

- Получаем данные не напрямую
  sleep(), benchmark
- Количество строк в ответе

«Слепые» инъекции

- Получаем данные по косвенным признакам:
  sleep(), benchmark()
- Количество строк в ответе
- Сам ответ (отличается в зависимости от результата запроса)

Error-based инъекции

- Получаем данные в ошибках от базы
- `SELECT COUNT(*) FROM (SELECT 1 UNION SELECT 2)x GROUP BY MID([YOUR_QUERY], FLOOR(RAND(33)*2), 64)`
- `SELECT updatexml(null, concat(char(123),VERSION(),char(125)))`
- `SELECT 2*(if((SELECT * from (SELECT (version()))s), 18446744073709551610, 18446744073709551610));`
- `select exp(~(select*from(select user())x));`
- …
- Google `sql injection error based`

https://rdot.org/forum/showthread.php?t=3167

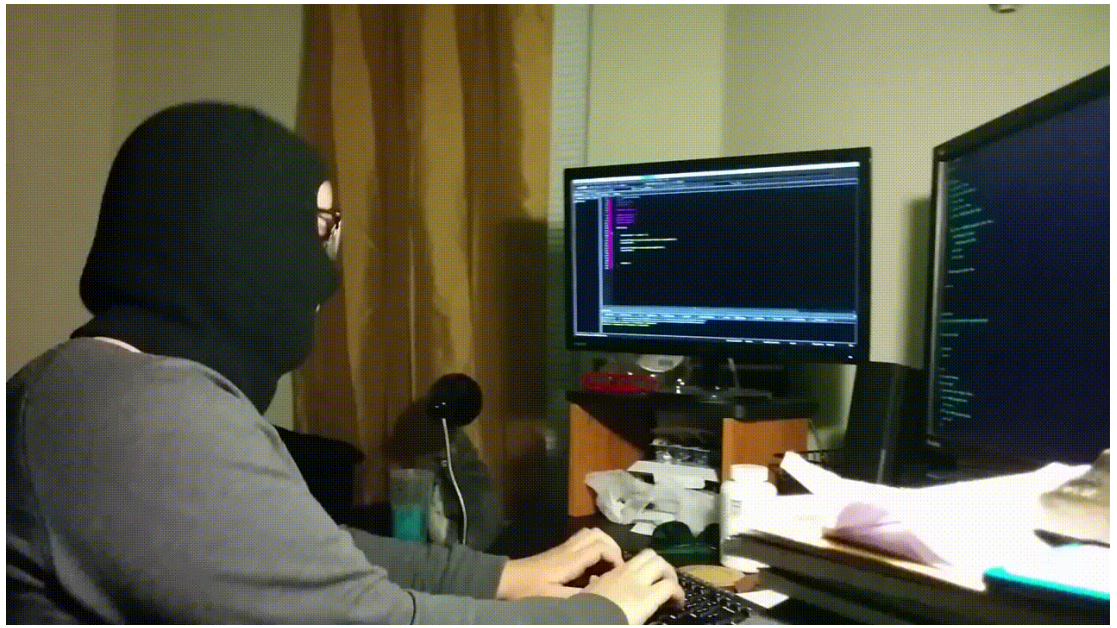
https://www.exploit-db.com/docs/english/37953-mysql-error-based-sql-injection-using-exp.pdf

https://intsystem.org/security/error-based-sql-injection-in-mysql/

sqlmap

https://github.com/sqlmapproject/sqlmap

- Делает хорошо и почти все сам
- Умеет в много баз
- Умеет в много типов инъекций
- Опенсурсный
- Может помочь при постэксплуатации



- Делает из тебя l33t h4x0r

Hack the world

sqlmap
-u “https://ifmo.ru”

--data “login=313373&password”

--level=5

--risk=3

https://github.com/sqlmapproject/sqlmap/wiki/Usage

ну давай разберем по частям, тобою написаное))

-u “https://ifmo.ru”	- что атакуем
--data “login=313373&password”	- пост параметры (без них автоматом гет)
--level=5	- «глубина» исследования
--risk=3	- насколько можно шуметь

Все остальное смотреть по ссылке (там много всего полезного)

https://github.com/sqlmapproject/sqlmap/wiki/Usage