Offensive Approach to Hunt Bugs

SQLi

Injection Point for SQL Injection

- SQL Injection can be GET Based
- SQL Injection can be POST Based
- SQL Injection can be Header Based
- SQL Injection can be Cookie Based

Learn SQLi Query Fixing

SQLI Get Based

- Find Injection Point
- Identify Vulnerability
- Balance the Query
- Try to Inject SQLI Statement there

Double Query SQLI

- error/double based sqli query -> hackbar->error/double->get database

Blind Boolien Based sqli

- Balance Query is necessary
- And 1=1 (True)
- OR 1=1 (True)
- And 1=2 (False)
- OR 1=2 (False)

Blind Time Based sqli

- Balance Query is necessary
- And SLEEP(10) if sleep then Vulnerable
- OR SLEEP(10) if sllep then vulnerable

Exploitation of SQLI

Post based SQLI

- Find Injection Point
- Identify Vulnerability
- Balance Query
- Try to execute any SQLI Statement there
- Inject Database Query

Header Based SQLI

- You have to look for Headers Parameter to find Injection Point such as Host | User-Agent | Referrer | Location

Cookie Based SQLI

- Find any Cookie parameter and try to execute any SQLi Statement

Waf Bypassing for SQLI

- Web application Firewall Bypassing
- WAF Filter Malicious illegal input
- There are many techniques to bypass waf

  Read this - https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF

Authentication Bypassing

- The error message includes the SQL query used by the login function. We can use this information to construct an injection attack to bypass authentication. The first account in a database is often an administrative user, we can exploit this behavior to log in as the first user in the database.

Automation of SQLI GET Based

- Requirement

  Install Python 2.7 on your Environment

- Download SQLMAP Zip file from sqlmap.zip
- Command for GET Based SQLI

  Basic Command

  python sqlmap.py -u “URL” --batch –banner

- Advance Command

  python sqlmap.py -u “URL” --level=5 --risk=3 --keep-alive --fresh-queries --random-agent --batch --banner

Automation of SQLI POST Based | Header Based | Cookie Based

- Requirement

  Install Python 2.7 on your Environment

- Download SQLMAP Zip file from sqlmap.zip
- Command for POST Based SQLI

  Basic Command

  python sqlmap.py -r requestfile.txt --batch –banner

- Advance Command

  python sqlmap.py -r requestfile --level=5 --risk=3 --keep-alive --fresh-queries --random-agent --batch --banner

Automation of SQLI with WAF Bypassed

- Requirement

  Install Python 2.7 on your Environment

- Download SQLMAP Zip file from sqlmap.zip

  Learn more -

  https://forum.bugcrowd.com/t/sqlmap-tamper-scripts-sql-injection-and-waf-bypass/423