Category 1: SQL Injection Variants (40 Items)

1. **SQL Injection (Classic)** – Exploiting unsanitized SQL queries. **Critical**
2. **SQL Injection (Union Variant 1)** – Using the UNION operator to extract data. **Critical**
3. **SQL Injection (Error Variant 1)** – Leveraging error messages to obtain database details. **Critical**
4. **SQL Injection (Blind Variant 1)** – Inferring data structure via blind techniques. **High**
5. **SQL Injection (Time-Based Variant 1)** – Using deliberate delays to infer information. **High**
6. **SQL Injection (Second-Order Variant 1)** – Injected payload stored and executed later. **High**
7. **SQL Injection (Stacked Queries Variant 1)** – Executing multiple queries in one request. **Critical**
8. **SQL Injection (Batch Query Variant 1)** – Combining multiple SQL statements. **Critical**
9. **SQL Injection (Conditional Error Variant 1)** – Triggering specific errors conditionally. **Critical**
10. **SQL Injection (Extended Union Variant 1)** – Extended use of UNION to match column counts. **Critical**
11. **SQL Injection (Comment-Based Variant 1)** – Using comments to alter query logic. **Critical**
12. **SQL Injection (MySQL Specific Variant 1)** – Targeting MySQL peculiarities. **Critical**
13. **SQL Injection (PostgreSQL Specific Variant 1)** – Exploiting PostgreSQL query constructs. **Critical**
14. **SQL Injection (MSSQL Specific Variant 1)** – Taking advantage of Microsoft SQL Server features. **Critical**
15. **SQL Injection (Oracle Specific Variant 1)** – Exploiting Oracle SQL syntax. **Critical**
16. **SQL Injection (SQLite Specific Variant 1)** – Targeting SQLite's query processing. **Critical**
17. **SQL Injection (Stored Procedure Variant 1)** – Injecting into stored procedures. **Critical**
18. **SQL Injection (Parameterized Query Bypass Variant 1)** – Bypassing weak parameterization. **Critical**
19. **SQL Injection (Dynamic SQL Variant 1)** – Exploiting dynamically built queries. **Critical**
20. **SQL Injection (ORM Bypass Variant 1)** – Evading ORM protections. **Critical**
21. **SQL Injection (URL Parameter Variant 1)** – Injection via query string parameters. **Critical**
22. **SQL Injection (Form Field Variant 1)** – Injection through user input fields. **Critical**
23. **SQL Injection (Header Injection Variant 1)** – Injecting via HTTP

headers. **Critical**

24. **SQL Injection (Cookie Injection Variant 1)** – Injection through cookie values. **Critical**
25. **SQL Injection (JSON Payload Variant 1)** – Exploiting JSON data passed to SQL. **Critical**
26. **SQL Injection (XML Payload Variant 1)** – Injecting SQL through XML data. **Critical**
27. **SQL Injection (Union Variant 2)** – A second variant of UNION-based injection. **Critical**
28. **SQL Injection (Error Variant 2)** – Alternate error-based injection method. **Critical**
29. **SQL Injection (Blind Variant 2)** – Another blind technique. **High**
30. **SQL Injection (Time-Based Variant 2)** – Alternate delay-based variant. **High**
31. **SQL Injection (Second-Order Variant 2)** – A second twist on stored injection. **High**
32. **SQL Injection (Stacked Queries Variant 2)** – Another stacked queries technique. **Critical**
33. **SQL Injection (Batch Query Variant 2)** – A different batch exploitation method. **Critical**
34. **SQL Injection (Comment-Based Variant 2)** – Variant using comment termination. **Critical**
35. **SQL Injection (Conditional Error Variant 2)** – Conditional error exploitation variant. **Critical**
36. **SQL Injection (Extended Union Variant 2)** – Another UNION extension technique. **Critical**
37. **SQL Injection (MySQL Specific Variant 2)** – A second MySQL-focused variant. **Critical**
38. **SQL Injection (PostgreSQL Specific Variant 2)** – Alternate PostgreSQL injection. **Critical**
39. **SQL Injection (MSSQL Specific Variant 2)** – Another Microsoft SQL Server variant. **Critical**
40. **SQL Injection (Oracle Specific Variant 2)** – Alternate Oracle SQL injection. **Critical**

Category 2: NoSQL Injection (20 Items)

1. **NoSQL Injection (Classic)** – Exploiting unsanitized NoSQL queries. **High**
2. **NoSQL Injection (MongoDB Variant 1)** – Targeting MongoDB query structures. **High**
3. **NoSQL Injection (CouchDB Variant 1)** – Exploiting CouchDB's JSON queries. **High**
4. **NoSQL Injection (Redis Variant 1)** – Manipulating Redis commands

via injection. **High**

5. **NoSQL Injection (Dynamic Query Variant 1)** – Injecting into dynamically built NoSQL queries. **High**
6. **NoSQL Injection (JSON Query Variant 1)** – Injection through JSON-based queries. **High**
7. **NoSQL Injection (URL Parameter Variant 1)** – Injection via URL parameters in NoSQL APIs. **High**
8. **NoSQL Injection (Form Field Variant 1)** – Using form input to manipulate NoSQL queries. **High**
9. **NoSQL Injection (Header Variant 1)** – Injecting via HTTP headers in NoSQL endpoints. **High**
10. **NoSQL Injection (Cookie Variant 1)** – Exploiting cookie data in NoSQL contexts. **High**
11. **NoSQL Injection (JSON Payload Variant 2)** – A second JSON payload variant. **High**
12. **NoSQL Injection (XML Payload Variant 1)** – Injection using XML that drives NoSQL queries. **High**
13. **NoSQL Injection (Error-Based Variant 1)** – Extracting data via error messages. **High**
14. **NoSQL Injection (Time-Based Variant 1)** – Using response delays for inference. **High**
15. **NoSQL Injection (Blind Variant 1)** – Blind techniques in NoSQL contexts. **High**
16. **NoSQL Injection (Cassandra Specific Variant 1)** – Targeting Cassandra query languages. **High**
17. **NoSQL Injection (Dynamic Parameter Variant 1)** – Injecting via dynamic query parameters. **High**
18. **NoSQL Injection (Aggregation Pipeline Variant 1)** – Manipulating NoSQL aggregations. **High**
19. **NoSQL Injection (Stored Query Variant 1)** – Injecting into stored NoSQL queries. **High**
20. **NoSQL Injection (Expression Variant 1)** – Exploiting expression-based queries. **High**

Category 3: LDAP Injection (10 Items)

21. **LDAP Injection (Classic)** – Exploiting unsanitized LDAP queries. **High**
22. **LDAP Injection (Bind Variant 1)** – Injection affecting LDAP bind operations. **High**
23. **LDAP Injection (Search Filter Variant 1)** – Altering LDAP search filters. **High**
24. **LDAP Injection (Username Field Variant 1)** – Injection through username inputs. **High**
25. **LDAP Injection (Email Field Variant 1)** – Injection via email-based

LDAP queries. **High**

26. **LDAP Injection (URL Parameter Variant 1)** – Using URL parameters to inject LDAP commands. **High**
27. **LDAP Injection (Form Field Variant 1)** – Injection through web form fields. **High**
28. **LDAP Injection (Header Variant 1)** – Manipulating HTTP headers to affect LDAP. **High**
29. **LDAP Injection (Cookie Variant 1)** – Injection via cookie data in LDAP queries. **High**
30. **LDAP Injection (Dynamic Query Variant 1)** – Exploiting dynamic LDAP query building. **High**

Category 4: OS Command Injection (20 Items)

31. **OS Command Injection (Classic)** – Injecting unsanitized commands into OS calls. **Critical**
32. **OS Command Injection (Shell Variant 1)** – Exploiting shell command contexts. **Critical**
33. **OS Command Injection (Bash Variant 1)** – Targeting Bash-specific command parsing. **Critical**
34. **OS Command Injection (PowerShell Variant 1)** – Exploiting PowerShell commands. **Critical**
35. **OS Command Injection (URL Parameter Variant 1)** – Injection via URL parameters leading to OS calls. **Critical**
36. **OS Command Injection (Form Field Variant 1)** – Injection through form inputs triggering OS commands. **Critical**
37. **OS Command Injection (Header Variant 1)** – Using headers to inject OS commands. **Critical**
38. **OS Command Injection (Cookie Variant 1)** – Exploiting cookie values to run commands. **Critical**
39. **OS Command Injection (Environment Variable Variant 1)** – Injection via environment variables. **Critical**
40. **OS Command Injection (Stacked Command Variant 1)** – Combining multiple commands. **Critical**
41. **OS Command Injection (Command Chaining Variant 1)** – Chaining commands with separators. **Critical**
42. **OS Command Injection (Conditional Command Variant 1)** – Executing commands based on conditions. **Critical**
43. **OS Command Injection (File Parameter Variant 1)** – Injection via file path parameters. **Critical**
44. **OS Command Injection (Log Parameter Variant 1)** – Exploiting logging parameters for command execution. **Critical**
45. **OS Command Injection (API Parameter Variant 1)** – Injection via API parameters that trigger OS commands. **Critical**

46. **OS Command Injection (OS-Specific Variant: Linux)** – Exploiting Linux command structures. **Critical**
47. **OS Command Injection (OS-Specific Variant: Windows)** – Exploiting Windows command processing. **Critical**
48. **OS Command Injection (OS-Specific Variant: macOS)** – Targeting macOS command vulnerabilities. **Critical**
49. **OS Command Injection (Remote Command Injection Variant 1)** – Triggering commands on remote systems. **Critical**
50. **OS Command Injection (Local Command Injection Variant 1)** – Executing commands locally via injection. **Critical**

Category 5: XPath Injection (10 Items)

51. **XPath Injection (Classic)** – Manipulating XPath queries in XML documents. **High**
52. **XPath Injection (Attribute-Based Variant 1)** – Injection via XML attributes. **High**
53. **XPath Injection (Element-Based Variant 1)** – Injection via XML elements. **High**
54. **XPath Injection (URL Parameter Variant 1)** – Injection through URL parameters. **High**
55. **XPath Injection (Form Field Variant 1)** – Injection via web forms. **High**
56. **XPath Injection (XML Payload Variant 1)** – Injection embedded in XML data. **High**
57. **XPath Injection (Cookie Variant 1)** – Using cookie data to affect XPath queries. **High**
58. **XPath Injection (Header Variant 1)** – Injection via HTTP headers affecting XPath. **High**
59. **XPath Injection (Conditional Variant 1)** – Injecting conditions to alter XPath logic. **High**
60. **XPath Injection (Blind Variant 1)** – Inferring XPath query structure by blind methods. **High**

Category 6: XML Injection (10 Items)

61. **XML Injection (Classic)** – Inserting or modifying XML content. **Medium**
62. **XML Injection (Data Injection Variant 1)** – Changing XML data values. **Medium**
63. **XML Injection (Structure Alteration Variant 1)** – Altering the XML document structure. **Medium**

64. **XML Injection (Form Field Variant 1)** – Injection through user input into XML. **Medium**
65. **XML Injection (URL Parameter Variant 1)** – Injection via URL parameters into XML. **Medium**
66. **XML Injection (Header Variant 1)** – Using headers to inject XML data. **Medium**
67. **XML Injection (Cookie Variant 1)** – Injection via cookies that affect XML processing. **Medium**
68. **XML Injection (Stored XML Injection Variant 1)** – Persisting injected XML data. **Medium**
69. **XML Injection (XML Entity Injection Variant 1)** – Abusing XML entity definitions. **Medium**
70. **XML Injection (Conditional XML Variant 1)** – Injection that triggers under certain conditions. **Medium**

Category 7: CRLF Injection (10 Items)

71. **CRLF Injection (Classic)** – Inserting carriage return/line feed characters to manipulate responses. **Medium**
72. **CRLF Injection (HTTP Header Variant 1)** – Injecting CRLF into HTTP headers. **Medium**
73. **CRLF Injection (Log Injection Variant 1)** – Altering log files with injected CRLF sequences. **Medium**
74. **CRLF Injection (SMTP Header Variant 1)** – Injection into email headers via CRLF. **Medium**
75. **CRLF Injection (URL Redirection Variant 1)** – Exploiting CRLF to force URL redirection. **Medium**
76. **CRLF Injection (Cookie Variant 1)** – Injection into cookies via CRLF. **Medium**
77. **CRLF Injection (XSS Triggering Variant 1)** – Using CRLF to enable cross-site scripting. **Medium**
78. **CRLF Injection (Form Field Variant 1)** – Injection through form fields affecting CRLF. **Medium**
79. **CRLF Injection (URL Parameter Variant 1)** – CRLF injection via URL parameters. **Medium**
80. **CRLF Injection (Conditional CRLF Variant 1)** – CRLF injection under certain conditions. **Medium**

Category 8: Server-Side Template Injection (SSTI) (10 Items)

81. **SSTI (Classic)** – Injecting code into server-side templates. **Critical**
82. **SSTI (Jinja2 Variant 1)** – Targeting Jinja2 template expressions. **Critical**

83. **SSTI (Twig Variant 1)** – Exploiting Twig templating in PHP. **Critical**
84. **SSTI (Mako Variant 1)** – Injection in Python's Mako templates. **Critical**
85. **SSTI (ERB Variant 1)** – Injecting Ruby code into ERB templates. **Critical**
86. **SSTI (Handlebars Variant 1)** – Exploiting Handlebars template syntax. **Critical**
87. **SSTI (EJS Variant 1)** – Injection in Embedded JavaScript templates. **Critical**
88. **SSTI (Liquid Variant 1)** – Exploiting Liquid templating systems. **Critical**
89. **SSTI (Mustache Variant 1)** – Injection via Mustache templating. **Critical**
90. **SSTI (Velocity Variant 1)** – Exploiting Apache Velocity templates. **Critical**

Category 9: Code Injection (10 Items)

91. **Code Injection (Classic)** – Forcing an application to execute injected code. **Critical**
92. **Code Injection (Dynamic Eval Variant 1)** – Exploiting unsafe eval usage. **Critical**
93. **Code Injection (Config File Variant 1)** – Injection into configuration files that are executed. **Critical**
94. **Code Injection (Environment Variable Variant 1)** – Using environment variables to pass code. **Critical**
95. **Code Injection (URL Parameter Variant 1)** – Injection via URL parameters that get evaluated. **Critical**
96. **Code Injection (Form Field Variant 1)** – Code injection through user inputs. **Critical**
97. **Code Injection (Header Variant 1)** – Injection via HTTP headers leading to code execution. **Critical**
98. **Code Injection (Cookie Variant 1)** – Using cookies to inject executable code. **Critical**
99. **Code Injection (Command-Line Parameter Variant 1)** – Injection via CLI parameters. **Critical**
100. **Code Injection (Script Tag Variant 1)** – Injection through script tags in templates. **Critical**

Category 10: Object Injection (10 Items)

101. **Object Injection (Classic)** – Exploiting unsafe deserialization of objects. **Critical**

102. **Object Injection (PHP Variant 1)** – Exploiting PHP unserialize() vulnerabilities. **Critical**
103. **Object Injection (Java Variant 1)** – Injection via Java deserialization. **Critical**
104. **Object Injection (Python Variant 1)** – Exploiting pickle deserialization issues. **Critical**
105. **Object Injection (Ruby Variant 1)** – Exploiting Ruby's object deserialization. **Critical**
106. **Object Injection (ASP.NET Variant 1)** – Injection in .NET deserialization routines. **Critical**
107. **Object Injection (Serialized Data Variant 1)** – Manipulating serialized data. **Critical**
108. **Object Injection (Custom Object Variant 1)** – Injection into custom object formats. **Critical**
109. **Object Injection (Stored Object Injection Variant 1)** – Persisting malicious objects. **Critical**
110. **Object Injection (Remote Object Injection Variant 1)** – Exploiting remotely deserialized objects. **Critical**

Category 11: Expression Language Injection (10 Items)

111. **Expression Language Injection (Classic)** – Exploiting dynamic expressions in templates. **High**
112. **Expression Language Injection (Java EL Variant 1)** – Injection via Java's Expression Language. **High**
113. **Expression Language Injection (Spring Expression Variant 1)** – Targeting Spring's expression language. **High**
114. **Expression Language Injection (JSP EL Variant 1)** – Exploiting JSP Expression Language. **High**
115. **Expression Language Injection (OGNL Variant 1)** – Injection in OGNL (Object-Graph Navigation Language). **High**
116. **Expression Language Injection (Freemarker Variant 1)** – Exploiting Freemarker expressions. **High**
117. **Expression Language Injection (Razor Variant 1)** – Injection in ASP.NET Razor views. **High**
118. **Expression Language Injection (Thymeleaf Variant 1)** – Exploiting Thymeleaf templates. **High**
119. **Expression Language Injection (Velocity Variant 1)** – Using Apache Velocity's expressions. **High**
120. **Expression Language Injection (Custom EL Variant 1)** – Injection in custom expression languages. **High**

Category 12: HTML Injection (10 Items)

121. **HTML Injection (Classic)** – Inserting malicious HTML into pages. **Medium**
122. **HTML Injection (Comment Injection Variant 1)** – Injecting HTML comments. **Medium**
123. **HTML Injection (Attribute Injection Variant 1)** – Manipulating HTML attributes. **Medium**
124. **HTML Injection (Tag Injection Variant 1)** – Inserting new HTML tags. **Medium**
125. **HTML Injection (URL Injection Variant 1)** – Injecting into link URLs. **Medium**
126. **HTML Injection (Form Field Variant 1)** – Injection via form inputs affecting HTML. **Medium**
127. **HTML Injection (Scriptless Injection Variant 1)** – Non-script HTML injections. **Medium**
128. **HTML Injection (Style Injection Variant 1)** – Altering inline styles via injection. **Medium**
129. **HTML Injection (Iframe Injection Variant 1)** – Inserting iframes maliciously. **Medium**
130. **HTML Injection (Conditional HTML Variant 1)** – HTML injection that triggers conditionally. **Medium**

Category 13: JavaScript Injection (10 Items)

131. **JavaScript Injection (Classic)** – Injecting malicious JavaScript into pages. **High**
132. **JavaScript Injection (Inline Script Variant 1)** – Injection into inline scripts. **High**
133. **JavaScript Injection (Event Handler Variant 1)** – Injecting code into event handlers. **High**
134. **JavaScript Injection (DOM-Based Variant 1)** – Exploiting client-side DOM processing. **High**
135. **JavaScript Injection (Callback Function Variant 1)** – Hijacking callbacks. **High**
136. **JavaScript Injection (Eval Variant 1)** – Exploiting unsafe eval() usage in JS. **High**
137. **JavaScript Injection (JSONP Variant 1)** – Manipulating JSONP responses. **High**
138. **JavaScript Injection (Script Tag Injection Variant 1)** – Injecting through dynamic script tags. **High**
139. **JavaScript Injection (URL Parameter Variant 1)** – Injection via URL affecting JS execution. **High**
140. **JavaScript Injection (Form Field Variant 1)** – Injection via form inputs leading to JS code execution. **High**

Category 14: CSS Injection (10 Items)

141. **CSS Injection (Classic)** – Injecting malicious CSS to alter appearance. **Low**
142. **CSS Injection (Style Tag Variant 1)** – Exploiting style tags. **Low**
143. **CSS Injection (Inline Style Variant 1)** – Injection into inline style attributes. **Low**
144. **CSS Injection (External Stylesheet Variant 1)** – Manipulating linked stylesheets. **Low**
145. **CSS Injection (URL Injection Variant 1)** – Injection affecting CSS URL references. **Low**
146. **CSS Injection (Attribute Injection Variant 1)** – Altering style-related attributes. **Low**
147. **CSS Injection (Conditional CSS Variant 1)** – CSS injection that fires conditionally. **Low**
148. **CSS Injection (HTML-Linked CSS Variant 1)** – Injection via HTML that alters CSS. **Low**
149. **CSS Injection (URL Parameter Variant 1)** – Injection via URL affecting CSS loads. **Low**
150. **CSS Injection (Advanced Selector Exploit Variant 1)** – Exploiting CSS selectors for injection. **Low**

Category 15: JSON Injection (10 Items)

151. **JSON Injection (Classic)** – Manipulating JSON structures through injection. **Medium**
152. **JSON Injection (Property Injection Variant 1)** – Changing JSON object properties. **Medium**
153. **JSON Injection (Array Injection Variant 1)** – Injecting into JSON arrays. **Medium**
154. **JSON Injection (Nested Object Variant 1)** – Injection into nested JSON objects. **Medium**
155. **JSON Injection (URL Parameter Variant 1)** – Injection via URL into JSON APIs. **Medium**
156. **JSON Injection (Form Field Variant 1)** – Injection via form inputs affecting JSON. **Medium**
157. **JSON Injection (API Payload Variant 1)** – Exploiting API endpoints with JSON injection. **Medium**
158. **JSON Injection (Stored JSON Injection Variant 1)** – Persisting malicious JSON. **Medium**
159. **JSON Injection (Dynamic JSON Injection Variant 1)** – Injection into dynamically generated JSON. **Medium**

160. **JSON Injection (Conditional JSON Injection Variant 1)** – Triggered JSON injection under conditions. **Medium**

Category 16: YAML Injection (10 Items)

161. **YAML Injection (Classic)** – Exploiting unsanitized YAML content. **Medium**
162. **YAML Injection (Mapping Injection Variant 1)** – Injection into YAML mappings. **Medium**
163. **YAML Injection (Sequence Injection Variant 1)** – Altering YAML sequences. **Medium**
164. **YAML Injection (Config File Variant 1)** – Injection via configuration files in YAML. **Medium**
165. **YAML Injection (URL Parameter Variant 1)** – Injection through URL parameters in YAML APIs. **Medium**
166. **YAML Injection (Form Field Variant 1)** – Injection via forms affecting YAML. **Medium**
167. **YAML Injection (API Payload Variant 1)** – Injection into YAML-based API data. **Medium**
168. **YAML Injection (Stored YAML Injection Variant 1)** – Persisting malicious YAML. **Medium**
169. **YAML Injection (Conditional YAML Injection Variant 1)** – Triggered under certain conditions. **Medium**
170. **YAML Injection (Complex Structure Variant 1)** – Exploiting deeply nested YAML. **Medium**

Category 17: CSV Injection (Formula Injection) (10 Items)

171. **CSV Injection (Classic)** – Inserting malicious formulas in CSV files. **High**
172. **CSV Injection (Excel Formula Variant 1)** – Injection targeting Excel's formula parsing. **High**
173. **CSV Injection (Google Sheets Variant 1)** – Exploiting CSV formulas in Google Sheets. **High**
174. **CSV Injection (Exported Data Variant 1)** – Injection in data exported to CSV. **High**
175. **CSV Injection (User Input Variant 1)** – Injection via user-provided CSV data. **High**
176. **CSV Injection (Stored CSV Injection Variant 1)** – Persisting injected CSV formulas. **High**
177. **CSV Injection (Dynamic Formula Injection Variant 1)** – Dynamic injection of spreadsheet formulas. **High**

178. **CSV Injection (Conditional CSV Injection Variant 1)** – Triggered under specific conditions. **High**
179. **CSV Injection (File Upload Variant 1)** – Injection via CSV file uploads. **High**
180. **CSV Injection (Advanced Formula Injection Variant 1)** – Sophisticated exploitation of CSV formulas. **High**

Category 18: Stored SQL Injection (10 Items)

181. **Stored SQL Injection (Classic)** – Malicious SQL stored and executed later. **Critical**
182. **Stored SQL Injection (User Data Variant 1)** – Injection via user profile data. **Critical**
183. **Stored SQL Injection (Parameter Variant 1)** – Stored injection via query parameters. **Critical**
184. **Stored SQL Injection (Comment Injection Variant 1)** – Injecting SQL through comment fields. **Critical**
185. **Stored SQL Injection (Hidden Field Variant 1)** – Injection via hidden form fields. **Critical**
186. **Stored SQL Injection (Cookie Injection Variant 1)** – Persisting injection via cookies. **Critical**
187. **Stored SQL Injection (Header Injection Variant 1)** – Injection stored from HTTP headers. **Critical**
188. **Stored SQL Injection (Delayed Execution Variant 1)** – Injection that executes later. **Critical**
189. **Stored SQL Injection (Conditional Variant 1)** – Stored injection that triggers under conditions. **Critical**
190. **Stored SQL Injection (Aggregation Variant 1)** – Exploiting aggregation queries via stored injection. **Critical**

Category 19: Union-Based SQL Injection (10 Items)

191. **Union-Based SQL Injection (Classic)** – Using UNION to combine query results. **Critical**
192. **Union-Based SQL Injection (Simple UNION Variant 1)** – A basic UNION-based technique. **Critical**
193. **Union-Based SQL Injection (Multiple UNION Variant 1)** – Exploiting multiple UNIONs in a query. **Critical**
194. **Union-Based SQL Injection (Column Count Variant 1)** – Adjusting injection based on column count. **Critical**
195. **Union-Based SQL Injection (Data Extraction Variant 1)** – Using UNION to retrieve data. **Critical**

196. **Union-Based SQL Injection (Blind UNION Variant 1)** – Blind data extraction using UNION. **Critical**
197. **Union-Based SQL Injection (Extended UNION Variant 1)** – Extended use of UNION for deeper extraction. **Critical**
198. **Union-Based SQL Injection (Comment-Based UNION Variant 1)** – Using comments to facilitate UNION injection. **Critical**
199. **Union-Based SQL Injection (OS-Specific UNION Variant 1)** – Targeting OS-specific query behavior with UNION. **Critical**
200. **Union-Based SQL Injection (Conditional UNION Variant 1)** – Triggering UNION-based injection conditionally. **Critical**

Category 20: Blind SQL Injection (10 Items)

201. **Blind SQL Injection (Classic)** – Inferring data via blind techniques. **High**
202. **Blind SQL Injection (Boolean Variant 1)** – Using boolean conditions to extract information. **High**
203. **Blind SQL Injection (Time-Based Variant 1)** – Leveraging delays to reveal data. **High**
204. **Blind SQL Injection (Conditional Variant 1)** – Blind injection triggered by conditions. **High**
205. **Blind SQL Injection (URL Parameter Variant 1)** – Blind injection via URL parameters. **High**
206. **Blind SQL Injection (Form Field Variant 1)** – Using form inputs for blind SQL injection. **High**
207. **Blind SQL Injection (Cookie Variant 1)** – Blind injection via cookies. **High**
208. **Blind SQL Injection (Header Variant 1)** – Blind injection via HTTP headers. **High**
209. **Blind SQL Injection (Stored Blind Injection Variant 1)** – Persisting blind SQL injection payloads. **High**
210. **Blind SQL Injection (Advanced Blind Variant 1)** – More sophisticated blind injection techniques. **High**

Category 21: Error-Based SQL Injection (10 Items)

211. **Error-Based SQL Injection (Classic)** – Using database errors to extract data. **High**
212. **Error-Based SQL Injection (Malformed Query Variant 1)** – Triggering errors with malformed queries. **High**
213. **Error-Based SQL Injection (Conditional Error Variant 1)** – Conditional triggering of errors for data retrieval. **High**

214. **Error-Based SQL Injection (Database-Specific Variant 1)** – Exploiting errors in specific DBMSs. **High**
215. **Error-Based SQL Injection (URL Parameter Variant 1)** – Using URL parameters to force error messages. **High**
216. **Error-Based SQL Injection (Form Field Variant 1)** – Injection through forms that cause errors. **High**
217. **Error-Based SQL Injection (Header Variant 1)** – Using HTTP headers to provoke errors. **High**
218. **Error-Based SQL Injection (Cookie Variant 1)** – Injection via cookies that generate errors. **High**
219. **Error-Based SQL Injection (Stored Error Injection Variant 1)** – Persisting error-based injections. **High**
220. **Error-Based SQL Injection (Dynamic Error Injection Variant 1)** – Dynamically triggering errors for exploitation. **High**

Category 22: Time-Based SQL Injection (10 Items)

221. **Time-Based SQL Injection (Classic)** – Using delays to infer query results. **High**
222. **Time-Based SQL Injection (URL Parameter Variant 1)** – Injection via URL causing delays. **High**
223. **Time-Based SQL Injection (Form Field Variant 1)** – Using form inputs to trigger time delays. **High**
224. **Time-Based SQL Injection (Header Variant 1)** – HTTP header–based time injection. **High**
225. **Time-Based SQL Injection (Cookie Variant 1)** – Injection via cookies that delay responses. **High**
226. **Time-Based SQL Injection (Conditional Delay Variant 1)** – Delays triggered under certain conditions. **High**
227. **Time-Based SQL Injection (Database-Specific Variant 1)** – Exploiting timing issues in specific databases. **High**
228. **Time-Based SQL Injection (Stored Time Injection Variant 1)** – Persisting payloads that delay execution. **High**
229. **Time-Based SQL Injection (Advanced Time Variant 1)** – More complex delay-based techniques. **High**
230. **Time-Based SQL Injection (Dynamic Delay Variant 1)** – Dynamically adjusting delays for data inference. **High**

Category 23: Second-Order SQL Injection (10 Items)

231. **Second-Order SQL Injection (Classic)** – Payload stored initially then executed unsafely later. **High**

232. **Second-Order SQL Injection (User Data Variant 1)** – Injection via user data fields stored for later use. **High**
233. **Second-Order SQL Injection (Parameter Variant 1)** – Stored parameters exploited later. **High**
234. **Second-Order SQL Injection (Session Data Variant 1)** – Injection via session variables. **High**
235. **Second-Order SQL Injection (Cookie Variant 1)** – Stored cookie data later triggering SQL injection. **High**
236. **Second-Order SQL Injection (Header Variant 1)** – Injection via headers stored and later executed. **High**
237. **Second-Order SQL Injection (Delayed Execution Variant 1)** – Injection with postponed execution. **High**
238. **Second-Order SQL Injection (ORM-Based Variant 1)** – Exploiting ORM layers with stored injections. **High**
239. **Second-Order SQL Injection (Dynamic Query Variant 1)** – Stored data altering dynamic queries. **High**
240. **Second-Order SQL Injection (Conditional Variant 1)** – Payload triggers under certain conditions later. **High**

Category 24: Shell Injection (10 Items)

241. **Shell Injection (Classic)** – Inserting shell commands into vulnerable calls. **Critical**
242. **Shell Injection (Command-Line Variant 1)** – Injection targeting command-line utilities. **Critical**
243. **Shell Injection (URL Parameter Variant 1)** – Shell injection via URL inputs. **Critical**
244. **Shell Injection (Form Field Variant 1)** – Injection through form inputs that trigger OS commands. **Critical**
245. **Shell Injection (Header Variant 1)** – Using headers to inject shell commands. **Critical**
246. **Shell Injection (Cookie Variant 1)** – Injection via cookie values that end up in shell calls. **Critical**
247. **Shell Injection (Stacked Command Variant 1)** – Combining multiple commands in one injection. **Critical**
248. **Shell Injection (Chained Command Variant 1)** – Command chaining through injection. **Critical**
249. **Shell Injection (OS-Specific Variant: Linux)** – Exploiting Linux shell behaviors. **Critical**
250. **Shell Injection (OS-Specific Variant: Windows)** – Exploiting Windows command interpreters. **Critical**

Category 25: Bash Injection (10 Items)

251. **Bash Injection (Classic)** – Exploiting Bash's command parsing flaws. **Critical**
252. **Bash Injection (Shellshock Variant 1)** – Leveraging Shellshock-like vulnerabilities. **Critical**
253. **Bash Injection (Environment Variable Variant 1)** – Injection via environment variables in Bash. **Critical**
254. **Bash Injection (Script Parameter Variant 1)** – Using script parameters to inject commands. **Critical**
255. **Bash Injection (URL Parameter Variant 1)** – Injection via URL parameters affecting Bash execution. **Critical**
256. **Bash Injection (Form Field Variant 1)** – Injection through forms that trigger Bash code. **Critical**
257. **Bash Injection (Header Variant 1)** – Exploiting HTTP headers to affect Bash commands. **Critical**
258. **Bash Injection (Cookie Variant 1)** – Using cookie data for Bash injection. **Critical**
259. **Bash Injection (Conditional Bash Variant 1)** – Injection that activates under certain conditions. **Critical**
260. **Bash Injection (Advanced Bash Variant 1)** – More sophisticated Bash exploitation. **Critical**

Category 26: Perl Injection (10 Items)

261. **Perl Injection (Classic)** – Exploiting unsafe eval in Perl scripts. **High**
262. **Perl Injection (URL Parameter Variant 1)** – Injection via URL affecting Perl code. **High**
263. **Perl Injection (Form Field Variant 1)** – Injection through web forms into Perl. **High**
264. **Perl Injection (Header Variant 1)** – Using HTTP headers to inject Perl code. **High**
265. **Perl Injection (Cookie Variant 1)** – Injection via cookies targeting Perl. **High**
266. **Perl Injection (Dynamic Code Execution Variant 1)** – Triggering dynamic eval in Perl. **High**
267. **Perl Injection (Stored Perl Injection Variant 1)** – Persisting malicious Perl payloads. **High**
268. **Perl Injection (Conditional Variant 1)** – Injection that triggers conditionally in Perl. **High**
269. **Perl Injection (OS Command Variant 1)** – Using Perl to execute OS commands via injection. **High**
270. **Perl Injection (Extended Perl Variant 1)** – Advanced techniques in Perl injection. **High**

Category 27: Python Injection (10 Items)

271. **Python Injection (Classic)** – Exploiting unsafe use of eval/exec in Python. **Critical**
272. **Python Injection (Dynamic Code Execution Variant 1)** – Injection that leverages Python's dynamic evaluation. **Critical**
273. **Python Injection (URL Parameter Variant 1)** – Injection via URL parameters in Python apps. **Critical**
274. **Python Injection (Form Field Variant 1)** – Injection through user input in Python. **Critical**
275. **Python Injection (Header Variant 1)** – Injection via HTTP headers in Python environments. **Critical**
276. **Python Injection (Cookie Variant 1)** – Using cookies to inject Python code. **Critical**
277. **Python Injection (Stored Python Injection Variant 1)** – Persisting malicious Python payloads. **Critical**
278. **Python Injection (Conditional Variant 1)** – Injection that activates under conditions in Python. **Critical**
279. **Python Injection (Custom Function Injection Variant 1)** – Injecting code into custom Python functions. **Critical**
280. **Python Injection (Configuration File Variant 1)** – Injection via configuration files in Python apps. **Critical**

Category 28: Deserialization Injection (10 Items)

281. **Deserialization Injection (Classic)** – Exploiting unsafe deserialization to execute code. **Critical**
282. **Deserialization Injection (PHP Variant 1)** – Exploiting PHP unserialize vulnerabilities. **Critical**
283. **Deserialization Injection (Java Variant 1)** – Injection via Java deserialization flaws. **Critical**
284. **Deserialization Injection (Python Variant 1)** – Exploiting Python pickle deserialization. **Critical**
285. **Deserialization Injection (Ruby Variant 1)** – Injection in Ruby's deserialization routines. **Critical**
286. **Deserialization Injection (ASP.NET Variant 1)** – Exploiting .NET deserialization. **Critical**
287. **Deserialization Injection (Serialized Data Variant 1)** – Injection via manipulated serialized data. **Critical**
288. **Deserialization Injection (Custom Object Variant 1)** – Exploiting custom object formats. **Critical**
289. **Deserialization Injection (Stored Deserialization Variant 1)** – Persisting malicious serialized objects. **Critical**

290. **Deserialization Injection (Remote Deserialization Variant 1)** – Exploiting remotely supplied serialized data. **Critical**

Category 29: ERB Template Injection (Ruby) (10 Items)

291. **ERB Template Injection (Classic)** – Injecting Ruby code into ERB templates. **Critical**
292. **ERB Template Injection (Inline Code Variant 1)** – Injection via inline Ruby code. **Critical**
293. **ERB Template Injection (Dynamic Content Variant 1)** – Exploiting dynamic ERB content. **Critical**
294. **ERB Template Injection (URL Parameter Variant 1)** – Injection via URL affecting ERB templates. **Critical**
295. **ERB Template Injection (Form Field Variant 1)** – Injection through forms into ERB templates. **Critical**
296. **ERB Template Injection (Stored ERB Injection Variant 1)** – Persisting injected Ruby code. **Critical**
297. **ERB Template Injection (Conditional ERB Variant 1)** – Injection triggered under specific conditions. **Critical**
298. **ERB Template Injection (Header Variant 1)** – Using HTTP headers to affect ERB templates. **Critical**
299. **ERB Template Injection (Cookie Variant 1)** – Injection via cookies in ERB contexts. **Critical**
300. **ERB Template Injection (Advanced ERB Variant 1)** – More advanced exploitation of ERB templates. **Critical**

Category 30: JSP Injection (Java) (10 Items)

301. **JSP Injection (Classic)** – Exploiting dynamic content in Java Server Pages. **High**
302. **JSP Injection (Scriptlet Injection Variant 1)** – Injection into JSP scriptlets. **High**
303. **JSP Injection (Expression Injection Variant 1)** – Exploiting JSP expressions. **High**
304. **JSP Injection (Stored JSP Injection Variant 1)** – Persisting malicious JSP code. **High**
305. **JSP Injection (Conditional JSP Variant 1)** – JSP injection that triggers under conditions. **High**
306. **JSP Injection (URL Parameter Variant 1)** – Injection via URL parameters in JSP. **High**
307. **JSP Injection (Form Field Variant 1)** – Injection through form inputs in JSP. **High**

308. **JSP Injection (Header Variant 1)** – Using headers to inject into JSP pages. **High**
309. **JSP Injection (Cookie Variant 1)** – Exploiting cookie data in JSP contexts. **High**
310. **JSP Injection (Advanced JSP Variant 1)** – More advanced JSP injection techniques. **High**

Category 31: Twig Template Injection (PHP) (10 Items)

311. **Twig Template Injection (Classic)** – Injecting code into Twig templates. **Critical**
312. **Twig Template Injection (Inline Expression Variant 1)** – Exploiting inline expressions in Twig. **Critical**
313. **Twig Template Injection (Dynamic Template Variant 1)** – Injection into dynamically generated Twig templates. **Critical**
314. **Twig Template Injection (Stored Twig Injection Variant 1)** – Persisting Twig injection payloads. **Critical**
315. **Twig Template Injection (Conditional Twig Variant 1)** – Injection triggered under specific conditions in Twig. **Critical**
316. **Twig Template Injection (URL Parameter Variant 1)** – Injection via URL into Twig templates. **Critical**
317. **Twig Template Injection (Form Field Variant 1)** – Exploiting form inputs in Twig contexts. **Critical**
318. **Twig Template Injection (Header Variant 1)** – Injection via HTTP headers in Twig. **Critical**
319. **Twig Template Injection (Cookie Variant 1)** – Using cookies to inject into Twig templates. **Critical**
320. **Twig Template Injection (Advanced Twig Variant 1)** – More sophisticated Twig injection techniques. **Critical**

Category 32: Handlebars Template Injection (Node.js) (10 Items)

321. **Handlebars Template Injection (Classic)** – Exploiting Handlebars templates. **High**
322. **Handlebars Template Injection (Inline Expression Variant 1)** – Injection via inline Handlebars expressions. **High**
323. **Handlebars Template Injection (Stored Handlebars Injection Variant 1)** – Persisting injection payloads in Handlebars templates. **High**
324. **Handlebars Template Injection (Conditional Variant 1)** – Injection that activates under conditions in Handlebars. **High**
325. **Handlebars Template Injection (URL Parameter Variant 1)** – Using URL parameters to inject into Handlebars. **High**

326. **Handlebars Template Injection (Form Field Variant 1)** – Injection via form inputs in Handlebars. **High**
327. **Handlebars Template Injection (Header Variant 1)** – Exploiting HTTP headers in Handlebars contexts. **High**
328. **Handlebars Template Injection (Cookie Variant 1)** – Using cookies to inject into Handlebars. **High**
329. **Handlebars Template Injection (Advanced Variant 1)** – Advanced exploitation of Handlebars templates. **High**
330. **Handlebars Template Injection (Dynamic Data Variant 1)** – Injection into dynamically bound data in Handlebars. **High**

Category 33: HTTP Header Injection (10 Items)

331. **HTTP Header Injection (Classic)** – Manipulating HTTP headers via injection. **Medium**
332. **HTTP Header Injection (Set-Cookie Variant 1)** – Injection affecting cookie headers. **Medium**
333. **HTTP Header Injection (Location Redirect Variant 1)** – Forcing redirects through header injection. **Medium**
334. **HTTP Header Injection (Custom Header Variant 1)** – Injection into non-standard headers. **Medium**
335. **HTTP Header Injection (URL Parameter Variant 1)** – Injection via URL parameters that modify headers. **Medium**
336. **HTTP Header Injection (Form Field Variant 1)** – Injection through forms altering headers. **Medium**
337. **HTTP Header Injection (Conditional Header Injection Variant 1)** – Headers injected conditionally. **Medium**
338. **HTTP Header Injection (Stored Header Injection Variant 1)** – Persisting header injection payloads. **Medium**
339. **HTTP Header Injection (Error Message Header Variant 1)** – Injection that manipulates error headers. **Medium**
340. **HTTP Header Injection (Advanced Header Variant 1)** – More complex header injection techniques. **Medium**

Category 34: SMTP Injection (10 Items)

341. **SMTP Injection (Classic)** – Injecting malicious commands into SMTP sessions. **High**
342. **SMTP Injection (Mail Header Variant 1)** – Injection targeting email headers. **High**
343. **SMTP Injection (Email Body Variant 1)** – Injection through the email body. **High**

344. **SMTP Injection (URL Parameter Variant 1)** – Injection via URL parameters in SMTP contexts. **High**
345. **SMTP Injection (Form Field Variant 1)** – Exploiting form inputs to inject SMTP commands. **High**
346. **SMTP Injection (Conditional SMTP Variant 1)** – SMTP injection that triggers conditionally. **High**
347. **SMTP Injection (Stored SMTP Injection Variant 1)** – Persisting SMTP injection payloads. **High**
348. **SMTP Injection (Dynamic SMTP Variant 1)** – Dynamically altering SMTP commands via injection. **High**
349. **SMTP Injection (Authentication Bypass Variant 1)** – Using injection to bypass SMTP authentication. **High**
350. **SMTP Injection (Advanced SMTP Variant 1)** – Advanced exploitation of SMTP injection. **High**

Category 35: FTP Injection (10 Items)

351. **FTP Injection (Classic)** – Exploiting FTP command processing via injection. **Medium**
352. **FTP Injection (Command Variant 1)** – Injection affecting FTP command execution. **Medium**
353. **FTP Injection (URL Parameter Variant 1)** – Injection via URL parameters in FTP interfaces. **Medium**
354. **FTP Injection (Form Field Variant 1)** – Injection through form inputs that affect FTP commands. **Medium**
355. **FTP Injection (Conditional FTP Variant 1)** – Injection that activates under certain FTP conditions. **Medium**
356. **FTP Injection (Stored FTP Injection Variant 1)** – Persisting FTP injection payloads. **Medium**
357. **FTP Injection (Anonymous FTP Variant 1)** – Exploiting anonymous FTP logins via injection. **Medium**
358. **FTP Injection (Authentication Bypass Variant 1)** – Using injection to bypass FTP authentication. **Medium**
359. **FTP Injection (File Path Injection Variant 1)** – Injection that manipulates file paths in FTP. **Medium**
360. **FTP Injection (Advanced FTP Variant 1)** – More advanced techniques in FTP injection. **Medium**

Category 36: HTTP Parameter Pollution (HPP) (10 Items)

361. **HTTP Parameter Pollution (Classic)** – Inserting multiple parameters to confuse logic. **Medium**

362. **HTTP Parameter Pollution (URL Parameter Variant 1)** – HPP via URL query strings. **Medium**
363. **HTTP Parameter Pollution (Form Field Variant 1)** – HPP through form inputs. **Medium**
364. **HTTP Parameter Pollution (Cookie Variant 1)** – Injecting multiple cookie values. **Medium**
365. **HTTP Parameter Pollution (Header Variant 1)** – Using headers to pollute parameters. **Medium**
366. **HTTP Parameter Pollution (Stored HPP Variant 1)** – Persisting polluted parameters. **Medium**
367. **HTTP Parameter Pollution (Conditional HPP Variant 1)** – HPP triggered under conditions. **Medium**
368. **HTTP Parameter Pollution (Dynamic HPP Variant 1)** – Dynamic manipulation of parameters. **Medium**
369. **HTTP Parameter Pollution (Multiple Injection Variant 1)** – Combining multiple injection points. **Medium**
370. **HTTP Parameter Pollution (Advanced HPP Variant 1)** – More sophisticated HPP techniques. **Medium**

Category 37: Format String Injection (10 Items)

371. **Format String Injection (Classic)** – Exploiting format functions to leak data or execute code. **Critical**
372. **Format String Injection (C/C++ Variant 1)** – Injection in C/C++ printf-style functions. **Critical**
373. **Format String Injection (Python Variant 1)** – Exploiting Python's string formatting. **Critical**
374. **Format String Injection (Java Variant 1)** – Injection in Java's format methods. **Critical**
375. **Format String Injection (PHP Variant 1)** – Exploiting PHP's sprintf functions. **Critical**
376. **Format String Injection (Conditional Format Variant 1)** – Injection triggered conditionally. **Critical**
377. **Format String Injection (Stored Format Injection Variant 1)** – Persisting malicious format strings. **Critical**
378. **Format String Injection (Dynamic Format Variant 1)** – Dynamic alteration of format specifiers. **Critical**
379. **Format String Injection (URL Parameter Variant 1)** – Injection via URL parameters affecting format strings. **Critical**
380. **Format String Injection (Advanced Format Variant 1)** – Advanced exploitation of format string vulnerabilities. **Critical**

Category 38: Redis Injection (10 Items)

381. **Redis Injection (Classic)** – Exploiting unsanitized input in Redis commands. **High**
382. **Redis Injection (Command Variant 1)** – Injection affecting Redis command execution. **High**
383. **Redis Injection (Pipeline Injection Variant 1)** – Exploiting Redis command pipelines. **High**
384. **Redis Injection (URL Parameter Variant 1)** – Injection via URL parameters in Redis interfaces. **High**
385. **Redis Injection (Form Field Variant 1)** – Injection through form inputs in Redis commands. **High**
386. **Redis Injection (Header Variant 1)** – Injection via HTTP headers that influence Redis. **High**
387. **Redis Injection (Cookie Variant 1)** – Injection via cookies in Redis contexts. **High**
388. **Redis Injection (Conditional Redis Variant 1)** – Injection that triggers under certain conditions. **High**
389. **Redis Injection (Stored Redis Injection Variant 1)** – Persisting malicious Redis commands. **High**
390. **Redis Injection (Advanced Redis Variant 1)** – Advanced techniques in Redis injection. **High**

Category 39: GraphQL Injection (10 Items)

391. **GraphQL Injection (Classic)** – Injecting malicious queries or mutations in GraphQL APIs. **High**
392. **GraphQL Injection (Query Variant 1)** – Injection altering GraphQL query structure. **High**
393. **GraphQL Injection (Mutation Variant 1)** – Injection affecting GraphQL mutations. **High**
394. **GraphQL Injection (Stored GraphQL Injection Variant 1)** – Persisting injected GraphQL payloads. **High**
395. **GraphQL Injection (Conditional GraphQL Variant 1)** – Injection triggered by conditions in GraphQL queries. **High**
396. **GraphQL Injection (URL Parameter Variant 1)** – Injection via URL parameters in GraphQL endpoints. **High**
397. **GraphQL Injection (Form Field Variant 1)** – Injection through forms affecting GraphQL. **High**
398. **GraphQL Injection (Header Variant 1)** – Injection via HTTP headers in GraphQL APIs. **High**
399. **GraphQL Injection (Cookie Variant 1)** – Injection via cookies in GraphQL contexts. **High**
400. **GraphQL Injection (Advanced GraphQL Variant 1)** – More advanced GraphQL injection techniques. **High**

Category 40: RPC Injection (10 Items)

401. **RPC Injection (Classic)** – Exploiting unsanitized input in RPC calls. **High**
402. **RPC Injection (XML-RPC Variant 1)** – Injection via XML-RPC interfaces. **High**
403. **RPC Injection (JSON-RPC Variant 1)** – Injection in JSON-RPC protocols. **High**
404. **RPC Injection (URL Parameter Variant 1)** – Injection via URL parameters in RPC calls. **High**
405. **RPC Injection (Form Field Variant 1)** – Injection through forms targeting RPC. **High**
406. **RPC Injection (Header Variant 1)** – Injection via HTTP headers in RPC contexts. **High**
407. **RPC Injection (Cookie Variant 1)** – Injection through cookies in RPC calls. **High**
408. **RPC Injection (Conditional RPC Variant 1)** – RPC injection triggered under conditions. **High**
409. **RPC Injection (Stored RPC Injection Variant 1)** – Persisting RPC injection payloads. **High**
410. **RPC Injection (Advanced RPC Variant 1)** – Advanced exploitation of RPC injection. **High**

Category 41: XXE Injection (XML External Entity) (10 Items)

411. **XXE Injection (Classic)** – Exploiting XML parsers via external entity definitions. **Critical**
412. **XXE Injection (Parameter Entity Variant 1)** – Injection using parameter entities. **Critical**
413. **XXE Injection (Stored XXE Variant 1)** – Persisting XXE payloads. **Critical**
414. **XXE Injection (Conditional XXE Variant 1)** – XXE injection triggered under certain conditions. **Critical**
415. **XXE Injection (URL Parameter Variant 1)** – Injection via URL parameters affecting XML parsers. **Critical**
416. **XXE Injection (Form Field Variant 1)** – Injection through form inputs in XML contexts. **Critical**
417. **XXE Injection (Header Variant 1)** – Injection via HTTP headers affecting XML parsing. **Critical**
418. **XXE Injection (Cookie Variant 1)** – Injection via cookies into XML parsers. **Critical**
419. **XXE Injection (Blind XXE Variant 1)** – Inferring data via blind XXE techniques. **Critical**

420. **XXE Injection (Advanced XXE Variant 1)** – Advanced exploitation of XXE vulnerabilities. **Critical**

Category 42: SVG Injection (10 Items)

421. **SVG Injection (Classic)** – Injecting malicious code into SVG files. **Medium**
422. **SVG Injection (Inline SVG Variant 1)** – Injection within inline SVG elements. **Medium**
423. **SVG Injection (Attribute Injection Variant 1)** – Altering SVG attributes via injection. **Medium**
424. **SVG Injection (Script Injection Variant 1)** – Inserting script code into SVG files. **Medium**
425. **SVG Injection (Conditional SVG Variant 1)** – SVG injection that triggers conditionally. **Medium**
426. **SVG Injection (Stored SVG Injection Variant 1)** – Persisting malicious SVG payloads. **Medium**
427. **SVG Injection (URL Parameter Variant 1)** – Injection via URL parameters in SVG contexts. **Medium**
428. **SVG Injection (Form Field Variant 1)** – Injection through form inputs affecting SVG. **Medium**
429. **SVG Injection (Header Variant 1)** – Using HTTP headers to alter SVG content. **Medium**
430. **SVG Injection (Advanced SVG Variant 1)** – Advanced techniques in SVG injection. **Medium**

Category 43: Server-Side JavaScript Injection (10 Items)

431. **Server-Side JavaScript Injection (Classic)** – Exploiting Node.js or similar environments via unsanitized input. **Critical**
432. **Server-Side JavaScript Injection (Dynamic Eval Variant 1)** – Injection via unsafe eval() in JS on the server. **Critical**
433. **Server-Side JavaScript Injection (URL Parameter Variant 1)** – Injection via URL parameters affecting server JS. **Critical**
434. **Server-Side JavaScript Injection (Form Field Variant 1)** – Injection through form inputs in server JS contexts. **Critical**
435. **Server-Side JavaScript Injection (Header Variant 1)** – Injection via HTTP headers in server JS. **Critical**
436. **Server-Side JavaScript Injection (Cookie Variant 1)** – Using cookies to inject server-side JS code. **Critical**
437. **Server-Side JavaScript Injection (Stored JS Injection Variant 1)** – Persisting JS injection payloads on the server. **Critical**

438. **Server-Side JavaScript Injection (Conditional JS Injection Variant 1)** – Triggered under specific conditions. **Critical**
439. **Server-Side JavaScript Injection (Advanced JS Injection Variant 1)** – More advanced server-side JS injection. **Critical**
440. **Server-Side JavaScript Injection (Module Injection Variant 1)** – Injection into Node.js modules. **Critical**

Category 44: Cookie Injection (10 Items)

441. **Cookie Injection (Classic)** – Manipulating cookie values via injection. **Medium**
442. **Cookie Injection (Session Cookie Variant 1)** – Injection affecting session cookies. **Medium**
443. **Cookie Injection (Persistent Cookie Variant 1)** – Persisting malicious cookie values. **Medium**
444. **Cookie Injection (URL Parameter Variant 1)** – Injection via URL parameters that modify cookies. **Medium**
445. **Cookie Injection (Form Field Variant 1)** – Injection through forms affecting cookies. **Medium**
446. **Cookie Injection (Stored Cookie Injection Variant 1)** – Persisting cookie injection payloads. **Medium**
447. **Cookie Injection (Conditional Cookie Variant 1)** – Injection activated under conditions. **Medium**
448. **Cookie Injection (Dynamic Cookie Variant 1)** – Dynamically altering cookie values via injection. **Medium**
449. **Cookie Injection (Cross-Site Cookie Injection Variant 1)** – Exploiting cross-site cookie vulnerabilities. **Medium**
450. **Cookie Injection (Advanced Cookie Variant 1)** – Advanced techniques in cookie injection. **Medium**

Category 45: DNS Injection (10 Items)

451. **DNS Injection (Classic)** – Manipulating DNS queries or responses via injection. **Medium**
452. **DNS Injection (DNS Query Variant 1)** – Injection altering DNS query parameters. **Medium**
453. **DNS Injection (URL Parameter Variant 1)** – Injection via URL affecting DNS lookups. **Medium**
454. **DNS Injection (Form Field Variant 1)** – Injection through form inputs into DNS systems. **Medium**
455. **DNS Injection (Conditional DNS Variant 1)** – DNS injection that triggers under certain conditions. **Medium**

456. **DNS Injection (Stored DNS Injection Variant 1)** – Persisting malicious DNS entries. **Medium**
457. **DNS Injection (Dynamic DNS Injection Variant 1)** – Dynamic manipulation of DNS responses. **Medium**
458. **DNS Injection (Advanced DNS Variant 1)** – Advanced exploitation of DNS injection. **Medium**
459. **DNS Injection (Header Variant 1)** – Using HTTP headers to manipulate DNS-related functions. **Medium**
460. **DNS Injection (OS-Specific DNS Injection Variant 1)** – Exploiting OS-specific DNS query processing via injection. **Medium**

Marking What You've Learned

For tracking, organizing, and "marking off" the injection vulnerabilities you've learned about, **Obsidian** is an excellent choice. It's a Markdown-based note-taking and knowledge management tool that lets you:

- Create individual notes for each vulnerability or category.
- Use tags and links to connect related topics.
- Visually map your learning with graph views.
- Easily mark, check off, or update your progress as you master each topic.

Other popular alternatives include Notion or Roam Research, but many security professionals appreciate Obsidian for its local storage, flexibility, and robust linking features.

Final Note:
This list is designed for educational and illustrative purposes. In practice, many of these "variants" overlap, and mitigating one injection vector often helps mitigate several related ones. Always refer to current best practices and security guidelines when learning and applying these concepts.

Happy learning and stay secure!