Vulnerabilities of Python

When working with Python, there are many vulnerabilities programmers must look out for. These vulnerabilities often include:

**SQL Injections (SQLI)** – A malicious user controls the execution of SQL statements for an application at the backend database server. There are several types of subclasses in SQLI, these consist of:

* In-band SQL Injection – Most common and easy to exploit where the attacker can use the same communication channel to both launch the attack and gather results.
* Error-based – This relies on error messages thrown by the database server to obtain information about the structure of the database.

**Cross Site Scripting (XSS)** – In XSS, a malevolent user can trick any web application to steal stored cookies, saved passwords, and script code that served unsuspecting users of that application.

**Cross Site Request Forgery (CSRF)** – This security vulnerability occurs when a compromised website is forced to perform an action by another user that is logged in that clicks on a button. Also, it includes the hacking or logging into of a website with others’ login details.

**Lightweight Directory Access Protocol Injections (LDAP)** – This vulnerability occurs when a malicious user inserts/modifies LDAP statements that lead to the speculations.

**Command Injections** – This is where a malicious user executes OS commands on a web server by abusing it in order to insert their own commands to gain complete control over a server.

**XPATHI** – This occurs when a malevolent user intentionally passes data to a website. They can use that interaction find out how the data is structured in the XML, or they can access secured data that they couldn’t access before.

**Timing Attacks** – These are essentially a way of exposing the behavior of the algorithm by timing how long it takes to compare provided values. Timing attacks require precision, so they don’t typically work over a high-latency remote network.