Fealy J C (FCES)

18024092

Security Vulnerabilities

Port Vulnerabilities

Any port on a computer system can pose a potential threat to an actor wishing to penetrate the system unlawfully. Many computer users may overlook potential flaws in their ports. While the most commons and serious port are often lockdown and protected, many are overlooked as they are not seen as dangerous. However, any unsecured port will pose a security threat and this will be abused by malicious actors. Commonly hacked ports include: TCP port 21 and TCP port 22 (Beaver, n.d.), these ports respectively control the File Transfer Protocol and the Secure Shell, these are often over look, yet pose a serious security vulnerability.

Malicious actors will use port scanning tools, these tools echo out to listen for open ports on any network. Once an open port has been detected the actor will attempt to access this port and gain access into the system. However, the same tools malicious actors use to gain access the system administrators will use to watch for and close ports. By system admins scanning for open ports on their networks they can easily identify and close such ports reducing security threats on the network (Gelnaw, 2021). Simply by doing routine security scans on the admins systems the risk posed by port vulnerabilities is mitigated.

SQL Injection

SQL Injection is the principle of typing SQL code into a text entry field for this to be entered once the command reaches the database (SQL Injection, n.d.). As long as the actors knows the name of a table, such as “ ORDERS “ they can simply enter “ DROP ORDERS “ and this will result in an unsecure system having their orders database deleted. This vulnerability occurs from improper string error checking and just accepting any data the user enters without first validating the information.

However, a successful SQL injection can do more to a computer system then simply delete data. The injection can result in a data breach from the actor being able to pull user data from the database or copy the contents of the database (Academy and injection, 2021). Further attacks can result in a denial of service attack which will render the companies server inaccessible on the internet until the attack is over.

Steps, however, can be taken to mitigate SQL Injection attacks, these include (Rubiens, 2018):

* Privilege levels – having privilege levels inside the database, ensure that accounts connecting to the database do not have privileges they do not require at the time. This method of limited access can prevent larger amounts of data from the database during a successful SQL Injection.
* Constantly validating user fields – by validating the database inputted and to not just give the user blanket trust will ensure that any string taken in from the user will not contain any malicious code.
* As the actor cannot access any table without knowing the tables names initially, ensure that table names are obscure and kept secure from any body without a reason to know that information.

Software Vulnerabilities

Software vulnerabilities, or most commonly known as “ bugs “ can present in many different forms with each having a different level of severity. But can be non-serious, as a button on a GUI being non-functional; to more severe such as the login system encryption being bypassed. While many software bugs will just be inconvenient or displeasing to the user, some can be quite serious to the company. Many different types of vulnerabilities exist that present as software bugs, such as SQL Injections and Buffer Overflow, these each have their own solutions to solve. The main path to mitigating against many software vulnerabilities is to preform constant test and maintenance to fix the security vulnerability as soon as it is discovered.

Integer Overflow

This vulnerability occurs by real-time calculations or user inputted data exceeding the maximum or minimum value for an integer (Mohanty, 2018). For example for an 8-bit signed integer value ranges from -128 to +127. If any real time calculation went over or under this value then the result would wrap. For example with an 8-bit integer if 10 is added to +120 the result would be -126, not +130.

While at face value, this vulnerability doesn’t appear to be as threatening to a computer system, this can have great risk to a system. Such errors can cause program to crash or throw the rest of the program logic out.

Simple pre calculation methods can be employed to first ensure that calculations between 2 number will not result in an overflow.

Bibliography

Beaver, K., n.d. *Commonly Hacked Ports - dummies*. [online] dummies. Available at: <https://www.dummies.com/programming/networking/commonly-hacked-ports/> [Accessed 20 February 2021].

Gelnaw, A., 2021. *Open Port Vulnerabilities: What's the Big Deal?*. [online] BitSight. Available at: <https://www.bitsight.com/blog/open-port-vulnerabilities-whats-the-big-deal> [Accessed 20 February 2021].

W3schools.com. n.d. *SQL Injection*. [online] Available at: <https://www.w3schools.com/sql/sql\_injection.asp> [Accessed 20 February 2021].

Academy, W. and injection, S., 2021. *What is SQL Injection? Tutorial & Examples | Web Security Academy*. [online] Portswigger.net. Available at: <https://portswigger.net/web-security/sql-injection> [Accessed 20 February 2021].

Rubiens, P., 2018. *SQL Injection Prevention | How to Prevent an SQL Attack*. [online] eSecurityPlanet. Available at: <https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks/> [Accessed 20 February 2021].

Mohanty, S., 2018. *5 Important Software Vulnerabilities - DZone Security*. [online] dzone.com. Available at: <https://dzone.com/articles/5-important-software-vulnerability-and-attacks-tha> [Accessed 20 February 2021].