# Security Vulnerabilities

**SQL Injection**  
SQL injection is the process of sending in code through strings to execute one or multiple processes at once.  
Often this method is used for exploitation and unauthorised access of the database by sending in malicious code.  
The 3 types of SQL injection attacks are:-  
Union-Based – A popular type. It makes the use of the UNION statement of the SQL to extract data.  
Error Based – Simplest type. The hacker causes an error and uses the error message to extract the data.  
Blind – The hacker extracts data by asking questions to the database for Boolean-based and inducing a time delay for Time-based. For the latter method, it is used to achieve basic test for determination of vulnerability presence within the database. Both the above mentioned methods are types of Blind SQL Injection [(Latest Hacking News, 2019).](https://latesthackingnews.com/2017/10/31/types-of-sql-injection/)   
  
In the past SQL injection has been used to [steal the personal details](http://securityaffairs.co/wordpress/36528/hacking/anonymous-breached-wto-db.html) of World Health Organization employees, [grab data](http://www.pcworld.com/article/2457240/sql-injection-flaw-in-wall-street-journal-database-led-to-breach.html) from the Wall Street Journal, and [hit the sites](http://www.darkreading.com/attacks-and-breaches/fbi-blames-federal-hacks-on-anonymous-campaign/d/d-id/1112650) of US federal agencies (Motherboard, 2019).

**Policies**  
Policies are formal set of statements that ensure rules and regulation depending on what the specific policies are set for. Some of them serve as security of data, rights of the software and even consumers. For example Data Protection Act, General Data Protection Regulation, etc.  
These policies have been implemented within the software to ensure security of data as the PIM software deals with personal information of several employees and can be accessed after authentication. If an employee or anyone with the access has malicious intent with the data they can be held accountable by law.  
The policies implemented within the program were referenced using the source, (Itgovernance.co.uk, 2019).

**Database inconsistencies**Database inconsistencies is basically the lack of consistency within the database between tables.  
It is an administrative problem and not a database technology problem.  
The said vulnerability is taken care by consistently looking after databases and avoiding data redundancy (Osborne, 2019).

**INFORMATIONAL: ROBOT Vulnerability**

This is active when a “robot” is used to forcefully retrieve data from a particular environment. This affects web servers that are configured to use RSA encryption key exchange. By exploiting the vulnerability, an attacker can recover the session key used for one or more sessions, and thereby decrypt communications to and from the web server (Knowledge.digicert.com, 2019).

**Zero-day vulnerability**

This is a newly discovered vulnerability that needs to be fixed. It is when one has no day to fix the problem and the attacks existing in between the fix and the day of the vulnerability getting public is called a zero-day attack (Us.norton.com, 2019).

**Integer wrap**

When using an integer it is common that the number can become too large for the computer to handle and will end up turning the number into something completely different. This happens due to the binary space that has been allocated for the variable. Usually integer numbers comprise of 32 bits which means that it can reach a maximum value of 2,147,483,647, which has a binary number of 1111111111111111111111111111111, meaning if we add any more numbers to it, it will reset back to 0 and carry on from there.

An easy way to combat this is to have checks in the maths to ensure that the numbers don't go beyond this value and cause issues across the program. There have been several issues with this vulnerability, especially the Y2K problem (Wikipedia, 2019), also known as the year 2000 problem. This arose because the integer for the date only went up to 31st of January 1999, and would wrap around to January 1st 1900, instead of 2000. This was a major issue at the time because it would cause systems all over the planet to fail. Another example of this vulnerability was the game Pac-Man, if a player reached level 256(Pac-Man wiki, 2019). Once the player has reached this level, the screen would interpret code wrong and the screen would become a jumbled mess of characters instead of a real screen.

# 

# ****Memory allocation****

There can be a few issues with memory allocation that can cause an error, specifically with restriction of space. If a program starts to use more space than it has been allocated it will overwrite program data and cause issues at run time(Deckard, 2019). This is known as a buffer overflow. This can happen to any program that doesn’t have any of the relevant security measures. To combat this we allocate a fixed size for the buffer to stop any other data from disrupting or other corrupting program data.

# ****Type casting Error****

Type casting or type conversion is an error that occurs when the user inserts data that isn’t compatible with the code(Wikipedia, 2019). This is very common if security measures are not implemented. A very common example of this is where the computer is expecting a whole number, but receives letters or even decimals. This causes the program to crash if not taken care of in the code with safeguards.

# 

# References

* Latest Hacking News. (2019). *Types of SQL Injection - Latest Hacking News*. [online] Available at: https://latesthackingnews.com/2017/10/31/types-of-sql-injection [Accessed 3 May 2019].
* Motherboard. (2019). *The History of SQL Injection, the Hack That Will Never Go Away*. [online] Available at: https://motherboard.vice.com/en\_us/article/aekzez/the-history-of-sql-injection-the-hack-that-will-never-go-away [Accessed 3 May 2019].
* Itgovernance.co.uk. (2019). *UK data protection law: GDPR, DPA 2018 and PECR | IT Governance | IT Governance UK*. [online] Available at: https://www.itgovernance.co.uk/data-protection?fbclid=IwAR1bzwZ5XBXLEx8fk3J8EyUmFgGXJ4nUGQ33QwuwCldASpmjx9T2qVxyIEs [Accessed 3 May 2019].
* Osborne, C. (2019). *The top ten most common database security vulnerabilities | ZDNet*. [online] ZDNet. Available at: https://www.zdnet.com/article/the-top-ten-most-common-database-security-vulnerabilities [Accessed 3 May 2019].
* Knowledge.digicert.com. (2019). *INFORMATIONAL: ROBOT Vulnerability*. [online] Available at: https://knowledge.digicert.com/alerts/ALERT2552.html [Accessed 3 May 2019].
* Us.norton.com. (2019). *Zero-day vulnerability: What it is, and how it works*. [online] Available at: https://us.norton.com/internetsecurity-emerging-threats-how-do-zero-day-vulnerabilities-work-30sectech.html [Accessed 3 May 2019].
* Wikipedia (2019) *Year 2000 problem*. Available at: <https://en.wikipedia.org/wiki/Year_2000_problem> [Accessed 1/4/2019]
* Pac-Man Fandom (2019) *Map 256 Glitch*. Available at: <https://pacman.fandom.com/wiki/Map_256_Glitch> [Accessed 1/4/2019]
* SANS Institute (2019) *Defeating overflow Attacks*. Available at: <https://www.sans.org/reading-room/whitepapers/securecode/defeating-overflow-attacks-1403> [Accessed 1/4/2019]
* Wikipedia (2019) *Type Conversion*. Available at: <https://en.wikipedia.org/wiki/Type_conversion> [Accessed 2/4/2019]