**Threats and risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Threat or risk** | **Description** | **Consequences** | **Prevention** |
| 1 | Acts of human error: incorrect data entry | Incorrect data entry is a risk where the developer or user enters the wrong data into a system. For example, a user enters a height incorrectly. Another example could be that a business manager might enter an employee’s time worked wrong which may be 125 but they meant to enter 115. | The system may not behave in the way it is intended.  An incorrect data entry can lead to many errors and data integrity is compromised. Data is misrepresented which could lead to inaccurate business decisions and subsequent loss of revenue and an inability to accurately assess business performance. | Systems can include as much manual and electronic validation for data entry.  Manual validation: Taking extreme precaution when entering data such as reading it over whilst and after the data has been entered to double check it is right.  Electronic validation: format masks such date format, enforcing numeric for integer fields.  Thorough training for all users of the system to knowledge of correct data entry procedures. |
| 2 | Acts of human error: inadequate training | Inadequate training is when a stakeholder internally in the organisation does not have enough training to be suitable for the work that they are | Data may not be safely stored and will be more vulnerable to attacks. | Restricting user access until the user has completed adequate training is a simple and effective method of preventing this risk.  Run frequent, relevant, and timely training and ensure that the employee is fit more the job they are assigned. |
| 3 | Acts of human error: inexperience | Inexperience is a risk where the developer or user is not experienced enough with the software that they are using. | A consequence of being inexperienced with software could be that developers do not know how to mitigate problems easily or could easily slip in bugs as they don’t have experience with the whole system first. | Create development branches which undergo code review before they are merged to prod in workspaces for unexperienced developers, and make sure developers have experience with the toolchain before hiring them. |
| 4 | Acts of human error: ignorance of policy requirements | Ignorance of policy requirements is where the developer or user is not fully aware of the policy requirements and does their job without abiding by the policy. |  |  |
| 5 | Intellectual property compromised: piracy | Compromised Intellectual property can be the result of a data breach. | This could result in piracy of a user's intellectual property. This could result in financial implications for that users as they may have been able to sell that property. This could also have legal and reputational consequences. | This could firstly be prevented ensuring that no unauthorised users have access to the data. This piracy could also be prevented by implementing digital rights management on the files that need to be protected. Another solution would be to include a watermark on any of the data that needs to be protected. This would result in anyone that sees the material, knowing who the creator / owner is. |
| 6 | Intellectual property compromised: copyright infringement | Intellectual property theft or Copyright infringement is where an individual or group use a third parties’ software or idea with very little to no alterations. | Infringement on intellectual property or copyright can have legal and public consequences such as financial ramifications or possibly prison time.  Having your intellectual property or copyright infringed could result in loss of credit, revenue or client interest. | Common intellectual property or copyright infringement prevention practices include:   * Watermarks * DRM |
| 7 | Espionage: unauthorised access | Unauthorised access is when a user can either gain access to a file or an entire system that they do not have permission to access. | This could result in data becoming compromised and accessed by unauthorised individuals or organisations | This could be prevented by forcing all users to use two factor authentication in order to access the system. This would prevent attacks as the system could not be accessed without the 2FA code.  Unauthorised access can also be prevented by ensuring that all traffic is encrypted. |
| 8 | Espionage: data collection |  |  |  |
| 9 | Extortion: ransomware | Ransomware is when malicious code is injected into a system. This code then encrypts all data on the system and forces the owners of the system to may a ransom before the attackers will unlock the data. | This could result in major data loss for the organisation that has been attacked. The organisation would likely have to pay a large sum in order to unlock their files. There is also the risk that even after the money is paid the hackers may not unlock the files. | This could be prevented by keeping regular backups of the data that could be quickly restored if there was an attack. Ensuring that all employees do not execute any unauthorised applications that could include ran |
| 10 | Sabotage or vandalism | A Sabotage attack is a deliberate action aimed at a weakness in an application designed to obstruct usage or destruct the software. | This could result in an outage of the application that could result in paying customers being unable to access the application. | This could be prevented by ensuring that no single person can take down the application. Sabotage can also be prevented by ensuring that users only have access to files that are essential to their tasks. |
| 11 | Theft | Theft is the act of stealing. Aside from the stealing of physical assents, attackers can take Intellectual property and data. | This could result in sensitive information or data being released, infringement of law, loss of data or loss of equipment.  In more specific scenarios, companies may lose money, productivity, and customer loyalty. |  |
| 12 | Malware: viruses | Viruses are any malicious code the attaches itself to executable files and travels with them. **Payload** is activated by human actions. Generally, disrupts regular functions of the program or file.  Viruses also replicates itself by modifying presently existing code and can damage programs, delete/corrupt data, change data, or simply replicate itself to reduce a computer’s performance.  **Payload** - refers to the destructive potential of the malware; these are pieces of code that perform actions beyond simply spreading the virus, which is what makes it destructive. | If the company’s software were infected by a virus, this could result in a loss of productivity, loss of data and therefore a loss of income or profit for the company.  In dire situations, if access to data may be a life-or-death reliance – such as in a hospital. | Since viruses attach themselves to files, they are likely to travel through emails, chats, or files downloaded from the internet. Thus, isolating a computer access to the internet- keeping it connected to only essential services to perform its intended job- will reduce the chance of viruses and any malware from infecting software.  Having an antivirus software could be implemented to rat out any currently present viruses. |
| 13 | Malware: worms | Worms are a type of virus that can spread across multiple computer systems, and can modify, corrupt, or delete data or information.  These programs can sometimes run undetected to grant the attacker remote control of the machine, turning it into a “zombie bot” which together with other machines created a “botnet”. | Computer worms can have a variety of consequences, generally depending on the intent of the payload.  A typical worm may harm host networks by consuming bandwidth and overloading servers if it has the intent to do so or could result in a loss of data or information. **Basically, any other malware as payload**  Worms that create zombie bots generally have the consequence of making computers slower and reducing performance. | An organisation must maintain good cyber hygiene, which is the day-to-day practice of maintaining the basic health and security of software and hardware. This includes routine check-ups and analysis of the systems.  Consistent patches and system updates also help, along with anti-virus software that can detect suspicious activity. Sensitive data can also be encrypted for extra security. |
| 14 | Malware: trojans | Trojans are a form of malware that take the form of other “harmless” software but contain malware. This is commonly in the form of “free virus-checking software”.  Trojans generally get the user to authenticate their installation themselves, so the user must be convinced to enter the appropriate auth to activate installation.  These attacks can be difficult to counter, as the weakness is the human themselves, who have administrative permissions to install the software. | Trojans, like viruses or worms, carry **payloads**. This means they could have almost any other malicious intent or carry other forms of malware. | Most preventative measures for trojans are defensive, such as full incremental backups of the system and user files, as well as anti-virus software.  The disadvantage of this is that newer malware will not be detected and will go unnoticed.  The only other preventative measure to deter trojans is to train staff or disallow installations from unknown sources. This could mean blocking all non-approved Hi Vihan applications on a system. |
| 15a | Malware: Spyware | Spyware is software that gathers information and sends it to another entity in some way. | Generally, spyware violates the privacy of the party that is being spied on, which can lead to larger more dire consequences such as the exposure of sensitive data. | Once spyware is installed, only an antivirus can identify and get rid of it. |
| 15b | Malware: keyloggers | Keyloggers are spyware that logs all keyboard commands that the user initiates and returns that information to the attacker. | As a result of the attacker having information about what the user has been entering into their device, from this information they may be able to find ways to access the system that the user was using, and launch an attack | Keyloggers can be prevented by ensuring that anti-virus software is up to date and that only trusted programs are being installed on the system. |
| 16 | Web application risk: XSS | XSS, also known as Cross-Site Scripting are a type of injection, where malicious scripts are injected into otherwise trustworthy sources.  This normally occurs when a website takes user input without any validation or checking.  An example of this could be sending someone a file on Messenger, and that causing their browser to execute JavaScript, and since it’s from a “trusted” source, this would give the script access to sensitive information such as cookies or private tokens.  Three types of XSS flaws:  ***Stored attacks*** are when the injected script is permanently stored on the target servers, such as in a database, message forum, visitor log or comment field. The victim then retrieves the malicious script from the server when it requests information.  ***Reflected attacks*** are those where the injected script is reflected off the web server, such as in an error message, search result or other response. This could be by clicking on a link, submitting a specially crafted form, or browsing to a malicious site.  ***Dom-based XSS*** or ***client-side XSS*** is where the local user browser is modified so that it behaves in an unexpected manner. The page looks the same, but the browser is modified in some way, and is different from other XSS attacks in which the server is modified in some way. | The consequences of an XSS exploit range in their severity.  If no sensitive information is exposed and scripting ability is limited, the issue could be at most a minor inconvenience, such as a popup window with an ad, or social engineering scam.  More severe consequence would involve the exposure of a user’s private token (presumably session token), allowing the attacker to completely compromise their account by taking it over. | The easiest form of prevention would be to make sure the server us checking and validating all user input before processing it in other forms. This sanitisation and validation of input fields will likely prevent most attacks.  Further measures can be taken by keeping software up to date, having a Content Secure Policy (CSP) along with CORS rules, and implementing a web application firewall. |
| 17 | Web application risk: SQL injections |  |  |  |
| 18 | Web application risk: XML injections (XPath) |  |  |  |
| 19 | Web application risk: DDoS |  |  |  |
| 20 | Software attacks: Bots and web scraping |  |  |  |
| 21 | Social engineering attacks: phishing |  |  |  |
| 22 | Social engineering attacks: spearphising |  |  |  |
| 23 | Social engineering attacks: pharming | Pharming redirects users to false websites that imitate the legitimate URL. Pharming  can affect large numbers of users simultaneously by a ‘poisoned’ DNS server that  re-directs to the wrong website. |  |  |
| 24 | Forces of nature |  |  |  |
| 25 | Technical hardware failure | Hardware failures can occur as a result of human error, natural disaster, accidents or |  |  |
| 26 | Technical software threat: back door |  |  |  |
| 27 | Technical software threat: Brute force |  |  |  |
| 28 | Technical software threat: password dictionary |  |  |  |
| 29 | Technical software threat: man-in-the-middle |  |  |  |
| 30 | Technical software threat: rootkit |  |  |  |
| 31 | Technical software threat: sniffer |  |  |  |