Jason Less
404-640-158
E96A, Reiher

Internet of Things: System and Security

Introduction:

For many decades, the Internet and the technological devices developed were created to allow interactivity between people and the technology. For example, personal computers, smart phones, entertainment systems, etc. were all designed with the purpose of allowing people to physically interact with the system to exploit the various functionalities and capabilities of the system that the user desired. In this modern era, the technological field has begun to see a shift to a new arena known as the Internet of Things (IoT), which created a new idea of creating technological devices that are able to connect to the Internet without the necessity for human interaction with the device.

These IoT devices are able to be controlled remotely, and not necessarily require a human who owns the device to physically provide the system with commands for it to do its designed purpose. For example, the Nest Smart Thermostat is a IoT device that is connected to the Internet, and studies a "family's routines and will automatically adjust the temperature based on when you're home or away…" (Albright, 1). These IoT devices, while very powerful in their capabilities to change our everyday lives, as well as the future of the technology, present a new set of challenges and difficulties for their designers that come in the form of system design and security. Unlike the typical Internet and the associated technological devices, IoT devices are an entirely different breed of technology that require a different approach to designing the devices to function without the typical human interaction, and present new security problems to defend against potential attackers who wish to take control of the system (e.g. an attacker who gains access to a smart car, and locks the breaks from functioning properly).

IoT Design -- Issues and Choice:

When designing an IoT device or system, there are many ideas to consider about how best to design the system. In addition, there are various issues and challenges as well, in which tradeoffs must be considered and taken in order to produce a successful IoT system. Two main design issues to consider are user interfaces and security. It is important to provide a system that is enjoyable and easy to use by the client, and thus, is able to sell and be successfully marketed to the target audience. At the same time, there needs to be a set of security measures taken to ensure that the system maintains the integrity and privacy of the person who owns or uses the device.

First of all, it is crucial that the system itself is easy for the client to use it, and doesn't require an immense amount of work to operate the system or get it to work each time it is to be used. For example, if there is an IoT device that unlocks a door using various sensors (e.g. with buttons, keypad, light sensors, etc.), it would be very inconvenient to the user if entering the password was a very complex process (i.e. it might involve using each of the numerous sensors each time, and the password needs to be entered every ten minutes or so). Therefore, it is important that the system be easy to use, and is convenient for the general public; otherwise, it is highly unlikely that people would be willing to buy the product. This idea corresponds to making sure that the user interface is user-friendly, which means that it should be easy to use, in addition to being aesthetically pleasing to the user. For instance, smart phone applications such as Instagram and Twitter are designed with the general public in mind, and the companies make sure that the applications are easy to use, navigate, and are aesthetically designed for the user experience. Thus, an IoT system when being designed, should keep in mind making the User Interface (UI) and the User Experience (UX) as simple, convenient, and as creative as possible to attract customers.

The advantages of UI and UX are that it provides a system that is enjoyable for the public to use it. If it is enjoyable to use, then the product will sell more, which would make it more successful. In addition, if the system is easy to use, and is convenient for people, this will just add to the success of the system. A disadvantage is that providing a system that is easy to use my hurt the strength of other design ideas such as security protocols of the system (as discussed below).

In addition to having a user-friendly IoT system, it is important to keep in mind the security aspect of things when creating the system. Any device or system that can connect to the Internet and contains or transfers sensitive or private information is vulnerable or at risk of a cyber attack if not designed properly. Especially with IoT devices, which involve systems that are designed to have sensors that observe human behavior, take in large amounts of data, and are designed to perform a specific (perhaps important) task, there needs to be security measures taken to ensure that the system is well-defended against potential attackers or malware that could disrupt the integrity and reliability that the system is supposed to provide. For example, even with a simple IoT device such as a "smart doll", which responds to a user with certain catch phrases or carries on a conversation in response to certain buzzwords spoken by the user, it is important that such a device not be able to be hacked or hijacked. If such a system had an audio system (which it most likely does), or potentially even a camera set up, it is important that such a system not be compromised by an attacker; otherwise, the attacker could get access to spying on the owner both visually and verbally, which is obviously not desired by any person who is buying a doll for their child. Therefore, proper security measures should be taken to keep IoT systems secure.

The advantages of having a secure system are obvious. Security allows a system to maintain the CIA principles of confidentiality, integrity, and availability. The system would be able to maintain the privacy of a user's sensitive information, and would be secure against potential attacks that could disable the system or corrupt information. The disadvantages of providing security to a system are that it could make a system more difficult to use, or require more work from a user. More security might mean having a user authenticate themselves to the system more frequently, which would be very inconvenient for an owner of the system.

However, the two issues described above often conflict with one another's goals. For example, if the IoT system accepts a password to unlock some other system, it is important that the system be secure, but it is also important that the system not be too hard to use or requires a lot of work each time to use the system, otherwise people won't want to use the system. Therefore, there are tradeoffs to be made when designing a system with these two ideas in mind. The system should be secure to protect the integrity of it, but it should not be made to difficult for someone to use it. To resolve the conflict, sacrifices obviously have to be made without compromising the system's integrity. While both are very important, a company might decide that the user interface and experience are more important as this is the whole reason people will buy the product, and the product will be successful. Then after making a friendly UI and UX, security measures can be taken that may not be the strongest, but are in a sense, good enough for the overall functionality of the system.

IoT Security -- Problems:

There are plenty of security issues that must be considered when designing an IoT system. Three such problems that could occur are hackers attempting to crack/steal passwords, hackers trying to steal sensitive information, and downloading malware or viruses to the system.

First of all, many IoT systems might have some form of authentication that needs to take place at some point in an owner's use of the system. These forms of authentication may come in the form of providing a username/password, or using some key cryptography. Either way, this is authentication based on what you know (and possibly what you have depending on the method of authentication). Any form of authentication is prone to attacks such as brute-force or dictionary attacks that attempt to crack passwords, so that they can gain access to the system. Therefore, it is necessary that a password be as secure as possible to deter such attacks from taking place. An example of this form of security problem is the IoT device that we created that uses light sensors to send a password to a remote server to unlock a door. For our secure design, we implemented a stronger password by increasing its length, number of characters, and enacting a timeout, so that a user can't make an unlimited amount of attempts. By doing this, an attacker would be deterred from performing brute-force and dictionary attacks, as the attack would take a large amount of time to execute. The characteristics of the IoT device that would make it vulnerable to these forms of attacks are the authentication process (as just discussed). IoT devices that don't require any authentication wouldn't have to worry about these security problems.

Second, if an IoT device involves transmitting data back and forth between the device and some remote site (e.g. a cloud server or handling site), then this would make the system vulnerable to attacks such as man-in-the-middle or replay attacks. For this case, the system involves sending data over a network that at often times isn't secure, and data can be compromised easily if the correct security measures aren't taken. Data should never be sent over a network in plain text, and thus, some form of encryption should be performed on the data to make it unreadable by potential attackers who are snooping on the network. For example, the IoT system that we created involved sending a username/password across a network to a remote server to authenticate that the user of the system was who they said they were. This data was initially sent over the system in plain text, but anyone who was snooping on the given network channel could easily see the password being sent over. Therefore, the data should be encrypted to make it unreadable to third parties that shouldn't be involved in the transmission (i.e. only the client and server should know the data being sent/received). The characteristics involved in this particular IoT device involve how the data is sent to and from the system. As discussed, encryption should be performed on the data to secure the data. This involves the use of some form of cryptography (i.e. symmetric key or public key) to protect the integrity of the data. There are a large number of IoT systems that involve transmitting data to and from the system; however, systems that remain local and don't need to send data over a network wouldn't be at risk of this security problem.

Lastly, as with any technological device, there are updates and upgrades that need to be constantly made to keep up-to-date on security issues and patch bugs found with the system. These updates involve downloading some software packages that are sent to the system across a network. In addition, the packages are downloaded from a website (most likely the creators of the IoT device) that must be authenticated to ensure that they are who they say they are. It would be catastrophic to download viruses or malware to a system from a website that is malicious, and thus, could gain access to the system. The updates to the system are required, otherwise the system would be out-of-date, and would be prone to other forms of attacks. Thus, using the ideas of authentication and cryptography as discussed above, the website from which the upgrade needs to be downloaded from should be validated to assure authenticity. An example of this problem would be downloading a package from a website who is impersonating a trusted company that sends viruses to the system to hijack it. The characteristics of this IoT device are similar to the previous security problems discussed, and involve how the system performs authentication. Most IoT devices need to update their systems from time to time, and thus most need to be concerned with this problem.

IoT Security -- Solutions:

To design a secure IoT system to combat the security issues discussed in the last section, it is appropriate to provide secure authentication of usernames and passwords, encryption of data sent over untrustworthy networks, and authentication of websites (e.g. use of Certificates and cryptography) to download update packages. These concepts are widespread and most, if not all, IoT systems integrate these security measures into their systems (if required).

First of all, to prevent hackers from cracking passwords, several approaches can be taken. The first approach is to increase the length of the password and to use more characters for each character of the password. For example, a password that is four digits long, and only involves the use of numbers is a very weak password. It can easily be cracked by guessing (brute-forced) as there are a very low number of possible passwords. Thus, the password should be made up of many digits (perhaps 8 or so), and should involve the use of many different characters (i.e. not just digits, but also alphanumeric characters as well as special symbols). By doing this, it would be harder to brute-force the attack, and also would deter against dictionary attacks. Another approach is to encrypt the username and password on the server that it is stored. Some websites (less secure ones) store the passwords of users in plain text. Thus, if anyone got ahold of the password files of the system, then all the users of the website would be compromised. To solve this problem, the data should be stored in a form that is unreadable to attackers. This can be achieved by using a cryptographic hashing algorithm to encrypt the data. Then the hash of the username and password can be stored on the server (e.g. the cipher text), instead of the easy to steal and take advantage of plain text version. Lastly, storing a salt in addition to the password can strengthen a password even further. A salt is 32 or 64-bit random string (also encrypted) that is stored with a password to deter dictionary attacks even more. For example, every entry of a dictionary now needs to have 2^32 or 2^64 more possible entries for each salt combination. This would make the time to perform a dictionary attack incredibly lengthy, and perhaps even unreasonable to even attempt to perform. All three of the solutions provided would make the system as secure as possible for this security issue; and thus, all solutions should be used at the same time.

Next to prevent hackers from performing man-in-the-middle and replay attacks; the data being sent over the server needs to be encrypted. One solution is to only send non-sensitive data over the network. In this approach, it doesn't matter if attackers see the information being sent across the network, as it doesn't contain any sensitive data. Therefore, no encryption of the data needs to take place at all, and the information can just be sent over normally. Another solution is use key cryptography to communicate over the network by exchanging the keys between the server and the client. In this solution, the data is being encrypted, but the keys being exchanged only takes place between the client and the server, and not some authentication server (which will be discussed next). This is a safe way to keep the data safe, but the only problem is the exchanging of keys between the parties, which could be compromised. Lastly, an authentication server could be used (i.e. KERBEROS), which would set up a secure communication network between the client and the server. Using this approach, the communication would be secure (perhaps with SSL), and would involve a secure way of exchanging keys as well, as it was all monitored by the authentication server. Someone might choose the first solution, if they don't have any sensitive data to send over the network, and thus, it would be simpler to set up the system, without having to deal with encrypting the data. Someone might choose the second solution, if they already know each other, and perhaps they already have each other's keys. Lastly, an IoT system might use the last solution as the parties don't know who they are communicating with (i.e. they don't know if the server or client is who they say they are), and thus, want to set up a secure communication channel to exchange keys securely, and perform secure transmission of data.

Lastly, to deal with protection against viruses and malware due to downloading updates or upgrades to the system, several approaches could be taken. The first is to only download packages from the vendor of the IoT system. This makes the most sense, as the vendor is the one who created the system, and thus we should be able to trust that they are providing secure updates to the system. Another solution to the problem is by utilizing Internet certificates to validate the authenticity of the websites where the packages are taken from. Certificates allow for websites to be authenticated, and thus trusted by the IoT system. Thus, this approach would allow for the packages to be downloaded from other third party websites that might not be trusted, but can be validated by requesting and authenticating the information provided by their certificates. Lastly, the IoT system can employ some constant scanning of the system using an anti-malware system, which would prevent the system from downloading questionable data, find unknown suspect-data, and then proceed to remove the data (viruses) to clean the system. By continuously scanning the system, it can be better defended against malware that might have been unknowingly downloaded or injected into the system. A system may only use the first solution, if the system doesn't require too many updates, and the updates that it does require only need to come from the vendor of the system. Other IoT systems may need to download various packages from some other trusted third party websites, but those only requiring infrequent updates from the creators of the system would only employ the first solution. Systems may use the second solution if they need third-party website updates. Certificates allow for these third-party websites to be trusted, and thus, this solution is a very common one. Lastly, systems can use the third solution as a secondary defense to either of the first two. Scanning the system for viruses is a backup, in case some malicious software was downloaded accidentally, and scanning can help to clean up this occurrence.

4. Evaluating IoT Design:

IoT systems, like any piece of technology, can be evaluated using many different metrics from latency to throughput to the system performing its desired functionality. First and foremost, an IoT system was created to perform a certain action or function. Therefore, it should always be evaluated on its completion or correctness of it performing its desired functionality. In addition, IoT systems should be evaluated not only by doing the right thing, but also on how well it performs its function. People don't want a system to perform its function if it is going to do it very slowly and inefficiently. For example, when someone searches something on a search engine, they want it to load instantaneously (or very quickly), and not have to wait five or more seconds each time something new was searched. Therefore, the performance of the IoT system should be evaluated, which can come in the form of latency, throughput, and network bandwidth (if the system communicates with a network). Another metric to evaluate an IoT system is security. The system should be able to protect the privacy and sensitive information of the user. Security can be evaluated by using the CIA principles. The systems should preserve the confidentiality (e.g. protecting the password of the user), integrity (e.g. the data is not corrupted when being sent over a network), and the availability (e.g. the system doesn't get blocked from use by an attacker).

Conclusion:

The Internet of Things presents a whole new set of possibilities to improve the every day lives of people, and advance technology in a new and impactful way. However, as with all great inventions, there are various challenges and issues that arise in order to make the system as successful as possible, while providing the necessary securities and defenses to ensure the integrity of the system and protect the people who use the system.

References:

Albright, Dann. "15 Examples of Internet of Things Technology in Use Today." *Beebom.* N.p., 19 Feb. 2017. Web. 16 June 2017.

Reiher, Peter. "ENGR 96A Lecture Slides." 16 June 2017.