Combating IoT cyber threats

Security best practices for IoT applications

David Whitelegg

September 04, 2017
(First published September 30, 2015)

The Internet of Things is changing the way that businesses operate. These changes make the security of IoT devices even more crucial, considering the time and money that is required if a hacker breaks through the defenses. This article outlines the best practices for securely developing robust IoT solutions.

Innovation, efficiency, and cost-savings are driving the rapid emergence of the Internet of Things (IoT). Away from the media limelight of smart home appliances and wearable gadgets, IoT is revolutionizing business operations within every industry sector and is a proven game changer within logistics, warehousing, production lines, pipeline transportation, and traffic management. The networking and remote control of connected devices is providing new levels of productivity and information management and capabilities.

For example, Continental Tires previously relied upon hand written notes to find carts of rubber within its large tire production plants; the system meant staff often found it troublesome to locate the carts, which translated costly production delays for the business. The problem was solved with IoT technology, which allowed the carts to be directly integrated and connected with the business inventory management system over a wireless network. Staff using a mobile device app, could locate any of the carts in the large plant in real-time, while the plant management team could know each cart's contents and the time involved in moving carts from their current locations to the production lines, allowing management to improve the efficiency of plant production exponentially.

Another example is the Kenya Pipeline Company, which upgraded its pipeline infrastructure with connected IoT devices that could sense oil pressure levels, temperatures, and flow speeds. These simple networked devices not only provided real-time metrics and alerts to the operational management team, but also sensed and automatically reacted by shutting down the pipeline when oil leaks were detected. Whether the leaks were caused by accident or by oil thieves, the fast detection and response saved the company millions of dollars and limited the environmental impact caused by any leaks by instantly closing oil flows to compromised pipes.

This new age of IoT connectivity, data collection, and management of physical world objects introduces new security risks, from sophisticated automated cyber attacks to hackers who are hell-

bent on stealing data and causing havoc. The security of IoT devices is heavily dependent on the software and applications that are used to manage them, which puts software developers on the cyber front-line and which requires that they know how to develop secure IoT applications.

This article outlines the best practices for secure coding techniques and security functions that will help development teams to produce resilient IoT applications that mitigate IoT security risks.

Examples of IoT cyber threats

IoT devices, like all networked computers, are attackable, either directly over the network to which they are connected to or indirectly through the applications that control them. Hackers target IoT devices for several reasons. Cyber criminals seek to profit by gaining control of IoT devices to steal data, blackmail the business, or use the IoT devices to perform massive Distributed Denial of Service (DDoS) attacks on behalf paying customers. Nation-states are known to target IoT devices involved with critical national infrastructure and for espionage. Some hackers just enjoy causing chaos for community kudos and self-worth. Even cyber terrorists are taking an interest in attacking IoT devices, because it is more than perceivable that some IoT cyber attacks could cause real world damage, including injury and death. Consider these examples of IoT cyber threats:

- A smart doll for children called “My Friend Cayla” was released in 2014, the doll used speech recognition, a mobile app, and the internet to interact with children, holding conversations and answering their questions. By definition, the doll is an IoT device and soon after its release, it was hacked by a security researcher to initially use foul language and then as a creepy and rather sinister spying tool, where by an attacker could use the Bluetooth default setting on the doll to communicate with the child from outside the child’s home. The security researcher had taken advantage of both the poor firmware security on the doll (the IoT device) and the poor application layer security on the application that controlled the doll. In February 2017, the German government advised parents to destroy the Doll.
- In October 2016, internet DNS service provider Dyn was taken down by one of the largest Distributed Denial of Service (DDoS) attacks in history. The cyber attack resulted in GitHub, Twitter, Reddit, Netflix, Airbub, Starbucks, PayPal, and many other company websites going offline. Behind this mass cyber attack was a large botnet of compromised devices, which included over 152,000 malware-compromised IoT devices. It was the first time that IoT devices were used in such high numbers in a DDoS attack. Each individual IoT device had been compromised with malware called Mirari, which allowed the attacker to remote control the device and have them send streams of network traffic on mass to Dyn. The security of the IoT devices was found to be poor, and had not been designed and developed to secure coding practices, which meant they were easily and quickly compromised on mass.
- In February 2015, a 60 Minutes episode demonstrated a hacking, namely an application buffer overflow attack, and then subsequent remote control of a car, where the hacker even managed to disable the car's brakes. And, in May 2015, a computer security expert hacked an airborne airplane by using the entertainment application; a subsequent FBI investigation reported he briefly controlled the aircraft. In most of the known IoT incidents, the security weakness that was exploited was at the application layer. Therefore, how IoT applications are developed might just be a matter of life and death.

- In June 2010, a highly sophisticated and unique computer worm called Stuxnet came to the world's media attention. Stuxnet was designed to target only specific software controls that are used at an Iranian nuclear plant. It used zero-day vulnerabilities in Microsoft Windows to propagate across the nuclear plant's network and to scan for the presence of Siemens Step7 software. The Stuxnet malware successfully compromised the Siemens application and issued instructions to rapidly increase and decrease the velocity of the spinning centrifuges, which caused vibrations that led to the centrifuges tearing apart. Between November 2009 and January 2010, it is estimated that over 1000 centrifuges were destroyed by the Stuxnet malware, setting back the Iranian nuclear program significantly. The Stuxnet malware demonstrates that connected machines in the physical world can be damaged by compromising the connected application that controls them.

What can we learn from these IoT cyber threats? The application layer of an IoT device provides the largest attack surface for hackers. The application layer includes any application that has connectivity with the IoT device, which can include local web applications, cloud-based applications, and smartphone or tablet apps. Therefore, application security must be an intrinsic part of the software development lifecycle (SDLC) for all IoT applications, particularly within the design, development (code writing), and testing stages.

Designing secure IoT Applications

Within the planning or design stage of an IoT application, there must be a formal "top to bottom" assessment of the planned application's security and privacy requirements. IoT application development requires a "Security by Design" approach. This approach means considering the security requirements of all the IoT application functions as part of the design stage, rather than assuming and applying security features later in the development process. Like any other bug or issue, it is more expensive and takes longer to correct security issues in later development stages. Therefore, at the application design phase, it is imperative to consider and plan for all the possible security requirements for the IoT application.

Security requirements review

Within the design stage, review the security requirements for your IoT application by completing a security requirements review.

- Plan the user account management functions. Ensure that the IoT application will have an appropriate level of account customization.
- Design a secure password reset mechanism. This process is often overlooked in favor of user convenience, but a weak password reset process can provide an easy backdoor into the system.
- Design a user account structure to limit administrative account privileges on a need-to-have basis. It is wise to separate administrative actions and rights from standard user accounts because this configuration limits the risk of misconfiguration by users, which can cause serious security holes.
- Determine how account passwords will be stored by the application. The storage of plain text passwords in databases and flat files must be avoided. The best practice is to use a salted

hash algorithm, such as SHA-256 and a salt, to compute passwords into a unique hash value, which means the passwords cannot be reversed back into plain text.
- Consider adding a two-factor authentication function, especially for applications that will process confidential data that is intended to be accessed from untrusted networks.
- Consider adding support to integrate the IoT application with account management systems, such as Microsoft Active Directory Services. This capability helps integrate IoT applications into business enterprise environments.
- If you plan to store personal or confidential data in third-party or untrusted environments, consider using encryption to protect the data at rest.
- Design a software update capability that ensures that only digitally signed (genuine) updates can be applied, and if feasible, consider an automatic update process.
- Provide a security notification function that allows the application to send security alerts, such as failed login attempts, with an enterprise security monitoring system or syslog server.
- Invite a security professional to audit and approve the application security functions and design.

Privacy impact assessment

Where IoT applications collect, store, and process personal data, they need to do so in compliance with the data protection and privacy laws that apply. Data privacy laws can have various degrees of stringent requirements depending on which country the citizen's data belongs to. A privacy impact assessment is required to ensure the applicable laws are adhered by the application and IoT device, including any cloud and third party storage or processing of personal data.

- Document and justify all planned personal data usage by the application.
- Limit personal data collection to only what is absolutely necessary.
- Consider using routines to anonymize the personal data; a process that, when done correctly, can remove the burden of meeting legal data privacy requirements.
- Plan to write routines or use solutions (like SSL) that encrypt all personal data that is stored and also as that data is transmitted over networks, including private networks.
- Ensure that there will be privacy transparency with consumers. Any personal data collection, processing, and storage, including cloud storage, must be made clear to the consumer within a privacy statement. Before you use personal data, obtain explicit user consent, and ensure that the user signs or agrees to a privacy agreement, which explains all personal data usage by the application.
- Consider other data types that are subject to industry regulations. Ensure that any regulatory requirements are fully understood and are intended to be complied with as part of the application design.
- Consider mobile application privacy. A mobile application can interface with GPS, SIM cards, device identification numbers, device data, and data from third-party mobile applications. All of these interactions can have privacy implications on the application and must be considered.

Coding secure IoT applications

View a sideshow on Open Web Application Security Project (OWASP) Internet of Things Top Ten

Many different application types can control and manage IoT devices, such as cloud-based and local web applications, mobile applications, and the software that runs on the IoT device itself. The Open Web Application Security Project (OWASP) is an organization that focuses on how to improve the security of software. This organization has an Internet of Things Project, which brings together the unique aspects for IoT security. They are currently drafting IoT Security Guidance for manufacturers, developers, and consumers.

IoT web applications: secure coding tips

Web applications are commonly used to manage IoT devices. Whether the web application is intended to be hosted directly from the IoT device, from an internal network server, or in the cloud, the development (coding) of those web applications must adhere to web application security development best practices, such as the OWASP Top Ten, which the approved 2013 list is currently under review with a planned update for late in 2017.

A common mistake with IoT web application development is to not diligently secure private-network web applications to the same degree as public-facing web apps. Internal networks can be compromised, however, and become untrusted environments. Therefore internal-facing web application vulnerabilities can be exploited by hackers and malware.

Consider these secure coding tips for IoT web applications:

- Sanitize all user inputs. This coding technique can prevent the most common web application attacks, which take advantage of poorly coded user data entry validation to inject malicious scripts, run SQL database commands, and perform buffer overflows. Therefore, it is a fundamental application coding practice for all user input fields to be cleansed to accept only a "white list" of expected characters. By all means, code client-side validation within the web browser (such as Java scripts), but input sanitization must always occur within the (server) application because most hacks are attempted by injecting code through the URL.
- Use cookies securely. When the application uses SSL to ensure that the application marks the cookies as secure, SSL automatically encrypts them too. When the web application is not intended to be SSL protected, code a routine to encrypt the cookie values with an industry recognized encryption algorithm, such as AES 256.
- Lock down application error reporting. These application error messages can provide hackers with clues on how to break into the application and user accounts. Even the most simple error message, such as "`incorrect user password`," can give away that a user account name was correctly guessed.
- Use data encryption routines. If you need the application to encrypt personal or confidential data, do not try to write your own encryption routines. Instead, use libraries that use industry recognized encryption methodologies and algorithms.
- Assess and document the security of third-party libraries that the application uses before you use them, and then continually monitor them for security vulnerabilities and patches. For example, OpenSSL, an open source library that is used to encrypt data communications, was recently found to have critical flaws, making web applications vulnerable to attack. The issue was quickly resolved by the release of an updated version of the library.

- Make sure all developers, including contractors and any third parties, are qualified and trained to use web application secure coding techniques.

IoT mobile device applications: secure coding tips

Mobile IoT applications, specifically smartphone and tablet apps, are actively targeted by hackers. The same secure coding techniques that are used with web applications are also required when you develop mobile IoT apps. However, additional application security considerations, such as mobile device authentication, telecom and SMS data communications, and further privacy risks, do exist.

Consider these secure coding tips for IoT mobile device applications:

- Assess the application integration with the features and capabilities of your mobile platform and operating system, which can vary with each mobile platform type and operating system. Integrating with mobile functions can enhance security, but they can also weaken the security of the application. For example, using the biometric fingerprint scanner on Apple devices can enhance a mobile application's authentication process; however, relying solely on a mobile device authentication system that can be disabled, might seriously weaken the application.
- Make sure that your IoT mobile application protects all personal and sensitive data while in transit and that the data is stored on the mobile device with encryption. The application needs to force the use of encrypted network services over Internet and wifi connections (that is, assume that they are public), and even include data that is sent over cellular networks. Also, make sure that the application encrypts all personal data that is stored on removable media.

IoT device software: secure coding tips

In addition to the secure application techniques already covered, you have a few additional considerations in the development of software that will operate on IoT devices, such as firmware usage and access control of physical interfaces.

Consider these secure coding tips for IoT device software:

- Ensure that you use the latest version of the IoT device firmware.
- Monitor the firmware manufacturers for notifications of firmware updates and the discovery of firmware security vulnerabilities.
- Test new releases of firmware with the application. New firmware releases might require an update in the application code.
- Review the physical interfaces requirements for your IoT device, which can require the application to provide an access control function.
- IoT applications must be able to be updated with security patches. Patches need to be digitally signed and verified by the application's update process before installation to ensure that malicious patches cannot be installed.
- IoT applications need to encrypt all personal data, and also sensitive IoT device and application data, that is intended to be stored on the IoT device's removable media (that is, memory cards).

Testing security in IoT applications

One of the most effective techniques in identifying security flaws and weakness in the development phase, regardless of the IoT application type, is to complete a code review. Code reviews need to be performed by suitably qualified coder writers who are ideally independent of the development project. Ensure that the code review is a formal thorough process, and consider using developers who work on different projects and using tools such as IBM Security AppScan Source. AppScan Source can scan for and identify web and mobile application source code vulnerabilities early within SDLC (ahead of the testing stage). It supports code that is written in JavaScript, HTML5, Cordova, Java, Objective-C, or the enterprise mobile platform IBM Mobile Foundation.

Code reviews might appear to be a costly and time-consuming addition to the SDLC, but they can pay big dividends in avoiding costly retesting, avoiding post-release security patches, and avoiding damage to your reputation by having an insecure IoT application compromised. In addition to code reviews in the development stage, the test stage of SDLC must include intensive security testing, which needs to be specific for each IoT application type.

Security testing tips for IoT Web applications

Performing vulnerability scans and penetration tests to detect software security flaws is a vital final step to take before you release IoT applications. Testing the application code prevents common application vulnerabilities such as SQL injection, cross-site scripting, cross-site request forgery, and buffer overflow attacks.

Consider these security testing tips for IoT web applications

- Use an application vulnerability scanning tool like IBM Security AppScan to security test the application and identify web application vulnerabilities. See the developerWorks tutorial, "Scan your app to find and fix OWASP Top 10 2013 vulnerabilities," for more information about this approach.
- Ensure that developers understand the problems when insecure application coding is discovered.
- Do not forget to secure all test APIs.
- Perform a penetration test, which replicates the approach that skilled hackers take. Penetration testing must be performed by certified and experienced ethical hackers (CEH). The approach is considered a security best practice for any internet-facing application. Complete penetration testing before application release, after any significant coding change, and on an annual basis thereafter.
- Ensure that penetration testers can adequately explain the vulnerabilities that are found and their method of exploitation to developers.

Security testing tips for IoT mobile applications

Use a specialized mobile application vulnerability scanning tool, such as IBM Application Security on Cloud, which is specifically designed to security test mobile applications. IBM Application Security not only detects mobile device application vulnerabilities, but it shows in detail the vulnerability to developers, along with the level of risk and solutions to address the vulnerability.

Security testing tips for IoT device software

IoT device software must be subjected to testing by security professionals and companies that specialize in finding vulnerabilities in IoT device software.

Conclusion

The development of secure IoT applications can push development teams outside their traditional comfort zone. Taking the time to analyze security functions and privacy requirements in the planning or design stage pays big dividends in developing a secure IoT application over the long term. By performing a code review with IBM Security AppScan Source in the development stage, developers can detect and correct code vulnerabilities early in development, which is more efficient than detecting and correcting vulnerabilities at the testing stage. While in the testing stage, development teams need to replicate the application layer attacks that hackers perform by using tools like IBM Security AppScan and IBM Application Security on Cloud. AppScan includes helpful video tutorials, explanations of vulnerabilities, and secure coding examples that all educate developers to improve their secure coding techniques and their confidence in writing secure IoT applications.

These development techniques benefit development teams by reducing overall development time and cost and by significantly reducing the likelihood of IoT application vulnerabilities. Application vulnerabilities that are discovered in IoT applications after their release, especially when discovered by a hacker, tend to be costly to resolve, and might even prove damaging to both the business and to the development team's reputation.

Related topics

- When combined with the IBM Cloud platform, **IBM Watson IoT Platform** provides simple, but powerful application access to IoT devices and data. IBM Watson IoT Platform is compliant with ISO 27000 security standards.
- The **IEEE Standards Association - Internet of Things** has a number of standards, projects, and events that are directly related to creating the environment needed for a vibrant IoT solution.