A Thorough Analysis of Improper Access Control

Saptarshi Laha
M.Sc. Cybersecurity
School of Computing
National College of Ireland
Dublin, Ireland
x18170081@student.ncirl.ie

***Abstract*—Access control mechanisms are often overlooked at an organizational level, which makes it vulnerable to a variety of improper access control exploits. This paper discusses the requirement of enforcing proper access control mechanisms and building robust authorization, verification, and logging system by first highlighting the business impact that was caused by some recent attacks and then delving into the technical aspects of a few common improper access control exploits that are possible if the systems are not hardened against such attacks. Whilst discussing the abovementioned, we also briefly examine the devices that are most susceptible to getting exploited by improper access control attacks due to the presence of faulty access control mechanisms or due to the complete absence of any access control mechanism being implemented within the device.**

Keywords—access, breach, exploit, vulnerability, hacker

I. Introduction

The need for access control mechanisms increases as the trust factor between individuals decreases. In today's world, the amount of data an individual or organization has access to serves as a determining factor in establishing their value in the market. With an ever-increasing production of data on a daily basis and the perpetually dropping costs of digital storage media, collection of data by different entities seem rather natural for multiple purposes ranging from behaviour analysis, trend modelling, estimation of the probability of situations and much more. However, it is mandatory for these entities to handle such sensitive information with care because if this information lands in the wrong hands, it can be exploited for monetary benefit, fame in the black-hat hacker community, social profiling to enforce outcomes, etc. Access control mechanisms are a means to limit the access to such sensitive or other insensitive data and privileges to safeguard them from malicious, unintended or unauthorized activity.

Access control primarily deals with the segregation of data and privileges amongst the hierarchy of an organization, device, or any layer of accessories in between the two. This is a necessary step that is undertaken to ensure that no single person knows enough to steal and replicate the business model of an organization or has enough privileges to exploit all the devices or manipulate all the users on a device into leaking information. Access control mechanisms are one of the primary methods of a strong defense against social engineering attacks, which are on the rise because credential cracking is getting more robust due to increasingly complex cryptographic algorithms being used for data protection. A hacker's best bet is to trick a user into revealing information to a dubious website made to look legitimate rather than exploiting a system to access the data due to tightened cybersecurity practices. Due to the lack of quality cybersecurity training, a lot of victims eventually end up keying in their credentials to one such legitimate-looking portal crafted with malicious intent. This is where access control mechanisms come in to contain the damage that a malicious user can cause to the system with the victim's privilege level or limit the amount of information the attacker has access to when logged in as the victim onto a device. However, this is an ideal scenario and if a system is not robust enough and does not enforce active access control mechanisms, an attacker can compromise the information of other users in an organization or a device at a higher or lower privilege level. This is also known as a faulty implementation of an access control mechanism, an improper access control attack, or an exploit for improper access control.

II. Business Impact

Improper access control exploits generally result in massive financial losses for the industry or the loss of their reputation in the market. Examples of situations where improper access control is exploited include, but are not limited to, data breaches, privilege escalation, spoofing attacks, social engineering, and manipulation, denial of service attacks, etc.

A single instance of a data breach at Capital One Bank reported on 29th July 2019, compromised data worth over 100 million as reported by the New York Times. The personal data of over 100 million people were stolen by the hacker, who exploited an improper access control vulnerability in the system to gain access to the information[1].

Another instance that exploited improper access control vulnerability of a database on the server this year and led to a data breach was the client database leak of Hostinger internet hosting provider which affected 14 million customers as the hacker had access to usernames, hashed passwords, emails, first names and IP addresses of Hostinger's clients as reported by PC Mag on 26th August, 2019[2].

There are instances where proper access control is implemented well in advance to reduce the risk related to improper access control exploits. A prime example for the same being the distributed denial of service attack against GitHub with traffic of 1.35 terabits per second was mitigated by their smart business practice of implementation of a distributed denial-of-service mitigation service named Akamai Prolexic as reported by Wired on 3rd January 2018[3]. This enabled them only to have a downtime of 8 minutes and then routing all malicious packets to the intermediary and allowing access to legitimate users. In this case, what could have been a devastating improper access control exploit was mitigated and fixed entirely due to proactive business-wide cyber security-specific implementations.

The threat is not always an external factor; it can be internal as well. This is the reason why privileges to systems and access control mechanisms are in place so one can mitigate the damage caused in the case of internal espionage. A recent case highlighted the event where a rogue employee

caused a data breach of 2.9 million members from Desjardins, a Quebec based financial company, as reported by Montreal Gazette on 20th June 2019[4]. This data contained sensitive information like names, dates of birth, social insurance numbers, addresses, and phone numbers. The company also mentioned that 173,000 out of the 2.9 million were business customers, which was a reputation loss for the financial institution.

Engaging in excellent team composition and cyber-safe business practices can uplift the reputation of a company while luring in more customers and thereby increasing profits. Such an incident occurred with a cryptocurrency start-up named Komodo. The Komodo Platform realized the presence of a backdoor in one of their older wallet application called Agama and exploited it to safeguard funds by moving them to a secure location before a hacker could misuse it, thereby saving an approximate figure of $13 million in bitcoins and Komodo coins as reported by ZDNet on 6th June, 2019[5]. This particular case emphasizes the need for a security audit or cybersecurity team even within a financial institution to come up with last-minute solutions in the safeguarding of sensitive data of any type by any means necessary, in this case by exploiting a backdoor to gain privileges in turn utilizing improper access control to move funds to a secure location.

III. Devices Affected

Every variant of every device in today's world encompasses some access control mechanisms in software or hardware format. These devices range from Internet of Things equipment containing low power microcontrollers embedded in regular use appliances to make the task easier, to enterprise-server standard hardware and software, and supercomputers.

Access control mechanisms in household devices deal with denying access to unauthenticated users from controlling the appliance or stealing data from it, thereby preventing personal data leaks of the users of the appliance. They also run additional programs to assist with the correct functioning of the device they are embedded within, but at a lower privilege level to restrict the amount of tampering a malicious user can do to that system if they gain unauthorized access to it, hence preventing privilege escalation attacks due to faulty or absence of access control mechanisms.

Enterprise-server standard hardware and software generally have multiple layers of access control mechanisms in place, which are harder to penetrate, starting from hardware firewalls to role-assigned users on systems. This allows the enterprise to not only contain the damage in case of a breach but also study it in order to prevent another instance of the same from happening by implementing the required rules into their currently implemented systems.

A single enterprise-server standard hardware or software, if compromised, can lead to massive data leaks, as mentioned in the previous section. However, what people tend to neglect is the data leak possible due to mass exploitation and querying of multiple embedded devices. On a singular scale, it only leaks the data of a few users, but when it is scaled to exploit systems in a larger area, it can lead to massive unnoticeable data leaks as generally, there are no logging systems in place.

Such an event occurred with the Internet of Things devices being added to a botnet network due to a lack of proper implementation of access control mechanisms, in this case, username and password in place. The malware named Mirai added the embedded devices running the default username and password to the botnet and further queried for other embedded devices on the internet using the same credentials to be infected and added to the botnet network. This botnet network was also used for a series of distributed denial-of-service attacks with a bandwidth peak of 623 gigabits per second on a security blog run by journalist Brian Krebs which was mitigated by an Akamai firewall, while most of the attacks after the source code of Mirai was publicly released was under 100 gigabits per second as reported by Chad Seaman for Akamai[6].

Although none of the devices can be neglected based on how small scale their operations are, there are a few prime candidates for profitable improper access control exploits, as mentioned below.

A. Smartphones

Smartphones are one of the most prime targets of any malicious attacker. This is because today's world has shifted from the requirement to use a computer or a laptop to do tasks digitally and currently mostly depend on smartphones to achieve most of the same. This, in turn, results in the production and storage of a lot of potentially personal and sensitive information on the smartphone, which can be accessed by the attacker. A recent attack of this sort in which over 100,000 iPhones were compromised because they simply visited a website was reported by Inc on 30th August 2019[7]. This attack used 14 different security flaws, including zero-day vulnerabilities in the iOS platform, to attack iPhone users. The attack was able to monitor every aspect of the phone and even take control of it, which means there was indeed a privilege escalation vulnerability that could be exploited, thus exploiting a faulty access control mechanism implemented in iOS.

B. Databases

Online databases or server linked databases are a lucrative source of personal and sensitive information. Some of the websites and enterprises follow less than the bare minimum required security rules necessary to keep the data safe, and some others are a victim of well-crafted exploitation ploys by ingenious hackers. The data breaches in the first half of 2019 itself exposed over 4.1 billion records, as reported by Forbes on the 20th August 2019[8]. There have been 3800 publicly disclosed data breaches only in the first six months. The exposed data contained emails in 70% of the breaches and passwords in 65% of the breaches. This was only possible because robust access control mechanisms were not in place to safeguard the data in these organizations, and thus, the attackers ended up exploiting improper access control mechanisms to get access to the data. Since databases are generally present on hosting servers. This might as well highlight an issue with the hosting provider's servers, which were not hardened enough to block or mitigate such attacks.

C. Security Cameras

Anyone who has access to live footage can pretty much have surveillance over an organization, an area, a state, or

even a country. Hence a lot of hackers are inclined to hack the security cameras. There was an incident of criticism against Amazon for promoting security cameras, which can be easily hacked, as an Amazon recommended product. These security cameras sported weak passwords and unencrypted data transfer, which allowed it to be hijacked by cybercriminals rather easily as reported by Independent on 1st October 2019[9].

Another incident that happened recently that highlighted the fact that security cameras are the new target for hackers around the world was dozens of Canon security cameras being hacked in Japan, which was possible because factory default passwords weren't changed. Over 60 cameras were reported to be illegally accessed nationwide, as reported by South China Morning Post on 7th May 2018[10]. This is a classic example of faulty implementation of access control by not setting a password, just like the Mirai malware previously mentioned.

IV. Common Improper Access Control Exploits

There are a lot of attacks related to improper access control; however, some of them have been recently used and cater more to the audience of the current date. A few of these are listed below, along with their fixes.

A. Buffer Overflow Attacks

There are two types of buffer overflow attacks; namely stack buffer overflow and heap buffer overflow attacks. However, both have the same underlying principle. A buffer overflow occurs when a program while writing data to a buffer overruns the buffer's boundary and overwrites adjacent memory locations[11]. A stack buffer overflow occurs in the stack and can be exploited by overwriting the EIP register with the memory location of a piece of malicious or privilege escalation code in a privileged application to gain improper access to the system. Whereas, a heap buffer overflow occurs in the heap and is of three different types, which are, classic heap buffer overflow, double free, and use after free. The heap buffer overflow can be exploited by arbitrary code execution on a privileged application via GOT PLT and __fini_array__ depending on Linux or Windows systems, respectively.

Even though this is quite a common and old exploit, till this date there are multiple applications reported by the Common Vulnerabilities and Exposures website that contain a buffer overflow vulnerability of various kinds such as ImageMagick 7.0.8-35 Q16, multiple Lexmark products, PuTTY versions before 0.71 on Unix and many more[12], which can be used for arbitrary privileged code execution thereby exploiting access control mechanisms incorporated into the application.

Buffer overflow attacks can be avoided using safe libraries and typesafe functions. Multiple operating system based solutions are also available to mitigate or eliminate the effects of buffer overflow, such as data execution prevention, pointer protection and executable space protection[11].

B. Session Hijacking

Session hijacking or cookie hijacking is the exploitation of a valid computer session—sometimes also called a session key—to gain unauthorized access to information or services in a computer system[13]. In particular, it is used to refer to the theft of a magic cookie used to authenticate a user to a remote server[13]. In this manner, an attacker can gain privileges assigned to any other user by impersonating them, thereby bypassing the access control mechanisms present in the application.

Multiple applications are prey to session hijacking attacks, including phpBB version 3.2.7, Afterlogic Aurora 8.3.9-build-a3, IBM Jazz for Service Management 1.1.3, etc., as listed by the National Vulnerability Database website[14].

Although session hijacking attacks can be devastating in nature, they can easily be avoided by using good programming practices including encryption of data traffic passed between parties, correlating the application session with the SSL/TLS credentials, using long random number or string as session key, regenerating session ID after a successful login, etc[13].

C. SQL Injection

An SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution[15]. SQL injections can be used to extract sensitive data present in databases and this information obtained can further be used to bypass access control mechanisms such as login forms or payment methods.

SQL injection, like buffer overflow, is quite a common and old exploit, but the number of applications and websites that still have this vulnerability makes this exploit deserve special mention. There are multiple instances of recent SQL injection vulnerability such as Harmis JE Messenger component 1.2.2 for Joomla!, RockOA 1.8.7, PHPSHE 1.7, etc., as enumerated by the Common Vulnerabilities and Exposures website[16].

SQL injection can be easily avoided by incorporating escaping, input sanitization, using parameterized statements and binding variables to them apart from other sophisticated methods such as pattern checking and database permission setup[15].

D. Remote Code Execution

A remote code execution vulnerability occurs when an application uses user-controlled input without sanitizing it. There are two ways in which remote code execution can be exploited, which are, by executing shell commands or by executing functions in the programming language that the vulnerable application uses or relies on[17]. Sometimes multiple other vulnerabilities lead to a possible remote code execution vulnerability, such as path traversal. In such cases, the primary vulnerability should be fixed first before fixing the remote code execution vulnerability as it might act as a gateway to multiple other possible critical exploits.

Multiple applications have been a victim to remote code execution vulnerabilities, including Quadbase EspressReport ES (ERES) v7.0 update 7, GStreamer before 1.16.0, Caret before 2019-02-22, Jector Smart TV FM-K75 devices, etc., as listed by the Common Vulnerabilities and Exposures website[18].

Remote code execution can be defeated by following a set of well laid out ground rules such as avoiding using user input inside evaluated code, never letting a user edit the content of files that might be parsed by the respective

languages which includes not letting a user decide the name and extensions of files he or she might upload or create in the web application[19].

CONCLUSION

Most of the vulnerabilities mentioned here that are capable of exploiting access control mechanisms are a product of poor coding practices employed by software developers or coding by inexperienced developers. These can also be detected and rectified before the production phase of the software development lifecycle by incorporating a penetration testing subphase to the testing phase and making the necessary changes to secure the application to a higher level. It is also essential to realize that no application is bulletproof irrespective of how many positive security measures it has taken. However, this does not imply that one should not incorporate security features in an application. It only tries to highlight the fact that given enough time, testing and knowledge, an attacker will eventually be able to bypass any specific security mechanism present in an application. This is the sole reason why new exploits and vulnerabilities are discovered on a daily basis. Such discovery only helps strengthen the standards of implementations of security mechanisms employed, after its public disclosure. And to achieve that, one must set up proper logging services to study, and take necessary measures to counteract the exploit.

ACKNOWLEDGMENT

I want to thank the authors of the referred links for providing me with relevant information for me to be able to complete this CA Assignment well within time.

I would also like to thank my lecturer Mr. Vikas Sahni for allotting me this fascinating topic that encompasses multiple domains of cyberattacks. This topic helped get me up to speed with the current affairs in the cybersecurity world.

REFERENCES

[1] E. Flitter and K. Weise, "Capital One Data Breach Compromises Data of Over 100 Million", *Nytimes.com*, 2019. [Online]. Available: https://www.nytimes.com/2019/07/29/business/capital-one-data-breach-hacked.html. [Accessed: 09- Oct- 2019].

[2] M. Humphries, "Hostinger Security Breach Impacts 14M Customers", *PCMAG*, 2019. [Online]. Available: https://www.pcmag.com/news/370387/hostinger-security-breach-impacts-14m-customers. [Accessed: 09- Oct- 2019].

[3] L. Newman, "A 1.3-Tbs DDoS Hit GitHub, the Largest Yet Recorded", *Wired*, 2019. [Online]. Available: https://www.wired.com/story/github-ddos-memcached/. [Accessed: 10- Oct- 2019].

[4] F. Tomesco, "Desjardins: Rogue employee caused data breach for 2.9 million members", *Montreal Gazette*, 2019. [Online]. Available: https://montrealgazette.com/business/desjardins-rogue-employee-caused-data-breach-for-2-9-million-members. [Accessed: 10- Oct- 2019].

[5] C. Cimpanu, "Cryptocurrency startup hacks itself before hacker gets a chance to steal users funds | ZDNet", *ZDNet*, 2019. [Online]. Available: https://www.zdnet.com/article/cryptocurrency-startup-hacks-itself-before-hacker-gets-a-chance-to-steal-users-funds/. [Accessed: 10- Oct- 2019].

[6] C. Seaman, "Threat Advisory: Mirai Botnet | Akamai", Akamai.com, 2019. [Online]. Available: https://www.akamai.com/uk/en/resources/our-thinking/threat-advisories/akamai-mirai-botnet-threat-advisory.jsp?gclid=Cj0KCQjwrfvsBRD7ARIsAKuDvMPYqAyOPeWsPITGWn9PxmIut2LV_xzmtfJWPNReGChzdYtnwomOZvMaAr3iEALw_wcB&ef_id=Cj0KCQjwrfvsBRD7ARIsAKuDvMPYqAyOPeWsPITGWn9PxmIut2LV_xzmtfJWPNReGChzdYtnwomOZvMaAr3iEALw_wcB:G:s&utm_source=google&utm_medium=cpc. [Accessed: 10- Oct- 2019].

[7] T. Koulopoulos, "A Mass Cyberattack on More Than 100,000 iPhones Was Just Reported. Here's How to Protect Yourself", Inc.com, 2019. [Online]. Available: https://www.inc.com/thomas-koulopoulos/a-mass-cyberattack-on-more-that-100000-iphones-just-reported-heres-how-to-protect-yourself.html. [Accessed: 10- Oct- 2019].

[8] D. Winder, "Data Breaches Expose 4.1 Billion Records In First Six Months Of 2019", Forbes.com, 2019. [Online]. Available: https://www.forbes.com/sites/daveywinder/2019/08/20/data-breaches-expose-41-billion-records-in-first-six-months-of-2019/#c875fb0bd549. [Accessed: 10- Oct- 2019].

[9] A. Cuthbertson, "Amazon is promoting 'extremely creepy' security cameras that can be hacked to spy on you", The Independent, 2019. [Online]. Available: https://www.independent.co.uk/life-style/gadgets-and-tech/news/amazon-security-camera-hack-privacy-which-a9127501.html. [Accessed: 10- Oct- 2019].

[10] "Dozens of Canon security cameras hacked in Japan", South China Morning Post, 2019. [Online]. Available: https://www.scmp.com/news/asia/east-asia/article/2144960/dozens-canon-security-cameras-hacked-japan-possibly-because. [Accessed: 10- Oct- 2019].

[11] "Buffer overflow", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Buffer_overflow. [Accessed: 12- Oct- 2019].

[12] "CVE -Search Results", Cve.mitre.org, 2019. [Online]. Available: http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Buffer+Overflow. [Accessed: 12- Oct- 2019].

[13] "Session hijacking", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Session_hijacking. [Accessed: 12- Oct- 2019].

[14] "NVD - Results", Nvd.nist.gov, 2019. [Online]. Available: https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=Session+Hijacking&search_type=all. [Accessed: 12- Oct- 2019].

[15] "SQL injection", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/SQL_injection. [Accessed: 12- Oct- 2019].

[16] "CVE -Search Results", Cve.mitre.org, 2019. [Online]. Available: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=sql+injection. [Accessed: 12- Oct- 2019].

[17] P. Yaworski, Real-World Web Hacking : A Field Guide to Web Hacking. No Starch Press, Incorporated, 2019, p. 119.

[18] "CVE -Search Results", Cve.mitre.org, 2019. [Online]. Available: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Remote+Code+Execution. [Accessed: 12- Oct- 2019].

[19] "Remote Code Evaluation (Execution) Vulnerability", Netsparker.com, 2019. [Online]. Available: https://www.netsparker.com/blog/web-security/remote-code-evaluation-execution/. [Accessed: 12- Oct- 2019].