Authentication of IoT Devices using 2-Factor Authentication and Virtual Keyboards

Andrew Lucero[#1], Joeseph Kracz[#2], Snehal Kenjale[#3], Wenhe Li[#4], Zami Talukder[#5]

***Abstract*- Internet of Things (IoT)[1] devices contain personalized information by nature. Devices like Smart Webcam, sweeping robots, and home assistants are quite likely to collect and discourse private information. Thus, in such cases, the authentication will be much more important. It is crucial for an IoT device to identify who is sending a command, monitoring the device and has the correct access to the device. Moreover, IoT devices have direct access to the Internet and often use a weak password or even no password for authentication. Thus, hackers have a chance to gain substantial control of the devices by simply using Telnet or SSH. Under such circumstances, guaranteeing the authentication of specific users under different use cases is a huge concern. The goal is to use better authentication mechanisms to so as to provide more security to the devices. This paper provides a security mechanism using virtual keyboards.**

***Keywords*- Internet of Things, Virtual keyboard, Middleware, Two-Factor Authentication, Keylogger**

I. INTRODUCTION

The Internet of Things (IoT) refers to the network of devices that exist which can connect, collect and transfer data in real time over the Internet. Applications include smart home devices, smart city devices, wearables, environment monitors, medical devices, etc. Often, these IoT devices can be remotely monitored and operated. Authentication of these devices can be done using username and password credentials, OTP, access tokens or with the help of digital certificates and protocols like MQTT[5]. An OAuth-based framework can be used to authorize the access.

By nature Internet of Things (IoT) devices also contain personal information. Devices like smart webcams, cleaning robots, and home assistants are all likely gathering personal information on their users. Information stored can range from what the user looks like, to their credit card info or address. In such cases, strong authentication for access to those devices is incredibly important. It is crucial for an IoT device to identify who is sending a command, monitoring the device or accessing the device. Moreover, IoT devices will almost always have direct access to the Internet and will often use a weak default password or even no password for authentication to maximize ease of access for users. This means that often times, attackers looking to compromise IoT devices that, for example, have Telnet capabilities will be able to do so by simply using common default credentials which can offer the attacker full control of the device.

However, the problem that we will be discussing the solution to in this study, is the problem of attackers gathering credentials through malware (primarily keyloggers). To combat this problem, we propose using two-factor authentication (2FA), where the pin/pass for that step is inputted using a virtual keyboard. The fact that we are using 2FA means that a compromised set of credentials is not as devastating as it should be, and the virtual keyboard allows for protection against keylogging malware.

II. Background

A common vulnerability present in Authentication of IoT devices is the use of Telnet which is a fairly outdated login protocol that allows for users to remotely access devices for which they have the credentials.
The vulnerability lies in the fact that often, these

credentials have defaults, that the everyday user might not know or care about. This means that there are many devices in the "wild", that can be remotely accessed after doing a quick search for the credentials. This problem was exacerbated and made more public, when a list of thousands of such credentials was left public on the web.

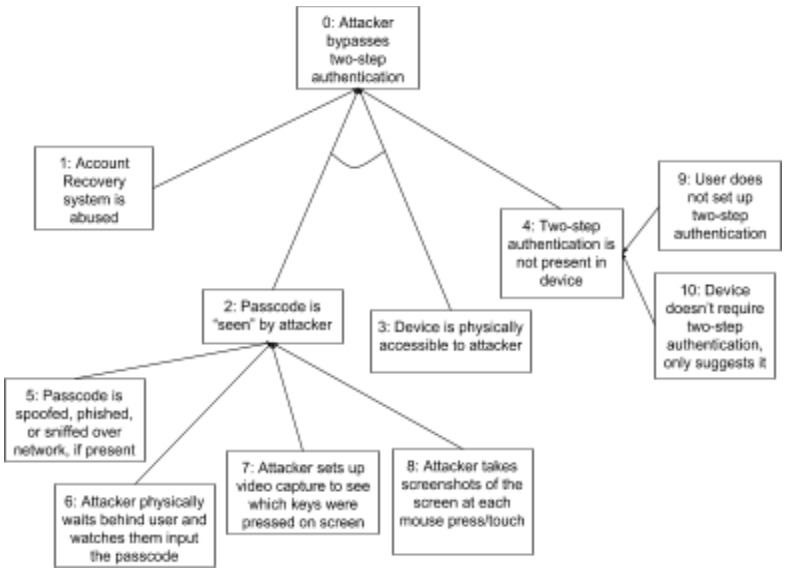
The reason that this problem exists, is that it was expected that those with the password, are those that should be allowed to access the device. however, as we are moving forward, it is becoming more and more clear that there needs to be better authentication, since it is becoming easier and easier for attackers to obtain these credentials; especially for devices that are outdated but still in operation.

Two common tools used to steal these credentials, are malware and phishing[3]. Phishing refers to the attacks that attackers will employ on potential victims. To explain simply, phishing is an attack in which attackers try to get victims to reveal private information. A common attack would be in the form of an email. The email would contain a malicious link that masquerades as a legitimate websites registration page, and the email would read something like “Hello, make sure to update your password for this site you might use”. Of course, there are many users that are able to identify these phishing attempts and can protect against them by simply ignoring the forms. However, this is why it is often said that phishing is a numbers game. The more people you can attack, the more likely it is that you will find someone uninformed enough to get information from.

Malware can also be used to steal credentials. One of the more common types of malware, are keyloggers. Though a keylogger is not always malicious, they can be used in such a manner. Usually, a keylogger in the form of malware, is software that records all the keystrokes made by the infected user. The goal is operate in the background, hidden from the user, so that the attacker can get as much information as possible. This can lead to stolen credentials, but can also lead to stolen credit cards, bank accounts, and more.

We attempt to tackle both of these problems with our solution. The 2FA portion of the solution handles the phishing side, as credentials alone are not enough to become authenticated, and the virtual keyboard helps to protect against forms of malware, like keyloggers, which can’t log keystrokes accurately if the keyboard is virtual, and even so, since the keyboard layout changes every time, recording clicks alone is not enough to lift the pass used for the 2FA.

III. THREAT MODEL



The threat model shows the risks that are faced when using a virtual keyboard as a form of two step authentication.

Node 1 is a problem that’s faced for all

authentication purposes. The information required to recover an account/user and determining if the information is coming from the correct user is a separate issue. This is because recovery happens less often and ease of use is less relevant for this.

Node 2 presents us with the fact that the inputs can still be seen by others. This threat is always present for all user-generated fixed passwords. Virtual keyboards help to lessen the children of this node by making attacks with keyloggers, hand-movement, etc. ineffective. However, the node still has some other children present. Node 3 works together with node 2 because the process requires inputs by a user to the operating system.

Node 4 shows us a design flaw that appears in many applications where security is optional and users ignore it.

IV. DESIGN

We proposed a two-factor authentication with randomized virtual keyboards solution with the following security properties:

- Improved Security
- Essential authentication
- Middleware Security
- Enhanced input security

Our system includes three major layers as follows:

A. Virtual Keyboards:

The virtual keyboards will act as a middleware[2] between the user interface and the system keyboard. Whenever any input is made by the user, the information flow will be processed by the middleware then cast to the system input. With this middleware, we can ensure the keylogger like tools won't get the real key.

B. Independent Validation Unit:

After the key is processed, an independent validation unit will validate the input. If it passes, the unit will sign a session with a system generated UUID to ensure the uniqueness.

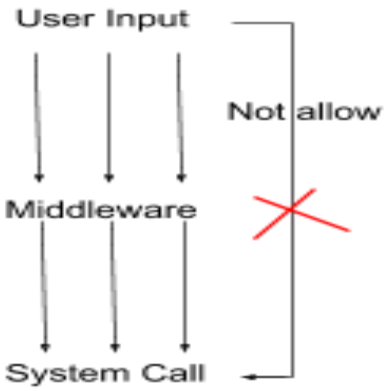
C. Runtime Validation Unit:

This unit will check if the session is valid based on the UUID and the session's timestamp when a core function is called.

Below is the overall architecture, the user's input and function call will all be processed by the self-designed unit to ensure the authentication security.

V. IMPLEMENTATION

We introduced the concept of middleware in the previous section to be the core part of our solution. Here we will explain the detail of our design. First, we will develop the middleware part that will receive the input and coordinate with the system I/O. And we will not allow any direct operations onto the system that means only the middleware has the right to perform operations. So for this part, we will use a strict access control list to perform the restriction policy.



(Middleware design)

Another part of our solution is about the session assignment and validation to ensure it is the right user to perform a certain operation. Here on this part, we will use a two-factor authentication alongside with a signed session carrying a timestamp and UUID to guarantee it is validated. In practice, we will use a virtual keyboard to protect the user's original input from being monitored and get the two-factor authentication.

Finally, our top-level (User Interface) implementation will ask the user to set a pin code first and the pin code will be stored encrypted locally. After the setting, every time, the user wants to perform any operation, the user needs to input the pin code on a randomized virtual keyboard.

VI. EVALUATION

Most modern authentication systems only require a single sign in or pin before granting access to the user. An attacker could watch a user enter their credentials or steal sensor data from a smartwatch to help them estimate a pin number. The attacker would then be able to gain unauthorized access to the system using someone else's credentials.

The design and implementation for a randomized virtual keyboard and two-factor authentication system demonstrate a considerable improvement upon a majority of authentication systems in use today. When an IoT device doesn't have a screen to show the virtual keyboard, an application can be used on the users phone to enable two step authentication. Using two factors instead of one to authenticate would make it considerably more difficult overall to gain unauthorized access to a system. Additionally, attackers would have a harder time gleaning accurate information from watching someone enter their credentials if the keypad changes after every input. Any tracked user hand motion would be useless as well because the data would not correlate to static key locations as well.

It is important to note that some difficulties arise when implementing the suggested system. Adding an additional layer of security requires users to remember another credential and enter it. This becomes even more of a pain when the system does not utilize a touchscreen, meaning the user must enter their pin via a series of mouse clicks. The randomized keyboard also has the potential to throw off users who rely on muscle memory to remember their pins. Additionally, accessibility becomes an issue when integrating the virtual keyboard with public devices such as ATMs. It is not currently possible to randomize a keyboard and have it display braille for the visually impaired, which means they could not benefit from the added security a virtual keyboard provides.

VII. RELATED SOLUTIONS

A. Using colour coded virtual keyboard and hiding keys

In paper [4], the proposed technique is a virtual keyboard with hidden key feature and colour coding scheme. The keys are shuffled after every click to prevent spoofing of the password by any person standing in the vicinity of the device. In addition to this, there is a hide key feature introduced to hide the keys before clicking so that if there is a spyware introduced to monitor keys on clicks, then it would not be able to capture and detect the symbol. In addition to this, a colour scheme is assigned to every symbol on the keyboard, so that the user finds it easier to differentiate the keys when they are hidden. Also, this method can be introduced with current technologies. There is

surely a drawback of proposed method that it consumes a lot of time but ensures that the credentials are well protected.

B. Anti-Screenshot Virtual Keyboard

In paper[6], the anti-screenshot virtual keyboard is proposed in order to overcome flaws in virtual keyboards such as shoulder surfing or vulnerability in screenshot capture. In the anti-screenshot virtual keyboard, when the mouse pointer moves to a certain row, all the keys in that row are changed to a symbol like an asterix. After user clicks, the keys are restored to their original values. Additionally, the keys can also be randomized. This proposed model is capable of solving screen captures by Trojans, and more secure than traditional keyboards.

VIII. CONCLUSION

In this paper, we discuss a solution to enhance the authentication system using two-factor authentication and virtual keyboards.

Our solution develops a scheme that is using middleware to perform the interaction onto the system and process the user input with authentication. Our solution utilizing the concept of middleware makes it easier to be applied to any OS. In addition, this solution is extensible. If any functionality or a more strict security policy wants to apply, we just need to make a new middleware and plug it between the original middleware and system layer.

REFERENCES

1. Wikipedia - Internet of Things https://en.wikipedia.org/wiki/Internet_of_things
2. Middleware https://en.wikipedia.org/wiki/Middleware
3. Phishing https://www.incapsula.com/web-application-security/phishing-attack-scam.html
4. Secure authentication using dynamic virtual keyboard layout https://www.researchgate.net/publication/220027075_Secure_authentication_using_dynamic_virtual_keyboard_layout
5. Authentication protocol - MQTT https://en.wikipedia.org/wiki/MQTT
6. Secure Authentication using Anti-Screenshot Virtual Keyboard http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.402.8503