Defending against USB Rubber Ducky Payloads

By Hao Cheong

Project Summary

USB Rubber Ducky's are a electronic device which when connected to a user's system, will inject keystrokes into the users systems usually at imperceptible speeds and doing harm to the users system. The destructiveness of a payload varies depending on the payload. If it can be done using just a keyboard, it can be done using a USB Rubber Ducky.

For my Something Awesome Project, I have designed and written several different payloads in an each with varying levels of capabilities. During this time, I have also discovered a few ways my payloads could be defended against, the reasons they break, and the changes, or at least proposed changes which can be used to defend against them. All payloads are written in a variation of C/C++ for Arduino boards specifically.

I will be only covering payloads which I have written but many of the learnings can be used in potentially other rubber ducky payloads. I will cover there purpose very generally, how they can be detrimental to a user system and how they can be defended against both in practical terms and theoretical terms where I am able.

Payloads

The payloads which I have written have designed were designed in order of complexity and each was to achieve to only do one goal. Payloads were designed in increasing levels of complexity. Early payloads were to be able to control the user's system locally and affect the data within the system. Later payloads then focused on more complicated features such as forcing a user to download and uploads files to the internet.

An assumption I made for the payloads I write is that they will work on any Window's system without the need to have previously downloaded any software excluding Mozilla Firefox browser.

Writer

Description

This payload purpose was to be able to write a file into the user system and being allowed the decide the content of the file as well as the decide the location where the file is to be written to. This was to show the ability to inject data into system which could later be used to inject malicious code or save downloaded malicious data.

Defense Techniques

The defense against this payload is relatively easy given the limitations of PowerShell. PowerShell commands are used primarily within the payload, but the problem is that the commands are all case sensitive. By enabling the Caps Lock onto the system will very easily stop the execution of the commands as they would not be recognized.

While there are commands in PowerShell to determine whether caps lock was on, there was no method I could find to parse the value into the Arduino board to disable the system.
Accessing the PowerShell and the "Run" feature on windows is the main reason the majority of how the payloads work. PowerShell is by default installed onto a windows device so by deleting or somehow disabling PowerShell would also be viable defense against this payload

Deleter

Description

This payload works opposite to the "writer" payload. Given a known location of a user's file, the payload will delete the file permanently, not simply in the recycling bin but permanently. This was to show the ability to remove data from the users system which could be used to sabotage a user's content or even potentially remove a critical file which may result in the users system being unusable.

Defense Techniques

The defense against this attack is the same for writer because of its use of PowerShell. If the defender's system has PowerShell, caps lock enabling would render the injection useless else disabling PowerShell all together.

Shutdowner

Description

This payload is a quite simple. Once plugged in, the user’s system will shut down, if left in the user’s system during turning on, it well shutdown the user system again. This was to explore and find rudimentary control over the user’s system.

Defense Technique

For how simple this payload was, this was a very difficult payload to find a defense against because it did not use any other application besides shortcuts provided by Windows. A registry edit on a window system would need to be done to disable the keyboard shortcut.

Rickroller

Description

This payload is also quite simple. Given a website link (in this case, the Rick Roll video on youtube, hence the name), the user is forced to access the website using the user’s default browser. The purpose was to demonstrate the ability to force users to access a website which forces them to a potentially malicious website.

Defense Technique

The defense is obvious, disable internet connection on the system. Without it, the website redirection would not happen. The Window’s run feature is used prominently in website redirection and a permanent disabling of the feature would require a registry edit. However, because of how much run is used in a lot of other things which may beneficial to the user, by disabling the run feature, the user would be quite limited in their control over their directory traversal

Wallpaper Setter

Description

This payload uses a combination of Rick Roller and the writer. Given a link to an image online, the image will be forcefully set as the user’s wallpaper. The purpose was to demonstrate the ability to force a user to download files off the internet and have the files affect the user’s system in some way.

Defense Technique

Since this payload uses methods from two previously mentioned payloads, the methods of defense will work for this payload. Disabling the internet connection, PowerShell and the Window Run feature would all protect against this payload. One application which was used to set the wallpaper was Window’s Paint program as it provided a simple shortcut to setting an image as a wallpaper, so by deleting the application, this will stop the payload.

Firefox Safety Disabler

Description

The payload disables security features present within the current version of Firefox. When used, the payload will enter the Firefox and begin disabling security measures. The purpose was to be able to disable user security system which in turn could be used to the attacker's advantage if they decide to force the user to download malicious files from the internet.

Defense Techniques

Previous methods as mentioned can be used to defends against this payload such as disabling the internet connection and PowerShell. By nature of this payload, it may very be that the payload would eventually stop working. The payload "traverses" the Firefox security page and disables all the security feature but it is running on the assumption that the security page would be identical on all system. If Firefox decided to update their security page layout, the payload would not be able to operate at all. Keystroke injection can be thought as almost an attacker trying to type malicious command into the defender's system blindfolded. If the attack has no prior knowledge on how the page layout would be, they would have no method of traversing the page.

Password Sender

Description

This payload was the most complicated of all the previous payloads and could be thought of the culmination of all the lessons of the previous payload. The payloads enter the users Firefox login page and proceed to screenshot the details of their login and passwords. The screenshot is saved in the system and then uploaded to an attacker's Gmail account.

Defense Technique

Since it is a combination of a lot of previous payloads, the defense against this system is very easy. As long as the user had previously: disabled their internet, disabled or uninstalled PowerShell, disabled window run feature, or turn on caps lock, the whole payload would fail. As payloads increase in complexity, the points of failure also increases.

Conclusion

Overall, there are many different methods which payloads can be defended against, whether that is simply turning on Caps Lock or disabling the internet. Payloads also themselves have an expiration date as many of them are depended on the certain programs to operate as according the when the payload was built. If changes to the programs were made by the original developer (I.E Altering the page layout of the security page on Firefox), the application will fail. However, since many of the defense against the payloads require disabling of a lot of important features which the user will need such as internet connection, it would practically render their system relatively useless.

If a suspected USB rubber ducky device was presented, the best course of action is to test the device by plugging it into a system with said features disabled. Even so, there are many ways a payload can be written and attackers are always finding new way of working around defenses so while payloads may expire due to new defense techniques, the techniques themselves would also expire.