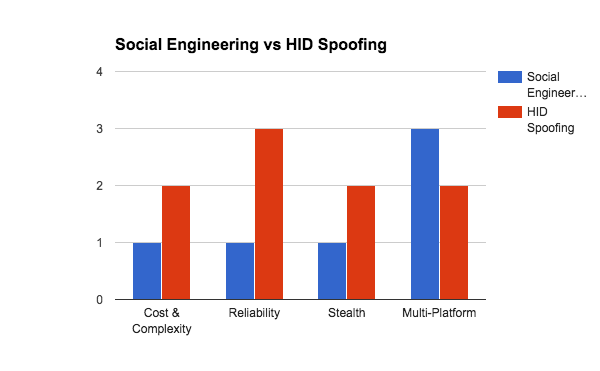
Max Conrad

SIP

As technology becomes more interconnected and devices such as wine coolers are becoming connected to the internet, security must be a main focus. Many people are aware they need to be careful with things they download on the internet. However, there are many people who do not know they can be hacked simply by plugging in a usb drive. This type of attack is not quite hard to make, but very easy to pull off. All a hacker has to do is drop a usb drive in a public place, and then the victim picks it up and plugs it into their computer. The purpose of this project is to raise awareness about this type of attack and cyber security. According to Identity Theft Center, in 2015 there were 781 data breaches that were known.[[1]](#footnote-0) This number is only going to go up as more organizations collect data about their customers, and hacks become harder to find.

There are two different types of attacks this paper will cover that can be carried out via a usb. Social engineering attacks and HID spoofing attacks can both be pulled off with a single usb drive. Social engineering can encompass many different types of attacks. In this case, the usb drive can contain a file that might interest the user. If the user clicked and opened the file, malware could be secretly loaded onto the user’s computer. A normal user would probably open a file called “survey.html” without giving it another thought. In HID spoofing attacks, the usb drive is disguised as an emulated keyboard and then can unload its payload onto the victim's computer as soon as it is plugged in. This attack is hard to pull off with windows 7 and up. In 2009 in order to prevent the spread of malware, Microsoft disabled by default the autorun feature on usb drives. However, they did not completely disable it, but rather gave the user the option to turn it back on. Microsoft created an entire howto if one wanted to permanently disable autorun on their website.[[2]](#footnote-1) To combat this, hackers started making HID usb devices that spoofed themselves in order to autorun a program as soon as the usb is plugged in. These devices are hard to make, since the exploit has only been released for devices with a Phison micro-controller. There are guides that step by step show how to execute and make this exploit.[[3]](#footnote-2) For hackers or people that do not want to go through the trouble of making their own, a device called rubber ducky usb costs only $40 and it is all set up for the buyer.[[4]](#footnote-3) More importantly it works on all operating systems. Depending on the person, one attack or the other might benefit them more. The following graphic shows strength and weaknesses between the two.



Social engineering attacks are not very reliable and can be discovered very quickly, however the benefit is that it is cheap to do and depending on the setup, it can reach many different operating systems. This allows the hacker to make a lot of social engineering drives at a low cost in order to counter its low success rate. HID spoofing on the other hand, comes at a higher cost, but ensures the user that it is a lot more reliable to infect victim’s computers.

As a Christian living in today’s world, we are faced with many challenges every day. In the cyber security field it is even harder. Depending on the job, the task at hand could require hacking into a user or organization’s computer or defending against a cyber attack. This is where as a Christian, one must hold their morals close and be able to decide what is right or wrong and be able to stand up for what they believe in. The purpose of this project is to bring awareness of a form of hacking that one might encounter. We Christians live in a fallen world and we cannot expect that everyone will abide by the rules. Instead we must be aware and educated in cyber security. To this date, there are countless data breaches and hacks that have been and are going on.[[5]](#footnote-4) My goal as a Christian is to not stand by and let this happen. There must be a resistance against enemy cyber agents or hackers so that we do not lose our personal information or become their bots. The first step is to bring awareness not just to Christians but to everyone. This will solve many problems, but not the main issue. To fully defend against hacks, one must understand how viruses work and understand all the different hacks going on. Even though this borders the line between right and wrong, I believe that it is what is done with the program that determines whether it is used for the right purpose or not. The program I created is malicious, however the purpose of it is to bring awareness to the underlying problem that people don’t know usb drives can be malicious. Being a Christian partly means spreading God’s Word. I believe that spreading the knowledge of something bad and potentially harmful is my duty as a Christian as well. Therefore in part to accomplish this, I decided to do this project. One might argue that as Christians we should not have anything to hide. Therefore we should be fine with our personal information leaked or stolen. In all cases this is wrong. In Genesis, God gave Adam and Eve clothes to wear to cover up their bodies. Such is the same with our data. It is personal information that should only be shown and accessible to the owner. Finally, as Christians we are taught to lead by example and people will see Christ within us. Christians in the cyber security field that are going against Christian ethics, should stand up for what they believe in and lead by example. Of course this can lead to potentially losing one's’ job, and then the question is would you stand up for what you believe in, or go against one’s Christian beliefs. Cyber security is definitely a hard field to be a Christian in. One could be asked to hack an opposing organization, or asked to defend against an attack. This also is not limited to cyber warfare, but to creating ethical programs that are secure and do not secretly spy on the consumer. It is my role as a Christian in this upcoming technology age to lead by example and raise awareness so that hackers are not running rampant everywhere.

The goal of this project is to educate and bring awareness to people on cyber security. While some may not think that cyber security is very important or does not pertain to them, this project intends to show that something could happen to them if they are not very careful. By selecting students primarily as the target audience, the idea to drop usb drives around campus and other student areas sounded like a good idea. Hopefully by educating students about cyber security early on, they will remember it as this world becomes more and more technology oriented with many different methods for hackers to use. Starting October, I started the first two phases of my project by sitting in different academic buildings around campus gathering the data I needed. Once I had enough data, I created the example program.

In order to defend against usb attacks, the user or victim must be educated or trained on cyber security first. By having little to no background about cyber security the victim probably do things resulting in a compromise of their system. In this case, most people do not think that a usb drive could contain a malicious virus that is hidden away. They instead think that all virus’ come from the internet or magically appear on their system. Secondly depending on the user and the system, it might be beneficial to physically block all usb ports.[[6]](#footnote-5) Any exposed usb port in any system is vulnerable to a usb attack like this. All it takes is one person plugging in a usb that is infected and the system is compromised. A tool called GoodUSB aims to counter HID attacks like this. This is done “by enforcing permissions based on user expectations of device functionality.” [[7]](#footnote-6) By creating a system of permissions to allow only certain usb drives to work, malicious usb drives can be profiled out. Lastly in order to figure out how exploits like this work, the exploit has to be reversed engineered to see exactly what it is infecting. Only then can professionals create countermeasures.

For the project portion of this SIP, I chose to show how this exploit can work in order to bring awareness to this exploit and cyber security in general. This project is composed of three parts. 1. Observing how many people leave their computer unattended. 2. Surveying how many people would plug in a usb drive they found. 3. Create a small program to demonstrate one of the very many things that could happen if a user plugged in a usb drive. For the first and second parts I sat in different academic buildings for a period of two hours across a few weeks. As well as a Starbucks in Alpharetta, Georgia.

In an academic setting, students typically feel safe around their peers, or in academic buildings. In a setting such as Covenant College, students typically have a lot of trust and tend to do things security experts might question. The main thing is leaving their computer unattended and logged in. This is a huge security risk to the victim, and a easy target for a hacker. While observing students that were using their laptop, I found that roughly 65% left their laptop at some point and approximately 30% of those left their laptop open. Besides leaving their laptop unattended, for anyone to take, the more shocking result is that 30% of students that I observed left their laptop open and available for potential malicious activity. Keeping to this project, regardless if the usb a hacker had, contained a social engineering attack or an HID usb attack they could have plugged it into the computer and had a 100% success rate in infecting their victim. Thus increasing the amount of possible people they could infect.

How effective is dropping usb drives in hopes that an unsuspecting person would put it into their computer? In order to provide an answer to this question, I dropped a usb drive 50 times around Covenant College and waited to see if someone picked it up. In all cases I waited two hours then concluded that session. Out of 50 drops, a student picked up the drive 42 of those times. Once picked up, the student was asked the following questions. 1. Would you have plugged the usb into your computer to see what was on it? Or to see whose it was? The following questions are assuming the student said they would plug it in. 2. If there was a file on the drive, would you have opened it? 3. If there was nothing on it, would you have kept it? The results varied student to student with multiple answers for question 2. In total, 19 students said that they would have plugged it in equaling to around 45%. While 10 students, around 23%, claimed they would have opened a file on it if there was one. These are alarmingly high numbers. Any hacker would be happy to do very little work in order to spread his or her virus. In order to produce more accurate results, I sat in Starbucks for a period of an hour over the course of a few days. I surveyed 20 people there and the results were pretty similar. With 35% of people saying they would have plugged the usb drive in, and 57% saying they would have opened a file if there was one. I was able to roughly get around 50% students and 50% adults to participate in my survey. The adults gave more no’s to the question about plugging in the drive. Overall these numbers add up to, 37% would have plugged in the usb and 20% opening a file. With more accurate numbers, it can be seen that a simple method of dropping malicious usb drives around public areas works.

For the final part I created a small Java program that shows something that could happen if a victim plugged in the usb drive. This is assuming the target computer has a version of java installed. Potentially more dangerous and malicious programs could be created if it was created in c or c++. This is partly because not everyone has java. Secondly java runs in it’s own environment so it does not have access to the whole computer. For example it is extremely difficult/impossible to create a keylogger in java because of the way it runs. However creating the program in java means it could be run on any machine that has java installed, regardless of the operating system. Since it is harder building malicious programs for mac, I targeted Windows machines for this program. For this program, I coded a few examples. The first, a virus that freezes mouse movement. Secondly a virus that deletes all the victims files. Thirdly and finally, a virus that targets the registry in Windows and makes the program run at startup. These hacks could be expanded to sending the user’s files to the hacker. Or turning on the webcam and microphone of the victim’s computer. More malicious programs could freeze up the entire computer and encrypt the user’s files and demand a ransom.

Since creating a usb that acted like a HID device was not the main focus of the project, social engineering was the better option. To disguise the program, I created a html file called survey.html in hopes that a potential victim might click on it. Then I put the jar file in the same place as the html file on the usb drive. After that, I created a shortcut to the jar file and called it survey. I made both the actual html file and the jar file hidden files so that only the shortcut was available to see. Now to the average user they only see one file that does not look in any way malicious. As soon as the shortcut is clicked, the program executes and the user becomes a victim.

I’ve learned many things from this project, how viruses are made, how hard some are to create, and statistics about how many people are not aware of malicious usbs. Researching different kinds of hacks I could pull off with a BadUSB showed that there was hard work involved with creating such devices. Initially picking Java as the language to make the program turned out it was limited in what it could do. Creating the program in c or c++ would have yielded in far more dangerous kinds of hacks. One of the hardest parts is figuring out what operating system the computer is running when the usb is plugged in. The program needed to be able to detect the operating system and then execute the payload all before the victim pulls out the usb. Since the BadUSB exploit requires that the usb be faked as a HID keyboard, I learned that it can only execute so many keystrokes a second. By doing this, the program can then open up an invisible shell on the computer and connect it back to the hacker’s server allowing him to proceed with malicious attacks.

Dropping usb drives in public places actually does work. It does not take a large amount of victims to get the virus started. Once infected the virus could spread to other computers by infecting other flash drives that the victim plugs in or through emails or other methods. Creating usb drives like this however is hard to do. If infected by something like this, the hacker most likely knew what they were doing and dedicated time to it. Simply having an anti-virus will not save a user from this attack. Instead, awareness about the subject does. Being aware of the different ways one could be hacked could save hours of troubleshooting or maybe thousands of dollars in the future. Potentially more dangerous attacks that can be done via usb are zero day exploits. These are less common and only used by organizations that discover them. With technology advancing as fast as it is, it is important to secure our devices and our applications. An even higher priority is raising awareness about cyber security and hacks that could happen to the everyday user. In the 4 years I have spent at Covenant I realized there is a lot that needs to be done. Most people are ignorant towards technology. This can be seen everywhere in every business. By creating this project I hope to bring awareness to one aspect of cyber security by addressing an issue that can potentially affect everyone.

1. "2015 Data Breaches | Data Breaches." ID Theft Resource Center 888-400-5530. n.d. http://www.idtheftcenter.org/ITRC-Surveys-Studies/2015databreaches.html. [↑](#footnote-ref-0)
2. Support.microsoft.com. <https://support.microsoft.com/en-us/kb/967715>. [↑](#footnote-ref-1)
3. How To, Make Your Own Bad USB. <http://null-byte.wonderhowto.com/how-to/make-your-own-bad-usb-0165419/> [↑](#footnote-ref-2)
4. HakShop, USB Rubber Ducky Deluxe. <https://hakshop.com/products/usb-rubber-ducky-deluxe>. [↑](#footnote-ref-3)
5. IdentityForce. "2016 Data Breaches." IdentityForce. N.p., 29 Nov. 2016. Web. <https://www.identityforce.com/blog/2016-data-breaches>. [↑](#footnote-ref-4)
6. Andra Zaharia. "How to fix the Critical BadUSB Security Flaw in Less than 10 Minutes - Heimdal Security Blog." Heimdal Security Blog. 2016. https://heimdalsecurity.com/blog/badusb-exploit-vulnerability-fix/. [↑](#footnote-ref-5)
7. Tian, Dave, Adam Bates, and Kevin Butler. Defending Against Malicious USB Firmware with GoodUSB. N.p., n.d. Web. <http://cise.ufl.edu/~butler/pubs/acsac15.pdf>. [↑](#footnote-ref-6)