CS 1122 Report

Team Members:
Jin Yan Ruan
Hui Wah Chiang
Zami Talukder
Alfred S Haber

Introduction

Keylogger

For the keylogger, we made a tool for Windows that would be able to log keystrokes to a txt file. We then took that txt file and sent it over a network to a remote server.

Network Scanner

For the network scanner, we made a tool that would scan for other computers on the same subnet. It would then scan the given network to find open ports, then compare those ports to a dictionary of common ports to predict what was being used.

File Recovery

The file recovery tool created is used to recover .png files and .pdf files that have already been deleted.

Background

Keylogger

Keylogging is when we log a person's keystrokes by intercepting the information passed when a key is pressed on a keyboard. In this situation, we want to save that information to a file, then have that file sent over a network. The benefit of this is that if we get a keylogger on a computer remotely, we can use it to monitor a person's keystrokes, and identify things such as logins and passwords. We can then use that and send it to a remote location, so we don't have to physically access that person's computer to gain access to the information.

Network Scanner

The network scanner needed to be able to ping other computers on the same subnet, and then find open TCP ports on those computers. Because each port has a typical common usage, we can then map those ports to a dictionary and predict what that is currently being used for. A subnet contains a list of ips, and each ip may have several ports open, so we needed to scan the subnet, then loop through the ips and scan for open ports.

File Recovery

File recovery tools can be used to recover something previously deleted by the user. The user may have accidently deleted the file and would want to have control of the data in the file once more. Alternatively, someone may delete files to hide information and cyber security experts may use such a tool to get the information, given that it was not yet overwritten. When a file is deleted, the information of the file is not removed, but instead thought of as removed. The space the file takes up is never recovered, it is just used by something else when something new is created. Because of this, the written portion can be read and the pointer to the memory location can be recreated if the space was not taken up by something new.

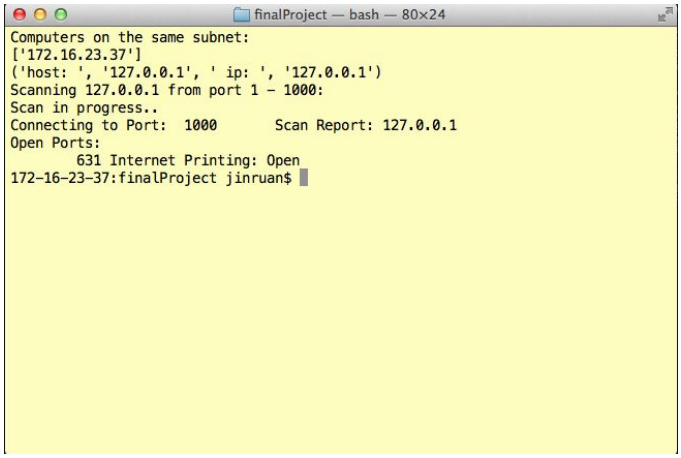
Implementation

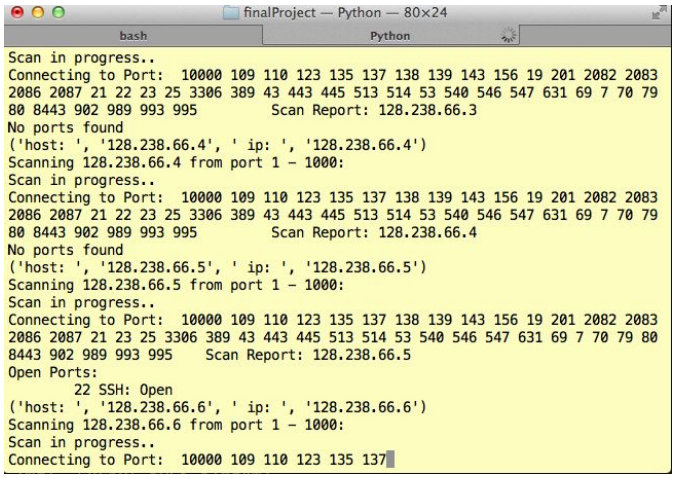
Keylogger

The code for the keylogger program was divided into separate parts; first, a text file called “keylog.txt” was created onto the user’s desktop. Each character that the user typed was then saved to the text file as a string, and that string was then sent to the remote server.

Network Scanner

First, our program needed to import the needed modules, such as socket, which was what was needed to create a socket connection. Looped through the given subnet and saved each ip to an array. Then, we did the following for each ip in the array: We then get the host name to scan for the open ports. We then convert the hostname into an ip address. If the startport and given endports to scan for are defined, we can then start to scan for open ports. Next, we have a dictionary of the common ports, which we will then map to, so that we can predict what does what. Then, we have a function to connect to a port to check if it is open or closed. We set a timeout for the socket, so that it won’t wait forever to connect. If the connection to the socket is successful, we return that result and close the socket. We then have a function that takes in the port as a parameter and returns the corresponding prediction in the dictionary. We then loop through all of the ports and attempt to connect to each one, and if the port is open, we append it to a list of open ports. Finally, we loop through the list of ports that are open and then display it. We also have a case statement for if the purpose of the port was not found.





File Recovery

For this project, the file recovery tool only finds .pdfs and .png files because the file structure of these files have both headers and footers. This means the bytes can tell us when the pdf file begins and when the pdf file ends. If the bytes in between the footers and headers are taken and written to a file, the created file will be an exact replica of what was deleted. The code is broken up in two portions: one for pdfs and one for pngs. They both essentially run the same way, but have different values stored for headers and footers. A loop runs through each bit of the larger file provided. Each bit is compared to the default header. Once the header is found, each bit is stored in a list and the footer is searched for instead. Once the full footer is found, the list is used to write to a file using the bytes. There were multiple footers for pdf files, as a result, each bit is compared to each footer for pdfs.

Future Work

Keylogger

Future work includes making it easier to spread the keylogger in more conspicuous ways so that it will be easier for the user to gain more keystrokes. One method would be to find a way to attach the keylogger to a file and then when the person downloads and opens that file, the keylogger will automatically run in the background. Another way would be to find a way to remotely have the keylogger activate the moment the person opens up his or her PC, so that we can get as many keylogs as possible, not just for that one time.

Network Scanner

Some small features that could be added to the tool would be to allow for user inputs, so that they can specify what ip they want to scan. Another could be to allow for passing in a single port to see if that port was open. Perhaps another would be to pass in multiple ip's to scan and return the results for all of them in a txt file. Additionally, it would be nice to find a way to scan the ports more quickly, since it has to do it for one at a time. We later modified it to only scan for the ports included in our common dictionary to speed things up.

File Recovery

The simplest way to improve the file recovery tool is to make it compatible for more file types. Since it currently only works for pdfs and pngs, it has limited use. If it could function for more varied file types and more common ones, such as .docx or jpegs, it would also prove to be more useful. An additional improvement would be to add more code so multiple headers and footers in the file do not distort the outputted file.

Results

Keylogger

The results that came out of running the program were that we were able to log a person's keystrokes to a text file on the system.

For the second part of the program, we were able to successfully send the text file over the internet to the remote server. The file containing the keystrokes were saved onto the computer, and then each keystroke was sent to a remote server as the keystroke was recieved.

Network Scanner

Our network scanner was able to work for the individual local ip that we tested it on. It was able to scan and detect that port 631 was open, which was then displayed as internet printing. We

were also able to test it on the keylogger ip and it detected that ssh and http were open. We then modified the program to scan through the subnet and scan each individual ip. The only caveat is that this is slow, and because it needs to scan 1000 ports, it would take a long time to scan the subnet.

Our network scanner was able to loop through and print out all the computers that were on the same subnet, so our program was also successful on that front.

File Recovery

The file recovery tool was mostly successful. It was able to recover all of the png files and nearly all the pdf files as well. One of the pdf files contained a footer in the middle of the file, which led the program to believe the file ended at that point. As a result, the final pdf was not complete and did not open as a result. The video only shows the pdfs and some pngs being opened. This is due to phone abnormalities. However, the rest of the pngs are clearly visible inside the video. If there was a problem with the pngs, the icons would not be a preview of the image, similar to the last pdf file, which shows that the pdf can't be shown.

The program could be improved by checking after the initial header for a following header, or a footer for a following footer. This would largely complicate the structure of the files as a large file can essentially contain multiple headers and footers and there would be no way to assert the actual header and actual footer of the file. The only way to find out would be to mix and match the headers and footer until a correct file is outputted.

Link to Video Demo

https://www.youtube.com/playlist?list=PLCwFTkBokDXhsEgp6AlcfuFAowpHDr3SG