**YS-5-RED Senior Project Cybersecurity in Malware Analysis**

***Website:*** *https://www.sites.google.com/view/ys5red*

*Authors: Connor Skidmore, Jonathan Tarrant, Andy Guo  
Project Owner: Yong Shi*

**8/20/2022**

**Abstract**

With society becoming more computerized each day, the paradigm of cybersecurity is emerging as a crucial field to the maintenance of the quality of life we all have come to enjoy. Every aspect of our lives seems to involve some sort of human-machine interaction, from the constant checking of our mobile phones to our growing reliance on smart appliances and devices. Industry has shown an even more eager attitude towards the adoption of automated systems. All this is to say, if our computer systems go down, life as we know it ceases to exist. Malicious actors have taken note of this as well and have begun to innovate new and more effective ways to hinder or shut down their target systems, threatening the stability of our society. We propose five algorithms to detect five different types of malwares, with a focus on the network traffic anomalies of Distributed Denial of Service (DDoS) attacks and Port Scanning Attacks. Our other modules include Internet of Things (IoT) Attacks, Artificial Intelligence (AI) Attacks, and Crypto Jacking attacks. This paper will explore the background of each of these kinds of malware, as well as an overview of the research we conducted and an exploration of how Artificial Intelligence and Machine Learning is being used to stop malware attacks today***.***

**Table of Contents**

Requirements

Analysis

Development of Software

Results

Contract

Planning and Management

Conclusion

References

4

5

6

14

16

17

24

25

**Requirements**

1. 2 research modules related to “Network traffic anomaly detection using AI and ML approaches”, in specific DDoS and Port Scanning.
2. Three modules, one of each in AI attacks, Crypto-Jacking, and Internet of Things (IoT) attack. This makes a total of 5 research modules.
3. 2 coding stimulations and solution of an AI/ML program, one for DDoS and Port Scanning.
4. 3 non-AI/ML coding stimulation and solution, one for AI attacks, Crypto-Jacking, and Internet of Things attack.
5. An IEEE research paper.
6. A documentation of the Project Plan report, which includes but not limited to Agenda, Gantt chart, Source code, Dataset, Milestones, Weekly Reports, Website link, and References.
7. GitHub page includes all researched modules with substantial material.

**Analysis**

**Tech platform**

* Colab
  + Short for Colaboratory, this is a product from Google Research. By using Colab our team can effectively write and execute python code through the browser. It allows easy access to team members, which makes collaborating more accessible.
* PyTorch
  + An library that provides powerful tools for Machine learning and Artificial Intelligence. In short, it is an incredibly comprehensive library to use for programming, training, and running AI and Machine Learning Programs.
* GitHub
  + Provides a remote and flexible workplace for our team to store, track, and collaborate on software projects. It also stores our documentations for coding and research for our senior project.
* Google Pages
  + Google owned site, its simplistic but effective layout and tools allow us to make a webpage that is visible to the public when we need it and is sufficient in presenting our results.
* Discord
  + Communication software that provides effective communication and allows us to keep track of files through different chat functions. It’s voice chat and text function allow us to communicate near real time even if we are not on campus.

**Development of Software**

***DDoS and Port Scanning***

During the first phase of development our team was not aware of the requirement to develop artificial intelligence and machine learning stimulations of the malware attacks. After a few weeks, our team formatted a contract and sent it to our project owner, Yong Shi.

We were then notified to attend a meeting and was assigned the DDoS and Port Scanning module. At that point we got a completely different requirement sheet. It had more requirements and overall remained somewhat vague. Professor Yong Shi then explained in the next meeting that our team was responsible for creating an AI/ML stimulation. Essentially make a mock DDoS and Port Scanning attack.

***AI Attack***

The situation our team has chosen in relation to this attack is the ability of AI to mimic voices. First, we implemented a text to speech to mimic three potential but popular calls, which are Bank, Internet, and Mobile. Using a package, we can record an audio file with the robotic sound of google to make it more realistic. Also, implementing a speech to text function where we get the user’s voice when they make a response to the specified service. This completes the attack.

Overall, this module was more doable the libraries online and tutorials helps tremendously in creating the code. The difficult parts were the ways to implement the audio files into Colab and access them within the code.

***Internet of Things Attack***

The situation our team has chosen is a user who has been a victim of a phishing attack, that attack is aiming to take the login information on the application of a fictious smart lightbulb company. First, we implemented the login function for the smart lightbulb. In this case we made an email phishing attempt which sends an email from our email account, it will ask for the user’s name and password as an update to the system and require the company to take in the previous login information. The solution to the issue is taking a list of the recent email phishing and comparing it to the official email accounts the company uses. The main distinction being the domain like, .gov, .net, .org, etc.

***Crypto Jacking***

The situation our team has chosen is a computer that uses a certain amount of memory it may fluctuate. With the implementation of a crypto currency which steals RAM thereby making the computer slower but at the same time only taking a small amount of RAM to mine cryptocurrency. The user is taking the place of the malicious party, and is asked how much of the total system memory they want to steal from the victim of the attack. Once they give the value, the simulation is run, and the time it takes for the program to run is tracked. Once the simulation is complete, the code calculates the amount of currency was mined using the victimized system during the execution time.

**AI Attack Source Code**

!pip install gTTS #Install through Command Prompt

from gtts import gTTS #Import Google Text to Speech

from IPython.display import Audio #Import Audio method from IPython's Display Class

#language to use

language = "en"

#random variable

num = input("Please enter a number 0, 1, 2: ")

#Choice of speech depending on number

if num == "0":

  speech = "Welcome to XYZ bank, please say your account number and pin."

elif num == "1":

  speech = "Welcome to XYZ internet, please say your account number and pin"

elif num == "2":

  speech = "Welcome to XYZ mobile, please say your account number and pin"

else:

  speech = "Invalid account number and/or pin"

#Provide the string to convert to speech

tts1 = gTTS(text = speech, lang = language, slow =False)

tts1.save('1.wav') #save the string converted to speech as a .wav file

sound\_file = '1.wav'

Audio(sound\_file, autoplay=False)

#Autoplay = True will play the sound automatically

#If you would not like to play the sound automatically, pass Autoplay = False.

#Cant run in Colab, using cmd to run

!pip install SpeechRecognition pydub

!pip install pyaudio

import speech\_recognition as sr

speech\_r = sr.Recognizer()

#Loading audio file and converting speech to text

with sr.Microphone() as source:

    print('Talking...')

    # read the audio data from the default microphone

    audio\_data = speech\_r.record(source, duration=5)

    print("Recognizing...")

    # convert speech to text

    text = speech\_r.recognize\_google(audio\_data)

    print(text)

**Internet of Things Attack (IoT Attack) Source Code**

def login():

  #Creation of user1 with a 6 digit password

  user1 = "Andy"

  pw1 = "123a@S"

  #Key to exit while loop and error key to stop new login

  key = True

  ekey = 0

  #Loop for login

  while key == True:

    print("Enter your login information")

    user = input("Enter your username: ")

    pw = input("Enter your password: ")

    if user == user1 and pw == pw1:

      key = False

      print("login successful... Welcome to your SMART LIGHT")

    else:

      print("Error username or password incorrect...")

      ekey+=1

    if ekey == 3:

      key = False

      print('Login failure')

login()

#importing the Yagmail library

!pip install yagmail #Install through Command Prompt

import yagmail

def sende():

  try:

    #initializing the server connection

    yag = yagmail.SMTP(user='ys5red2@gmail.com', password='ltbcomkwymjrwqfd')

    #sending the email

    yag.send(to='ys5red2@gmail.com', subject='Smart Life - Smart Living',

             contents='<img src="https://images.unsplash.com/photo-1606812667169-0e1991ed3742?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1170&q=80" alt="Lightbuld" width="70" height="60"> <br></br>' +

             '<h2> Due to the unexpected breach, we are requiring a mandatory password change. ' +

             '</h2> <br></br> Please verify username and old password to recive a temporary password link to reset your password.')

    print("Email sent successfully")

  except:

    print("Error, email was not sent")

#Uncomment as needed

sende()

#Importing modules/packages

import smtplib

import imaplib

import email

import traceback

#Defining email used to send phishing attack

FROM\_EMAIL = "ys5red2@gmail.com"

#Using APP password to log into gmail

FROM\_PWD = "ltbcomkwymjrwqfd"

#Connecting to Internet Message Access with gmeil.com

SMTP\_SERVER = "imap.gmail.com"

SMTP\_PORT = 993

#Reading the emails in gmail

def read\_gmail():

    try:

        mail = imaplib.IMAP4\_SSL(SMTP\_SERVER) #Connecting to server, using the logins from above, and targeting 'inbox

        mail.login(FROM\_EMAIL,FROM\_PWD)

        mail.select('inbox')

        data = mail.search(None, '(SUBJECT "Smart Life - Smart Living")') #Sifting emails for targeted subject

        mail\_id = data[1]

        id\_list = mail\_id[0].split()

        first\_email = int(id\_list[0]) #Making a list of the emails

        latest\_email = int(id\_list[-1])

        for i in range(latest\_email, first\_email, -1): #Searching through emails and fetching requirements

            data = mail.fetch(str(i), '(RFC822)' )

            for response in data:

                array = response[0]

                if isinstance(array, tuple): #Traversing data list of emails

                    msg = email.message\_from\_string(str(array[1],'utf-8')) #Pulling email message

                    e\_subject = msg['subject'] #Pulling subject

                    e\_from = msg['from'] #Pulling from whom

                    e\_body = str(msg.get\_payload(0)) #Pulling email body

                    print('From : ' + e\_from + '\n')

                    print('Subject : ' + e\_subject + '\n')

                    print('Body: ' + e\_body + '\n')

    except Exception as e:

      print(str(e))

#Solution algorithm to authenticate emails

def solve\_email():

    try:

        mail = imaplib.IMAP4\_SSL(SMTP\_SERVER) #Connecting to email server and logging in

        mail.login(FROM\_EMAIL,FROM\_PWD)

        mail.select('inbox')

        data = mail.search(None, '(SUBJECT "Smart Life - Smart Living")') #Selecting suspicious emails

        mail\_id = data[1]

        id\_list = mail\_id[0].split() #Creating list of emails from subject

        first\_email = int(id\_list[0])

        latest\_email = int(id\_list[-1])

        for i in range(latest\_email, first\_email, -1): #Traversing email list

            data = mail.fetch(str(i), '(RFC822)' )

            for response in data:

                array = response[0]

                if isinstance(array, tuple): #Checking if message from component is credible

                    msg = email.message\_from\_string(str(array[1],'utf-8'))

                    e\_from = msg['from']

                    #Checking to see if the domain is from an organization or business

                    if e\_from.\_\_contains\_\_('.inc') or e\_from.\_\_contains\_\_('.org') or e\_from.\_\_contains\_\_('.net'):

                      print('Credible')

                    else:

                      print('DO NOT TRUST ' + e\_from)

    except Exception as e:

      print(str(e))

#Calling functions

read\_gmail()

solve\_email()

**Crypto Jacking Attack Source Code**

from random import sample

from time import time, sleep

from math import floor

from copy import deepcopy

class CPU:

    RAM = 24

    startTime = 0

    sortList = list()

    def \_\_init\_\_(self, memory, sortList) -> None:

        self.memory = memory

        self.sortList = deepcopy(sortList)

    def makeArray(self, infection):

        multiplier = 1 if infection < 1 else infection

        # print(multiplier)

        nums = sample(range(0,10000\*multiplier), (9000\*multiplier))

        for x in range(len(nums)):

            self.sortList.append(nums[x])

        size = len(self.sortList)

        self.startTime = time()

        self.quickSort(self.sortList, 0, size - 1)

        # print("Array Sorted!")

    def partition(self, array, low, high):

        pivot = array[high]

        i = low - 1

        for j in range(low, high):

            if array[j] <= pivot:

                i = i + 1

                (array[i], array[j]) = (array[j], array[i])

        (array[i + 1], array[high]) = (array[high], array[i + 1])

        return i + 1

    def quickSort(self, array, low, high):

        # print("Sorting")

        if low < high:

            pi = self.partition(array, low, high)

            self.quickSort(array, low, pi - 1)

            self.quickSort(array, pi + 1, high)

class CryptoJacker:

    cryptoRam = 0

    run = True

    coinsMined = 0

    def \_\_init\_\_(self) -> None:

        pass

    def stealRAM(self):

        self.cryptoRam = input("How much RAM do you want to use? ")

        return int(self.cryptoRam)

    def mineCoins(self, seconds):

        coinsPerSec = float(self.cryptoRam) \* 0.05

        self.coinsMined += coinsPerSec \* seconds

        print(f"Total ShiCoins mined: {self.coinsMined}")

class Detector:

    def \_\_init\_\_(self) -> None:

        pass

    def detect(self, expectedTime, actualTime):

        if actualTime != expectedTime:

            print("\n-------------------------------------------\n" +

                  "  WARNING: CRYPTOJACKER RUNNING ON SYSTEM\n" +

                  "-------------------------------------------\n" +

                  f"Expected Execution Time: {expectedTime}\n" +

                  f"Actual Execution Time: {actualTime}")

def Menu():

    print("\n------Main Menu------\n" +

          "1. Run Program\n" +

          "2. Exit Program\n")

    return int(input("Enter an Option: ") or 0)

run = True

sortList = list()

cpu = CPU(10, sortList)

crypto = CryptoJacker()

cpu.makeArray(0)

expectedTime = time() - cpu.startTime

while(run):

    userInput = Menu()

    if userInput == 1:

        del cpu

        cpu = CPU(10, sortList)

        detector = Detector()

        stolen = crypto.stealRAM()

        infection = floor((stolen / cpu.RAM) \* 250)

        cpu.makeArray(infection)

        totalTime = time() - cpu.startTime

        crypto.mineCoins(totalTime)

        formatTime = "{:.2f}".format(totalTime)

        print(f"Exectution Time: {formatTime} sec")

        detector.detect("{:.2f}".format(expectedTime), formatTime)

    elif userInput == 2:

        print("Done! Thank you")

        run = False

    else:

        print("Please input a valid menu option (1 or 2)")

**Results**

**DDoS**

DDoS, or Distributed Denial-of-Service is a growing and concerning trend in both academic and industrial settings concerned with cybersecurity and malware. A DDoS attack is, quite simply, an orchestrated take-down of an organization's computer networks and infrastructure using a flood of requests, typically HTTP requests, to a server or network. The reason why the topic of malware comes into play here is that most of these attacks are accomplished using many separate subordinate computing devices and/or servers that have been infected with a program, or piece of malware, to effectively turn them into DDoS bots that the originator of this attack can use to create this flood of HTTP requests, hence, the word "distributed" in the DDoS acronym. The result of a successful DDoS attack is always either slowing the target server/network down to a speed at which it can no longer respond to legitimate user requests, or the crashing of the server/network completely. Once one sees the dire consequences of such an attack, especially for critical networks and servers that support medical, law enforcement, or national security related industries, it becomes necessary to investigate how to stop and prevent DDoS attacks from happening.

**Port Scanning**

Till this day the use of the internet is still growing. Many applications have been transformed into some sort of digital product. For example, purchasing from online stores, making appointments, and various services. For a computer to communicate and transfer requests, it requires a tunnel to send that data. The tunnel is uses is ports, there are 65,536 ports numbered from 0 to 65,535. One growing concern is the usage of port scanning. Port scanning is a method for determining what ports on a network are operating. But what makes port scanning so important? All computers must use ports to communicate and transfer digital data. This is the key, with computers being so abundant and mandatory in today’s age, data is constantly being transferred. This creates opportunities for hackers to steal and send information from and to ports. One can imagine that if an organization that relies on accurate information, such as a hospital, if false information is provided it would lead to lethal consequences. That is just one possibility of what port scanning could cause. With this potential in mind, it becomes necessary to investigate how to prevent port scanning attacks.

**AI Attacks**

The category of malware known as AI Attacks is indeed a broad category, as it covers any malware that utilizes Artificial Intelligence to carry out some sort of cyber-attack on a system or to gain unauthorized access to information. Artificial Intelligence is a fast-growing and powerful field in the world of computing, so malicious actors are beginning to realize its potential to target everyone from large agencies and corporations down to individual users.

**Crypto Jacking**

Crypto Jacking is the practice of using a person's computer to mine cryptocurrencies without the person's authorization. It makes use of the victim's hardware to mine minuscule amounts of the desired cryptocurrency. There are a couple of symptoms of Crypto Jacking that can alert victims to the presence of malware on their system. Victims might experience a slowdown in their device's performance, overheating of components such as batteries and processors, or unexpected shutdowns due to insufficient computing resources. This can cause outages and issues for non-infected users if the victim is a server that hosts web services.

**Internet of Things Attack**

An IoT Device Attack is any malware attack whose target is an IoT device. Such devices include smartphones, home assistants, and any "smart" device (light bulbs, TVs, door locks, switches, appliances, etc.). Many of these devices have vulnerabilities that are constantly being exploited by malicious parties. In today's technological space, smartphones are the prime target for these kinds of attacks. Many home-based IoT devices are controlled through apps on a person's phone. This means that gaining control over a smartphone with attached IoT devices gives the malicious party access to all the devices without having to hack each one.

Graphical user interface, text, application

Description automatically generated**Contract**

Text

Description automatically generated**Project Planning and Management**

**#1 Milestone due date 10/22/2022**

* 09/29/2022 Thursday 2:00pm in person
* 10/06/2022 Thursday 2:00pm in person
* 10/13/2022 Thursday 2:00pm in person
* 10/20/2022 Thursday 2:00pm in person

**#2 Milestone due date part 1 - 11/19/2022**

* 10/27/2022 Thursday 2:00pm in person
* 11/03/2022 Thursday 2:00pm in person
* 11/10/2022 Thursday 2:00pm in person
* 11/17/2022 Thursday 2:00pm in person

**#3 Milestone due date part 2 - 12/01/2022**

* 11/24/2022 Thursday 2:00pm in person
* 12/01/2022 Thursday 2:00pm in person
* 12/08/2022 Thursday 2:00pm in person
* 12/15/2022 Thursday 2:00pm in person

Weekly Reports and Notes in Chronological Order

**YS5-Red Cybersecurity in Malware Analysis Weekly Report**

Connor Skidmore, Jonathan Tarrant, Andy Guo

Date: 9/23/2022

**Meeting Notes from This Week:**

**Meeting 1 (Thursday 9/22/2022 2 PM):**

In this meeting we discussed the following:

• Created Google Site Framework to contain both research and Github links for source code programs.

• Discussed deliverable due dates and planned out workload distribution throughout the semester

• Settled on which malware modules we would be completing for the project

• Discussed the project scope with Dr. Perry and determined that we would be unable to do AI/ML programs for all 5 of the modules

• Discussed and looked at various sources for data sets of malware attacks, including Kaggle

**Meeting 2 (Friday 9/23/2022 12 PM)**

In this meeting we discussed the following:

• Discussed project scope with Dr. Perry

• Discussed need to do source code program for all 5 modules

• Discussed using non-AI/ML source code for 3 of the Modules

• Decided on doing in-depth research and AI/ML analysis/detection for DDoS and Port Scan attacks

• Decided on doing non-AI/ML source code program for other 3 modules

**Goals for Next Week:**

• Begin filling in Google Site with preliminary research for DDoS, Port Scan, Cryptojacking, AI Attacks, and IoT Attacks

• Continue searching for and gathering data sets on these malware topics from various sites such as Kaggle

• Continue learning about ML/AI, how they are used to detect malware, as well as how they apply to our specific modules

Next Meeting: Thursday 9/29/2022 2 PM

**YS5-Red Cybersecurity in Malware Analysis Weekly Report**

Connor Skidmore, Jonathan Tarrant, Andy Guo

Date: 9/30/2022

**Meeting Notes from This Week:**

**Meeting 1 (Thursday 9/29/2022 2 PM):**

In this meeting we discussed the following:

• Discussed progress on website

• Discussed AI Attacks and DDoS modules

• Discussed deadlines and organization of deliverables (possible Gantt chart development)

• Went over research conducted so far

• Discussed prototype presentation on 10/18/2022

**Goals for Next Week:**

• Getting one research module done for DDoS attacks

• Continuing to plan and discuss AI and non-AI code development

• Plan for prototype presentation to class

Next Meeting: Thursday 10/6/2022 2 PM

**YS5-Red Cybersecurity in Malware Analysis Weekly Report**

Connor Skidmore, Jonathan Tarrant, Andy Guo

Date: 10/7/2022

**Meeting Notes from This Week:**

**Meeting 1 (Thursday 10/6/2022 2 PM):**

In this meeting we discussed the following:

• Discussed upcoming progress presentations for Dr. Perry’s class

• Further hashed out workload distribution and discussed work that needed to be done over the weekend

• Went over research completed so far and discussed the plan for further research to be completed over the weekend.

**Goals for Next Week:**

• Being fully prepared to present our work up to this point to Dr. Perry’s class

• Getting a significant amount of research completed and up on the website to show both the class and Dr. Shi at our next meeting.

• Continuing to work with PyTorch to explore the development of our source code/programs to detect malware.

• Continuing to collect and search for data sets on our assigned Malware topics using sites such as Kaggle.

Next Meeting: Thursday 10/13/2022 2 PM

**YS5-Red Cybersecurity in Malware Analysis Weekly Report**

Connor Skidmore, Jonathan Tarrant, Andy Guo

Date: 10/14/2022

**Meeting Notes from This Week:**

**Monthly Meeting with Dr. Shi (Wednesday 10/12/2022 1 PM):**

In this meeting we discussed the following with Dr. Shi:

• Need to have working demonstrations ready by week of November 7th.

• Need to finish research by end of October

• Be ready to demonstrate prototypes to Dr. Shi by our next monthly meeting (11 November 2022)

• Additional deliverable required now: IEEE format research paper

• Reiterated AI/ML approaches for Network Traffic Anomalies DDoS and Port Scanning

• Reiterated Conventional non-AI/ML approaches for other 3 modules.

• Reiterated using Python/Pytorch/Colab for code/environment.

**Meeting 2 (Thursday 10/13/2022 2 PM)**

In this meeting we discussed the following:

• Discussed progress with research

• Discussed progress with code

• Assigned work for the IEE Research paper among the 3 group members

• Discussed preparation for milestone 1 demonstration.

**Goals for Next Week:**

• Prepare for presentations

• Start work on IEEE paper

• Work on Research so that it can be done by the end of October.

Next Meeting: Tuesday 10/18/2022 2 PM

**YS5-Red Cybersecurity in Malware Analysis Weekly Report**

Connor Skidmore, Jonathan Tarrant, Andy Guo

Date: 10/21/2022

**Meeting Notes from This Week:**

**Meeting 1 (Tuesday 10/18/2022 2 PM):**

In this meeting we discussed the following:

• Progress on IEEE paper

• Need to add Dr. Shi to google site as editor

• Jonathan’s progress on Pytorch to use for our AI/ML Programs

**Meeting 2 (Thursday 10/20/2022 2 PM)**

In this meeting we discussed the following:

• Continued to work on research and code

• Discussed current state of deadlines and workload

• Discussed goals for next week

**Goals for Next Week:**

• Near completion of research for our first milestone

• Have some AI/ML code completed for our Network Traffic Anomaly modules

Next Meeting: Monday 10/24/2022 5 PM

**YS5-Red Cybersecurity in Malware Analysis Weekly Report**

Connor Skidmore, Jonathan Tarrant, Andy Guo

Date: 10/29/2022

**Meeting Notes from This Week:**

**Meeting 1 (Monday 10/24/2022 5 PM):**

In this meeting we worked on the following:

• **Continued work on IEEE research paper finishing first few sections**

• **Jonathan continued work on understanding PyTorch to apply to AI/ML code**

• **Andy continued work on IoT Attack non-AI code**

**Meeting 2 (Tuesday 10/25/2022 5 PM)**

In this meeting we discussed the following:

• Continued to work on IEEE research paper

• Jonathan continued work on DDoS/Port Scanning

• Andy Continued work on IoT device attacks

**Meeting 3 (Thursday 10/25/2022 2 PM)**

In this meeting we discussed the following:

• Continued to work on IEEE research paper

• Jonathan continued work on learning AI/ML code/programming

• Andy Continued work on his phishing code for IoT device attacks

**Goals for Next Week:**

• Near completion of source code

• Have C Day Submission ready by Monday

• Have IEEE paper near complete or complete by end of next week

• Revamp website to showcase updated research/code

Next Meeting: Tuesday 11/1/2022 2 PM

**YS5-Red Cybersecurity in Malware Analysis Weekly Report**

Connor Skidmore, Jonathan Tarrant, Andy Guo

Date: 11/4/2022

**Meeting Notes from This Week:**

**Meeting 1 (Tuesday 11/1/2022 2 PM):**

In this meeting we discussed the following:

• How we can work towards the deadline of the 9 November meeting

• Finalizing the 5 modules to demonstrate to Dr. Shi

• Finalizing the website to show to Dr. Shi

• Finalizing the IEEE paper to show to Dr. Shi

**Meeting 2 (Thursday 11/3/2022 2 PM)**

In this meeting we discussed the following:

• Continued to finalize our modules for demonstration next week.

• Continued to finalize the website to show to Dr. Shi

• Continued to work on and finalize the IEEE paper to show to Dr. Shi

**Goals for Next Week:**

• Have modules ready to demonstrate to Dr. Shi

• Have website ready to show to Dr. Shi

• Have IEEE paper draft ready to show to Dr. Shi

Next Meeting: Tuesday 11/8/2022 2 PM

**YS5-Red Cybersecurity in Malware Analysis Weekly Report**

Connor Skidmore, Jonathan Tarrant, Andy Guo

Date: 11/11/2022

**Meeting Notes from This Week:**

**Meeting 1 (Tuesday 11/8/2022 2 PM):**

In this meeting we discussed the following:

• Finalizing items to present to Dr. Shi tomorrow

• Finalizing Code and draft of IEEE paper for Dr. Shi

• Determining how meeting and presentation will go tomorrow

**Meeting 2 (Wednesday 11/9/2022 1 PM):**

In this meeting we discussed the following with Dr. Shi:

• Demonstrated working code for non-AI modules

• Showed work so far on IEEE paper

• Showed sources

• Discussed plans and suggestions for what to add to project and how to modify project

**Meeting 3 (Thursday 11/10/2022 2 PM)**

In this meeting we discussed the following:

• Discussed Dr. Shi’s feedback on our project from yesterday.

• Discussed how workload for next deliverables will be distributed.

• Further worked on suggested improvements to project

**Goals for Next Week:**

• Implement Dr. Shi’s suggestions

• Continue working on finishing project for Nov 30 Meeting with Dr. Shi

• Prepare for Milestone 2 presentation next week.

Next Meeting: Tuesday 11/15/2022 2 PM

A picture containing table

Description automatically generated

Summary / Conclusions and feedback

***What can be done better***

-Senior project needs to have clear requirements and scope from the beginning of the project.

-Mandatory and more frequent meetings with project owner will be useful to see progress and keep on track.

-A well described product by project owner that can be subjected to questions by the students.

-An increased team size to minimal of 4 to reduce workload throughout semester.

-Introduction page to project needs to have specific requirements

1. Amount of team members minimal of 3

2. Detailed description of project

***What was frustrating***

-Throughout the semester the project deliverables were changed.

-Scope was defined but not achievable in a single semester long project.

-Knowledge related to project needs to be clearly stated.

-Each meeting needed to reexplain last meeting’s information.

-Project owner refused to alter requirements to better suit our team’s abilities.

-After contract was signed, other requirements were added that were not included in the original contract.

References

“2022 cyber security statistics trends & data,” *PurpleSec*, 17-Oct-2022. [Online]. Available: https://purplesec.us/resources/cyber-security-statistics/. [Accessed: 29-Nov-2022].

B. Guembe, A. Azeta, S. Misra, V. C. Osamor, L. Fernandez-Sanz, and V. Pospelova, “The emerging threat of AI-Driven Cyber Attacks: A Review,” *Applied Artificial Intelligence*, vol. 36, no. 1, 2022.

D. Tang and X. Kuang, “Distributed denial of service attacks and defense mechanisms,” *IOP Conference Series: Materials Science and Engineering*, vol. 612, no. 5, p. 052046, 2019.

Address false positives/negatives in Microsoft defender for endpoint,” *Microsoft Learn*. [Online]. Available: https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/defender-endpoint-false-positives-negatives?view=o365-worldwide. [Accessed: 29-Nov-2022].

E. Chickowski, “What is cryptojacking? how to prevent, detect, and recover from it,” *CSO Online*, 20-Jun-2022. [Online]. Available: https://www.csoonline.com/article/3253572/what-is-cryptojacking-how-to-prevent-detect-and-recover-from-it.html?page=2. [Accessed: 29-Nov-2022].

H. H. Hajri, B. M. Mughairi, M. I. Hossain, and A. M. Karim, “Crypto jacking a technique to leverage technology to mine crypto currency,” *International Journal of Academic Research in Business and Social Sciences*, vol. 9, no. 3, 2019.

H. Kumar, “Survey on Malware Detection Methods.”

K. Hahn, “The real reason why malware detection is hard--and underestimated,” *G Data Software*, 15-Jul-2022. [Online]. Available: https://www.gdatasoftware.com/blog/2022/06/37445-malware-detection-is-hard. [Accessed: 29-Nov-2022].

M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Surveying port scans and their detection methodologies,” *The Computer Journal*, vol. 54, no. 10, pp. 1565–1581, 2011.

O. Aslan and R. Samet, “A Comprehensive Review on Malware Detection Approaches,” *IEEE Access*, Jan. 2020.

Valeo Networks, “How is the internet of things (IOT) being impacted by malware?,” *Valeo Networks*, 10-Sep-2021. [Online]. Available: https://www.valeonetworks.com/how-is-the-internet-of-things-iot-being-impacted-by-malware/. [Accessed: 29-Nov-2022].

*Python : How to parse the body from a raw email , given that raw email does not have a "body" tag or anything*. Stack Overflow. Retrieved December 1, 2022, from https://stackoverflow.com/questions/17874360/python-how-to-parse-the-body-from-a-raw-email-given-that-raw-email-does-not

*Send email using yagmail in python*. GeeksforGeeks. (2021, October 7). Retrieved December 1, 2022, from https://www.geeksforgeeks.org/send-email-using-yagmail-in-python/

Rockikz, A. (2019, October 15). *How to convert speech to text in Python*. Python Code. Retrieved December 1, 2022, from https://www.thepythoncode.com/article/using-speech-recognition-to-convert-speech-to-text-python