# Sysinternals

### Process Explorer

1. Liệt kê một vài (tối thiểu 5) chức năng chính của chương trình.

- Kiểm tra metrics (CPU, GPU, RAM, …) của các processes chạy trên máy chủ

- Kiểm tra chi tiết của các Process. (parent process và các child processes của nó).

- Kiểm tra VirusTotal của từng process.

- Kiểm tra path của từng process.

- Có thể dùng để set priority, restart, kill hoặc suspend cho process chỉ định.

- Kiểm tra các tiến trình của process chỉ định.

- Kiểm tra các thư viện (.DLL files) được nạp vào trong process chỉ định.

1. Process có màu hồng là gì? Process có màu tím là gì?

- Process có màu hồng là process của hệ thống

- Process có màu tím là process khi chúng ta mở các ứng dụng lên kể từ khi khởi động máy chủ.

3. Session là gì? Session 0 có gì đặc biệt không?

Session là phiên bản làm việc của process đó, với session 1 có nghĩa là process đó đang trong trạng thái hoạt động và session 0 là process đó đang trong trạng thái pending.

Session 0 xảy ra trong trường hợp có vần đề gì đó xảy ra bên trong process làm cho process đó bị dừng hoạt động.

4. Cách xem các Thread của một Process.

- Cách 1: Chúng ta nhấn chuột phải vào một process trên column Process, ở đây chúng ta có thể thấy một cột tên Properties, Khi trang Properties hiện lên ta có thể thấy 1 trang Threads, ta nhấn vào đó thì chúng ta có thể xem các Threads bao gồm các thuộc tính của các Thread tương ứng.

- Cách 2: Khi chúng ta nhấn vào một Process bất kỳ, ở phía dưới sẽ hiển thị lên các tab như Handles, DLLs, Threads, chúng ta có thể bấm vào tab Threads thì sẽ xem được các Threads thuộc Process đó.

5. Cách xem các kết nối (Connections) của một Process.

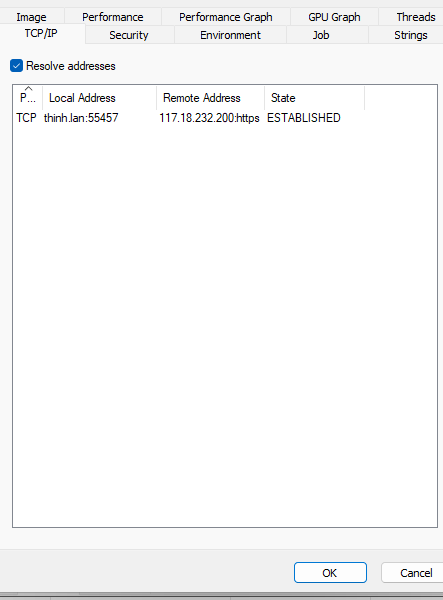
Chúng ta nhấn chuột phải vào một process trên column Process, ở đây chúng ta có thể thấy một cột tên Properties, Khi trang Properties hiện lên ta có thể thấy 1tab TCP/IP, ta nhấn vào đó thì chúng ta có thể xem các kết nối trong một process.

6. Cách xem tổng lưu lượng mạng gửi/nhận của một Process.

Chúng ta nhấn chuột phải vào một process trên column Process, ở đây chúng ta có thể thấy một cột tên Properties, Khi trang Properties hiện lên ta có thể thấy 1tab Performance, chúng ta có thể thấy ở ô I/O có hiển thị Reads/Writes, đây chính là tổng lưu lượng mạng gửi/nhận của một process.

7. Hãy viết một chương trình Go đơn giản download tập tin https://download.sysinternals.com/files/SysinternalsSuite.zip và sử dụng Process Explorer để theo dõi chương trình kết nối ra IP nào, dung lượng gửi nhận bao nhiêu.

**Done**

****

****

8. Cách để xác định một Process có chữ ký điện tử hợp lệ.

Chúng ta có thể kiểm tra một Process có chữ ký hợp lệ bằng cách xem cột Verified Signer của process chỉ định nếu có chữ **Verified** + **Signer** thì process đó đã có chữ ký điện tử hợp lệ.

9. Hãy viết một chương trình Go đơn giản để kiểm tra chữ ký điện tử của một tập tin thực thi.

**Done**

10. Một Process khi chạy sẽ nạp rất nhiều Module (.DLL) để hoạt động, cách xem các Module được một Process nạp.

Khi chúng ta nhấn vào một Process bất kỳ, ở phía dưới sẽ hiển thị lên các tab như Handles, DLLs, Threads, chúng ta có thể bấm vào tab DLLs thì sẽ xem được các Modules thuộc Process đó.

11. Module có chữ ký điện tử như Process không? Nếu có, nêu cách xác định Module có chữ ký điện tử hợp lệ.

Ở trong tab DLLs. Các Module cũng sẽ hiển thị cột Verified Signer, chúng ta có thể kiểm tra thông qua chữ **Verified** để xác định module đó có hợp lệ hay không.

12. Hãy viết một chương trình Go đơn giản liệt kê hết những tập tin .DLL đang được nạp và so sánh kết quả với Process Explorer.

**Done**

13. Một Process khi mở một đối tượng nào đó (tập tin, registry key, thread, token, ...) sẽ giữ một Handle của đối tượng đó. Nêu cách để xem các Handle mà process đang giữ. Một Process giữ quá nhiều Handle có ảnh hưởng gì tới hệ thống không?

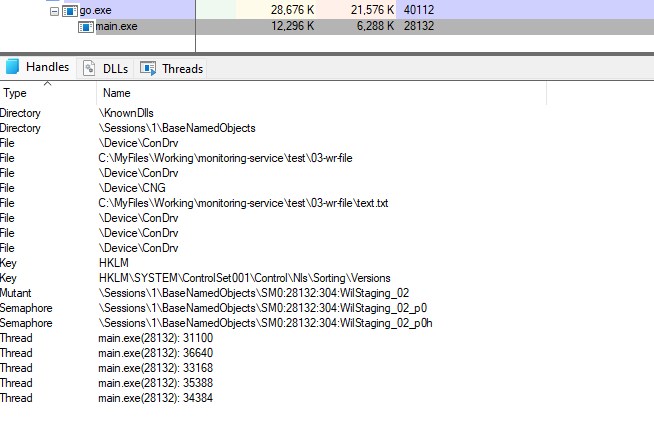
Khi chúng ta nhấn vào một Process bất kỳ, ở phía dưới sẽ hiển thị lên các tab như Handles, DLLs, Threads, chúng ta có thể bấm vào tab Handles thì sẽ xem được các đối tượng (tập tin, regisreykey, thread, token) thuộc Process đó.

Trong trường hợp một Process giữ quá nhiều Handle sẽ gây ra 2 vấn đề:

- Sẽ làm tăng memory của Process đó, làm giảm performance của Process.

- Nếu Process giữ quá nhiều tệp tin quan trọng, khi Process bị hack, thì toàn bộ những tệp tin quan trọng sẽ bị đánh cắp.

14. Hãy viết một chương trình Go đơn giản thực hiện đọc/ghi tập tin và theo dõi Handle được mở trong Process Explorer.

****

15. Base Address của Process và Module là gì? Cách để xem Base Address của Process và Module.

Base address của một process và module là địa chỉ bắt đầu của không gian địa chỉ mà hệ điều hành cấp cho tiến trình đó trong bộ nhớ. Nó là điểm khởi đầu từ đó tiến trình có thể truy cập vào bộ nhớ, bao gồm mã chương trình, dữ liệu, ngăn xếp, và heap.

Chúng ta có thể xem Base Address của Process bằng cách

Chúng ta có thể xem Base Address của Module bằng cách nhấn vào View -> Select Columns, nhấn vào tab DLL, tích chọn vào Image Base Address để hiển thị Address cho các Module trong một threads.

16. Làm cách nào để xác định nhanh 1 cửa sổ đang mở (ví dụ cửa sổ Microsoft Word đang mở) thuộc về Process nào?

Chúng ta có thể nhấn Crtl+F để tìm nhanh tên của Process đó

### Process Monitor (ProcMon)

1. Chương trình ProcMon dùng để làm gì?

Chương trình ProcMon dùng để xem các thao tác (operations) của một process, giúp cho chúng ta kiểm tra chi tiết khi một process gặp lỗi thì sẽ lỗi ở bước nào và tiến hành debug.

1. Những loại sự kiện nào có thể monitor bằng ProcMon?

- Khởi tạo, xóa hoặc modify registry key.

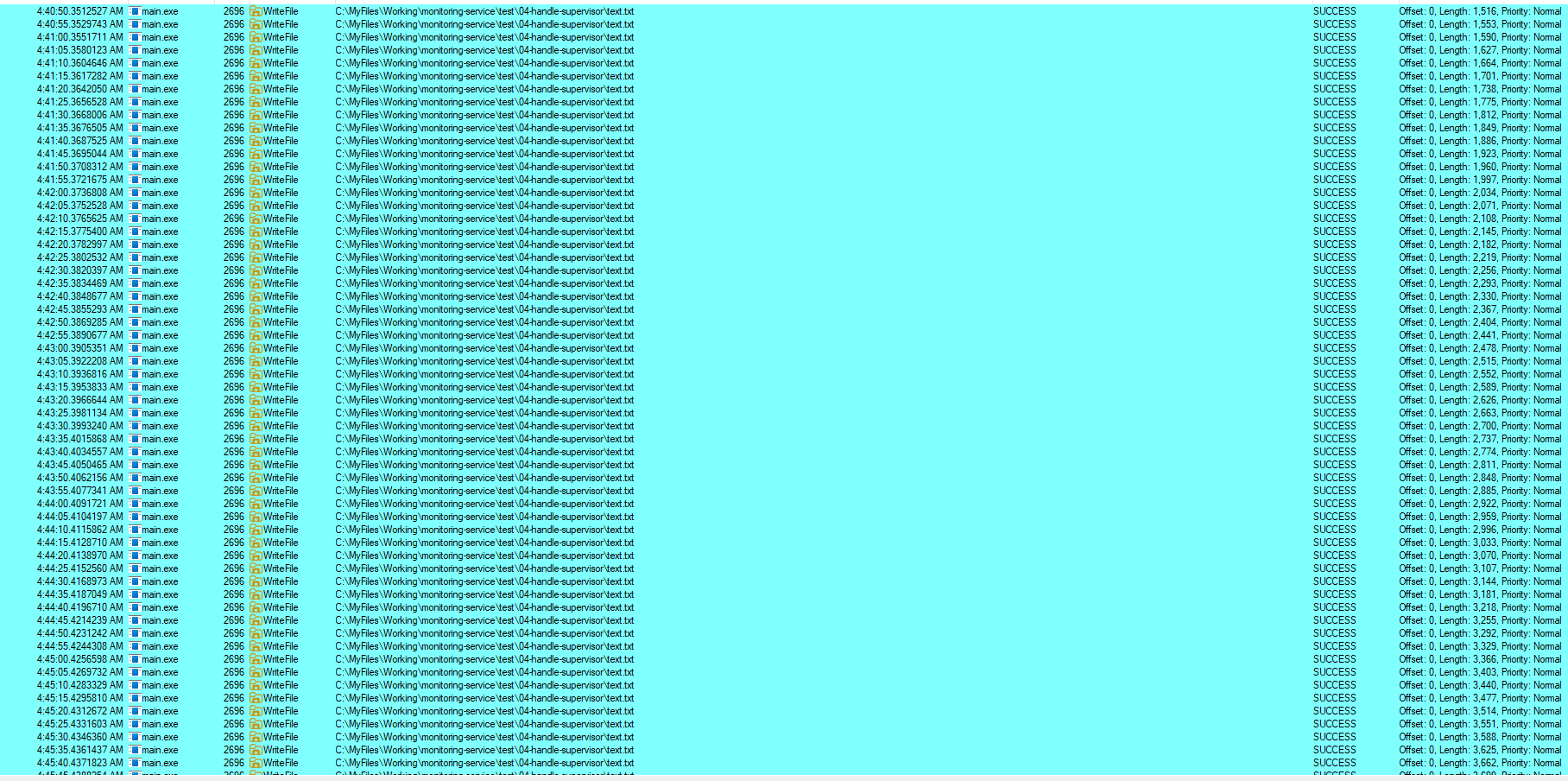
- Khởi tạo, xóa processes hoặc threads, thay đổi priority của processes hoặc threads.

- Các thao tác với các file (tạo, xóa, sửa, đổi tên, …)

- TCP/UDP network connections.

- Các service start hoặc stop events, config của service change.

3. Viết chương trình Go đơn giản thực hiện tuần tự các công việc sau: (1) Ghi ra tập tin text (.txt) có nội dung tùy ý; (2) Mở tập tin text vừa ghi ra bằng Notepad. Dùng ProcMon để monitor quá trình hoạt động của chương trình.

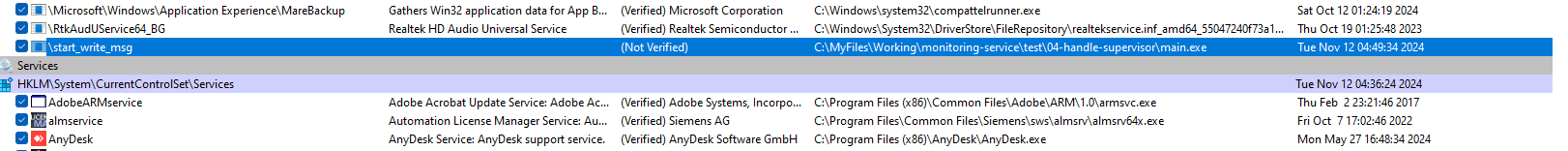
****

### Autoruns

1. Chương trình Autoruns dùng để làm gì?

Chương trình Autoruns dùng để kiểm tra các process nào đang tự động restart cùng hệ thống, từ đó chúng ta có thể set chế độ schedule cho các process chỉ định, điều này giúp tránh việc các chương trình nếu bị virus xâm nhập sẽ không restart theo máy chủ và lấy được tài nguyên từ máy chủ.

1. Dùng command line để đăng ký chương trình đã viết ở phần ProcMon 3 khởi động cùng hệ thống dưới dạng Task Scheduler. Sử dụng chương trình Autoruns để quan sát chương trình đã được đăng ký.

****

1. Trong phần khởi động bằng Services (HKLM\System\CurrentControlSet\Services) có rất nhiều tập tin .DLL tuy nhiên tập tin .DLL không thể hoạt động đơn lẻ, vậy chương trình nào đã nạp những tập tin này? Hình thức khởi động này có gì nguy hiểm không?

Service Control Manager (SCM), một phần của Windows chịu trách nhiệm quản lý các dịch vụ và tiến trình chạy trên hệ thống.

Hình thức khởi động này có một vài rủi ro như sau:

Tấn công Malware:

Nếu một dịch vụ bị tác động bởi phần mềm độc hại, nó có thể nạp các tập tin .DLL độc hại, cho phép kẻ tấn công thực hiện các hoạt động không mong muốn trên hệ thống.

Dịch vụ không xác định:

Nếu không kiểm soát hoặc xác nhận các dịch vụ đang chạy, có thể có các dịch vụ không rõ nguồn gốc hoặc không được tin cậy, khiến hệ thống dễ bị tổn thương.

Tiêu tốn tài nguyên:

Một số dịch vụ có thể tiêu tốn tài nguyên hệ thống không cần thiết, ảnh hưởng đến hiệu suất của máy tính.

Rủi ro bảo mật:

Nếu một dịch vụ có quyền truy cập cao (như Local System), việc nạp một .DLL độc hại có thể cho phép kẻ tấn công có quyền truy cập sâu vào hệ thống.

### TcpView và Wireshark

1. Chương trình TcpView dùng để làm gì?

Chương trình TcpView dùng để list toàn bộ những kết nối của các TCP endpoints đến từng process trong hệ thống, bao gồm các local address và remote address. Đồng thời chúng ta có thể xem số lượng packets, bytes truyền và nhận giữa các address với nhau.

1. State ESTABLISHED, LISTENING, SYN\_SENT có nghĩa là gì?

Trạng thái ESTABLISHED là trạng thái cho biết một kết nối TCP đã được thiết lập thành công, dữ liệu có thể truyền qua lại giữa hai thiết bị trong trạng thái này.

Trạng thái LISTENING là trạng thái một process đang chờ các kết nối từ các máy khác. Nó đang lắng nghe từ một cổng cụ thể để nhận các kết nối.

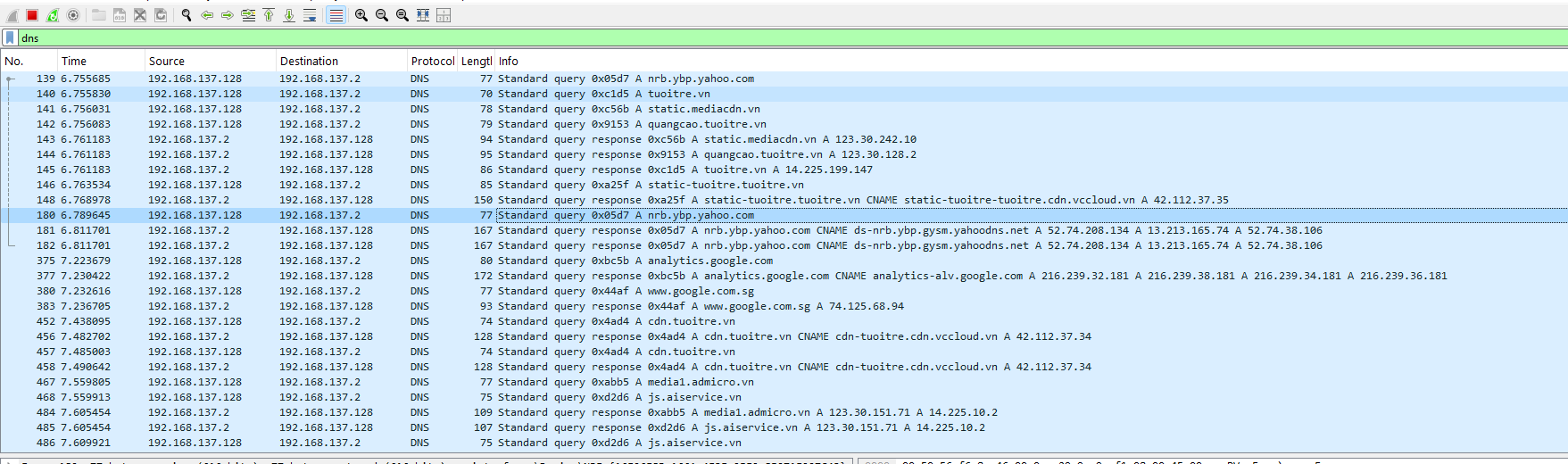
Trạng thái SYD\_SENT là trạng thái một process đã gửi yêu cầu kết nối (SYN packet) đến một máy tính khác nhưng không nhận được phản hồi xác nhận (Gói tin ACK).

3. Command nào trên Windows có tác dụng tương ứng với TcpView?

netstat -a

4. Cài đặt và sử dụng chương trình Proxifier để điều hướng mọi kết nối của máy tính thông qua chương trình này. Các kết nối thay đổi ra sao trên TcpView?

5. Cài đặt chương trình Wireshark. Sử dụng Wireshark để theo dõi quá trình phân giải tên miền tuoitre.vn.



6. Viết chương trình Go đơn giản nhận vào tham số là URL và tiến hành download nội dung từ URL này. Dùng chương trình download 2 URL sau:

- https://tuoitre.vn/nga-cong-bo-phe-duyet-vacxin-covid-19-dau-tien-tren-the-gioi-chuan-bi-san-xuat-hang-loat-2020081116001058.htm

- https://chinhphu.vn/gioi-thieu-chinh-phu

Dùng chương trình Wireshark để theo sniff đường truyền. So sánh kết quả giữa 2 lần sniff.

**Done code**

7. Có phương pháp nào để sniff được dữ liệu HTTPS hay không?

Tấn công Man-in-the-Middle (MitM): Kỹ thuật này có thể được thực hiện bằng cách tạo ra một chứng chỉ SSL giả mạo và lừa người dùng cài đặt nó. Điều này cho phép kẻ tấn công giải mã và đọc được dữ liệu mã hóa. Tuy nhiên, đây là một hoạt động phi pháp và vi phạm quyền riêng tư.

Sử dụng công cụ phân tích mạng: Các công cụ như Wireshark có thể ghi lại lưu lượng mạng, nhưng để xem được nội dung HTTPS, bạn sẽ cần có chứng chỉ phù hợp để giải mã. Thường thì điều này chỉ khả thi trong một môi trường kiểm thử mà bạn kiểm soát.

Proxy HTTP/HTTPS: Sử dụng các proxy như Burp Suite hoặc Fiddler có thể giúp bạn phân tích lưu lượng HTTPS bằng cách cài đặt chứng chỉ của proxy vào thiết bị. Điều này thường được sử dụng trong phát triển và kiểm thử ứng dụng.

Chứng chỉ SSL/TLS: Trong các môi trường phát triển, bạn có thể sử dụng các chứng chỉ tự ký để thiết lập một môi trường mà bạn có thể kiểm soát lưu lượng HTTPS.

# Mạng và internet

1. DNS là gì và cơ chế hoạt động như thế nào?

DNS (Domain Name System) là hệ thống cơ sở dữ liệu phân giải tên miền, nó có tác dụng biên dịch tên website hay hostname thành địa chỉ IP. Thông tin của từng tên miền ứng với địa chỉ IP nào sẽ được ghi lại trong một thư viện và thư viện này được lưu trên các server tên miền.

- Cơ chế hoạt động:

Người dùng gửi truy vấn: Khi người dùng nhập một tên miền vào Google, Bing, …, máy tính sẽ gửi một yêu cầu đến máy chủ DNS gần nhất.

Recursive DNS Server tìm kiếm: Recursive DNS Server tìm kiếm thực hiện tìm kiếm thông tin từ Root đến Authoritative DNS Server.

Authoritative DNS Server cung cấp địa chỉ IP trùng với tên miền.

Nhận địa chỉ IP: Recursive DNS Server trả lại thông tin cho máy tính của bạn và cập nhật vào bộ nhớ

Kết nối đến máy chủ: Máy tính của bạn sẽ sử dụng địa chỉ IP này để kết nối trực tiếp đến máy chủ của Google, Bing, … và tải website về.

1. Route là gì và cơ chế hoạt động như thế nào?

Route là tập hợp các thông tin định tuyến cho phép các gói tin di chuyển từ nguồn đến đích qua mạng. Nó thường được sử dụng để xác định cách mà các gói dữ liệu sẽ được gửi qua các thiết bị mạng như router và switch

1. Trình duyệt hoạt động như thế nào khi truy cập vào website tuoitre.vn?

Đầu tiên là nhập URL vào thanh địa chỉ của trình duyệt, tiếp theo trình duyệt sẽ gửi yêu cầu đến máy chủ DNS để phân giải tên miền tuoitre.vn thành địa chỉ IP, máy chủ DNS sẽ trả về địa chỉ IP tương ứng với tên miền. Tiếp theo, trình duyệt sẽ tiến hành thiết lập một kết nối TCP đến địa chỉ IP đã nhận được, thực hiện quá trình bắt tay TCP để thiết lập kết nối ổn định giữa client (trình duyệt) và server. Sau khi tạo kết nối TCP thành công, trình duyệt sẽ gửi một yêu cầu đến máy chủ web để lấy nội dung trang web. Sau khi nhận được mã HTML của trang web, trình duyệt sẽ xây dựng một cây DOM (Document Object Model) từ mã HTML để tìm kiếm để các nguyên bổ sung và nếu trong mã HTML có các liên kết đến các tệp CSS hoặc JavaScript thì trình duyệt sẽ yêu cầu đến máy chủ để tải các tệp đó.

4. Firewall là gì? Các tính năng của Firewall trên Windows gồm những gì?

Firewall là một bức tường hoặc một rào cản giữa máy tính cá nhân và thế giới mạng bên ngoài. Nó giúp cho việc ngăn chặn sự tấn công bởi các mối đe dọa trên mạng từ đó đảm bảo quá trình bảo mật thông tin cho người dùng.

Các tính năng của Firewall trên windows bao gồm:

+ Kiểm soát truy cập mạng.

+ Quy tắc định tuyến.

+ Chặn kết nối không an toàn.

+ Theo dõi và ghi log.

+ Hỗ trợ IPsec và VPN.

+ Cấu hình từ xa.

+ Tích hợp với Windows Defender.

+ Chế độ khôi phục.

5. Khi trình duyệt không truy cập được một website, làm cách nào để xác định nguyên nhân do đâu?

- Kiểm tra kết nối internet.

- Kiểm tra đường dẫn URL (kiểm tra đúng tên miền, ping tên miền bằng câu lệnh `ping` thử xem đúng tên miền hay không, truy vấn dns xem tên miền đã được phân giải hay chưa)

- Kiểm tra cài đặt proxy

- Kiểm tra tường lửa hoặc phần mềm virus có chặn website hay không

- Kiểm tra lưu lượng mạng thông qua wireshark để tìm ra nguyên nhân cụ thể.

6. Làm cách nào để xác định được người đã đăng ký hosting và domain?

Chúng ta có thể xác định được người đã đăng ký hosting và domain bằng cách sử dụng các website trực tuyến để kiểm tra hosting và domain (ví dụ whois.net hoặc hostingchecker.com, …)

# Mã hóa và an toàn đường truyền

1. Hash (mã băm) là gì? Các tính chất của hash? Hash thường được ứng dụng vào việc gì?

Hash (mã băm) là một đoạn mã có nội dung kích thước cố định sau khi chạy qua hàm băm, mã băm dùng để mã hóa dữ liệu, đảm bảo việc bảo mật dữ liệu một cách tốt nhất

Các tính chất của hash:

* Xác định: Một đầu vào cụ thể khi đi qua hàm băm sẽ luôn tạo ra cùng một giá trị băm. Điều này có nghĩa cùng một dữ liệu đầu vào sẽ luôn dẫn đến cùng một mã băm.
* Đơn hướng: Rất khó để tính hướng ngược lại đầu vào từ mã băm sau khi mã hóa, điều này duy trì tính bảo mật.
* Tính toàn vẹn: Một thay đổi nhỏ trong dữ liệu đầu vào sẽ tạo ra một mã băm hoàn toàn khác, giúp phát hiện bất kỳ thay đổi hoặc sửa đổi nào trong dữ liệu.
* Hiệu quả: Mã băm thường được tạo ra rất nhanh chóng bởi hàm băm, từ đó nó có thể được ứng dụng trong việc xử lý dữ liệu lớn.
* Không trùng lặp: Xác xuất để hai dữ liệu khác nhau có cùng mã băm (còn gọi là "collision") phải cực kỳ thấp, đảm bảo rằng mỗi mã băm là duy nhất cho một tập dữ liệu cụ thể.

Ứng dụng:

* Bảo mật mật khẩu: Các hệ thống lưu trữ mật khẩu thường băm mật khẩu người dùng trước khi lưu trữ để bảo vệ khỏi bị đánh cắp.
* Chữ ký số và chứng thực: Hash được sử dụng trong việc tạo và xác thực chữ ký số, đảm bảo tính toàn vẹn và nguồn gốc của tài liệu số.
* Blockchain và tiền mã hóa: Các khối trong blockchain được liên kết với nhau - thông qua các giá trị băm, giúp bảo vệ tính toàn vẹn và an ninh của toàn bộ chuỗi.
* Kiểm tra tính toàn vẹn của tệp: Hash được sử dụng để kiểm tra tính toàn vẹn của tệp tin khi tải xuống từ internet, đảm bảo rằng tệp không bị thay đổi hoặc bị lỗi trong quá trình truyền tải.

- Hệ thống quản lý dữ liệu và cơ sở dữ liệu: Hash tables được sử dụng để tăng tốc độ tìm kiếm và truy xuất dữ liệu trong các hệ thống quản lý dữ liệu và cơ sở dữ liệu.

2. Viết chương trình Go đơn giản để tính hash MD5, SHA-1, SHA-256 của một thư mục (không cần tính thư mục con). So sánh kết quả với kết quả tính hash của TotalCmd.

**Done**

3. Mã hóa đối xứng và bất đối xứng khác nhau như thế nào?

- Mã hóa đối xứng là một kỹ thuật đã được giới thiệu từ lâu, trong khi mã hóa bất đối xứng là kỹ thuật mới hơn.

- Mã hóa bất đối xứng mất nhiều thời gian hơn để thực hiện do logic phức tạp liên quan. Vì lý do này, mã hóa đối xứng vẫn được ưu tiên sử dụng khi truyền dữ liệu hàng loạt.

- Mã hóa bất đối xứng an toàn hơn vì nó sử dụng các key khác nhau cho quá trình mã hóa và giải mã.

4. Mã hóa đối xứng có những dạng nào?

AES-12́, AES-192, AES-256

5. Cipher Mode trong mã hóa khối là gì? Tại sao cần Cipher Mode? ECB với CBC khác nhau như thế nào?

Cipher Mode là một phương pháp sử dụng để mã hóa và giải mã dữ liệu trong các thuật toán mã hóa khối (block cipher). Mã hóa khối xử lý dữ liệu theo khối (thường là 128bit, 192bit hoặc 256bit) thay vì từng bit hoặc byte. Cipher modes định nghĩa cách mà các khối dữ liệu được xử lý và kết hợp với nhau để tạo ra một chuỗi dữ liệu mã hóa.

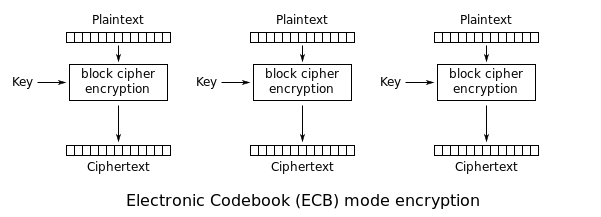
Chúng ta cần Cipher modes vì nó cho phép mã hóa dữ liệu có kích thước lớn hơn kích thước khối của thuật toán mã hóa, điều này giúp cho việc mã hóa các tệp, dữ liệu có kích thước lớn dễ dàng hơn. Đồng thời có một số cipher modes cung cấp tính năng bảo mật cao hơn so với mã hóa đơn giản, giúp giảm thiểu các loại tấn công như tấn công lặp lại. Cuối cùng, chúng có thể bảo vệ tính toàn vẹn của dữ liệu mã hóa, đảm bảo dữ liệu không bị thay đổi trong quá trình truyền tải.

Sự Khác Nhau Giữa ECB và CBC

1. ECB (Electronic Codebook Mode)

Cách hoạt động:

Trong ECB, mỗi khối dữ liệu được mã hóa độc lập với nhau. Điều này có nghĩa là nếu hai khối dữ liệu giống nhau, chúng sẽ có cùng một mã hóa.

****

Ưu điểm:

Dễ dàng triển khai và thực hiện.

Tốc độ mã hóa nhanh.

Nhược điểm:

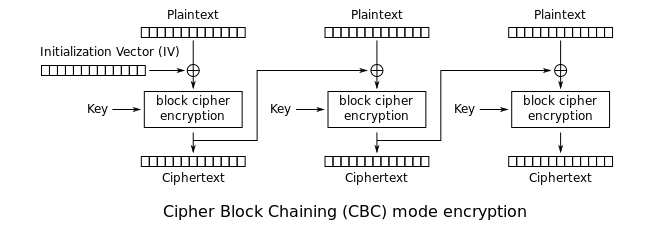
Không an toàn khi sử dụng cho dữ liệu lớn hoặc nhạy cảm. Vì cùng một khối dữ liệu sẽ luôn tạo ra cùng một mã hóa, điều này có thể dẫn đến việc rò rỉ thông tin và tấn công bằng cách phân tích.

Ví dụ: Nếu có một hình ảnh mà có một phần lặp lại, các phần đó sẽ có cùng một mã hóa, dễ dàng cho kẻ tấn công phát hiện.

2. CBC (Cipher Block Chaining Mode)

Cách hoạt động:

CBC sử dụng một vector khởi tạo (IV) cho quá trình mã hóa. Mỗi khối dữ liệu được mã hóa sẽ được kết hợp (XOR) với khối dữ liệu trước đó trước khi được mã hóa. Khối đầu tiên sẽ được kết hợp với IV.



Ưu điểm:

Cung cấp độ bảo mật cao hơn so với ECB. Mỗi khối dữ liệu phụ thuộc vào khối dữ liệu trước đó, vì vậy nếu có một thay đổi trong một khối, nó sẽ ảnh hưởng đến tất cả các khối tiếp theo.

Giảm thiểu khả năng lặp lại mã hóa cho các khối dữ liệu giống nhau.

Nhược điểm:

Cần phải quản lý IV, và nếu IV không được sử dụng đúng cách, nó có thể làm giảm bảo mật.

Tốc độ mã hóa chậm hơn một chút so với ECB do sự phụ thuộc vào khối trước.

6. IV trong mã hóa khối được dùng như thế nào?

IV trong mã hóa khối là vec-tơ khởi tạo ban đầu có chiều dài bằng với kích thước của block cipher, IV được dùng để XOR với thuật toán mã hóa hoặc với plaintext để tạo ra ciphertext ban đầu

7. Padding Mode trong mã hóa khối là gì? Tại sao cần Padding Mode? PKCS7 và ISO 10126 khác nhau như thế nào?

Padding Mode là một phương pháp dùng những bit đ để bù vào các plaintext có một vài bit sao cho plaintext có độ dài đúng. Bởi vì độ dài plaintext phải là bội số của độ dài quy định của block cipher thì mới có thể tiến hành đưa vào trong block cipher, vậy nên trong trường hợp plaintext có độ dài không bằng bội số của chiều dài của block cipher, ta phải cần padding mode để đệm thêm các bit còn thiếu vào cuối plaintext sao cho plaintext có độ dài phù hợp.

PKCS 7 và ISO 10126 là hai phương pháp khác nhau để thực hiện quy trình bổ sung (padding) trong mã hóa khối.

PKCS7: thêm một số byte vào cuối dữ liệu sao cho tổng số byte của dữ liệu là bội số của kích thước khối, nó có ưu điểm là dễ dàng để giải mã và loại bỏ padding, vì mình có thể biết chính xác số byte padđing dựa trên giá trị của byte cuối cùng. (tất cả các byte padding đều giống nhau)

ISO 10126: phương pháp này sẽ thêm một số byte để đạt được độ dài khối cần thiết nhưng các byte padding được chọn ngẫu nhiên và byte cuối cùng sẽ chứa số lượng byte padding. Nó có ưu điểm là làm tăng tính bảo mật trong một số tình huống vì nó làm cho mẫu dữ liệu trở nên khó đoán hơn. Tuy nhiên việc loại bỏ padding có thể trở nên phức tạp hơn, vì cần phải biết chính xác cách thức mà byte padding đã được thêm vào để loại bỏ chính xác.

8. Viết chương trình Go đơn giản thực hiện mã hóa/giải mã AES (Cipher mode CBC, Padding mode PKCS7) tập tin. Chương trình nhận vào các tham số:

- mode: encrypt/decrypt

- input: đường dẫn đến tập tin đầu vào

- output: đường đẫn đến tập tin đầu ra

**Done**

9. Dùng openssl generate ra một cặp public key/private key. Viết chương trình Go đơn giản thực hiện mã hóa/giải mã RSA tập tin. Chương trình nhận vào các tham số:

- mode: encrypt/decrypt

- input: đường dẫn đến tập tin đầu vào

- output: đường đẫn đến tập tin đầu ra

10. Dùng chương trình viết ở câu 8 và câu 9 mã hóa cùng một tập tin có kích thước > 200MB. So sánh tốc độ và lượng CPU tiêu tốn.

11. Chữ ký điện tử là gì? Chữ ký điện tử có các thành phần cơ bản nào?

Chữ ký điện tử (Digital Signature) là một phương pháp xác thực số được sử dụng để đảm bảo tính toàn vẹn, xác thực và không thể chối cãi của tài liệu điện tử hoặc thông điệp điện tử. Chữ ký điện tử sử dụng các thuật toán mã hóa để tạo ra một mã số duy nhất cho tài liệu, cho phép người nhận xác nhận rằng tài liệu không bị thay đổi và xác định được người gửi.

Chữ ký điện tử thường bao gồm các thành phần cơ bản sau:

Khóa riêng (Private Key):

Đây là một phần của cặp khóa trong hệ thống mã hóa bất đối xứng. Khóa riêng được giữ bí mật bởi người ký và được sử dụng để tạo chữ ký điện tử. Chữ ký này chỉ có thể được tạo ra bằng khóa riêng tương ứng với khóa công khai.

Khóa công khai (Public Key):

Đây là phần còn lại của cặp khóa, được phân phối công khai. Người nhận sử dụng khóa công khai để xác thực chữ ký điện tử. Khóa công khai không cần phải được bảo mật và có thể được chia sẻ với bất kỳ ai.

Tài liệu hoặc thông điệp (Document or Message):

Đây là dữ liệu mà người ký muốn ký. Tài liệu có thể là một văn bản, hợp đồng điện tử, email, hoặc bất kỳ loại dữ liệu nào khác.

Thuật toán băm (Hashing Algorithm):

Trước khi ký, tài liệu được xử lý qua một thuật toán băm (như SHA-256) để tạo ra một giá trị băm (hash value) duy nhất cho tài liệu đó. Giá trị băm này đại diện cho nội dung của tài liệu mà không thể đảo ngược. Việc sử dụng thuật toán băm giúp tiết kiệm không gian và thời gian, vì chỉ cần ký giá trị băm thay vì toàn bộ tài liệu.

Chữ ký điện tử (Digital Signature):

Chữ ký điện tử được tạo ra bằng cách mã hóa giá trị băm của tài liệu bằng khóa riêng của người ký. Kết quả là một chuỗi mã hóa (chữ ký) mà người nhận có thể sử dụng để xác thực tài liệu.

12. Chữ ký điện tử được sử dụng thế nào trong việc bảo đảm an toàn cho đường truyền.

Chữ ký điện tử là một công cụ quan trọng trong việc bảo đảm an toàn cho đường truyền thông tin trong môi trường kỹ thuật số. Nó giúp xác thực người gửi, đảm bảo tính toàn vẹn của dữ liệu, cung cấp tính không thể chối cãi, và bảo vệ thông tin nhạy cảm. Khi được kết hợp với các công nghệ bảo mật khác, chữ ký điện tử tạo ra một môi trường truyền thông an toàn và đáng tin cậy.

13. Viết 2 chương trình Go đơn giản A (server) và B (client) làm nhiệm vụ:

- A và B liên lạc thông qua gRPC, xác thực 02 đầu kết nối bằng certificate

- B nhận tham số input là đường dẫn của tập tin cần tải về từ A, A khi nhận được request trả về cho B tập tin.

**Done**

# Tương tác CSDL, API, Authentication

(Trên tư tưởng của ứng dụng Everything - cho phép tìm kiếm mọi thứ trên máy tính một cách nhanh chóng) Hãy viết 2 service FileIndex và FileSearch làm các nhiệm vụ:

### FileIndex (GRPC Service)

1. Liệt kê tất cả tập tin trong máy tính, thu thập các thông tin sau ghi vào CSDL (quan hệ hoặc không quan hệ): đường dẫn đầy đủ của tập tin, tên tập tin, phần mở rộng của tập tin, kích thước tập tin, ngày tạo, ngày sửa đổi, ngày truy cập cuối, các attributes (Read Only, Hidden). Nếu là một trong các loại tập tin Word, Excel, Powerpoint thì lấy thêm nội dung của tập tin (nếu tập tin không đặt mật khẩu).

2. Việc thu thập diễn ra liên tục nhằm bảo đảm khi có người dùng có thao tác mới trên hệ thống tập tin, nó đều được ánh xạ vào CSDL đã xây dựng.

3. Tạo method GetFileChecksum => input: []string{filepath}, output: map[string][string]{filepath: file checksum}

### FileSearch (Rest API Service)

1. Api login: trả về jwt để xác thực cho các api khác

a. Input: body json

{

    "username": "admin",

    "password": "admin"

}

b. Output:

{

    "status": 200,

    "error": "",

    "errors": **null**,

    "data": {

        "access\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjAsImlhdCI6MTY5OTg0MzU5NiwiaXBfYWRkcmVzcyI6IjEyNy4wLjAuMSIsImlzcyI6IiIsImp0aSI6IlVteWtOc2RxMnJjL2VWRUhxRkJGVEE9PSIsIm5iZiI6MTY5OTg0MzU5Niwic2Vzc2lvbl9pZCI6MjIsInVzZXJfYWdlbnQiOnsicGxhdGZvcm0iOiIiLCJvcyI6IiIsImVuZ2luZSI6IiIsImVuZ2luZV92ZXJzaW9uIjoiIiwiYnJvd3NlciI6IlBvc3RtYW5SdW50aW1lIiwiYnJvd3Nlcl92ZXJzaW9uIjoiNy4zNC4wIn0sInVzZXJfaWQiOjJ9.STdg4lK0DRyYKZ3HleM6w0LTgZkx4D0qbXv6ZBrsvDw"

    }

}

🡺 access\_token dùng để xác thực cho các api phía sau

2. Cung cấp Rest API cho phép tìm kiếm tập tin theo các tiêu chí: tên tập tin (tìm kiếm gần giống), phần mở rộng (tìm kiếm chính xác), kích thước tập tin (tìm kiếm trong khoảng), ngày tạo (tìm kiếm trong khoảng), ngày sửa đổi (tìm kiếm trong khoảng), ngày truy cập cuối (tìm kiếm trong khoảng), nội dung tập tin (tìm kiếm gần giống). Tìm kiếm kết hợp các điều kiện AND, OR. Kết quả trả về là danh sách gồm đường dẫn tập tin thỏa tiêu chí tìm kiếm và checksum của tập tin. (thông tin checksum thì call method GetFileChecksum của FileIndex GRPC Service để lấy thông tin có xác thực bằng certificate)

Output khi thành công

{

    "status": 200,

    "data":   [

        {

        "filepath": "",

        "checksum": ""

        }

    ],

    "meta": {

        "total":  500,

        "offset": 0,

        "limit":  10 // Nếu limit = 0 thì trả về tất cả record trong db

    }

}

3. Tạo table user

type User struct {

Id int `json:"id" orm:"column(id)"`

Email string `json:"email" orm:"column(email)" validate:"required"`

Username string `json:"username" orm:"column(username)" validate:"required"`

Password string `json:"password" orm:"-"`

PasswordHash string `json:"-" orm:"column(password)"`

Phone string `json:"phone" orm:"column(phone)"`

FullName string `json:"full\_name" orm:"column(full\_name)"`

Avatar string `json:"avatar" orm:"column(avatar)"`

State UserState `json:"state" orm:"column(state)"`

Role string `json:"role" orm:"column(role)"`

CreateAt time.Time `json:"create\_at" orm:"column(create\_at);auto\_now\_add"`

UpdateAt time.Time `json:"update\_at" orm:"column(update\_at);auto\_now"`

}

type UserState = uint8

const (

UserStateActive UserState = iota

UserStateBanned

UserStateDeleted

)

4. Api tạo mới user: mã hóa password dùng SHA256 + bcrypt, username tạo mới có giá trị unique trong những user có state khác delete đã tồn tại trong database.

5. Api update user:

6. Api delete user: chuyển state sang delete

7. Api get user by id

8. Api get list user

a. Input: hỗ trợ

- tìm kiếm gần giống (ilike) theo username và fullname

- tìm kiếm chính xác theo state

- limit, offset

- sắp xếp kết quả trả về theo nhiều cột với 2 param:

+orderby=id, username, …

+order=asc, desc, …

b. Output

{

    "status": 200,

    "data":   []User, // array of user object

    "meta": {

        "total":  500,

        "offset": 0,

        "limit":  10 // Nếu limit = 0 thì trả về tất cả record trong db

    }

}

9. Phân quyền user theo 2 role admin và operator (phân quyền theo mô hình rbac dùng thư viện <https://github.com/casbin/casbin>).

a. Admin: thực hiện được tất cả các api nêu trên

b. Operator: chỉ thực hiện được chức năng của api get thông tin của file, không có quyền thực hiện các api quản lý user.

10. Tốc độ tìm kiếm nhanh, kết quả chính xác, hỗ trợ tìm kiếm tiếng Việt.

# Triển khai

Đóng gói 2 service FileIndex, FileSearch; triển khai bằng Docker