# 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 khởi tạo các processes từ các ứng dụng.

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

Session là phiên bản làm việc của process đó. Session 0 là session mà process của hệ thống đang hoạt động mà người dung không có quyền để tác động vào process đó.

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 view -> lower view panel -> Threads, 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.

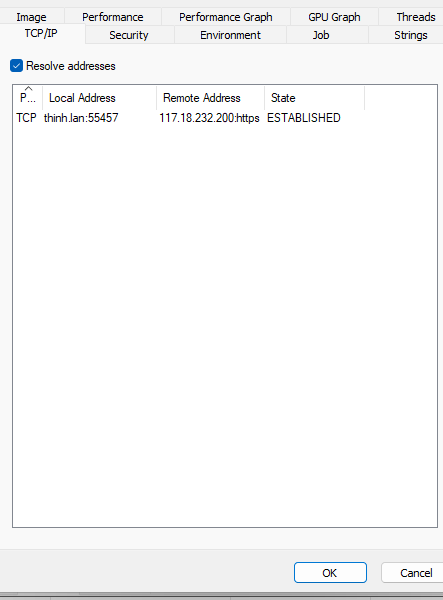
5. Cách xem các kết nối (Connections) của một Process.

Chúng ta nhấn chuột phải vào một process bất kỳ 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 incoming/outcoming đến process đã được chỉ định.

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.

****

****

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.

Nhấn vào view -> lower view pane -> DLLs, ở phía dưới sẽ hiển thị lên các tab như Handles, DLLs, Threads, chúng ta có thể bấm vào một process bất kỳ trên cột Process và nhấn vào tab DLLs ở phía dưới thì sẽ xem được các Modules đang được nạp trong 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?

Nhấn vào view -> lower view pane -> DLLs, ở phía dưới sẽ hiển thị lên các tab như Handles, DLLs, Threads, chúng ta có thể bấm vào một process bất kỳ trên cột Process và nhấn vào tab Handles thì sẽ xem được các đối tượng (tập tin, regisreykey, thread, token) thuộc Process đó.

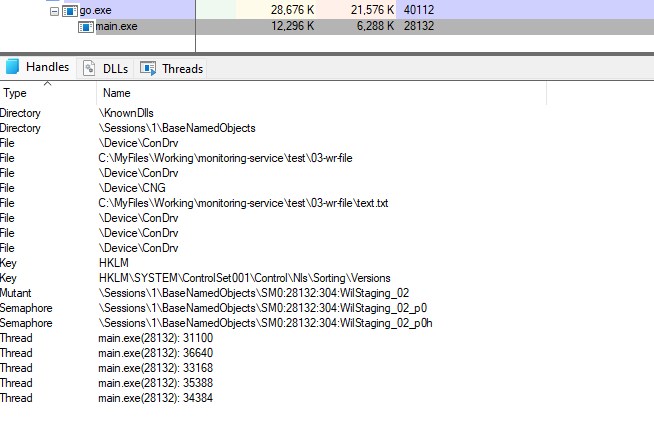
Trong trường hợp một Process giữ quá nhiều Handle sẽ gây ra các vấn đề:

- Mỗi handle đều tiêu tốn tài nguyên của hệ thống, nếu quá nhiều handle được mở mà không được giải phóng, điều này có thể gây ra tình trạng thiếu tài nguyên của hệ thống -> chậm hiệu suất của máy tính.

- Mỗi service có giới hạn về số lượng handle mà các process có thể sử dụng. Nếu quá nhiều handle được mở, service có thể không mở thêm được handle cho các process mới, dẫn đến lỗi.

- 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 Process là địa chỉ bộ nhớ mà hệ điều hành cấp phát để lưu trữ toàn bộ không gian địa chỉ của một tiến trình. Base Address giống như một điểm khởi đầu của mã thực thi của process đó trong bộ nhớ

Base Address của Module là địa chỉ bộ nhớ nơi các module (file .DLL hoặc file thực thi .exe) được load vào trong không gian bộ nhớ của process. Một process có thể chứa nhiều module, và mỗi module sẽ có một base address riêng.

Ví dụ nếu một file DLL được nạp vào process, thì base address của nó sẽ là địa chỉ mà DLL đó bắt đầu trong không gian địa chỉ của process.

Chúng ta có thể xem Base Address của Process bằng cách nhấp chuột phải vào tiến trình đó và chọn Properties. Trong tab Image, bạn sẽ thấy thông tin liên quan đến địa chỉ image base của tiến trình. Đây chính là Base Address của process.

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

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?

Cách 1: Chúng ta có thể nhấn Crtl+F để tìm nhanh tên của Process đó.

Cách 2: Trên thanh công cụ của Process Explorer, chúng ta sẽ thấy biểu tượng "Find Window's Process”

Kéo biểu tượng này và thả nó lên cửa sổ mà bạn muốn kiểm tra (ví dụ: cửa sổ Microsoft Word). Process Explorer sẽ tự động chọn và highlight process liên quan đến cửa sổ đó trong danh sách tiến trình.

### Process Monitor (ProcMon)

1. Chương trình ProcMon dùng để làm gì?

Truy cập file hệ thống: Ghi lại các hoạt động như mở, đọc, ghi, xóa file.

Tương tác registry: Theo dõi các thay đổi trong registry (key nào được truy cập, tạo, xóa, sửa đổi).

Tạo hoặc kết thúc process: Theo dõi việc khởi động và kết thúc tiến trình.

Hoạt động mạng: Ghi lại một số thông tin về kết nối mạng của tiến trình.

Phân tích phần mềm độc hại: Theo dõi hành vi của phần mềm độc hại và phát hiện sớm những thay đổi mà phần mềm độc hại đang tác động đến hệ thống.

Theo dõi chuỗi sự kiện của tiến trình.

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

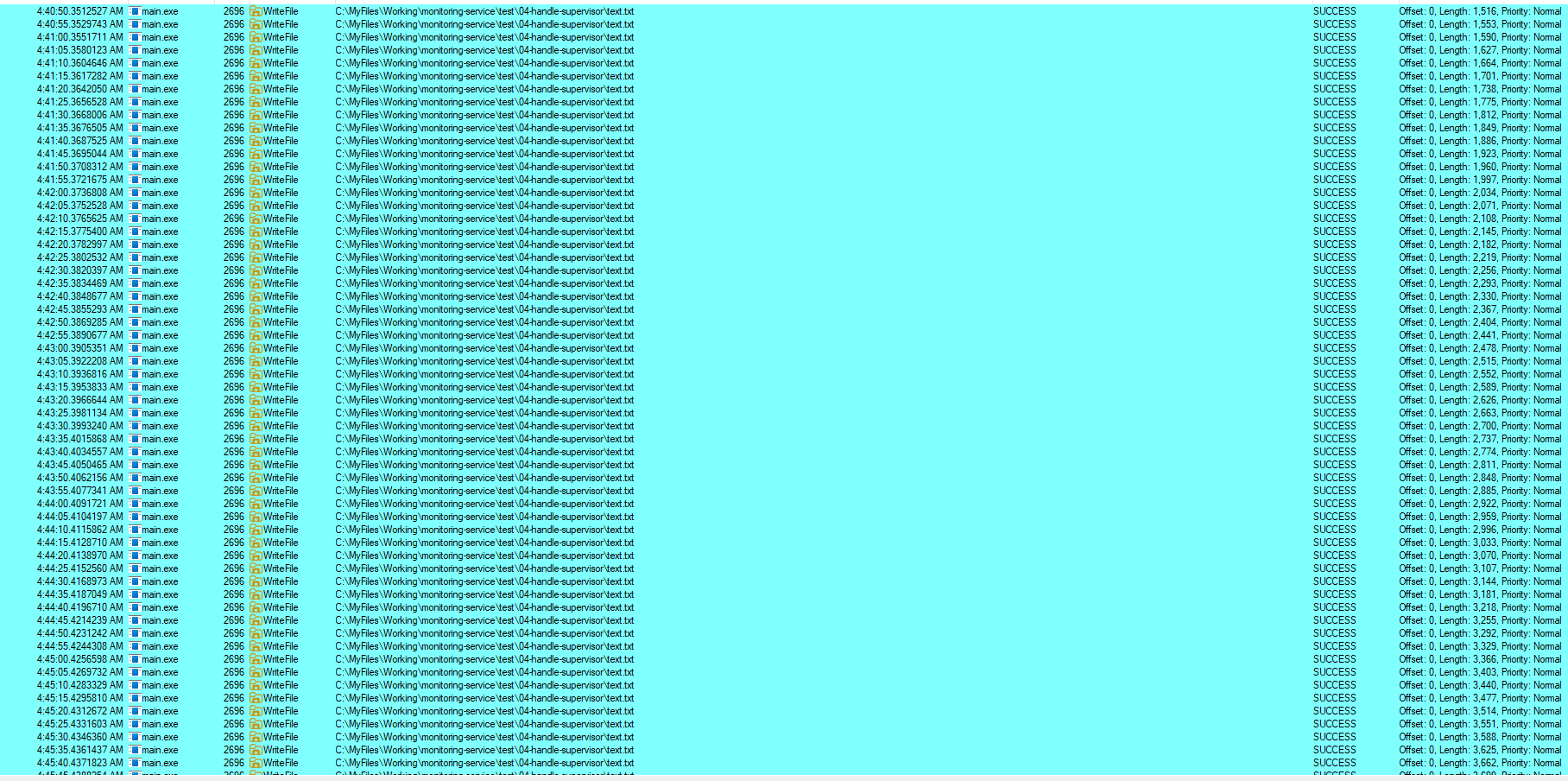
- Hoạt động với File System (open, read, write, create, delete, rename, close, ...)

- Hoạt động Registry (Open key, Create key, Set Value, Delete Key/Value, ...)

- Hoạt động mạng: theo dõi kết nối TCP/IP của process và hiển thị IP hoặc Port mà process đang sử dụng

- Kết quả của các events (success, access denied, file not found, ...)

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.

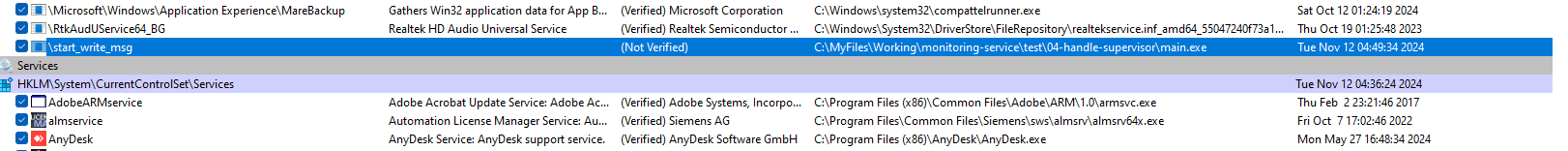
****

### Autoruns

1. Chương trình Autoruns dùng để làm gì?

Chương trình Autoruns dùng để kiểm tra các process nào đang tự động 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?

Trong phần khởi động bằng services có rất nhiều tập tin .DLL không thể hoạt động riêng lẻ, chương trình Service Control Manager (SCM) đã nạp những tập tin này. Đây là service quản lý tất cả các service của hệ thống khởi động cùng Windows. SCM kiểm tra các khóa trong HKLM\System\CurrentControlSet\Services để biết các dịch vụ nào cần khởi động và nạp các DLL tương ứng.

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

* Một số tập tin DLL có thể chạy với quyền hệ thống, nên nếu một tập tin DLL độc hại được nạp với đặc quyền này, nó có thể có toàn quyền điều khiển hệ thống và truy cập dữ liệu nhạy cảm.
* Nếu một tập tin DLL độc hại được đăng ký vào khóa HKLM\System\CurrentControlSet\Services, nó có thể tự động khởi động mỗi khi hệ thống được bật lên. Điều này làm mã độc chạy liên tục mà người dùng khó phát hiện.
* Các tập tin DLL không hiển thị trong Task Manager như các tiến trình độc lập, khiến việc phát hiện mã độc chạy dưới dạng DLL trở nên khó khăn hơn.

### TcpView và Wireshark

1. Chương trình TcpView dùng để làm gì?

Chương trình TcpView dùng để list toàn bộ những kết nối của các process đến với các IPAddress và Port, bao gồm các local address, local port, remote port 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. (thông thường socker này thuộc về server hoặc một process đang chờ yêu cầu kết nối đến).

Khi một client gửi yêu cầu kết nối, server sẽ chuyển trạng thái từ LISTENING sang SYN\_RECEIVED, và sau đó là ESTABLISHED nếu thành công.

Trạng thái SYD\_SENT là trạng thái một process (client) đã gửi yêu cầu kết nối (SYN packet) đến một máy tính khác (server). Nếu server phản hồi bằng gói tin SYN-ACK, trạng thái sẽ chuyển thành ESTABLISHED, còn nếu không nhận được phản hồi, kết nối có thể bị hủy hoặc timeout

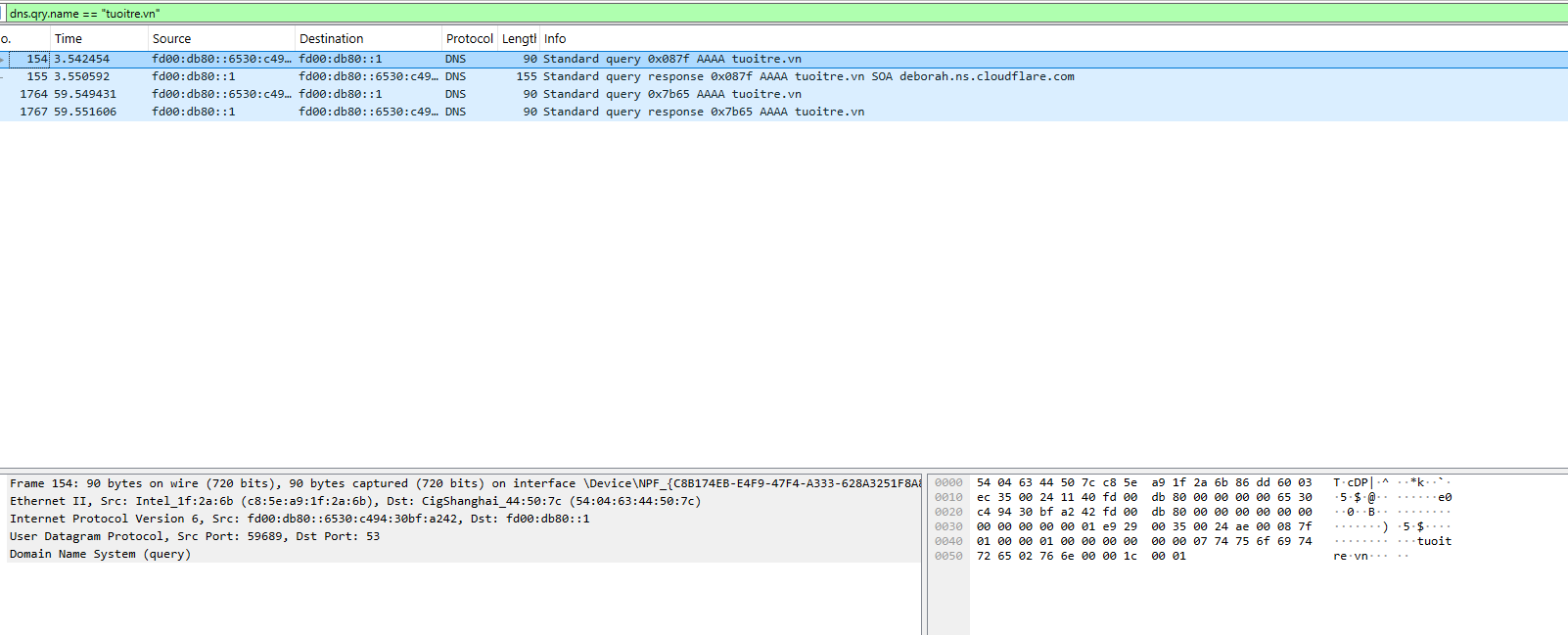
3. Command nào trên Windows có tác dụng tương ứng với TcpView?

**Netstat**

|  |  |
| --- | --- |
| **Lệnh** | **Mô tả** |
| netstat | Hiển thị danh sách các kết nối TCP hiện tại. |
| netstat -a | Hiển thị tất cả các kết nối TCP và UDP, bao gồm cả các cổng đang **LISTENING**. |
| netstat -n | |  | | --- | |  |  |  | | --- | | Hiển thị các địa chỉ và cổng dưới dạng số, thay vì dịch tên DNS. | |
| netstat -o | Hiển thị thêm **PID** (Process ID) của tiến trình tương ứng với kết nối. |
| netstat -b | Hiển thị các tiến trình hoặc ứng dụng đang sử dụng các kết nối. (**Yêu cầu quyền Administrator**). |
| netstat -an | |  | | --- | |  |  |  | | --- | | Kết hợp hiển thị tất cả kết nối dưới dạng số, cùng trạng thái TCP/UDP. | |
| netstat -ano | Hiển thị danh sách tất cả kết nối kèm **PID**, địa chỉ IP, và trạng thái TCP/UDP dưới dạng số. |
| netstat -r | Hiển thị bảng định tuyến (Routing Table). |

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.



Kết quả trong Wireshark trả về khi phân giải tên miền của tuoitre.vn bao gồm 2 gói tin:

Gói DNS Query:

Gói tin gửi từ máy tính của bạn đến máy chủ DNS để yêu cầu phân giải tên miền tuoitre.vn.

Gói DNS Response:

Gói tin từ máy chủ DNS trả về với địa chỉ IP của 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**

**Kết quả so sánh 2 lần sniff 2 URL phía trên:**



**Bộ lọc DNS:**

* tuoitre.vn có thể sử dụng nhiều máy chủ để phân phối nội dung, nên DNS có thể trả về nhiều địa chỉ IP hơn.
* Chinhphu.vn trả về ít địa chỉ IP hơn, vì đây là trang chính phủ.

**Giao thức:**

Kiểm tra giao thức sử dụng khi truy cập hai trang:

* Tuoitre.vn: Dùng TLS1.3.
* Chinhphu.vn: Dùng TLS1.2.

**Kích thước dữ liệu tải về**

* Tuoitre.vn thường tải nhiều tài nguyên động (hình ảnh, quảng cáo, JS).
* Chinhphu.vn tải ít tài nguyên hơn, chủ yếu là nội dung tĩnh.

**Nội dung được mã hóa**

* Tuoitre.vn có thể sử dụng nhiều gói tin TLS hơn vì tải nhiều tài nguyên động.
* Chinhphu.vn ít gói TLS hơn, tập trung vào nội dung tĩnh.

**Tên miền phụ**

* Tuoitre.vn có thể truy cập các miền phụ như quangcao.tuoitre.vn, media.tuoitre.vn.
* Chinhphu.vn chủ yếu truy cập miền chính.

**Kết quả:**

Tuoitre.vn có lưu lượng mạng phức tạp hơn, bao gồm nhiều miền phụ, nhiều kết nối TCP, và dữ liệu tải về lớn hơn do sử dụng hình ảnh, video, và quảng cáo.

Chinhphu.vn tập trung vào nội dung tĩnh, tải ít tài nguyên hơn, và ít kết nối mạng hơn.

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

Tấn công Man-in-the-Middle (MitM): Phương pháp MITM yêu cầu bạn làm trung gian giữa máy khách và máy chủ. Bạn sẽ thay đổi chứng chỉ SSL của máy chủ bằng chứng chỉ của bạn để giải mã và phân tích dữ liệu.

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.

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:

* **Truy vấn DNS**: Khi người dùng gõ một tên miền (ví dụ: tuoitre.vn) vào trình duyệt, hệ thống DNS trên máy tính của người dùng sẽ bắt đầu tìm kiếm địa chỉ IP tương ứng của tên miền đó.
* **Cache DNS**: Trước tiên, hệ thống sẽ kiểm tra bộ nhớ cache của DNS trên máy tính hoặc trong trình duyệt. Nếu địa chỉ IP của tên miền đã được lưu trữ trước đó, thì quá trình phân giải sẽ kết thúc ở đây mà không cần truy vấn thêm. Tuy nhiên, nếu không có sẵn trong bộ nhớ cache, yêu cầu sẽ được gửi tới máy chủ DNS của nhà cung cấp dịch vụ Internet (ISP). Đây là máy chủ DNS chịu trách nhiệm phân giải các tên miền phổ biến.
* **Truy vấn DNS Recursive**: Nếu máy chủ DNS của ISP không biết địa chỉ IP của tên miền, nó sẽ thực hiện một truy vấn gọi là recursive query. Máy chủ này sẽ liên lạc với một loạt các máy chủ DNS cấp cao hơn, bắt đầu từ máy chủ DNS root.
* **Máy chủ DNS Root**: Máy chủ DNS root là máy chủ cấp cao nhất trong hệ thống DNS, và có thể định hướng yêu cầu đến máy chủ DNS của các TLD (Top-Level Domain), ví dụ như .com, .vn, .org...
* **Máy chủ DNS của TLD (Top-Level Domain):** Sau khi yêu cầu đến máy chủ root, hệ thống sẽ tiếp tục yêu cầu tới các máy chủ DNS của TLD (ví dụ: .com, .vn). Các máy chủ TLD này sẽ hướng dẫn truy vấn tiếp theo đến máy chủ DNS của tên miền cụ thể, chẳng hạn như tuoitre.vn.
* **Máy chủ DNS Authoritative:** Cuối cùng, máy chủ authoritative DNS sẽ cung cấp kết quả cuối cùng về địa chỉ IP của tên miền. Máy chủ này chứa bản sao dữ liệu của tên miền và sẽ trả về địa chỉ IP chính xác (ví dụ: 203.119.8.111).
* **Cập nhật bộ nhớ cache**: Sau khi nhận được địa chỉ IP từ máy chủ authoritative, kết quả sẽ được lưu vào bộ nhớ cache DNS của hệ thống (máy tính người dùng, máy chủ DNS của ISP) để có thể sử dụng cho các lần truy vấn sau mà không cần phải thực hiện lại toàn bộ quá trình.
* **Kết quả trả về cho người dùng:** Trình duyệt sẽ nhận địa chỉ IP và sử dụng nó để kết nối với máy chủ web và tải nội dung của trang web. Khi kết nối thành công, người dùng sẽ thấy trang web được hiển thị.

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

Route (hay còn gọi là "định tuyến") trong mạng máy tính là quá trình xác định và chỉ định các tuyến đường (hoặc lộ trình) mà dữ liệu (gói tin) phải đi qua từ nguồn đến đích trên một mạng. Định tuyến có thể xảy ra trong mạng cục bộ (LAN) hoặc mạng diện rộng (WAN), và nó thường được thực hiện bởi các thiết bị như router (bộ định tuyến) hoặc các công cụ phần mềm. Mỗi định tuyến xác định đường đi cụ thể mà dữ liệu cần phải qua để tới địa chỉ đích.

**Cơ chế Hoạt Động của Route:**

**Bước 1: Xác định Địa chỉ Đích**

Mỗi gói tin trong mạng sẽ chứa địa chỉ đích, chính là địa chỉ IP của thiết bị mà nó cần gửi đến. Router sẽ sử dụng địa chỉ đích này để xác định đường đi.

**Bước 2: Tạo Bảng Định Tuyến (Routing Table)**

Mỗi router có một bảng định tuyến, trong đó chứa thông tin về các mạng mà nó có thể tiếp cận và cách thức gửi dữ liệu đến những mạng đó. Bảng định tuyến sẽ có các dòng chứa:

**Địa chỉ đích (Destination IP):** Mạng hoặc thiết bị đích.

**Gateway:** Địa chỉ của router tiếp theo (nếu có).

**Giao thức định tuyến:** Xác định cách thức cập nhật bảng định tuyến (ví dụ: OSPF, BGP, RIP).

**Interface:** Cổng mà router sẽ gửi gói tin.

**Bước 3: Tìm kiếm trong Bảng Định Tuyến**

Khi một router nhận được một gói tin, nó sẽ kiểm tra địa chỉ đích của gói tin và tìm kiếm trong bảng định tuyến của nó. Nếu có một tuyến đường trực tiếp đến đích, router sẽ chuyển gói tin tới thiết bị đích. Nếu không, gói tin sẽ được chuyển tới một router khác (gateway) theo lộ trình phù hợp.

**Bước 4: Định Tuyến lại hoặc Tiếp Tục Đến Đích**

Nếu router không phải là đích cuối, nó sẽ tiếp tục chuyển tiếp gói tin đến router tiếp theo trong chuỗi. Quá trình này sẽ lặp lại cho đến khi gói tin đến đích cuối cùng.

**Bước 5: Đến Đích**

Khi gói tin đến được router có cổng đầu ra trực tiếp đến thiết bị đích (hoặc mạng đích), nó sẽ được chuyển đến thiết bị đó. Nếu gói tin là gói tin ICMP (ping), nó sẽ tạo phản hồi đến nguồn.

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

**Gửi Yêu Cầu DNS**

Khi nhập tuoitre.vn vào thanh địa chỉ của trình duyệt và nhấn Enter, trình duyệt sẽ cần phải biết địa chỉ IP của website đó. Vì vậy, nó sẽ thực hiện một yêu cầu DNS (Domain Name System) để phân giải tên miền tuoitre.vn thành một địa chỉ IP cụ thể.

**Kết Nối TCP/IP**

Khi địa chỉ IP được biết đến, trình duyệt sẽ bắt đầu thiết lập kết nối TCP với máy chủ lưu trữ website (máy chủ web) thông qua cổng 80 (HTTP) hoặc 443 (HTTPS).

**Gửi Yêu Cầu HTTP/HTTPS**

Khi kết nối TCP đã được thiết lập, trình duyệt gửi một yêu cầu HTTP hoặc HTTPS đến máy chủ web. Yêu cầu này bao gồm các thông tin như: **phương thức HTTP, URL, Header**

**Máy Chủ Web Xử Lý Yêu Cầu**

Máy chủ web nhận yêu cầu và bắt đầu xử lý nó. Nếu trang web sử dụng HTTPS, máy chủ sẽ giải mã yêu cầu trước khi xử lý. Sau khi xử lý, máy chủ sẽ gửi một phản hồi HTTP/HTTPS lại cho trình duyệt, bao gồm: Status Code HTTP, Header, Body của trang web, thường là mã HTML.

**Trình Duyệt Nhận và Xử Lý Phản Hồi**

Khi nhận được phản hồi từ máy chủ web, trình duyệt sẽ bắt đầu xử lý mã HTML trong body của phản hồi:

Trình duyệt sẽ phân tích cú pháp HTML, xây dựng DOM (Document Object Model) của trang.

Trình duyệt sẽ tải thêm tài nguyên cần thiết, chẳng hạn như CSS, JavaScript, hình ảnh, video, v.v., mà trang web yêu cầu.

**Tải và Hiển Thị Trang Web**

Sau khi tải xong tất cả các tài nguyên cần thiết, trang web sẽ được trình duyệt render (hiển thị) cho người dùng.

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

Firewall (Tường lửa) là một hệ thống bảo mật mạng dùng để giám sát và kiểm soát lưu lượng mạng vào và ra khỏi một hệ thống hoặc mạng. Firewall có thể được triển khai dưới dạng phần mềm hoặc phần cứng và có nhiệm vụ bảo vệ hệ thống khỏi các cuộc tấn công hoặc truy cập trái phép, đồng thời cho phép hoặc từ chối các kết nối dựa trên các quy tắc bảo mật đã được xác định trước.

Các tính năng của Firewall trên windows bao gồm:

+ Giám sát và kiểm soát lưu lượng mạng

+ Quy tắc định tuyến. (Inbound và Outbound)

+ Chặn kết nối không an toàn.

+ Theo dõi và ghi log.

+ Cấu hình từ xa.

+ Tích hợp với Windows Defender Virus.

+ 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.

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?

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Mã hóa đối xứng** | **Mã hóa bất đối xứng** |
| Khái niệm | Sử dụng một khóa duy nhất cho cả quá trình mã hóa và giải mã. | Sử dụng một cặp khóa: khóa công khai để mã hóa và khóa riêng để giải mã. |
| Hiệu suất | Thường nhanh hơn do thuật toán đơn giản. | Chậm hơn do tính toán phức tạp hơn |
| Bảo mật | Ít an toàn hơn nếu khóa bị lộ, vì cùng một khóa được dùng cho cả mã hóa và giải mã. | An toàn hơn do hai khóa khác nhau; khóa công khai không thể dùng để suy ra khóa riêng. |
| Quản lý khóa | Khó khăn trong việc phân phối khóa ban đầu; yêu cầu kênh bảo mật để truyền khóa. | Dễ dàng phân phối khóa công khai mà không cần kênh bảo mật. |
| Ứng dụng | Thường dùng trong mã hóa dữ liệu lớn hoặc trong các hệ thống cần tốc độ cao, như AES trong lưu trữ đám mây và nhắn tin an toàn | Thường dùng trong các hệ thống yêu cầu tính bảo mật cao, như trao đổi khóa và chữ ký số, ví dụ RSA. |

4. Mã hóa đối xứng có những dạng nào?

**Mã hóa dòng** (Stream Cipher): Mã hóa dữ liệu theo từng bit hoặc byte, thích hợp cho dữ liệu liên tục. Ví dụ: RC4, Salsa20.

**Mã hóa khối** (Block Cipher): Mã hóa dữ liệu theo các khối cố định. Ví dụ: AES, DES, Blowfish.

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.

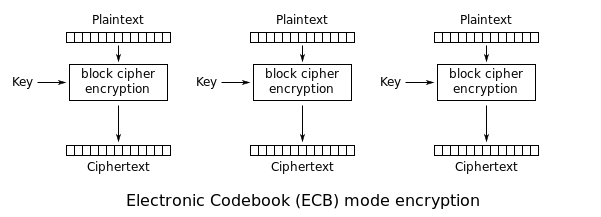
Chúng ta cần Cipher modes vì nó cho phép mã hóa dữ liệu có kích thước lớn hơn kích thước khối của thuật toán mã hóa, điều này giúp cho việc mã hóa các tệp, dữ liệu có kích thước lớn dễ dàng hơn. Đồng thời có một số cipher modes cung cấp tính năng bảo mật cao hơn so với mã hóa đơn giản, giúp giảm thiểu 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:**

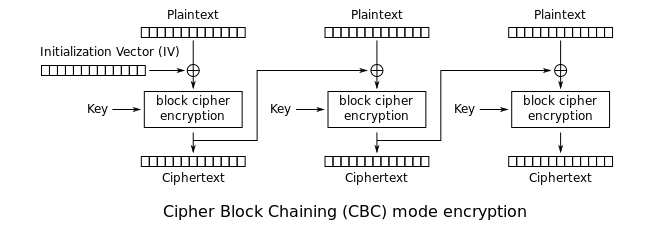
Không an toàn khi sử dụng cho dữ liệu lớn hoặc nhạy cảm. Vì cùng một khối dữ liệu sẽ luôn tạo ra cùng một mã hóa, điều này có thể dẫn đến việc rò rỉ thông tin và tấn công bằng cách phân tích.

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. Kết quả mã hóa của khối đầu tiên sẽ được kết hợp (XOR) với plaintext ở khối tiếp theo trước khi được mã hóa.



**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 (Initialization Vector) trong mã hóa khối được sử dụng để đảm bảo tính ngẫu nhiên và bảo mật khi mã hóa dữ liệu. IV được sử dụng trong các chế độ hoạt động của mã hóa khối như CBC (Cipher Block Chaining) hoặc CTR (Counter Mode) để tạo sự biến đổi cho mỗi khối dữ liệu mã hóa. IV cần phải là ngẫu nhiên và không lặp lại cho mỗi lần mã hóa, để tránh tấn công theo kiểu "replay" và tăng cường bảo mật.

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 padding 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

**Done**

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. **Private key** 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