1. Lý do chọn đề tài

Mã độc ( Malicious software hay còn gọi là Malware) là một phần mềm được bí mật chèn vào hệ thống nhằm thực hiện các hành vi xấu như chiếm quyền kiểm soát, ăn cắp dữ liệu, mã hóa tống tiền, …

Dọc theo chiều dài phát triển của lịch sử máy tính, những ý tưởng đầu tiên về malware được nhen nhóm từ năm 1949 trong cuốn “Lý thuyết và tổ chức của dữ liệu tự động phức tạp” và được hiện thực hóa vào năm những năm 1950. Năm 1986, lần đầu tiên thuật ngữ “virus” xuất hiện trong một cuốn luận văn Tiến sĩ của Fred Cohen. Tại thời điểm đó, virus vẫn là một chương trình lành tính và đơn giản với mục đích là “Một chương trình có thể lây nhiễm sang các chương trình khác bằng cách sửa đổi chúng để bao gồm một phiên bản, có thể đã tiến hóa, của chính nó”. [1]

Sau hơn 70 năm, từ những chương trình cơ bản, mã độc đã trở thành một ngành công nghiệp đồ sộ và kỳ quặc. Theo The Anti-Phishing Workgroup, mã độc đã lây nhiễm cho một phần ba số máy tính trên thế giới. Theo báo cáo của Cybersecurity Ventures, chỉ tính riêng năm 2021, thiệt hại do mã độc gây ra là 20 tỉ đô và con số này sẽ là 42 tỉ đô trong năm 2024.

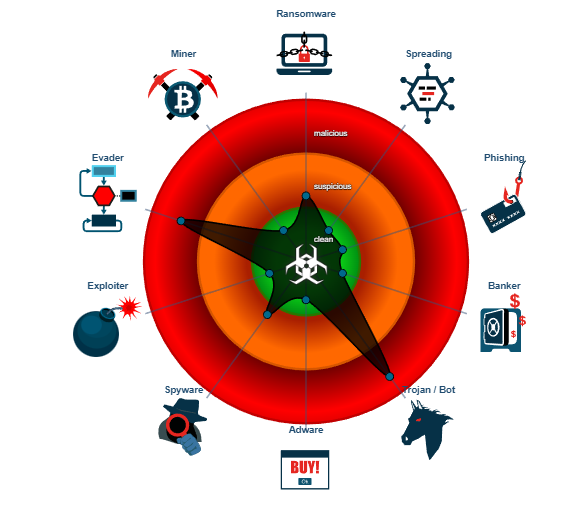
Ảnh có chứa văn bản

Mô tả được tạo tự động

Trước những thiệt hại tỉ đô đó, rất nhiều phần mềm diệt virus ra đời và phát triển song song với sự tiến bộ về công nghệ của mã độc. Nếu như đã có phần mềm diệt virus, mã độc đã bị chặn từ trước khi nó kịp thực thi vậy thì **tại sao ta lại phải phân tích mã độc** ? **Việc phân tích mã độc quan trọng đến mức nào** ?

Ngày nay, ngay cả khi thế giới có đến hàng trăm phần mềm diệt virus lớn nhỏ khác nhau, vẫn tồn tại một số lượng lớn mã độc vượt mặt các phần mềm đó. Không thể nào cải tiến được phần mềm nếu như ta không hiểu rõ về chúng. Từ đó, ta cần tới các malware researcher, các data sciencetist của ngành mã độc.

Đầu tiên, khi kẻ tấn công gửi một mã độc tới máy tính nạn nhân. Bước đầu tiên, ta cần giải quyết bài toán, nó có hay không là mã độc. Kết quả của bài toán chỉ đơn giản là có hoặc không. Sau khi xác định được đó là mã độc, chúng được chuyển tới các malware researcher. Ở đó, mã độc được bóc tách, phân tích xem chúng có những hành vi gì, thực thi như thế nào, thuộc loại mã độc nào, có nằm trong một cuộc tấn công có chủ đích hay không.



Phân loại mã độc

Ảnh có chứa văn bản, trong nhà, kim loại, ảnh chụp màn hình

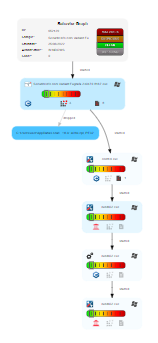
Mô tả được tạo tự động

Ma trận hành vi (MITRE ATT&CK matrix)

Ảnh có chứa văn bản, bản đồ

Mô tả được tạo tự động

Địa chỉ IP kết nối đến (có hay không nằm trong một cuộc tấn công có chủ đích)



Cây thực thi (Tree process)

Trong một bài toán lớn như vậy, đồ án của em tập trung vào đề tài **sử dụng học máy để phát hiện Ma trận hành vi (MITRE ATT&CK matrix).**

Theo the AV-TEST Institute, mỗi ngày có hơn 450000, mã độc mới được phát hiện và 99% trong số đó chỉ được nhìn thấy một lần trước khi được sửa đổi và sử dụng lại. Rất khó cho các malware researcher để phân tích toàn bộ số mã độc đó khi mà mỗi con mã độc lại có một đặc điểm rất riêng. Để bắt kịp với các cuộc tấn công không ngừng như vậy, năm 2015, MITRE ATT&CK ra đời. Đó là một ma trận bao gồm tất cả các hành vi có thể có của một con mã độc.

MITRE ATT&CK bao gồm các bước khác nhau tạo nên chuỗi tấn công - như truy cập ban đầu, leo thang đặc quyền, truy cập thông tin xác thực, di chuyển bên và lọc - cùng với các kỹ thuật có thể được sử dụng trong từng giai đoạn. Bằng cách phân loại các cuộc tấn công thành các phân đoạn cụ thể, các malware researcher sẽ dễ dàng hơn trong việc xem các mẫu, tìm ra kẻ đã phát động các cuộc tấn công khác nhau và theo dõi cách một phần mềm độc hại đã phát triển theo thời gian. Nó đặc biệt quan trọng với các doanh nghiệp để xác định trước các lỗ hổng trong hệ thống phòng thủ mạng của mình, thay vì tìm hiểu về chúng một cách khó khăn - sau mỗi cuộc tấn công.

Bởi vì tầm quan trọng, cùng như tính cấp thiết của việc **tìm ra MITRE ATT&CK** một các **nhanh chóng** và **chính xác**. Em chọn đề tài **ứng dụng học máy để phân tích MITRE ATT&CK của mã độc thông qua lệnh gọi hệ thống ( API call)**

1. Tính mới

* Trong phần lớn các nghiên cứu hiện tại, các bài toán về mã độc chủ yếu xoay quanh 2 bài toán lớn là phát hiện và phân loại theo hành vi. Có khá ít bài toán liên quan đến ứng dụng machine learning trong MITRE ATT&CK
* Các phương pháp gần đây được sử dụng chủ yếu là chuyển behavior sang dạng MIST và sử dụng N-grams. Điểm yếu của phương pháp này là các hành vi ảnh hưởng lẫn nhau bởi thứ tự thời gian và dễ bị bypass khi mã độc cố tình làm các hành vi trở lên lộn xộn. Trong đề tài của bọn em, các hành vi được xét rời rạc và được xếp cạnh nhau nếu có liên quan đến nhau (sẽ giải thích chi tiết hơn ở phần sau)
* Dữ liệu sẽ được cập nhật mới và khác so với dữ liệu ở các nghiên cứu đã có.

1. Cách làm

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Dữ liệu thu thập từ <https://www.joesandbox.com/> (đã hoàn thành)

- Dữ liệu gồm 43849 file của 10962 mã độc. Mỗi mã độc có 4 file json là:

* Signature
* processTree
* Behavior
* MITRE

Cấu trúc tên file là :

<phần trăm là mã độc>.<1 hoặc nhiều, phân loại mã độc>.<hệ điều hành><dạng file>-<tên mã độc><loại file>

Ví dụ :

* + *mal100.phis.troj.spyw.expl.evad.winEXE-7aa19913253d9a036b10df1f8f0bdb25567edda11fe99050d85a47249142bec2-mitre*

100% là mã độc, thuộc 5 loại: phis, troj, spyw, expl, evad

là file có đuôi .exe chạy trên hệ điều hành Windows, có tên 7aa19913253d9a036b10df1f8f0bdb25567edda11fe99050d85a47249142bec2. Trong file có lưu MITRE ATT&CK của mã độc

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Chuẩn hóa dữ liệu (đang làm)

Để train một model ta cần có 2 phần input và output theo cấu trúc



Trong đó:

INPUT: file behavior.json biểu diễn dưới dạng vector

OUTPUT: vector 191 chiều, biểu diễn cho 191 techniques của MITRE ATT&CK[4]  (đã xong)

Công việc

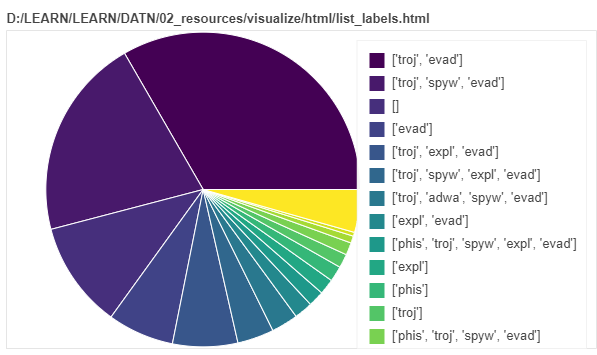
* Chuẩn hóa ma trận từ Joesandbox về ma trận chuẩn mới nhất rồi duỗi ra thành vector (đã xong)
* Lưu vào một file pkl có cấu trúc: tên, vector MITRE (đã xong)
* Chuyển file \_behavior.json sang vector() chưa xong

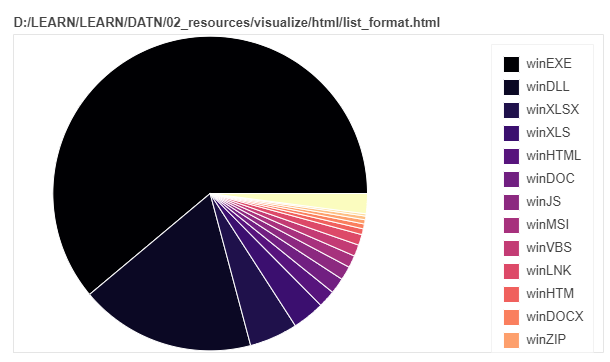
1. Biểu diễn dữ liệu (đang thực hiện)

* Mục đích:

Công việc:

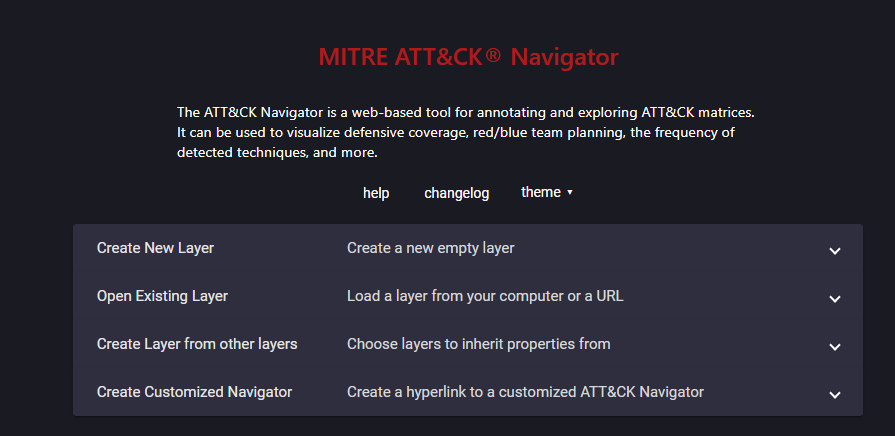
* Biểu diễn thành phần dữ liệu thành các biểu đồ (sẽ làm tiếp sau khi xong các công việc khác)

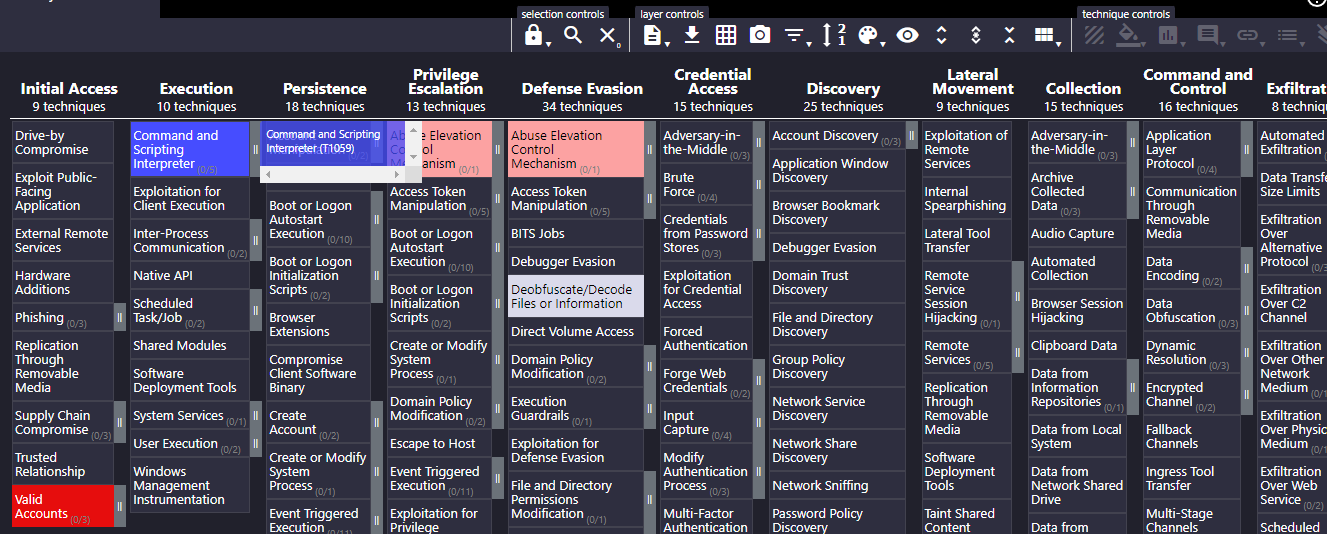




* Biểu diễn output dưới dạng MITRE ATT&CK:

Công cụ online: <https://mitre-attack.github.io/attack-navigator/>





Ảnh có chứa bàn

Mô tả được tạo tự động

Công cụ offline (làm sau)

1. Huấn luyện model (sẽ làm trong tuần tới)

Sử dụng neural network, cấu trúc: MLP

1. Đánh giá kết quả
2. Kết quả hiện tại

Ký hiệu màu:

Xanh lá : đã ổn

Đỏ : cần hoàn thành gấp

Xanh biển : có thời gian thì làm thêm

|  |  |  |
| --- | --- | --- |
| MỤC | Công việc | Tiến độ |
| Câu chuyện cho bài toán |  | Đã xong |
| Crawl dữ liệu |  | Đã xong và tiếp tục crawl thêm 5000 mẫu nữa |
| Huấn luyện mô hình | Chuyển input (behavior) sang vector | Xong trong tuần sau |
| Chuyển output (behavior) sang vector | Đã xong |
| Training model | Làm ngay sau khi chuyển được behavior sang vector |
| Biểu diễn dữ liệu | Biểu diễn dữ liệu tổng hợp | Xong 1 phần. Đánh giá ko quan trọng lắm nên làm cuối cùng |
| Biểu diễn MITRE bằng output của model | Đã xong |
| Biểu diễn MITRE offline | Chưa làm |
| Đánh giá kết quả |  | Chưa làm |
| Demo | Viết chương trình hoàn toàn tự động bằng UWP/WPF | Có thời gian sẽ làm |
| Viết quyển đồ án |  | Chưa làm |

reference

[1]: <https://www.lastline.com/blog/history-of-malware-its-evolution-and-impact/>

[2]: <https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031>

[4]: <https://attack.mitre.org/techniques/enterprise/>