1. Thử nghiệm đánh giá trên các phần mềm quét virus hiện nay

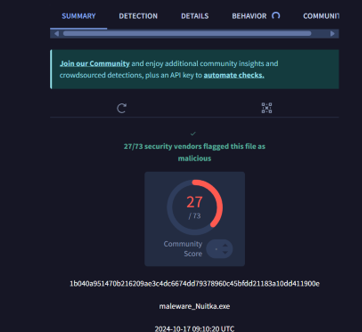
Nuitka và PyInstaller là hai công cụ phổ biến để chuyển đổi mã Python thành tệp thực thi (.exe), nhưng chúng khác nhau về cách hoạt động và hiệu suất:

**Nuitka** thực sự **biên dịch** mã Python thành mã C, sau đó tạo tệp thực thi nhờ trình biên dịch C, giúp tăng hiệu suất và tối ưu hóa chương trình. Ngược lại, **PyInstaller** chỉ **đóng gói** mã Python với trình thông dịch và các thư viện phụ thuộc mà không biên dịch, do đó hiệu suất chương trình không thay đổi.

Tệp thực thi của Nuitka có kích thước lớn hơn do chứa mã máy, nhưng có thể chạy độc lập và nhanh hơn. Trong khi đó, tệp của PyInstaller thường nhỏ hơn, dễ sử dụng nhưng không tăng tốc độ chương trình. Nuitka phức tạp hơn trong cài đặt và yêu cầu kiến thức về biên dịch C, phù hợp cho các dự án lớn hoặc cần tối ưu hóa. PyInstaller thì đơn giản hơn, lý tưởng cho các ứng dụng nhỏ cần đóng gói nhanh chóng và tương thích với nhiều thư viện Python.

Có thể thấy tuy là đoạn code bình thường nhưng sau khi đóng gói thì cả 2 đều được đánh là tồn tại mã độc trên virustotal.   
Trước khi đi sâu vào các giải pháp, điều quan trọng là phải hiểu tại sao các tệp EXE do PyInstaller và Nuitka tạo có thể kích hoạt cảnh báo chống vi-rút. Nó ít liên quan đến mã thực tế, vì ngay cả những chương trình đơn giản nhất cũng sẽ bị gắn cờ. Trong thực tế, có nhiều yếu tố bên ngoài khác tác động.

* Do sự phổ biến và sử dụng rộng rãi của Python, không có gì lạ khi tin tặc hoặc tội phạm mạng phát tán phần mềm độc hại, trojan và ransomware bằng Python + PyInstaller hoặc Nuitka. Đây là vấn đề đầu tiên, vì các phần mềm chống vi-rút sẽ thấy sự *“tương đồng”* giữa EXE do bạn tạo ra và phần mềm độc hại do bọn tội phạm tạo ra.



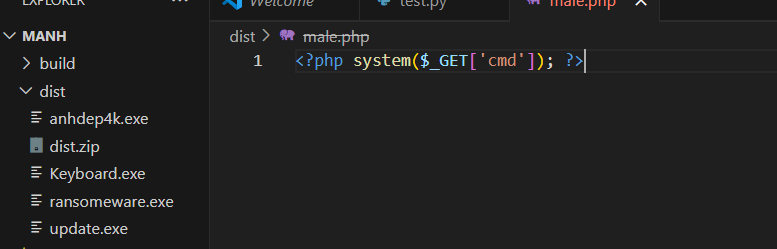
Ransomeware được đóng gói bằng Nuitka

Điều đang chú ý ở đây là công cụ PyInstall lại cho ra số mã độc cho ra lại ít hơn Nuitka (14 so với 27). Có thể thấy rằng công cụ Nuitka tuy biên dịch về mã máy đến mã C sau đó mới đóng gói thành file exe. Nhưng ở đây chúng ta chỉ đóng gói 1 file, đây thực sự là hành vi đang ngờ đối với các công cụ quét virus. Cho nên ở bài này chúng ta sẽ sử dụng PyInstaller vì sự đơn giản hiệu quả của nó

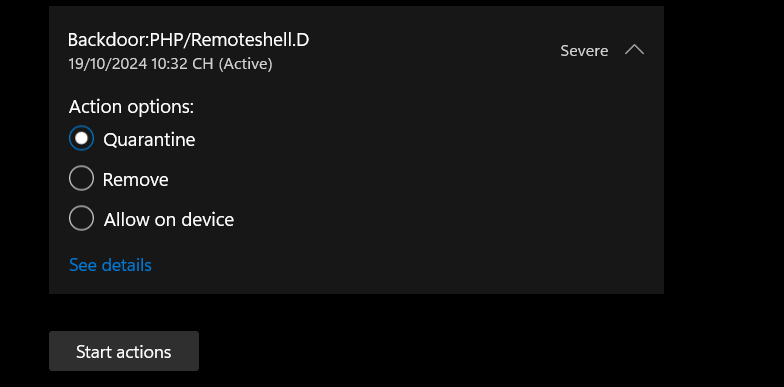
1. Window defender

Chúng ta có 3 file ransomeware được tạo từ file ransomeware.py và icon trong đó icon ảnh và icon khiên bảo vệ là mặc định của window còn icon keyboard là không phải mặc định. Tất cả được đóng gói bằng PyInstaller như phân tích ở trên

Kết quả thu về khá bất ngờ khi cả 4 file exe đều vượt qua được window defender.

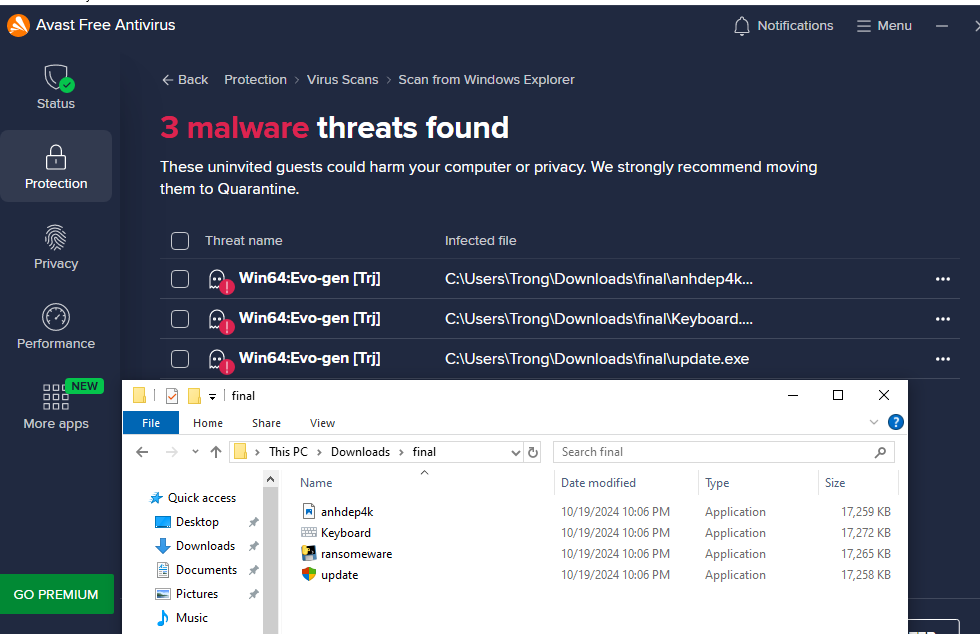


Đoạn mã php được thêm vào để chứng tỏ window defender hoạt động tốt



1. Antivirus

Với 3 file chứa fake icon trên thì cả 3 đều không thể vượt qua được antivirus với category: Win64:Evo-gen[Trj]



Và file ransomeware.exe giống các file .exe trên nhưng không chèn thêm icon nào(là sản phẩm của ransomeware.py+ PyInstaller)  
Vậy chỉ sự khác biệt là thêm icon chèn vào mà antivirus đã xếp nó vào dạng Win64:Evo-gen[Trj]. Nó là gì?

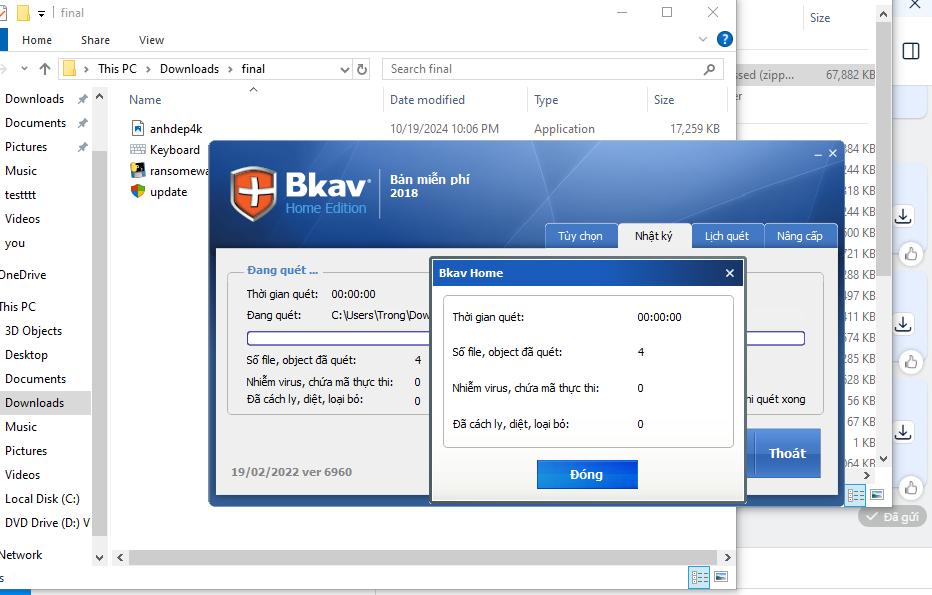
Win64: Evo-gen là tên phát hiện chung được sử dụng bởi phần mềm chống vi-rút để xác định các tệp không mong muốn hoặc đáng ngờ tiềm ẩn thể hiện các đặc điểm hoặc hành vi nhất định thường liên quan đến phần mềm độc hại. Điều quan trọng cần lưu ý là **Win64: Evo-gen không phải là một phần mềm độc hại hoặc vi rút cụ thể**, mà là một phát hiện cho thấy sự **hiện diện của một tệp có khả năng gây hại.**

Khi Win64: Evo-gen được phát hiện, điều đó có nghĩa là phần mềm chống vi-rút đã xác định được một tệp phù hợp với các mẫu hoặc hành vi nhất định có khả năng độc hại. Bản chất chính xác và mức độ nghiêm trọng của tệp được phát hiện có thể khác nhau, vì đó có thể là dương tính giả, biến thể phần mềm độc hại mới hoặc chưa biết hoặc chương trình không mong muốn tiềm ẩn.

Để xác định mức độ rủi ro liên quan đến một trường hợp cụ thể của Win64: Evo-gen, cần phân tích và điều tra thêm. Điều này có thể bao gồm chạy quét chi tiết hơn, gửi tệp cho nhà cung cấp phần mềm chống vi-rút để phân tích hoặc tìm kiếm sự trợ giúp từ chuyên gia an ninh mạng.

Để bảo vệ hệ thống của bạn khỏi Win64: Evo-gen và các mối đe dọa tương tự, điều cần thiết là sử dụng phần mềm bảo mật hoặc chống vi-rút có uy tín, cập nhật hệ điều hành và ứng dụng của bạn, thận trọng khi tải xuống và cài đặt tệp từ các nguồn không xác định và thường xuyên quét hệ thống của bạn để tìm phần mềm độc hại.

1. BKAV home



BKAV bản miễn phí này không quét được bất kì file ransomeware nào.

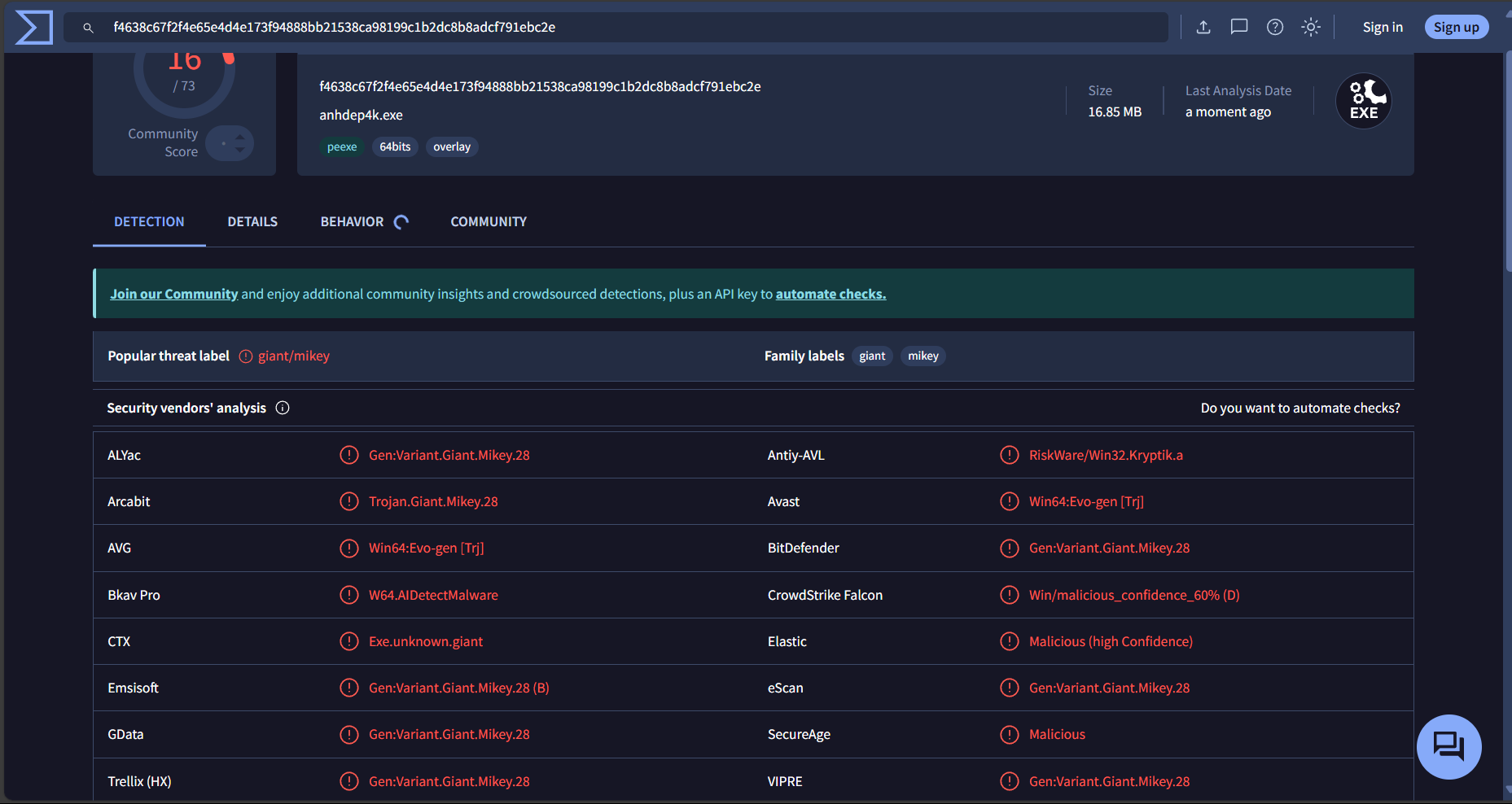
1. Virustotal

VirusTotal kiểm tra các mục với hơn 70 trình quét chống vi-rút và dịch vụ chặn URL / tên miền, ngoài vô số công cụ để trích xuất tín hiệu từ nội dung được nghiên cứu. Bất kỳ người dùng nào cũng có thể chọn một tệp từ máy tính của họ bằng trình duyệt của họ và gửi tệp đó đến VirusTotal. VirusTotal cung cấp một số phương pháp gửi tệp, bao gồm giao diện web công cộng chính, trình tải lên trên máy tính để bàn, tiện ích mở rộng trình duyệt và API lập trình. Giao diện web có mức độ ưu tiên quét cao nhất trong số các phương thức gửi có sẵn công khai. Bài gửi có thể được viết kịch bản bằng bất kỳ ngôn ngữ lập trình nào sử dụng API công khai dựa trên HTTP.

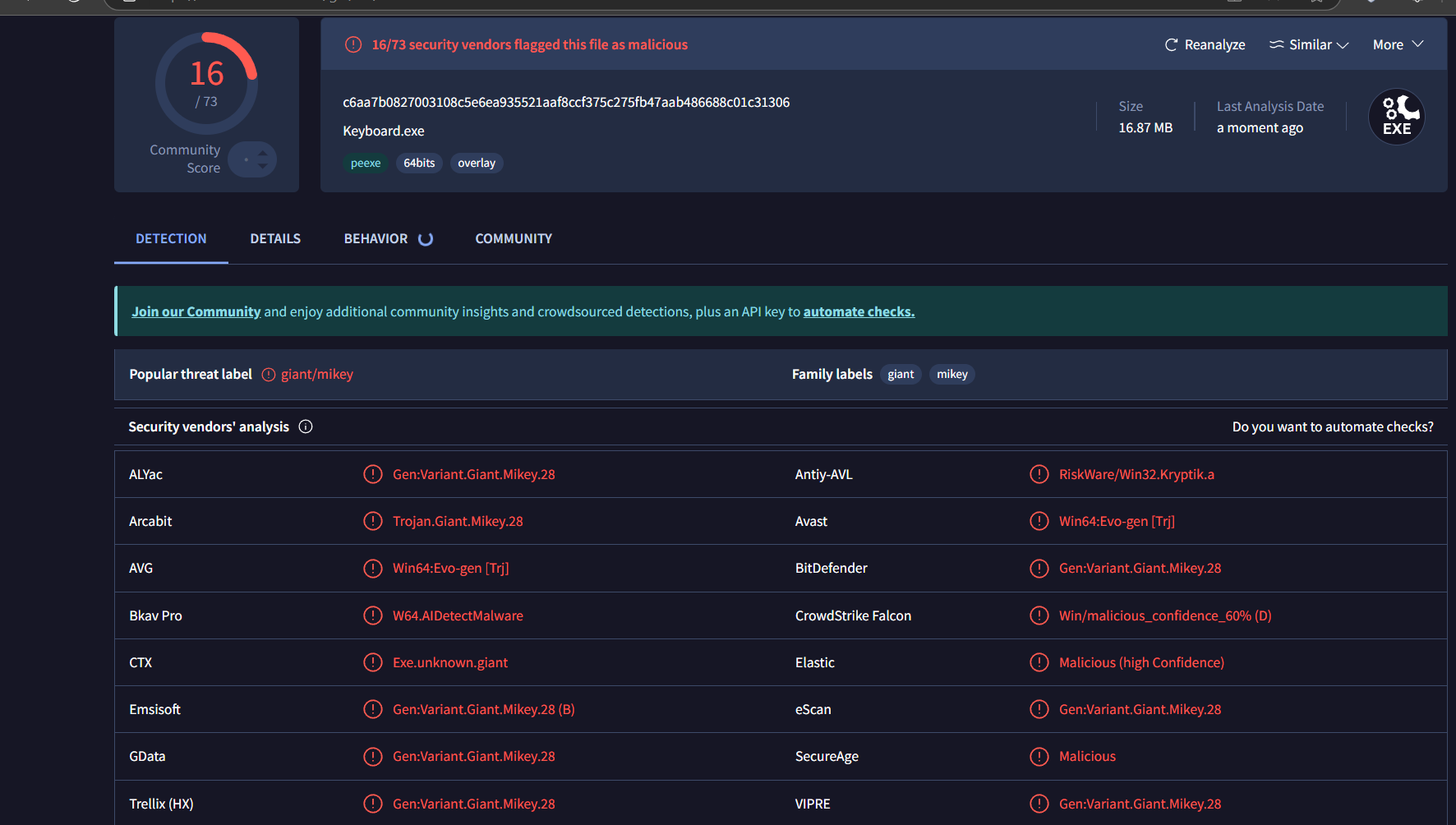
Phân tích cốt lõi này cũng là cơ sở cho một số tính năng khác, bao gồm Cộng đồng VirusTotal: một mạng cho phép người dùng nhận xét về các tệp và URL và chia sẻ ghi chú với nhau. VirusTotal có thể hữu ích trong việc phát hiện nội dung độc hại và cũng trong việc xác định dương tính giả - các mục bình thường và vô hại được phát hiện là độc hại bởi một hoặc nhiều máy quét.

VirusTotal cập nhật chữ ký phần mềm độc hại theo thời gian thực và cung cấp kết quả chi tiết từ nhiều công cụ, giúp người dùng hiểu rõ hơn về các nội dung có hại.  
  
Trên đã so sánh giữa 2 phương pháp đóng gói thì PyInstaller cho ra kết quả khả quan hơn

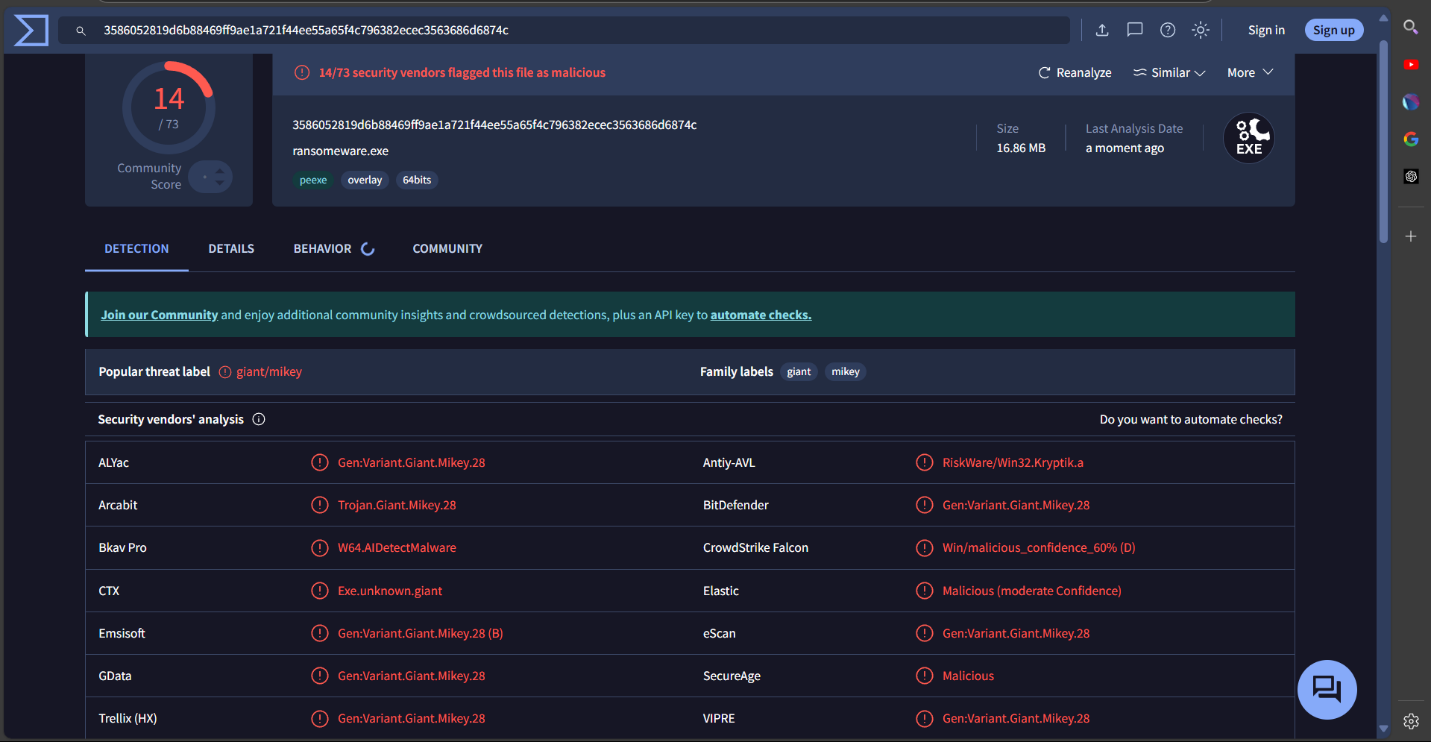
Bây giờ cùng xem xét 3 file anhdep4k.exe(icon Default win 11), KeyBoard.exe(icon internet),ransomeware (không icon)



Anhdep4k.exe



Keyboard.exe



ransomeware

|  |  |  |  |
| --- | --- | --- | --- |
|  | Anhdep4k.exe | KeyBoard.exe | ransomeware.exe |
| BKAV | pass | pass | pass |
| Antivirus | Win64:Evo-gen[Trj] | Win64:Evo-gen[Trj] | pass |
| Window Defender | pass | pass | pass |
| Virustotal | 16/73 security vendors flagged   * Gen:Variant.Giant.Mikey.28 * RiskWare/Win32.Kryptik.a * Trojan.Giant.Mikey.28 * Win64:Evo-gen [Trj] * Gen:Variant.Giant.Mikey.28 * W64.AIDetectMalware * Win/malicious\_confidence\_60% (D) * Exe.unknown.giant * Malicious (high Confidence) * Gen:Variant.Giant.Mikey.28 (B) * Gen:Variant.Giant.Mikey.28 x4 * Malicious | 16/73 security vendors flagged   * Gen:Variant.Giant.Mikey.28 * RiskWare/Win32.Kryptik.a * Trojan.Giant.Mikey.28 * Win64:Evo-gen [Trj] * Gen:Variant.Giant.Mikey.28 * W64.AIDetectMalware * Win/malicious\_confidence\_60% (D) * Exe.unknown.giant * Malicious (high Confidence) * Gen:Variant.Giant.Mikey.28 (B) * Gen:Variant.Giant.Mikey.28 x4 * Malicious | 14/73 security vendors flagged   * Gen:Variant.Giant.Mikey.28 * RiskWare/Win32.Kryptik.a * Trojan.Giant.Mikey.28 * Gen:Variant.Giant.Mikey.28 * W64.AIDetectMalware * Win/malicious\_confidence\_60% (D) * Exe.unknown.giant * Malicious (moderate Confidence) * Gen:Variant.Giant.Mikey.28 (B) * Gen:Variant.Giant.Mikey.28 x4 * Malicious |

Cả 3 file đều là kết quả của mã độc viết bằng Python và được đóng gói bằng PyInstaller. Sự khác biệt chính giữa chúng là có thêm biểu tượng hình ảnh (icon) nhằm giả mạo để lừa người dùng nhấp vào, và tệp còn lại chỉ là mã độc Python thông thường. Dưới đây là nhận xét cụ thể về từng tệp:

* Ransomeware.exe (Python + PyInstaller):
  + Đây là mã độc Python đã được đóng gói thành tệp thực thi bằng PyInstaller.
  + Phần lớn các phần mềm diệt virus như VirusTotal phát hiện ra tệp này dựa trên dấu hiệu của mã độc và dấu hiệu nhận biết của PyInstaller. Nó được nhận diện với các tên như Gen.Giant.Mikey.28, RiskWare/Win32.Kryptik.a, và Trojan.Giant.Mikey.28.
  + Điều này cho thấy rằng mã độc này đã bị phát hiện bởi các công cụ diệt virus dưới nhiều biến thể khác nhau, nhưng không có yếu tố bổ sung nào để lừa người dùng ngoài việc là một mã độc đơn giản.
* Anhdep4k.exe (Python + PyInstaller + icon giả mạo):
  + Tệp này tương tự tệp thứ nhất, nhưng kèm theo một biểu tượng hình ảnh nhằm mục đích giả mạo (ví dụ như biểu tượng giống ứng dụng phổ biến) để đánh lừa người dùng nhấp vào tệp. Đây là một kỹ thuật xã hội để tăng tỷ lệ thành công của mã độc.
  + VirusTotal đã phát hiện mã độc này với một mức độ tự tin cao hơn (“Malicious (high Confidence)”) và cũng xác định thêm một biến thể khác là Win64[Trj].
  + Điều này cho thấy việc thêm biểu tượng giả làm cho mã độc có tính lừa đảo cao hơn, dẫn đến mức độ nguy hiểm cao hơn trong mắt các phần mềm diệt virus.

Nhận xét chung:

* Sử dụng PyInstaller là một kỹ thuật phổ biến để đóng gói mã độc viết bằng Python thành các tệp thực thi, khó phát hiện hơn đối với người dùng không chuyên về công nghệ.
* Tệp thứ hai có nguy cơ cao hơn vì nó tận dụng kỹ thuật giả mạo hình ảnh để lừa người dùng nhấp vào tệp, đồng thời nó có mức độ nguy hiểm cao hơn trong mắt các phần mềm bảo mật. Điều này làm tăng nguy cơ lây nhiễm mã độc.
* VirusTotal đã nhận diện được cả hai tệp dưới nhiều biến thể khác nhau, chứng tỏ các công cụ diệt virus hiện tại đã có khả năng phát hiện những loại mã độc này khá tốt, đặc biệt khi các dấu hiệu như PyInstaller và các hành vi đáng ngờ khác được sử dụng.

Vì thế, mặc dù cả ba đều là mã độc Python, nhưng tệp anhdep4k.exe nguy hiểm hơn do có thêm yếu tố lừa đảo và mức độ tin cậy cao hơn từ phần mềm diệt virus.

1. Các loại mã độc mà virustotal cũng như các công cụ scan khác scan được

Những mã độc mà VirusTotal quét được từ các tệp tệp trên đều thuộc các loại mã độc phổ biến và nguy hiểm, bao gồm Trojan, RiskWare, và các biến thể mã độc như Kryptik hoặc Evo-gen. Dưới đây là nhận xét chi tiết về từng loại mã độc được phát hiện và lý do tại sao chúng bị phát hiện:

* Trojan.Giant.Mikey.28

Trojan là một loại mã độc mà hacker thường sử dụng để truy cập trái phép vào hệ thống máy tính. Sau khi người dùng vô tình chạy tệp thực thi, mã độc này có thể hoạt động ngầm trong nền và mở ra lỗ hổng bảo mật cho hacker kiểm soát máy tính từ xa.

Tại sao nó bị phát hiện: Trojan có những hành vi đáng ngờ, như tạo kết nối không được xác thực đến máy chủ từ xa, thao tác với các tệp hệ thống hoặc lén lút thu thập dữ liệu. Các phần mềm diệt virus nhận diện dựa trên các hành vi điển hình của Trojan và dấu hiệu nhận biết của các đoạn mã độc hại.

* RiskWare/Win32.Kryptik.a

Kryptik là một dạng biến thể của mã độc thường được sử dụng để cài đặt phần mềm độc hại khác trên hệ thống mà không cần sự cho phép của người dùng. Loại mã độc này có thể liên quan đến spyware, keylogger hoặc các công cụ khác nhằm thu thập dữ liệu nhạy cảm.

Tại sao nó bị phát hiện: Các công cụ bảo mật có thể phát hiện dựa trên các dấu hiệu mã độc đã biết từ các biến thể trước đó của Kryptik. Ngoài ra, mã độc này thường đi kèm với các tệp thực thi có hành vi độc hại, như tự sao chép hoặc giao tiếp với các máy chủ lạ.

* Win64[Trj]

Evo-gen là một thuật toán phát hiện dựa trên sự tiến hóa được nhiều phần mềm diệt virus sử dụng để nhận diện các biến thể mới của mã độc. Evo-gen chủ yếu nhằm phát hiện các Trojan mới, nhờ sử dụng các kỹ thuật phân tích heuristic (dự đoán dựa trên các mẫu hành vi).

Tại sao nó bị phát hiện: Phần mềm bảo mật sử dụng thuật toán heuristic để phát hiện các tệp có hành vi tương tự với các Trojan đã biết. Dù tệp chưa từng được xác định trước đó, thuật toán có thể nhận diện hành vi đáng ngờ.

* W64.AIDetectMalware

Đây là một loại mã độc được phát hiện bằng cách sử dụng trí tuệ nhân tạo (AI). Các phần mềm diệt virus hiện nay sử dụng AI để phân tích hành vi của các tệp và phát hiện những mẫu mã độc lạ, thậm chí là những tệp chưa từng được nhận diện trước đó.

Tại sao nó bị phát hiện: Bằng cách phân tích các hoạt động của tệp thực thi trong thời gian thực, AI có thể phát hiện ra những hành vi bất thường như truy cập các khu vực nhạy cảm của hệ thống hoặc thực hiện kết nối ra bên ngoài mà không có sự chấp thuận.

* Gen.Giant.Mikey.28

Gen là một loại phát hiện dựa trên dấu hiệu mã độc đã có. "Gen" ám chỉ các mẫu mã độc được phát hiện từ trước, và "Variant" cho biết đây là một biến thể của mã độc Giant.Mikey.

Tại sao nó bị phát hiện: Các phần mềm bảo mật sử dụng cơ sở dữ liệu với hàng triệu mẫu mã độc đã biết. Khi một tệp chứa các chuỗi mã hoặc hành vi tương đồng với mẫu đã biết, nó sẽ bị đánh dấu là độc hại.

Tại sao các tệp bị phát hiện?

* Dấu hiệu tĩnh: Các phần mềm diệt virus thường kiểm tra dấu hiệu tĩnh của tệp, bao gồm phân tích mã nguồn (nếu có thể), tìm kiếm các chuỗi mã độc đã biết, hoặc kiểm tra các chữ ký mã độc. Các tệp mã độc Python + PyInstaller thường dễ bị phát hiện do tệp thực thi chứa dấu hiệu của PyInstaller và có hành vi giống các mã độc đã được ghi nhận.
* Phân tích hành vi: Khi một tệp được thực thi, các phần mềm bảo mật sử dụng các công cụ để giám sát hành vi của tệp, như tạo kết nối mạng, ghi hoặc đọc các tệp nhạy cảm. Nếu phát hiện ra hành vi nghi ngờ, tệp sẽ bị đánh dấu là độc hại.
* Heuristic và AI: Nhiều phần mềm bảo mật hiện đại sử dụng thuật toán heuristic và AI để dự đoán xem một tệp có phải là mã độc hay không, dựa trên các hành vi hoặc mẫu mã.

1. Phương pháp vượt qua phần mềm quét virus và các phòng tránh
2. Phương pháp đóng gói

Phương pháp đóng gói tệp Python thành tệp thực thi (.exe) có thể ảnh hưởng lớn đến cách các phần mềm quét virus nhận diện và xử lý tệp của bạn. Có một số lý do chính cho điều này:

* False Positive (Dương tính giả) từ các phần mềm antivirus

Khi một tệp Python được đóng gói thành một tệp thực thi .exe, các công cụ như PyInstaller, cx\_Freeze, và py2exe sẽ nhúng tất cả mã Python, thư viện, và phụ thuộc vào một tệp duy nhất. Quá trình này có thể làm các phần mềm quét virus nghi ngờ vì:

* Tệp thực thi chứa rất nhiều mã, bao gồm cả các thư viện nhúng và mã được mã hóa, khiến nó trông như mã độc.
* Một số kỹ thuật đóng gói nén mã thực thi để giảm kích thước, điều này có thể giống với hành vi mã hóa của các phần mềm độc hại (malware).
* Các chương trình do người dùng tự tạo thường không có chứng nhận số (code signing), khiến chúng dễ bị coi là không đáng tin cậy.
* Sử dụng UPX và tác động đến Antivirus

UPX là một công cụ nén tệp thực thi mà PyInstaller và các công cụ khác có thể sử dụng để làm giảm kích thước tệp .exe. Tuy nhiên, khi tệp .exe được nén bằng UPX, điều này có thể khiến nhiều phần mềm antivirus đánh dấu tệp đó là nguy hiểm. Lý do là các virus cũng thường sử dụng kỹ thuật nén tương tự để tránh bị phát hiện.

Giải pháp:

* Tránh sử dụng UPX hoặc thử chạy lệnh đóng gói mà không có tùy chọn nén để giảm tỷ lệ phát hiện sai:

bash

Copy code

pyinstaller --onefile --noupx myscript.py

* Tạo file thực thi mà không nén

Thay vì sử dụng phương pháp đóng gói thành một tệp .exe duy nhất, bạn có thể tạo nhiều tệp (nhiều tệp thư viện riêng biệt cùng với file thực thi). Phương pháp này có thể làm giảm sự nghi ngờ từ các phần mềm quét virus vì mã không bị nén và rõ ràng hơn. Tuy nhiên, nhược điểm là bạn sẽ phải phân phối nhiều tệp.

Ví dụ:

bash

Copy code

pyinstaller --onedir myscript.py

* Quản lý các thư viện và phụ thuộc bên ngoài
* Các thư viện hoặc mã bên ngoài bạn sử dụng trong dự án cũng có thể bị phần mềm antivirus nhận diện sai. Vì vậy, khi đóng gói phần mềm, hãy chắc chắn rằng bạn đang sử dụng các thư viện từ nguồn đáng tin cậy.
* Nếu bạn sử dụng các thư viện mã hóa, bảo mật hoặc bảo vệ mã, các phần mềm quét virus có thể cho rằng chương trình của bạn là phần mềm độc hại.

1. Phần lập trình file ransomeware

m Kỹ thuật lập trình file malware là một phần quan trọng trong việc tạo ra và phân phối các phần mềm độc hại. Những kẻ tấn công thường sử dụng nhiều chiến thuật phức tạp để làm cho malware của họ khó bị phát hiện và loại bỏ. Một trong những kỹ thuật phổ biến nhất là **ngụy trang mã độc** dưới dạng các file hợp pháp, chẳng hạn như tệp tin PDF, hình ảnh, hoặc tài liệu Word. Bằng cách này, nạn nhân có thể vô tình mở file mà không nhận ra nó chứa mã độc.

Kỹ thuật **polymorphism** và **metamorphism** cũng được sử dụng để làm thay đổi cấu trúc hoặc mã nguồn của malware mỗi khi nó lan truyền, khiến các phần mềm diệt virus khó phát hiện. Với polymorphism, mã độc giữ nguyên chức năng nhưng thay đổi hình thức khi thực thi; còn metamorphism thì hoàn toàn viết lại mã nguồn trong quá trình lây lan.

Ngoài ra, kỹ thuật **code obfuscation** (làm rối mã) được sử dụng để mã hóa và che giấu các đoạn mã quan trọng, ngăn không cho chuyên gia bảo mật dễ dàng phân tích. Một số malware sử dụng **anti-debugging** hoặc **anti-virtualization** để tránh bị phân tích bởi các công cụ debug hoặc môi trường ảo hóa. Điều này giúp malware hoạt động trong môi trường thực tế mà không bị phát hiện.

Malware cũng thường sử dụng các phương pháp **persistence** để đảm bảo rằng nó tiếp tục tồn tại ngay cả sau khi hệ thống khởi động lại, ví dụ bằng cách chỉnh sửa registry của Windows hoặc tạo các file hệ thống ẩn. Sự kết hợp của các kỹ thuật này tạo nên các loại malware tinh vi, có khả năng vượt qua các biện pháp phòng thủ truyền thống và gây thiệt hại lớn cho các hệ thống bị lây nhiễm.

1. Cách hacker có thể bypass (vượt qua) các công cụ phát hiện:

* Obfuscation (Làm rối mã):
  + Hacker thường sử dụng kỹ thuật làm rối mã để mã độc trở nên khó hiểu hoặc khó phân tích. Các công cụ như PyArmor, Cython, hoặc sử dụng mã hóa code có thể giúp che giấu mã độc bên trong tệp.
  + Điều này làm cho các phần mềm diệt virus khó khăn hơn trong việc phát hiện mã độc vì các chuỗi mã đặc trưng bị che giấu.
* Packing (Đóng gói lại mã độc):
  + Hacker có thể sử dụng các packer hoặc cryptor để nén hoặc mã hóa mã độc trước khi phát hành. Điều này có thể làm mã độc thoát khỏi các công cụ quét chữ ký tĩnh.
  + Tuy nhiên, các phần mềm diệt virus ngày nay có thể giải nén hoặc giải mã các tệp đóng gói để phân tích.
* Sử dụng các biến thể mới:
  + Thay vì sử dụng mã độc đã biết, hacker có thể tạo ra các biến thể mới bằng cách thay đổi nhỏ trong mã nguồn hoặc dùng các thư viện mã độc cập nhật. Các biến thể mới có thể khó bị phát hiện nếu phần mềm diệt virus chưa cập nhật cơ sở dữ liệu của mình.
* Anti-VM hoặc Anti-debugging:
  + Một số mã độc có thể được lập trình để phát hiện xem nó có đang chạy trong môi trường ảo hóa (Virtual Machine) hoặc đang bị debug hay không. Nếu phát hiện đang trong môi trường kiểm tra, mã độc có thể không hoạt động, qua đó lẩn tránh các phần mềm bảo mật.
* Kỹ thuật xã hội:
  + Như trong tệp thứ hai, việc hacker sử dụng icon giả mạo để làm người dùng tưởng rằng tệp là ứng dụng vô hại (như file ảnh) là một trong những kỹ thuật xã hội phổ biến. Bằng cách dụ người dùng tự chạy mã độc, hacker có thể vượt qua phần lớn các cảnh báo bảo mật.