***Tài liệu***

**BÁO CÁO**

**XÂY DỰNG HỆ THỐNG RÀ QUÉT WEBSHELL**

**Mục Lục**

[**1.** **Giới thiệu giải pháp** 3](#_Toc71816799)

[**1.1.** **Mục tiêu** 3](#_Toc71816800)

[**1.2.** **Phạm vi triển khai** 3](#_Toc71816801)

[**1.3.** **Mô hình triển khai** 3](#_Toc71816802)

[**1.4.** **Phương thức thực hiện** 4](#_Toc71816803)

[**1.5.** **Các chức năng của chương trình** 5](#_Toc71816804)

[**1.6.** **Kế hoạch thực hiện** 6](#_Toc71816805)

[**2.** **Kết quả thí điểm** 6](#_Toc71816806)

[**2.1.** **Dữ liệu sử dụng** 7](#_Toc71816807)

[**2.2.** **Kết quả rà soát webshell** 7](#_Toc71816808)

[**3.** **Hướng phát triển và kết luận** 12](#_Toc71816809)

[**3.1.** **Hướng phát triển** 12](#_Toc71816810)

[**3.2.** **Kết luận** 12](#_Toc71816811)

1. **Giới thiệu giải pháp**
   1. **Mục tiêu**

Webshell là một tập lệnh có thể được tải lên máy chủ web để cho phép quản trị máy từ xa. Webshell có thể được viết bởi bất kì ngôn ngữ nào mà máy chủ web hỗ trợ, phổ biến nhất là PHP, ASP. Perl, Ruby và Python. Ngoài ra, webshell có thể được thực thi hoàn toàn trong suốt ở trên cả hai nền tảng Windows và Linux. Đây là hai nền tảng phổ biến được sử dụng cho các máy chủ hiện nay, cũng như các máy chủ của tập đoàn Vingroup.

Sau khi webshell được tải lên thành công, kẻ tấn công có thể sử dụng webshell để tận dụng các kỹ thuật khai thác khác nhằm nâng cao đặc quyền và ra lệnh từ xa. Các lệnh này có thể bao gồm khả năng thêm, xóa, thực thi các tệp, cũng như khả năng chạy các lệnh shell, các tệp thực thi khác hoặc các tập lệnh. Từ đây, kẻ tấn công có thể đánh cắp các dữ liệu nhạy cảm trên máy chủ, duy trì kết nối command-and-control và tải thêm các malware, virus độc hại cho mục đích khác.

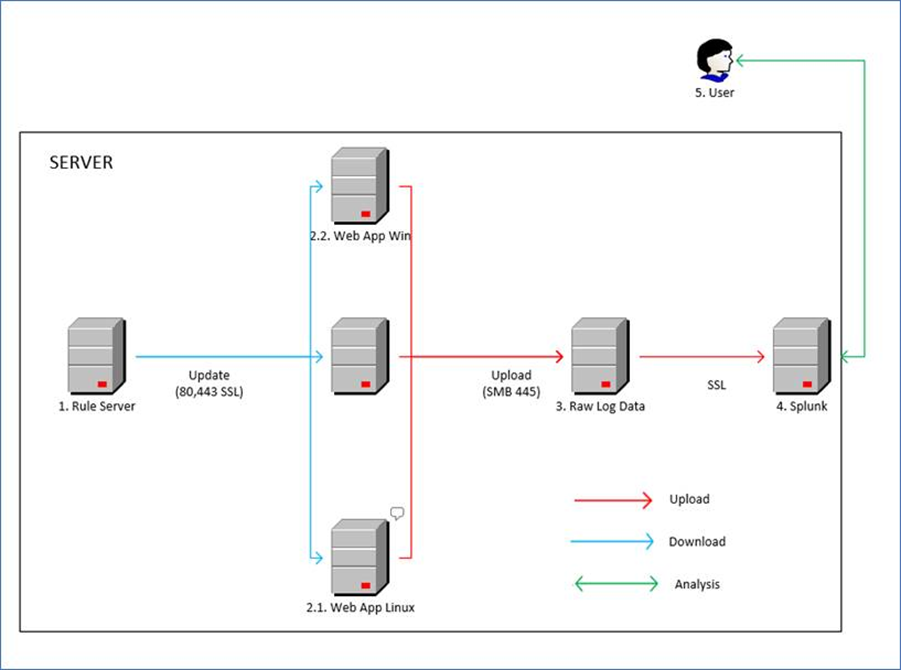
Với mục tiêu chủ động săn tìm phát hiện sớm dấu hiệu, chỉ dấu tấn công tồn tại trên hệ thống CNTT của tập đoàn Vingroup, Nhóm Giám sát và Điều tra phòng An ninh bảo mật công ty VINITIS đã tiến hành tìm hiểu và nghiên cứu các kỹ thuật phát hiện webshell hiện nay, từ đó đưa ra giải pháp phát hiện các file webshell nói trên. Giải pháp sẽ sử dụng một chương trình được xây dựng dựa trên ngôn ngữ Python, thực hiện rà quét các file trên web server. Sau đó kết quả rà quét file, bao gồm thông tin về các file nghi ngờ là webshell sẽ tự động được chuyển về kho lưu trữ tập trung, và tiến hành phân tích thông qua hệ thống quản lý và phân tích dữ liệu. Yêu cầu giải pháp là phải thu thập đầy đủ thông tin, có khả năng triển khai từ xa, tự động, ảnh hưởng thấp đến hiệu năng máy tính và trong suốt với người dùng, có khả năng phân tích dữ liệu lớn với tốc độ nhanh, chính xác và có thể tái sử dụng được nhiều lần.

* 1. **Phạm vi triển khai**

Với giai đoạn thử nghiệm giải pháp, thực hiện triển khai trên các máy tính phát triển giải pháp thuộc phòng An ninh bảo mật – Công ty VinITIS.

* 1. **Mô hình triển khai**

Mô hình triển khai giải pháp rà quét webshell như sau:



Hình 1: Mô hình triển khai hệ thống rà quét webshell.

**+ Máy chủ lưu trữ bộ luật (rule):** Máy chủ lưu trữ thông tin về các file đã có kết quả scan (thuộc blacklist hoặc whitelist) nhằm tiết kiệm thời gian rà soát cũng như nâng cao độ chính xác phát hiện. File cơ sở dữ liệu được chương trình đọc thông qua giao thức HTTP (port 80).

**+ Máy chủ lưu trữ mã nguồn website:** Các máy chủ webserver, chứa mã nguồn của website. Các máy tính này sẽ thực thi chương trình rà quét webshell được xây dựng dựa trên Python. Chương trình hiện có thể chạy trên hệ điều hành Windows và Linux.

**+ Máy chủ lưu trữ log:** Máy chủ lưu trữ log từ các máy tính được thực hiện rà soát. Thông tin về các file mới được quét mà không có trong cơ sở dữ liệu của máy chủ lưu trữ bộ luật sẽ được gửi đến máy chủ lưu trữ log này thông qua giao thức SMB (port 445). Trên máy chủ được cài đặt phần mềm Splunk Forwarder để đẩy dữ liệu log từ các máy tính về máy chủ Splunk. Đây là thành phần trung gian, giúp lưu trữ dữ liệu thô, đã qua giai đoạn tiền xử lý để đưa vào hệ thống Splunk và tiến hành phân tích.

**+ Máy chủ Phân tích:** Được cài đặt phần mềm Splunk Enterprise. Phần mềm Splunk Enterprise sẽ là công cụ thực hiện phân tích dữ liệu. Đồng thời có thể tạo các bảng biểu, biểu mẫu báo cáo về trạng thái săn tìm, kết quả săn tìm mối đe dọa.

* 1. **Phương thức thực hiện**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bước** | **Hành động** | **Kết quả** | **Ghi chú** |
| 1 | Thực thi chương trình rà quét webshell | Chương trình lấy CSDL về các file đã scan tại máy chủ lưu trữ bộ luật thông qua giao thức HTTP (port 80). Nếu gặp file đã có kết quả trong CSDL, chương trình sẽ không quét lại file và trả về kết quả tương ứng. Các file không nằm trong CSDL sẽ được rà quét và phát hiện dựa trên các đặc tính của webshell. Cuối cùng kết quả quét sẽ được đẩy về máy chủ lưu trữ log tập trung thông qua phương thức SMB (port 445). | Kết quả thử nghiệm trên máy tính: |
| 2 | Đẩy dữ liệu log tới máy chủ phân tích Splunk thông qua Spunk forwader | Dữ liệu log có định dạng json được Splunk hiểu và parse thành công ra các trường dữ liệu có thể tìm kiếm, sắp xếp, tính toán, phân tích. | Tác vụ tự động thực hiện bằng Splunk Forwarder |
| 3 | Thực hiện phân tích log trên máy chủ Splunk | Các dữ liệu log sẽ được phân tích và tạo sẵn các Report, Alert và hiển thị trên Dashboard. | Kết quả phân tích có thể thực hiện định kỳ và gửi báo cáo tự động. |

* 1. **Các chức năng của chương trình**

Công cụ có khả năng rà quét các file có phần mở rộng thường được các webshell sử dụng (.php, .asp, .aspx, .sh, .bash, …) và phát hiện các file là webshell dựa trên các dấu hiệu đặc trưng bao gồm:

* Dựa trên các từ khóa bất thường: chương trình sẽ kiểm tra xem có tồn tại chuỗi kí tự bất thường xuất hiện trong webshell hay không.
* Dựa trên các từ khóa bất thường trong chuỗi bị chia nhỏ: chương trình sẽ nối các chuỗi ký tự trong dấu “ ” hoặc ‘ ’ gần nhau và kiểm tra lại để phát hiện các trường hợp webshell chia payload thành các chuỗi ký tự nhỏ và nối chúng lại trong quá trình thực thi.
* Dựa trên các chuỗi có định dạng base64 đáng ngờ: phát hiện các trường hợp webshell mã hóa payload theo định dang base64 rồi giải mã chúng trong quá trình thực thi.
* Dựa trên các chuỗi có định dạng base32 đáng ngờ: phát hiện các trường hợp webshell mã hóa payload theo định dang base32 rồi giải mã chúng trong quá trình thực thi.
* Dựa trên dấu hiệu các file có chỉ số entropy cao: tính toán chỉ số entropy của file được kiểm tra để tìm kiếm các file có khả năng cao bị mã hóa. Các file có chỉ số entropy lớn hơn 6 theo thang đo 10 thì sẽ bị nghi ngờ. Webshell có thể sử dụng hình thức mã hóa payload để tránh bị phát hiện bởi các hệ thống phòng thủ.
* Dựa trên dấu hiệu các file có chỉ số nén cao: tính toán chỉ số nén của file được kiểm tra, nếu file có chỉ số nén lớn hơn 0.74 thì sẽ bị nghi ngờ. Tương tự như trường hợp trên, webshell cũng có thể nén payload trong source code để tránh bị phát hiện.
* Dựa trên nghi ngờ chuỗi ở định dạng hex: chương trình sẽ kiểm tra xem liệu file được kiểm tra có viết dưới dạng hex hay không. Đây cũng là kĩ thuật tránh bị phát hiện của webshell.
* Dựa trên dấu hiệu file bị xáo trộn nội dung (obfuscated): chương trình sẽ kiểm tra xem liệu file được kiểm tra có chứa các hàm nhằm xáo trộn nội dung file (obfuscate) hay không. Đây cũng là kĩ thuật khác nhằm tránh bị phát hiện của webshell.
* Dựa trên chuỗi có độ dài bất thường trong file: chương trình sẽ tìm kiếm các chuỗi ký tự có độ dài trên 64 nhằm phát hiện các trường hợp webshell mã hóa payload và để nó trong một chuỗi ký tự dài sau đó giải mã trong quá trình thực thi.

Công cụ được xây dựng bằng ngôn ngữ lập trình Python nên có ưu điểm chính là hỗ trợ đa nền tảng, bao gồm cả Windows và Linux. Đây là hai hệ điều hành chính được sử dụng trên các máy chủ hiện nay. Ngoài ra Python là ngôn ngữ có khả năng phát triển ứng dụng dễ dàng, cũng như cho phép cài đặt các gói thư viện hỗ trợ nhanh chóng.

Công cụ có khả năng tự động thực thi dựa trên lập lịch. Đối với Windows sẽ dựa trên tính năng Task Scheduler và với Linux sẽ dựa trên tính năng crontab của hệ điều hành. Người dùng có thể tùy chỉnh các cài đặt để chương trình chạy theo lịch như mong muốn. Khi đến lịch đã được thiết lập trước, chương trình sẽ tự động rà quét hệ thống và tự động gửi kết quả về máy chủ phân tích, mà không cần người dùng chạy thủ công lại công cụ này.

Công cụ sẽ quét các thư mục web do người dùng định nghĩa, ngoài ra sẽ mặc định quét các thư mục sau:

* Đối với hệ điều hành Windows:
  + C:\xampp\htdocs\
  + C:\inetpub\wwwroot\
  + C:\Apache24\htdocs\
* Đối với hệ điều hành Linux:
  + /var/www/html
  + /var/http/
  + /srv/http/
  + /etc/apache2/
  + /etc/nginx/
  + /etc/httpd/
  + /usr/local/apache2
  + /webapps/ROOT/
  + /applications/DefaultWebApp/
  + /opt/lampp/httpdocs/

Danh sách các thư mục mặc định được quét có thể được bổ sung trong tương lai.

* 1. **Kế hoạch thực hiện**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Task** | **Owner** | **Thời gian** |
| 1 | Thiết kế hệ thống |  | 1 tuần |
| 2 | Lập trình các chức năng |  | 3 tuần |
| 3 | Xây dựng môi trường thử nghiệm |  | 2 ngày |
| 4 | Thực hiện kiểm tra và đánh giá phương án |  | 3 ngày |
| 5 | Hoàn thiện các chức năng |  | 2 tuần |
| 6 | Gửi thông tin cho PnL đánh giá |  | 1 tuần |
| 7 | Viết tài liệu triển khai |  | 3 ngày |
| 8 | Phối hợp cùng PnLs triển khai giải pháp |  | 1 tuần |
| 9 | Theo dõi và đánh giá kết quả |  | 3 ngày |

* 1. **Cách thức cài đặt chương trình**

Chương trình có thể triển khai trên HĐH Windows và Linux. Để sử dụng chương trình, thực hiện theo các bước sau:

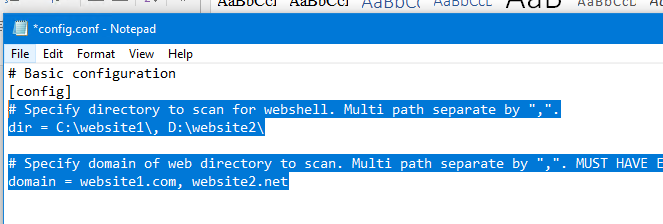
Mã SHA-1 kiểm tra toàn vẹn của chương trình:

|  |  |  |
| --- | --- | --- |
| **STT** | **Filename** | **SHA-1** |
| 1 | webshell\_scan.exe | 3EAE7265F05DD2C5D13AC1BBF8ADFF66D033EDE1 |
| 2 | webshell\_scan | 3702671C6D1491FDF11F217F60EA9B9571367A37 |

Bước 1: Kiểm tra kết nối từ máy chủ web tới địa chỉ máy chủ lưu trữ log tập trung qua giao thức SMB: IP- 10.111.177.41, Port – 445, 139.

Bước 2: Cấu hình cho chương trình:

* Chỉnh sửa cấu hình cho chương trình nhận biết được thư mục chứa mã nguồn của website cần được quét. Trong tệp tin “config.conf”. Tìm đến hai trường là trường “dir” và trường “domain”. Trong đó, “dir” là nơi lưu trữ mã nguồn của website, tương ứng với trường “domain”, nếu máy chủ lưu trữ nhiều mã nguồn của nhiều website.
* Trong ví dụ dưới đây, “website1.com” được lưu tại thư mục “C:\website1\”, “website2.net” được lưu tại thư mục “D:\website2\”.



Hình 2: Cấu hình ví dụ của chương trình.

* Đối với Windows, trường “days\_interval =” dùng để cấu hình số ngày mà chương trình sẽ được lập lịch để thực thi. Nếu để trống sẽ thực thi sau 30 ngày và vào 22h00 trên Windows. Khuyến cáo để mặc định trường này.
* Đối với Linux, trường “crontab =” dùng để cấu hình crontab cho chương trình lập lịch thực thi. Nếu để trống giá trị crontab mặc định là 0 22 1 \* \*, tức là chương trình thực thi vào ngày đầu tiên của tháng lúc 22h00 trên Linux. Khuyến cáo để mặc định trường này.
* Lưu lại tệp tin cấu hình.

Bước 3: Sau khi đã chỉnh sửa tệp tin cấu hình và lưu lại. Thực thi chương trình.

* Trên Windows, thực thi chương trình với quyền Administrator Local. Bấm chuột phải vào chương trình “webshell\_scan.exe” và chọn “Run as Administrator”.
* Trên Linux, thực thi chương trình bằng cách chạy câu lệnh “chmod +x webshell\_scan && ./webshell\_scan”.

1. **Kết quả thí điểm**
   1. **Dữ liệu sử dụng**

Dữ liệu sử dụng để đánh giá hiệu năng của công cụ scan webshell bao gồm mã nguồn các webshell độc hại từ Github và mã nguồn web sạch không chứa webshell:

Mã nguồn webshell:

* <https://github.com/xl7dev/WebShell>
  + Tổng cộng có 536 file, bao gồm các loại webshell thuộc định dạng phổ biến.
* <https://github.com/tennc/webshell>
  + Tổng cộng có 1110 file, bao gồm các loại webshell thuộc định dạng phổ biến.
* <https://github.com/JohnTroony/php-webshells>
  + Tổng cộng có 119 file, bao gồm các loại webshell thuộc định dạng phổ biến.
* <https://github.com/tanjiti/webshellSample>
  + Tổng cộng có 502 file, bao gồm các loại webshell thuộc định dạng phổ biến.
* <https://github.com/ysrc/webshell-sample>
  + Tổng cộng có 2476 file, bao gồm các loại webshell thuộc định dạng phổ biến.

Mã nguồn web sạch:

* <https://github.com/woocommerce/woocommerce>
  + Tổng cộng có 865 file lành tính, cùng thuộc các định dạng webshell phổ biến trên.
* <https://123code.vn/do-an/chia-se-mien-phi-source-code-web-ban-hang-gan-giong-tgdd-wordpress-full-data.html> (/banhangthegioididong)
  + Tổng cộng có 5985 file lành tính, cùng thuộc các định dạng webshell phổ biến trên.
* <https://123code.vn/do-an/chia-se-mien-phi-source-code-web-ban-hang-bang-wordpress-chuan-seo.html> (/antien123code.vn)
  + Tổng cộng có 3355 file lành tính, cùng thuộc các định dạng webshell phổ biến trên.
* <https://123code.vn/do-an/chia-se-source-code-website-tour-du-lich-wordpress-chuan-seo-mien-phi.html> (/tourdulich)
  + Tổng cộng có 4906 file lành tính, cùng thuộc các định dạng webshell phổ biến trên
  1. **Kết quả rà soát webshell**

Sau khi tải mã nguồn trên về, Nhóm giám sát và điều tra đã thực hiện chạy công cụ scan webshell trên các thư mục mã nguồn đó và cho kết quả như sau:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Mã nguồn** | **Tổng số file** | **Cho là webshell** | **Không cho là webshell** | **Tỉ lệ dương tính đúng** | **Tỉ lệ lành tính sai** |
| 1 | /xl7dev/WebShell | 535 | 447 | 88 | 83.55% | 16.45% |
| 2 | /tennc/webshell | 1110 | 936 | 174 | 84.32% | 15.68% |
| 3 | /JohnTroony/php-webshells | 119 | 111 | 8 | 93.28% | 6.72% |
| 4 | /tanjiti/webshellSample | 502 | 467 | 35 | 93.02% | 6.98% |
| 5 | /ysrc/webshell-sample | 2476 | 2115 | 361 | 85.42% | 14.58% |
| **Tổng cộng** | | 4742 | 4076 | 666 | 85.95% | 14.05% |

Tổng thời gian quét các thư mục trên là 406.18 giây. CPU trung bình sử dụng 13.23%, RAM sử dụng 77.52 MB.

* Đối với các thư mục web sạch:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Mã nguồn** | **Tổng số file** | **Cho là webshell** | **Không cho là webshell** | **Tỉ lệ dương tính sai** | **Tỉ lệ lành tính đúng** |
| 1 | /woocommerce | 865 | 210 | 655 | 24.27% | 75.73% |
| 2 | /banhangthegioididong | 5985 | 1223 | 4762 | 20.43% | 79.57% |
| 3 | /antien123code.vn | 3355 | 760 | 2595 | 22.65% | 77.35% |
| 4 | /tourdulich | 4906 | 977 | 3929 | 19.91% | 80.09% |
| **Tổng cộng** | | 15111 | 3170 | 11941 | 20.97% | 79.03% |

Tổng thời gian quét các thư mục trên là 309.54 giây. CPU trung bình sử dụng 14.71%, RAM sử dụng 46.3 MB.

Kết quả thí điểm trên đều là kết quả phát hiện thuần túy của công cụ, chưa sử dụng máy chủ lưu trữ thông tin các file đã scan (dựa trên danh sách blacklist và whitelist).

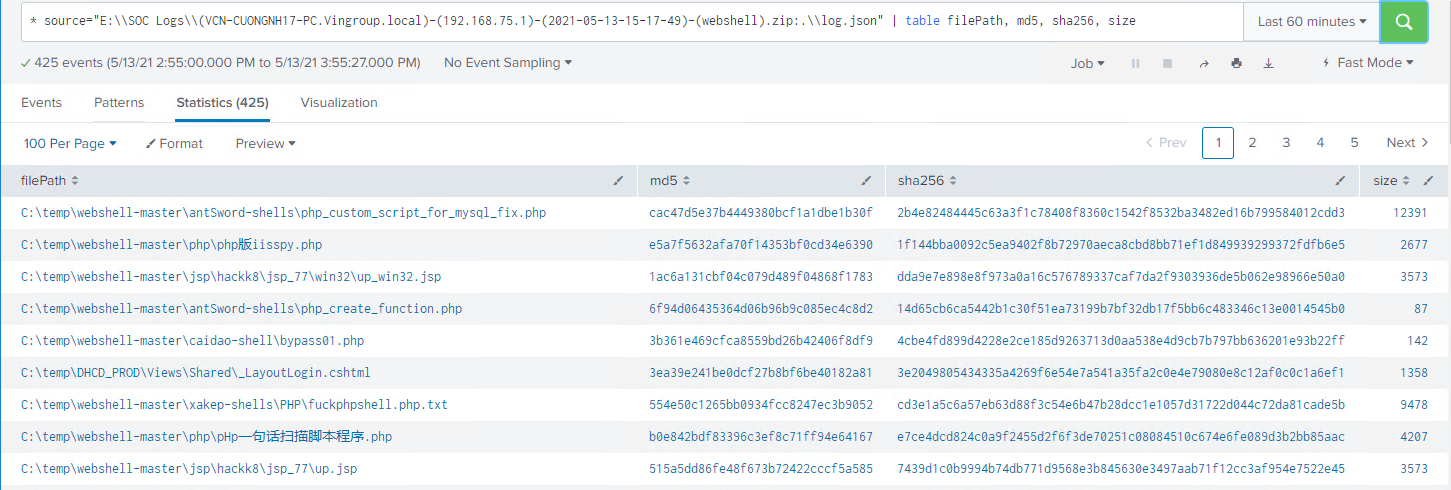
Dữ liệu phân tích sẽ được tự động đưa vào hệ thống Splunk như sau:



Hình 3: Kết quả tổng quan khi rà quét hệ thống.

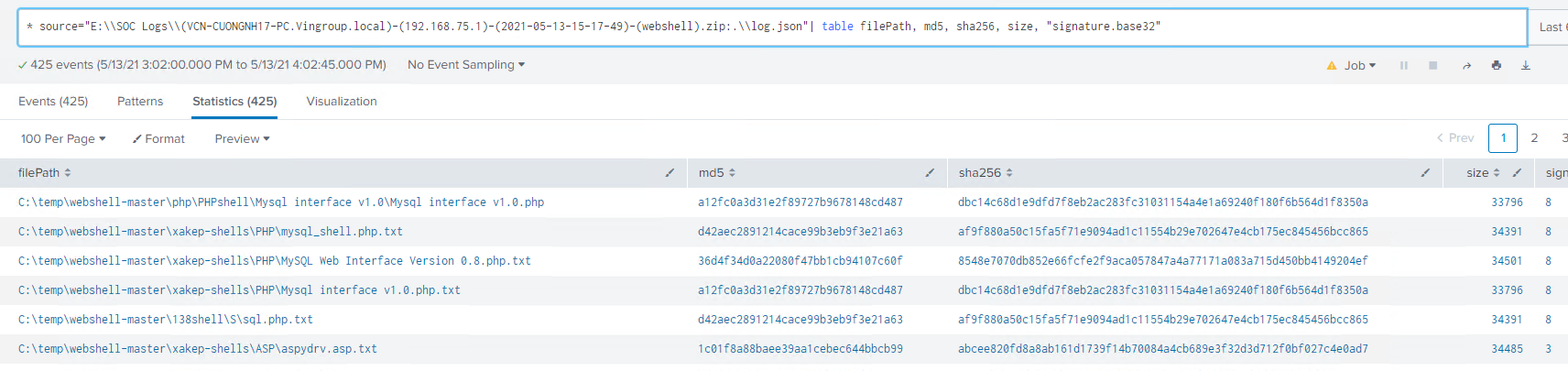


Hình 4: Kết quả khi phân tích một tệp tin của hệ thống.



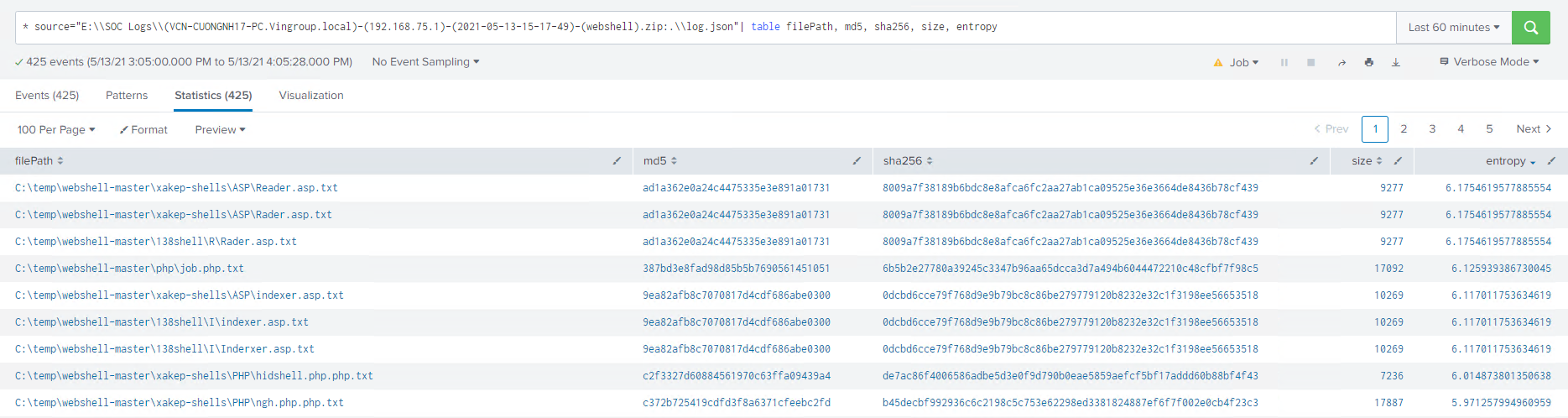
Hình 5: Kết quả phân tích trên Splunk.

Có thể thực hiện tìm kiếm tệp tin có những dấu hiệu bất thường. Ví dụ dưới đây tìm kiếm tệp tin sử dụng hàm mã hóa base32 trong mã nguồn.



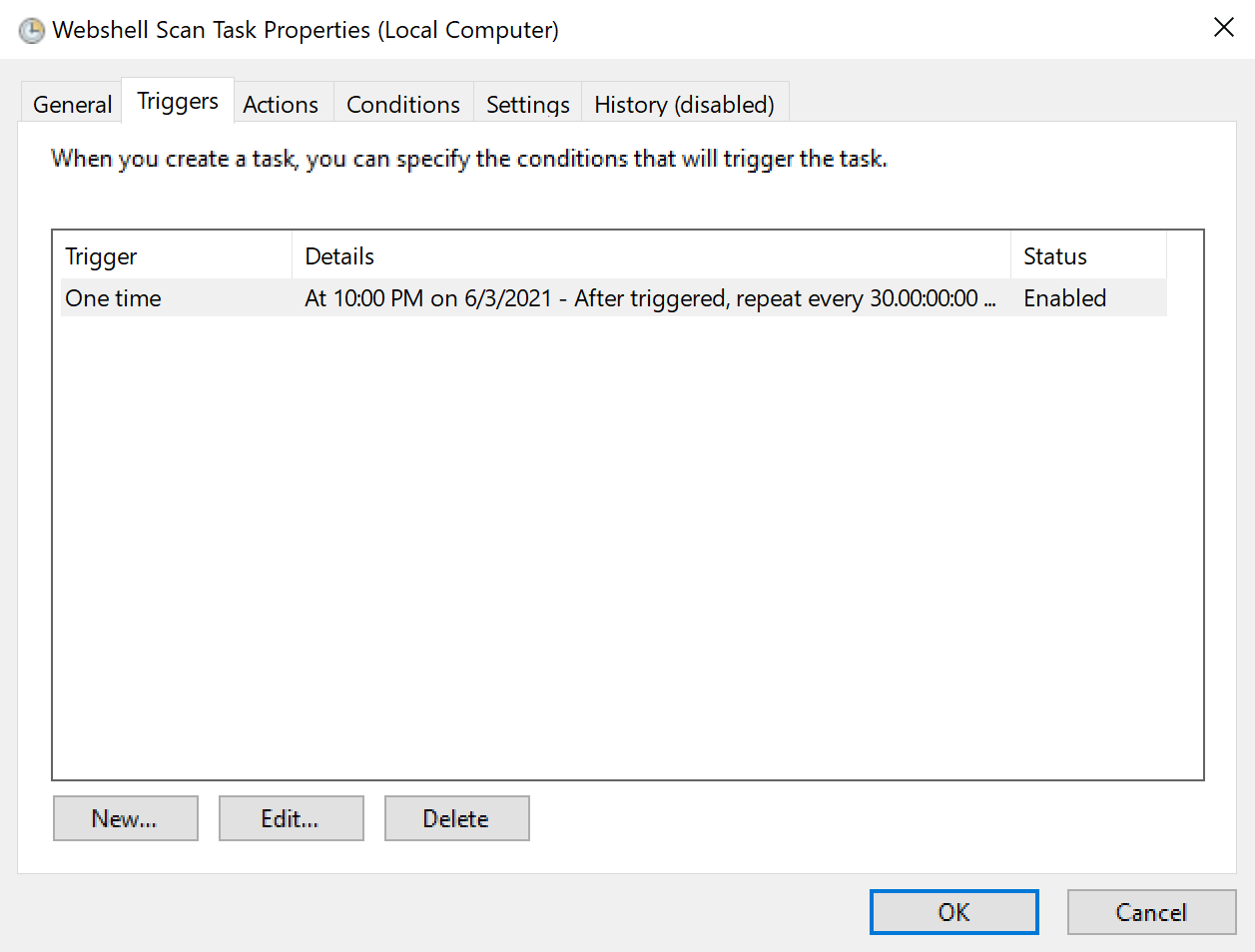
Hình 6:Tìm kiếm kết quả tệp tin có sử dụng hàm mã hóa base32 trong mã nguồn.

Hoặc tìm kiếm tệp tin nào có entropy >= 6.0.



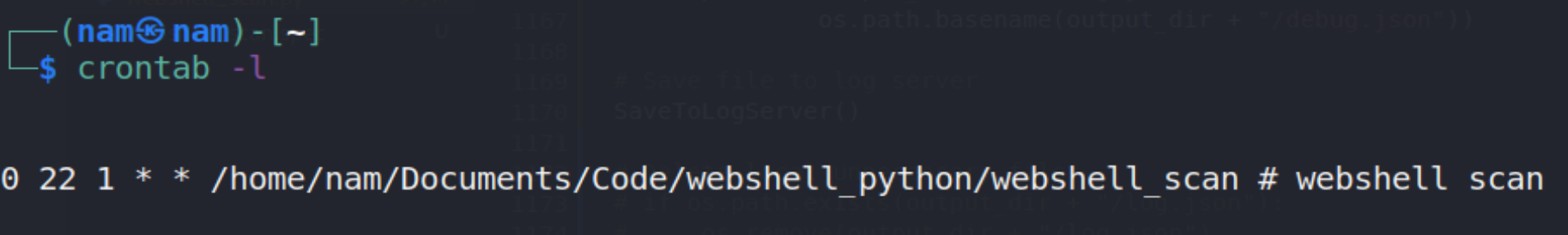
Hình 7:Kết quả tìm kiếm tệp tin có Entropy >= 6.0 trong mã nguồn.

Đối với Windows, chương trình tự khởi tạo Task Scheduler để lập lịch tự động chạy theo thời gian cấu hình. Mặc định sẽ chạy hàng tháng (sau 30 ngày tính từ ngày chạy công cụ, vào 22h00).



Hình 8: Task Scheduler tự khởi chạy chương trình sau mỗi 30 ngày.

Đối với Linux, chương trình sẽ tự tạo Crontab và lập lịch chạy công cụ tự động. Mặc định giá trị crontab được đặt là 0 22 1 \* \* (khởi chạy công cụ hàng tháng vào 22h00)



*Hình 9: crontab tự khởi chạy chương trình vào ngày đầu tiên của tháng.*

1. **Hướng phát triển và kết luận**
   1. **Hướng phát triển**

Với phiên bản hiện tại, chương trình đã có khả năng phát hiện webshell với độ chính xác tương đối cao. Tuy nhiên kết quả trên mới chỉ là kết quả khi chưa sử dụng máy chủ lưu trữ thông tin các file đã quét. Theo thời gian, nếu như được bổ sung càng nhiều thông tin trên cơ sở dữ liệu, thì tỉ lệ dương tính giả và âm tính giả sẽ ngày càng giảm, độ chính xác của công cụ ngày càng được nâng cao.

Chương trình có thể bổ sung tính năng cập nhật các dấu hiệu phát hiện webshell mới một cách tự động, mà không cần can thiệp chỉnh sửa mã nguồn. Điều này cũng giúp nâng cao độ chính xác phát hiện đối với các loại webshell mới sử dụng các kĩ thuật ngày càng tinh vi.

Ngoài ra trong tương lai cần tối ưu chương trình để chương trình có thể nâng cao khả năng phát hiện, giảm tiêu tốn tài nguyên hệ thống và thời gian thực thi.

* 1. **Kết luận**

Giải pháp đáp ứng được yêu cầu đặt ra đối với nhu cầu rà quét các file webshell. Giải pháp có khả năng triển khai dễ dàng trên diện rộng và không làm ảnh hưởng lớn đến hiệu năng hoạt động của máy tính người dùng, vô hình với người dùng. Giải pháp có thể triển khai trên toàn bộ máy chủ sử dụng hai hệ điều hành Windows và Linux. Hiệu năng thí điểm cho thấy chương trình sử dụng dưới 15% CPU, 100MB RAM và trong thời gian nhỏ hơn 10 phút. Khi thực hiện rà quét, người sử dụng chỉ cần thực thi chương trình duy nhất một lần đầu tiên, những lần kế tiếp chương trình sẽ tự động rà quét các thư mục thông qua cơ chế lập lịch (Task Scheduler, crontab). Trong quá trình chương trình thực thi không yêu cầu người dùng phải tương tác. Thông tin sẽ tự động được gửi về máy chủ phân tích dữ liệu.

Hệ thống phân tích có khả năng phân tích với lượng dữ liệu đáp ứng số lượng máy chủ được rà quét. Dữ liệu đầu ra của chương trình không quá lớn, với hệ thống hiện tại có thể đáp ứng được yêu cầu lưu trữ, phân tích, tìm kiếm với lượng dữ liệu này. Nếu thực hiện định kỳ hàng năm 4 lần, có thể xóa các dữ liệu đã thu thập trước đó để sử dụng dữ liệu mới.

Hệ thống đáp ứng khả năng tìm kiếm chính xác, với tỉ lệ dương tính giả và âm tính giả ở mức cho phép. Đồng thời để cải thiện độ chính xác phát hiện của công cụ, chuyên viên phân tích có thể dễ dàng bổ sung danh sách các file đã xác định là webshell hay không tại máy chủ cơ sở dữ liệu.

Giải pháp phù hợp và cần được đưa vào sử dụng./

-- HẾT --