

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



PHÒNG CHỐNG VÀ ĐIỀU TRA TỘI PHẠM MÁY TÍNH

Đề tài

Windows Forensics Analysis - chapter 6
PHÂN TÍCH TẬP TIN THỰC THI (Executable File Analysis)

Ngành: Công nghệ thông tin
Chuyên ngành: An toàn thông tin

Cán bộ hướng dẫn :

GV. Lại Minh Tuấn

Khoa An toàn thông tin – Học viện Kỹ thuật Mật Mã

Hà Nội, 11/2019

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



PHÒNG CHỐNG VÀ ĐIỀU TRA TỘI PHẠM MÁY TÍNH
Đề tài
Windows Forensics Analysis - chapter 6
PHÂN TÍCH TẬP TIN THỰC THI (Executable File Analysis)

Sinh viên thực hiện:

- Nguyễn Tuấn Anh
- Nguyễn Hữu Hải
- Nguyễn Văn Minh

Cán bộ hướng dẫn :

GV. Lại Minh Tuấn

Khoa An toàn thông tin – Học viện Kỹ thuật Mật Mã

Hà Nội, 11/2019

Nhận xét của giảng viên

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Mục lục

PHÂN TÍCH TẬP TIN THỰC THI (Executable File Analysis)	6
1. Giới thiệu	6
2. Phân tích tĩnh	7
2.1. Định vị tệp để phân tích	8
2.2. Tài liệu tập tin	10
2.3. Phân tích.....	13
2.3.2. <i>IMPORT Tables</i>	23
2.3.3. <i>EXPORT Tables</i>	26
2.3.4. Tài nguyên	28
2.3.5. <i>Obfuscation</i>	29
3. Phân Tích Động	36
3.1. Kiểm Tra Môi Trường	36
4. Tóm Tắt	49
5. Giải pháp theo dõi	50

DANH MỤC HÌNH ẢNH

Hình 6. 1 Notepad.exe Open in BinText	14
Hình 6. 2 Cấu trúc IMAGE_DOS_HEADER được xem trong Trình soạn thảo Hex	17
Hình 6. 3 IMAGE_NT_HEADERS Giá trị chữ ký	17
Hình 6. 4 IMAGE_FILE_HEADER được xem trong PEView	18
Hình 6. 5 IMAGE_OPTIONAL_HEADER được xem trong PEView	20
Hình 6. 6 Trích dẫn cấu trúc IMAGE_DATA_DIRECTORY được xem trong PEView	21
Hình 6. 7 IMAGE_SECTION_HEADER được xem trong PEView	22
Hình 6. 8 Trích từ Dependency Walker GUI.....	25
Hình 6. 9 Chọn hộp thoại giả nén từ Procdump32	31
Hình 6. 10 Biểu tượng cho nhị phân phần mềm độc hại rada.exe.....	32
Hình 6. 11 Rada.exe được tải vào PEiD	34
Hình 6. 12 Hộp thoại thông tin bổ sung PEiD với rada.exe	35
Hình 6. 13 .exe được tải trong Mandiant's Red Curtain.....	36
Hình 6. 14 Regshot GUI	45
Hình 6. 15 InControl5 GUI.....	46
Hình 6. 16 minh họa thành công cụ Process Monitor.....	48
Hình 6. 17 Process Monitor hiển thị quyền truy cập vào tệp tùy chọn thực thi tệp hình ảnh.....	48

DANH MỤC BẢNG

Bảng 6 1 IMAGE_FILE_HEADER Giá trị cấu trúc	18
---	----

PHÂN TÍCH TẬP TIN THỰC THI (Executable File Analysis)

Giải pháp trong chương này:

- Phân tích tĩnh
- Phân tích động

tóm tắt:

Giải pháp theo dấu nhanh

Các câu hỏi thường gặp

1. Giới thiệu

Đôi khi trong quá trình điều tra, bạn có thể bắt gặp một tệp thực thi đáng ngờ mà bạn muốn thực hiện một số phân tích để có ý tưởng về những gì nó làm hoặc chức năng nào phù hợp với nó. Nhiều lần, kẻ xâm nhập có thể để lại tập lệnh hoặc tệp cấu hình và các tệp này thường là tệp văn bản có thể mở và xem được. Trong trường hợp tập lệnh, một số kiến thức về lập trình có thể cần thiết để hiểu đầy đủ chức năng của tệp.

Trong Chương 5, chúng tôi đã thảo luận về phân tích chữ ký tệp, một phương pháp để xác định xem một tệp có phần mở rộng tệp chính xác hay không dựa trên loại tệp. Đây là một trong những phương tiện đơn giản nhất để kẻ tấn công sử dụng để che giấu hoặc che giấu sự hiện diện của các tệp trên hệ thống bị xâm nhập; bằng cách thay đổi tên tệp và phần mở rộng, kẻ tấn công có thể (nhiều lần, chính xác) cho rằng nếu quản trị viên phát hiện ra tệp, họ sẽ không thực sự truy cập và xác định bản chất thực sự của nó nếu tệp có phần mở rộng như .dll.

Trong chương này, chúng tôi sẽ thảo luận về các cách mà bạn, với tư cách là điều tra viên, có thể cố gắng xác định bản chất của một tệp thực thi. Tôi sẽ trình bày các công cụ và kỹ thuật bạn có thể sử dụng để thu thập thông tin về một tệp thực thi và nhận được manh mối về mục đích của nó. Thảo luận này sẽ không chỉ đơn giản là về phân tích phần mềm độc hại; thay vào đó, tôi sẽ trình bày các kỹ thuật để phân tích các tệp thực thi nói chung, trong đó phần mềm độc hại có thể chỉ là một loại tệp thực thi. Trong chương này, chúng tôi sẽ thảo luận về một số kỹ thuật phân tích, nhưng chúng tôi sẽ dừng ngay mọi cuộc thảo luận về việc dịch ngược mã nguồn hoặc sử dụng các công cụ như IDA Pro (www.hex-rays.com/idapro/). Việc sử dụng các ứng dụng dịch ngược mã nguồn là một chủ đề riêng biệt nằm trong một cuốn sách của riêng nó. Trong chương

này, chúng tôi sẽ bám sát các phương pháp và kỹ thuật mà hầu hết các quản trị viên và nhà phân tích pháp y sẽ có thể thực hiện.

Tuy nhiên, trước khi chúng tôi bắt đầu, bạn có thể hỏi tại sao chúng tôi làm việc này. Mục đích phân tích các tập tin phần mềm độc hại là gì? Đây có phải là những gì các nhà cung cấp chống vi-rút làm gì? Vâng, gần đây lịch sử đã chỉ ra rằng điều này có thể không còn là lựa chọn phù hợp. Ví dụ, cuối năm 2008 và đầu năm 2009 đã chứng kiến sự phổ biến của sâu Conficker (a.k.a. Downadup) thông qua các mạng công ty, do phần lớn là do thiếu cập nhật bảo mật (cụ thể là lỗ hổng Microsoft's MS MS08-067). Các tổ chức sử dụng các giải pháp chống vi-rút cho doanh nghiệp thấy mình bị nhiễm sâu máy tính khi một biến thể mới, chưa từng xuất hiện được phát hành. Tuy nhiên, đó là sự thiếu hiểu biết về bản chất của phần mềm độc hại nói chung, cũng như sự phụ thuộc gần như hoàn toàn vào một giải pháp chống vi-rút, gây ra nhiều rắc rối nhất cho cơ sở hạ tầng của công ty. Những người phản hồi đầu tiên có thể thực hiện một số phân tích ngay lập tức sẽ dẫn đến thời gian phản hồi và phục hồi nhanh hơn cho toàn bộ tổ chức, từ đó có thể hỗ trợ thêm cho nhóm phản ứng trong việc xác định nguyên nhân gốc của sự lây nhiễm. Phản ứng nhanh hơn cũng sẽ dẫn trực tiếp đến việc giảm khả năng truy cập dữ liệu nhạy cảm của kẻ xâm nhập (thông qua Trojan, backdoor hoặc bot) hoặc bởi chính phần mềm độc hại, vì việc phân tích phần mềm độc hại sẽ dẫn đến hiểu biết về sự lây nhiễm của nó (nó lây nhiễm như thế nào), cơ chế liên tục của nó (làm thế nào nó vẫn chạy trên các hệ thống) và bất kỳ dấu vết nào nó có thể để lại. Những dấu vết này sau đó có thể được sử dụng để định vị các hệ thống bị nhiễm khác, đặc biệt khi biến thể phần mềm độc hại không được các ứng dụng chống vi-rút nhận ra. Khả năng này có thể có nghĩa là sự khác biệt giữa thực hiện hành động quyết định, lý luận và thông báo ngay bây giờ và chờ vài ngày để đại diện nhà cung cấp xuất hiện, thu thập mẫu và sau đó tung ra một tệp chữ ký được cập nhật.

Điều đó đang được nói, hãy xem xét cách phân tích các tập tin thực thi.

2. Phân tích tĩnh

Phân tích tĩnh bao gồm thu thập thông tin từ một tệp thực thi mà không thực sự chạy hoặc khởi chạy tệp theo bất kỳ cách nào. Khi hầu hết mọi người mở một tệp thực thi trong Notepad (Tôi đã thực hiện việc này nhiều lần để minh họa một cái gì đó cho khách hàng) hoặc thậm chí là một trình soạn thảo hex, tất cả những gì họ thấy là một bó dữ liệu nhị phân dường như là rác vô nghĩa. Bây giờ và một lần nữa, bạn có thể thấy một từ mà bạn nhận ra, nhưng đối với hầu hết các từ đó không có ý nghĩa; nó có thể là bất cứ điều gì. Các nhà điều tra cần

lưu ý rằng các tệp thực thi phải tuân theo các quy tắc nhất định liên quan đến định dạng của chúng, vì có những điều cụ thể mà chúng ta có thể thấy trong một tệp thực thi được tìm thấy trên hệ thống Windows. Hiểu được các quy tắc đó cho phép chúng tôi đi sâu vào gobbledygook (Bạn có vui không khi tôi sử dụng thuật ngữ kỹ thuật khi tôi viết điều này?) của các tệp thực thi và trích xuất thông tin có ý nghĩa.

Tuy nhiên, trước khi chúng ta đào sâu vào một tập tin thực thi, có một vài điều chúng ta cần nói đến.

2.1. Định vị tệp để phân tích

Một trong những câu hỏi tôi được hỏi khá thường xuyên là làm thế nào để bạn xác định vị trí các tệp độc hại hoặc đáng ngờ trên một hệ thống hoặc trong một hình ảnh thu được từ một hệ thống?

Như chúng ta đã thảo luận trong Chương 3, một cách để xác định các tệp này yêu cầu bạn thu thập kết xuất bộ nhớ từ hệ thống, phân tích kết xuất bộ nhớ với một trong các công cụ được đề cập trong chương và xác định quy trình liên quan đến hoạt động đáng ngờ bằng công cụ Aaron Walters' Volatility. Khi bạn tìm thấy quá trình đó và phân tích khối EProcess, bạn sẽ có đường dẫn đến hình ảnh tệp thực thi. Sau đó, bạn có thể định vị tệp đó trong hệ thống tệp trong ảnh hệ thống dựa trên đường dẫn đó.

Một cách khác để xác định vị trí các tệp đáng ngờ là kiểm tra nội dung của các vị trí tự khởi động Windows Registry, như chúng ta đã thảo luận trong Chương 4. Nếu bạn tìm thấy một mục Đăng ký đáng ngờ, thì trong khóa Chạy, bạn có thể chỉ cần xác định vị trí tệp đó trong hệ thống hình ảnh.

Sử dụng các kỹ thuật phản hồi trực tiếp từ xa, như đã thảo luận trong Chương 1, bạn có thể tiếp cận với các hệ thống từ xa để thực hiện các truy vấn cần thiết để tìm kiếm các vị trí tự khởi động của Registry. Một phương pháp khác để thực hiện điều này là triển khai Phản hồi F trên hệ thống từ xa và khi bạn đã kết nối ổ đĩa từ xa, hãy sử dụng một công cụ như RegRipper để thu thập thông tin từ Registry của hệ thống từ xa, giống như bạn đã trích xuất các tệp trung tâm để phân tích. Bạn cũng có thể tự động trích xuất dữ liệu bằng cách đặt các lệnh cần thiết để triển khai rip.exe, phiên bản giao diện dòng lệnh (CLI) của RegRipper, vào một tập các tệp.

Một phương tiện khác để định vị các tệp thực thi độc hại hoặc đáng ngờ trên hình ảnh hệ thống là gắn hình ảnh dưới dạng ký tự ổ đĩa chỉ đọc trên hệ thống phân tích của bạn bằng Smart Mount (www.asrdata.com/SmartMount) hoặc Mount Image Pro, và sau đó quét ký tự ổ đĩa đó bằng một ứng dụng quét

chống vi-rút. Trong thực tế, do có nhiều trường hợp không tìm thấy các tệp độc hại thực tế bởi một ứng dụng chống vi-rút này hoặc một ứng dụng chống vi-rút khác, bạn có thể muốn quét ký tự ổ đĩa với nhiều hơn một trình quét chống vi-rút.

Claus Valca của blog Grand Stream Dreams ([http:// grandstreamdreams.blogspot.com/](http://grandstreamdreams.blogspot.com/)) đăng bài trên các ứng dụng chống vi-rút khác nhau có sẵn để sử dụng. Một số ứng dụng mà ông đề cập là miễn phí nhưng phiên bản ứng dụng đầy đủ tính năng hơn có sẵn với một khoản phí. Tuy nhiên, trong nhiều trường hợp, phiên bản có sẵn miễn phí cung cấp khả năng quét phần mềm độc hại, đây là điều chúng tôi thực sự quan tâm ở đây. Một số ứng dụng quét chống vi-rút mà Claus đề cập có thể được định cấu hình hoặc được viết riêng để chạy từ thumb drive, cho phép bạn tải xuống, cập nhật và sử dụng ứng dụng mà không cần phải giới hạn tất cả chúng trong một hệ thống. Truy cập blog của anh ấy và tìm kiếm phần mềm chống vi-rút và các công cụ phần mềm độc hại để xem danh sách các bài viết trên blog về các chủ đề đó.

Tuy nhiên, ngay cả với trạng thái của các ứng dụng quét chống vi-rút như ngày nay, chúng vẫn có một nhược điểm chính: Vì các ứng dụng này dựa trên chữ ký, tất cả các tác giả phần mềm độc hại cần làm là thực hiện một thay đổi nhỏ cho phần mềm của chúng, sau đó biên dịch lại và triển khai lại và phần mềm độc hại có thể không được phát hiện. Tôi đã thấy các trường hợp (tôi đã có một vài bản thân mình) và nghe nói về những người khác đã gửi phần mềm độc hại đến các trang web khác nhau (chẳng hạn như VirusTotal. Com) để xem xét và tìm thấy tệp thực thi không bị phát hiện hoặc nhận dạng bởi 35 (hoặc hơn) các ứng dụng quét chống vi-rút khác nhau. Như vậy, chúng ta cần phát triển các phương tiện khác nhau để có thể xác định các tệp thực thi độc hại, trên các hệ thống hoặc trong các hình ảnh hệ thống. Một kỹ thuật khác ngoài những gì đã đề cập sẽ là thực hiện một phiên bản sâu hơn của phân tích chữ ký tệp; nghĩa là, thay vì chỉ đơn giản là tìm kiếm các chữ cái MZ trong hai byte đầu tiên của một tệp thực thi tiềm năng và sau đó so sánh nó với phần mở rộng tệp, hy vọng tìm thấy exe, hoặc dll hay một phần mở rộng hợp lệ khác, chúng ta nên khai thác một chút sâu sắc hơn. Ngoài chữ ký tệp ban đầu, phần còn lại của tệp có cấu trúc tệp phù hợp cho loại tệp đó không? Bạn cũng có thể xác minh rằng các tệp được ký điện tử bằng cách sử dụng sigcheck.exe của Microsoft ([http://technet.microsoft.com/en-us/sysinternals / bb897441.aspx](http://technet.microsoft.com/en-us/sysinternals/bb897441.aspx)) hoặc sử dụng WFPCheck, như được mô tả trong Chương 5, để cố gắng xác định vị trí các tệp được bảo vệ bởi Windows File Protection (WFP) đã được thay thế hoặc sửa đổi.

Bất kể các phương tiện bạn sử dụng để định vị và xác định các tệp thực thi có thể đáng ngờ hoặc độc hại (lý tưởng là sử dụng nhiều hơn một trong các kỹ thuật đã nói ở trên), bạn nên chắc chắn ghi lại kỹ lưỡng những gì bạn làm, cũng như kết quả quét hoặc tìm kiếm.

2.2. *Tài liệu tập tin*

Trước khi phân tích hoặc đào sâu vào tập tin thực thi theo bất kỳ cách nào, điều đầu tiên bạn nên làm là ghi lại nó. Tuy nhiên, nó có một niềm tin được tổ chức rộng rãi mà những người định hướng kỹ thuật ghét phải ghi chép lại bất cứ điều gì. Vâng, điều này là đúng, ít nhất là một phần. Tôi không thể cho bạn biết số lần tôi đã phản hồi về một sự cố (tại chỗ hoặc từ xa) và được người trả lời cho biết, chúng tôi đã tìm thấy một tập tin. Khi được hỏi, bạn đã tìm thấy tập tin ở đâu? Người phản hồi trả lời với đôi mắt mở to, nhìn chăm chú. Trường hợp tệp được tìm thấy có thể cực kỳ hữu ích trong việc thêm ngữ cảnh vào thông tin khác và giúp bạn tìm hiểu điều gì đã xảy ra.

Vì vậy, điều đầu tiên bạn cần làm là ghi lại toàn bộ đường dẫn và vị trí của tệp bạn tìm thấy; nó nằm trên hệ thống nào, đường dẫn đầy đủ đến tập tin là gì và ai đã tìm thấy nó và khi nào.

Cảnh báo

Một điều mà nhiều người kỹ thuật dường như không nhận ra là trên hệ thống máy tính (không chỉ trên Windows), một tệp có thể được đặt tên gần như bất cứ thứ gì. Theo dõi bất kỳ danh sách công khai nào trong một khoảng thời gian và bạn sẽ tìm thấy các bài đăng mà ai đó sẽ nói, “tôi đã tìm thấy tệp này trên hệ thống của mình và một tìm kiếm của Google cho tôi biết rằng đó là vô hại ...” Tìm kiếm thông tin về một tệp chỉ dựa trên tên của tệp có thể xuất hiện một số thông tin thú vị hoặc hữu ích, nhưng thông tin đó không nên được coi là kết thúc cuộc điều tra. Tôi đã trả lời một sự cố một lần khi nhân viên công nghệ thông tin tại chỗ (IT) đã tìm thấy một số tệp trên một hệ thống bị nhiễm và sau đó Googled để biết thông tin về từng tệp. Nhập tên của một trong những tệp họ tìm thấy, họ thấy rằng tệp đó là hợp pháp, do Microsoft cung cấp và họ đã kết thúc cuộc điều tra của họ ở đó. Tuy nhiên, bằng cách kiểm tra thêm tệp bằng các kỹ thuật được trình bày trong chương này, tôi có thể xác định rằng tệp trên thực tế là phần mềm độc hại mà tôi đang phản hồi.

Tùy thuộc vào cách tệp đáng ngờ ban đầu được định vị, bạn có thể đã có tài liệu cho tệp có sẵn. Ví dụ, nếu bạn trả lời một hệ thống trực tiếp và đã sử dụng một (hoặc nhiều) kỹ thuật phản hồi được đề cập trong Chương 1, có khả

năng bạn đã có tài liệu, như đường dẫn đầy đủ đến tệp, có sẵn. Điều tương tự cũng đúng nếu bạn định vị tệp trong ảnh hệ thống bằng ProDiscover hoặc một số công cụ hoặc kỹ thuật phân tích pháp y khác.

Một khía cạnh khác của tệp quan trọng đối với tài liệu là hệ điều hành và phiên bản mà nó được đặt. Các hệ điều hành Windows khác nhau giữa các phiên bản và thậm chí giữa các Gói dịch vụ trong cùng một phiên bản. Hiệu ứng mà phần mềm độc hại gây ra cho mục tiêu có thể phụ thuộc hoặc thậm chí khác nhau tùy thuộc vào phiên bản Windows mà nó được đặt. Ví dụ: e-mail trò lừa bịp của gấu Teddy xác định tệp jdbgmgr.exe là phần mềm độc hại (nó được gọi là vi-rút Teddy Teddy Bear vì biểu tượng cho tệp này là một con gấu bông) và bảo người đọc xóa ngay lập tức tập tin. Nếu điều này được thực hiện trên Windows NT 4.0, tệp sẽ bị xóa. Tuy nhiên, trên Windows 2000, WFP sẽ ngay lập tức thay thế tệp. Tập hợp các tệp được bảo vệ bởi WFP khác nhau giữa Windows 2000 và Windows XP. Trở lại năm 2000, Benny và Ratter đã phát hành virus bằng chứng khái niệm W32.Stream sử dụng các luồng dữ liệu thay thế NTFS (ADS; xem Chương 5 để biết giải thích chi tiết về ADS). Nếu vi-rút xâm nhập vào hệ thống Windows có hệ thống tệp được định dạng là FAT / FAT32, thì vi-rút có vẻ hoạt động khác, nhưng chỉ vì hệ thống tệp FAT không hỗ trợ ADS.

Bên cạnh việc lưu ý nơi tìm thấy tệp trong hệ thống tệp và phiên bản Windows nào trong quy trình phản hồi của bạn, bạn cũng nên thu thập thông tin bổ sung về tệp, như thời gian MAC của tệp và mọi tham chiếu đến tệp đó trong hệ thống tệp (ví dụ: các phím tắt trong thư mục StartUp của người dùng) hoặc Registry, mà bạn có thể nhận thấy trong lần kiểm tra ban đầu của mình.

Cảnh báo

Các nhà điều tra cần phải rất cẩn thận khi ban đầu tiếp cận một hệ thống, đặc biệt là một hệ thống vẫn đang chạy. Trước đó trong cuốn sách này, chúng tôi đã thảo luận về Nguyên tắc trao đổi của Locard và thực tế là các tìm kiếm văn bản ASCII và Unicode không phải lúc nào cũng hoạt động trên các tìm kiếm của Registry, vì một số giá trị được lưu trữ ở định dạng nhị phân. Bất cứ điều gì một điều tra viên làm trên một hệ thống sẽ để lại các tác động trên hệ thống đó, vì vậy nếu bạn tìm thấy một tệp bất thường, hãy giới hạn tìm kiếm của bạn để biết thêm thông tin về tệp càng nhiều càng tốt. Bất kỳ hoạt động nào bạn tham gia nên được ghi lại kỹ lưỡng.

Tài liệu của bạn càng đầy đủ thì càng tốt. Đó là một ý tưởng tốt để tạo thói quen làm điều này cho mọi cuộc điều tra, vì nó sẽ giúp bạn tiết kiệm rất

nhiều đau khổ trong tương lai. Hơn nữa, điều này tạo thành một cách tiếp cận thực hành tốt nhất.

Một bước khác bạn sẽ cần phải làm theo để ghi lại tệp là tính toán băm mật mã cho tệp. Băm mật mã được sử dụng trong bảo mật thông tin và pháp y máy tính để đảm bảo tính toàn vẹn của tệp; nghĩa là, không có thay đổi nào được thực hiện đối với tệp. Một thuật toán băm phổ biến là hàm MD5, lấy đầu vào có độ dài tùy ý và tạo ra hàm băm đầu ra 128 bit thường được biểu thị bằng 32 ký tự thập lục phân. Mọi thay đổi đối với đầu vào, thậm chí chuyển đổi một bit đơn, sẽ dẫn đến hàm băm MD5 khác nhau. Mặc dù thiếu sót trong thuật toán MD5 cho phép va chạm đã được ghi nhận (<http://en.wikipedia.org/wiki/Md5>), thuật toán vẫn hữu ích cho pháp y máy tính. Một thuật toán băm phổ biến khác là SHA-1 (<http://en.wikipedia.org/wiki/Sha-1>). Các tổ chức như Thư viện tham chiếu phần mềm quốc gia (NSRL) tại NIST sử dụng thuật toán SHA-1 khi tính toán băm mật mã cho các đĩa CD tập dữ liệu tham chiếu (RDS). Các bộ tham chiếu như thế này cho phép các nhà điều tra một mô-đun giảm dữ liệu bằng cách lọc ra các tệp tin nổi tiếng (hợp pháp) và các loại dữ liệu nổi tiếng (đã biết là phần mềm độc hại) từ bộ dữ liệu.

Khi bạn đã tính băm MD5 của một tệp thực thi mà bạn cho rằng có thể có bản chất độc hại, bạn có thể truy cập trang web VirusTotal.com và đăng chính tệp đó hoặc băm MD5 để xem xét. Nếu bạn đăng tệp thực thi để phân tích, trang web sẽ quét tệp với khoảng 35 ứng dụng quét chống vi-rút khác nhau. Nếu bạn gửi hàm băm MD5, nó được so sánh với cơ sở dữ liệu băm được duy trì tại trang web. Trang web này là một tài nguyên tuyệt vời cho những người có quyền truy cập hạn chế vào nhiều hơn một hoặc hai ứng dụng quét.

Một thuật toán băm hữu ích khác đã được Jesse Kornblum triển khai trong công cụ của mình có tên là ssdeep (dựa trên spamsum của Tiến sĩ Andrew Tridgell), có sẵn từ <http://ssdeep.sourceforge.net/>. Ssdeep.exe tính toán ngữ cảnh được kích hoạt băm piecewise băm nhỏ (www.Dfrws.org/2006/proceedings/12-Kornblum.pdf), có nghĩa là thay vì tính toán một hàm băm mật mã trên toàn bộ tệp bắt đầu, nó sẽ tính toán băm bằng cách sử dụng một cách tiếp cận từng phần, băm các phần có kích thước ngẫu nhiên (ví dụ: 4 KB) tại một thời điểm. Kỹ thuật này không chỉ tạo ra một hàm băm mà sau đó có thể được sử dụng để xác minh tính toàn vẹn của tệp gốc mà còn có thể được sử dụng để xem hai tệp tương tự có thể như thế nào. Ví dụ: nếu một tài liệu Word được băm bằng ssdeep.exe và sau đó sửa đổi một chút (thêm / xóa văn bản, thay đổi định dạng, v.v.), sau đó băm được tính toán lại, ssdeep.exe sẽ

có thể hiển thị các tệp tương tự như thế nào . Bạn cũng có thể sử dụng kỹ thuật này với các loại tệp khác, chẳng hạn như hình ảnh, video và tệp âm thanh.

Khi bạn đã ghi lại thông tin về tệp, bạn có thể bắt đầu thu thập thông tin từ chính tệp đó.

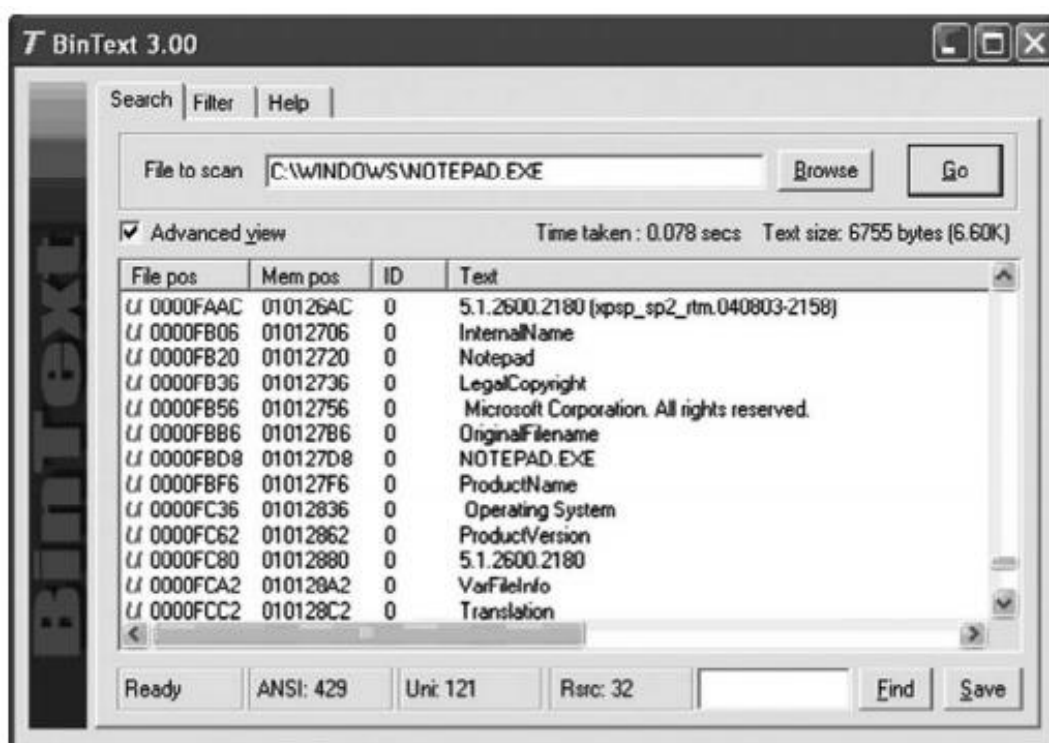
2.3. Phân tích

Một trong những bước đầu tiên của phân tích tĩnh mà hầu hết các nhà điều tra tham gia là quét tệp đáng ngờ bằng phần mềm chống vi-rút. Đây là một cách tuyệt vời để bắt đầu, nhưng đừng ngạc nhiên nếu quá trình quét chống vi-rút không có gì dứt khoát. Malcode mới đang được phát hành mọi lúc. Trên thực tế, một công ty chống vi-rút đã phát hành một báo cáo vào tháng 1 năm 2007, nhìn lại năm trước, trong đó họ đã xác định được tổng cộng 207.684 mối đe dọa khác nhau mà sản phẩm chống vi-rút của họ bảo vệ và 41.536 mẫu mã độc mới mà sản phẩm của họ phát hiện. Quét tệp tin đáng ngờ có thể cung cấp cho bạn cái nhìn sâu sắc về bản chất của tệp tin, nhưng đừng quá lo lắng nếu phản hồi bạn nhận được là không có virus phát hiện ra. Quét bằng nhiều công cụ chống vi-rút cũng có thể cung cấp một cái nhìn toàn diện hơn về tệp.

Bước tiếp theo mà hầu hết các nhà điều tra sẽ thực hiện với một tệp thực thi đáng ngờ là chạy nó thông qua chuỗi.exe (<http://technet.microsoft.com/en-us/sysinternals/bb897439.aspx>), trích xuất tất cả các chuỗi ASCII và Unicode của một chiều dài cụ thể. Điều này có thể rất hữu ích, trong đó điều tra viên có thể biết được bản chất của tệp từ các chuỗi trong tệp. Phiên bản mới nhất của chuỗi.exe (kể từ khi viết bài này) cho phép bạn tìm kiếm cả chuỗi ASCII và Unicode, cũng như in phần bù của vị trí trong tệp. Phần bù này sẽ cho bạn biết phần nào chuỗi xuất hiện trong và cung cấp ngữ cảnh cho chuỗi (chúng ta sẽ thảo luận về phần và phần tiêu đề sau trong chương này). Bạn thậm chí có thể chạy chương trình String.exe để tìm kiếm các chuỗi cụ thể trong tất cả các tệp, sử dụng dòng lệnh ví dụ được liệt kê tại trang Web cho ứng dụng.

Trước đây, tôi đã hỗ trợ điều tra một tệp tin được lấy từ một hệ thống đang tăng lưu lượng truy cập trên Internet từ trong cơ sở hạ tầng của công ty. Tệp hóa ra là vi-rút IE0199 (www.f-safe.com/v-descs/khángtc.shtml) sẽ lây nhiễm một hệ thống và bắt đầu gửi lưu lượng truy cập đến cơ sở hạ tầng viễn thông của Bulgaria. Chúng tôi đã tìm thấy các chuỗi ASCII trong tệp tạo ra một bản tuyên ngôn, và may mắn thay, ai đó trong nhóm của chúng tôi đã được đào tạo tiếng Nga trong Quân đội Hoa Kỳ và có thể diễn giải những gì chúng tôi đã tìm thấy. Rõ ràng, tác giả đã rất buồn với giá phải trả cho truy cập Internet ở Bulgaria và muốn tiến hành một cuộc tấn công từ chối dịch vụ (DoS) đối với cơ sở hạ tầng.

Một tiện ích hữu ích khác để tìm kiếm các chuỗi trong tệp nhị phân là BinText, được sử dụng từ Foundstone (thuộc sở hữu của McAfee, Inc.). BinText sẽ định vị tất cả các chuỗi ASCII, Unicode và chuỗi tài nguyên trong tệp nhị phân và hiển thị chúng trong giao diện người dùng đồ họa đẹp (GUI), cùng với phần bù với tệp nhị phân nơi tìm thấy chuỗi. Hình 6.1 minh họa một số chuỗi được tìm thấy trong notepad.exe.



Hình 6. 1 Notepad.exe Open in BinText

Mặc dù các chuỗi được tìm thấy trong tệp không vẽ ra một bức tranh hoàn chỉnh về những gì tệp thực hiện, chúng có thể cung cấp cho bạn manh mối. Hơn nữa, các chuỗi có thể nằm ngoài ngữ cảnh, ngoài vị trí của chúng. Ví dụ, trong Hình 6.1, chúng ta thấy các chuỗi là Unicode (xem phần U U ở bên trái của giao diện) và chúng dường như là một phần của thông tin phiên bản tệp (sẽ nói thêm về điều này sau trong chương này). Các chuỗi khác có thể không có cùng mức ngữ cảnh trong tệp. Một tùy chọn khác là các chuỗi xuất hiện kỳ lạ hoặc duy nhất (thực sự nghiêm trọng, tôi thực sự đã tìm thấy chuỗi siêu supercalifragilisticexpialidocious, trong một tệp; trung thực) trong tệp có thể được sử dụng để tìm kiếm trong các tệp khác, cũng như trên Internet. Kết quả của các tìm kiếm này có thể cung cấp cho bạn manh mối để hỗ trợ phân tích sâu hơn (tĩnh hoặc động) của tệp thực thi.

Rất nhiều trang web có sẵn trên phần mềm độc hại kỹ thuật đảo ngược hoặc thậm chí các ứng dụng hợp pháp, và thật kỳ lạ, tất cả đều chỉ ra một số kỹ

thuật cốt lõi tương tự để thu thập thông tin từ các tệp thực thi, cũng như sử dụng một số công cụ rất giống nhau. Hai trong số các công cụ mà chúng tôi sẽ sử dụng trong suốt các phần tiếp theo của chương này là pedump.exe và pevview.exe.

Vào tháng 2 năm 2002, bài viết đầu tiên trong số hai bài viết của Matt Pietrek, có tựa đề là An In-Depth Look into the PE File Format, đã được xuất bản. Trong các bài viết này, Matt không chỉ mô tả chi tiết các khía cạnh khác nhau của định dạng tệp thực thi di động (PE), mà còn cung cấp một công cụ CLI có tên pedump.exe (tìm thấy tại www.wheaty.net) mà bạn có thể sử dụng để trích xuất thông tin chi tiết từ tiêu đề của tệp PE. Thông tin được trích xuất bởi pedump.exe được gửi đến STDOUT, vì vậy nó có thể được xem dễ dàng tại bàn điều khiển hoặc được chuyển hướng đến một tệp để phân tích sau.

Bạn có thể tìm thấy phần 1 của các bài viết của Matt Pietrek, tại <http://msdn.microsoft.com/en-us/Magazine/cc495805.aspx>. Bạn có thể tìm thấy phần 2 tại <http://msdn.microsoft.com/en-us/magazine/cc495808.aspx>.

Một công cụ hữu ích khác để khám phá phần bên trong của các tệp Windows PE là pevview.exe (www.magma.ca/~wjr/), từ Wayne Radburn. Pevview.exe là một công cụ GUI cho phép bạn xem các thành phần khác nhau của tiêu đề PE (và các phần còn lại) theo định dạng được bố trí độc đáo. Phiên bản mới nhất của pevview.exe có sẵn tại thời điểm viết bài này là Phiên bản 0.96 và phiên bản đó không bao gồm khả năng lưu những gì được xem trong GUI vào một tệp.

Cả hai công cụ này đều không được cung cấp trên DVD đi kèm, do vấn đề cấp phép và phân phối. Ngoài ra, truy cập các trang web để có được các công cụ sẽ đảm bảo rằng bạn có các phiên bản mới nhất hiện có. Tuy nhiên, DVD chứa mã Perl để truy cập cấu trúc tệp PE. Tập lệnh Perl pedmp.pl sử dụng mô-đun File::ReadPE Perl để truy cập nội dung của tiêu đề PE và phân tích các cấu trúc khác nhau. Kịch bản và mô-đun Perl được cung cấp cho mục đích giáo dục và hướng dẫn để bạn có thể thấy những gì diễn ra sau hậu trường với các công cụ khác. Ngoài ra, mã Perl được viết là độc lập với nền tảng nhất có thể; nghĩa là, khi các giá trị byte được truy xuất từ tệp thực thi, hàm Perl unpack() được sử dụng với các chuỗi giải nén buộc các giá trị theo thứ tự littleendian. Bằng cách này, bạn có thể chạy các tập lệnh trên Windows, Linux và thậm chí cả Mac OS X (có lợi cho việc phân tích, vì không chắc là trên Linux hoặc Mac OS X, bạn sẽ vô tình sử dụng phần mềm độc hại Windows và lây nhiễm hệ

thống), vì vậy bạn không bị hạn chế thực hiện phân tích trên một nền tảng duy nhất.

2.3.1. Tiêu đề PE

Tại www.microsoft.com/whdc/system/platform/ffware/PMFFFF.msp, Microsoft đã ghi lại kỹ lưỡng định dạng của các tệp PE (cũng như Định dạng tệp đối tượng chung hoặc COFF, được tìm thấy trên các hệ thống VAX / VMS) và đã công khai tài liệu đó. Microsoft cũng đã công khai hầu hết các cấu trúc được sử dụng trong các tiêu đề tệp, như là một phần của tài liệu cho cấu trúc API ImageHlp (<http://msdn2.microsoft.com/en-gb/library/ms680198.aspx>). Với tài nguyên này và các tài nguyên khác, chúng tôi có thể hiểu cấu trúc của tệp PE, đi sâu vào độ sâu của nó và trích xuất thông tin có thể được sử dụng cho chúng tôi trong quá trình điều tra.

Một tập tin PE có thể được chia thành một số lĩnh vực quan tâm (tôi ngần ngại nói rằng các phần, vì chúng tôi sẽ sử dụng thuật ngữ này cho một mục đích cụ thể trong cuộc thảo luận của chúng tôi). Phần đầu tiên và có lẽ là quan trọng nhất của tệp PE (nếu không phải là phần quan trọng nhất, thì một trong những phần hay nhất của câu đố đam mê) là chữ ký của tệp. Đối với các tệp thi hành trên các hệ thống Windows, chữ ký tệp bao gồm các chữ cái MZ, được tìm thấy trong hai byte đầu tiên của tệp. Như đã lưu ý trước đó trong cuốn sách, hai chữ cái này là chữ cái đầu của Mark Zbikowski (http://en.wikipedia.org/wiki/Mark_Zbikowski), kiến trúc sư của Microsoft đã ghi công vào việc thiết kế định dạng tệp thực thi. Tuy nhiên, như bạn thấy, nó sẽ mất nhiều hơn hai chữ cái đó và một .exe siêu mềm ở cuối tên tệp để tạo tệp thực thi.

Tên viết tắt Mark Mark là chữ ký cho cấu trúc 64 byte được gọi là IMAGE_DOS_HEADER. Các yếu tố quan trọng của cấu trúc này là hai byte đầu tiên (số ma thuật của Wap, 0x5a4d ở định dạng thập lục phân nhỏ, hoặc MZ) và giá trị DWORD (4 byte) cuối cùng, được gọi là e_lfanew. Giá trị này được xác định trong tệp tiêu đề ntimage.h là địa chỉ tệp (offset) của tiêu đề EXE mới; đó là phần bù mà chúng ta sẽ tìm thấy chữ ký cho phần đầu của cấu trúc IMAGE_NT_HEADERS. Giá trị e_lfanew trở đến vị trí của tiêu đề PE, cho phép Windows thực hiện đúng tệp hình ảnh. Hình 6.2 minh họa các giá trị này từ một tệp thực thi được mở trong trình soạn thảo hex.

Size	Name	Description
2 bytes	<i>Machine</i>	Designates the architecture type of the computer; the program can be run only on a system that emulates this type
2 bytes	<i>Number of Sections</i>	Designates how many sections (IMAGE_SECTION_HEADERS) are included in the PE file
4 bytes	<i>TimeDateStamp</i>	The time and date that the linker created the image, in UNIX time format (i.e., number of seconds since midnight, 1 Jan 1970). This normally indicates the system time on the programmer's computer when he compiled the executable
4 bytes	<i>Pointer to Symbol Table</i>	Offset to the symbol table (0 if no COFF symbol table exists)
4 bytes	<i>Number of Symbols</i>	Number of symbols in the symbol table
2 bytes	<i>Size of Optional Header</i>	Size of the IMAGE_OPTIONAL_HEADER structure; determines whether the structure is for a 32-bit or 64-bit architecture
2 bytes	<i>Characteristics</i>	Flags designating various characteristics of the file

Bảng 6 1 IMAGE_FILE_HEADER Giá trị cấu trúc

Hình 6.4 minh họa IMAGE_FILE_HEADER của một ứng dụng mẫu được mở trong PView.

pFile	Data	Description	Value
000000BC	014C	Machine	IMAGE_FILE_MACHINE_I386
000000BE	0003	Number of Sections	
000000C0	40C2E1C2	Time Date Stamp	2004/06/06 Sun 09:20:02 UTC
000000C4	00000000	Pointer to Symbol Table	
000000C8	00000000	Number of Symbols	
000000CC	00E0	Size of Optional Header	
000000CE	010F	Characteristics	
	0001		IMAGE_FILE_RELOCS_STRIPPED
	0002		IMAGE_FILE_EXECUTABLE_IMAGE
	0004		IMAGE_FILE_LINE_NUMS_STRIPPED
	0008		IMAGE_FILE_LOCAL_SYMS_STRIPPED
	0100		IMAGE_FILE_32BIT_MACHINE

Hình 6. 4 IMAGE_FILE_HEADER được xem trong PView

Đối với các nhà điều tra pháp y, giá trị TimeDateStamp có thể có ý nghĩa khi điều tra một tệp thực thi, vì nó cho thấy khi trình liên kết tạo tệp hình ảnh (các nhà điều tra cũng nên biết rằng giá trị này có thể được sửa đổi bằng trình soạn thảo hex mà không có bất kỳ ảnh hưởng nào đến việc thực thi của chính tệp tin). Điều này thường cho biết thời gian hệ thống trên máy tính lập trình viên của chương trình khi lập trình viên biên dịch tệp thực thi và có thể là đầu mỗi khi chương trình này được xây dựng. Khi thực hiện phân tích tệp, số phần được báo cáo trong cấu trúc IMAGE_FILE_HEADER phải khớp với số phần trong tệp. Ngoài ra, nếu phần mở rộng tệp đã bị thay đổi, giá trị Đặc điểm sẽ cung cấp một số manh mối về bản chất thực sự của tệp; chẳng hạn, trong giá trị Đặc điểm được minh họa trong Hình 6.4, nếu cờ IMAGE_FILE_DLL được đặt (tức là 0x2000), tệp thực thi là thư viện liên kết động (DLL) và không thể chạy trực tiếp. Một lớp các tệp thường xuất hiện dưới dạng DLL là các đối tượng trợ giúp trình duyệt hoặc BHO (được thảo luận trong Chương 4). Đây là các DLL được tải bởi Internet Explorer và có thể cung cấp tất cả các cách thức chức năng. Trong một số trường hợp, các DLL này là hợp pháp (chẳng hạn như BHO được sử dụng để tải Adobe, Acrobat Reader khi tệp PDF được truy cập qua trình duyệt), nhưng trong nhiều trường hợp, các BHO này có thể là phần mềm gián điệp hoặc phần mềm quảng cáo. Trang MSDN cho IMAGE_FILE_HEADER cung cấp danh sách các giá trị không đổi có thể có thể bao gồm trường Characteristics.

Giá trị cho kích thước của cấu trúc IMAGE_OPTIONAL_HEADER (<http://msdn.microsoft.com/en-gb/library/ms680339.aspx>) rất quan trọng để phân tích tệp, vì nó cho bạn biết liệu tiêu đề tùy chọn có dành cho 32-bit hay ứng dụng 64 bit. Giá trị này tương ứng với số ma thuật của người Hồi giáo, cấu trúc IMAGE_OPTIONAL_HEADER, nằm ở hai byte đầu tiên của cấu trúc; giá trị 0x10b biểu thị hình ảnh thực thi 32 bit, giá trị 0x20b biểu thị hình ảnh thực thi 64 bit và giá trị 0x107 biểu thị hình ảnh ROM. Trong cuộc thảo luận của chúng tôi, chúng tôi sẽ tập trung vào cấu trúc IMAGE_OPTIONAL_HEADER32 cho hình ảnh thực thi 32 bit. Hình 6.5 minh họa IMAGE_OPTIONAL_HEADER của một ứng dụng mẫu được xem trong PEView.

pFile	Data	Description	Value
00000108	0007C000	Size of Image	
0000010C	00001000	Size of Headers	
00000110	0007F430	Checksum	
00000114	0002	Subsystem	IMAGE_SUBSYSTEM_WINDOWS_GUI
00000116	0000	DLL Characteristics	
00000118	00100000	Size of Stack Reserve	
0000011C	00001000	Size of Stack Commit	
00000120	00100000	Size of Heap Reserve	
00000124	00001000	Size of Heap Commit	
00000128	00000000	Loader Flags	
0000012C	00000010	Number of Data Directories	

Hình 6. 5 IMAGE_OPTIONAL_HEADER được xem trong PEView

Các giá trị hiển thị trong Hình 6.5 chỉ ra rằng ứng dụng mẫu được thiết kế cho hệ thống con GUI của Windows và giá trị Đặc tính DLL là 0000 chỉ ra rằng ứng dụng mẫu không phải là DLL.

Như bạn đã thấy trước đó, kích thước của cấu trúc IMAGE_OPTIONAL_HEADER được lưu trữ trong cấu trúc IMAGE_FILE_HEADER, chứa một số giá trị có thể hữu ích cho các phân tích chi tiết nhất định của các tệp thực thi. Mức độ phân tích này nằm ngoài phạm vi của chương này.

Tuy nhiên, giá trị quan tâm trong IMAGE_OPTIONAL_HEADER là giá trị Hệ thống con, cho hệ điều hành biết hệ thống con nào được yêu cầu để chạy hình ảnh. Microsoft thậm chí còn cung cấp một bài viết Cơ sở Kiến thức (90493, <http://support.microsoft.com/kb/90493>) mô tả cách (và bao gồm mã mẫu) để xác định hệ thống con của ứng dụng. Lưu ý rằng trang MSDN của cấu trúc IMAGE_OPTIONAL_HEADER cung cấp nhiều giá trị có thể hơn cho Hệ thống con so với bài viết Cơ sở tri thức.

Một giá trị khác mà các nhà điều tra sẽ quan tâm là giá trị addressofEntryPoint trong IMAGE_OPTIONAL_HEADER. Đây là một con trỏ đến chức năng điểm nhập liên quan đến địa chỉ cơ sở hình ảnh. Đối với các tệp thi hành, đây là nơi mã cho ứng dụng bắt đầu. Tầm quan trọng của giá trị này sẽ trở nên rõ ràng sau trong chương này.

Ngay sau cấu trúc IMAGE_OPTIONAL_HEADER là các cấu trúc IMAGE_DATA_DIRECTORY (<http://msdn.microsoft.com/en-us/library/ms680305.aspx>). Các thư mục dữ liệu này, được minh họa trong Hình 6.6, hoạt động như một cấu trúc thư mục để lấy thông tin trong tệp PE, chẳng hạn như các bảng IMPORT NAME và IMPORT ADDRESS (danh sách các hàm DLL được nhập vào và sử dụng bởi tệp thực thi), bảng EXPORT (đối với

DLL, vị trí của các hàm được xuất), địa chỉ bắt đầu và kích thước của thư mục Debug (<http://msdn.microsoft.com/en-us/library/ms680305.aspx>), nếu có, và thư mục Resource, để đặt tên cho một vài (trong số 16 thư mục có thể). Mỗi thư mục dữ liệu được liệt kê dưới dạng địa chỉ ảo tương đối (RVA) và giá trị kích thước và theo thứ tự cụ thể, được xác định.

00000138	00078004	RVA	IMPORT Table
0000013C	00000028	Size	
00000140	0007A000	RVA	RESOURCE Table
00000144	0000114C	Size	
00000148	00000000	RVA	EXCEPTION Table
0000014C	00000000	Size	
00000150	00000000	Offset	CERTIFICATE Table
00000154	00000000	Size	

Hình 6. 6 Trích dẫn cấu trúc IMAGE_DATA_DIRECTORY được xem trong PEXView

Hình 6.6 cho thấy bốn trong số 16 thư mục dữ liệu có sẵn trong ứng dụng mẫu. Các giá trị được liệt kê là các vị trí hoặc độ lệch trong tệp PE nơi chứa thông tin. Ví dụ, dòng đầu tiên trong Hình 6.6 cho thấy bảng IMPORT nằm ở offset 0x138, giá trị tại vị trí đó (0x78004) và tên của giá trị (RVA). Từ thông tin hiển thị trong Hình 6.6, chúng ta có thể thấy rằng ứng dụng mẫu có cả bảng IMPORT và bảng RESOURCE.

Một RVA được sử dụng trong một tệp thực thi khi một địa chỉ của một biến (ví dụ) cần được chỉ định nhưng không thể sử dụng địa chỉ mã hóa cứng. Điều này là do hình ảnh thực thi sẽ không được tải vào cùng một vị trí trong bộ nhớ trên mọi hệ thống. RVA được sử dụng do nhu cầu có thể chỉ định các vị trí trong bộ nhớ độc lập với vị trí nơi tệp được tải. Một RVA về cơ bản là một phần bù trong bộ nhớ, liên quan đến nơi tệp được tải. Công thức tính toán RVA như sau: $RVA = (\text{Địa chỉ Target}) - (\text{Địa chỉ Load})$ Để lấy địa chỉ bộ nhớ thực (a.k.a. Địa chỉ ảo hoặc VA), chỉ cần thêm địa chỉ Load vào RVA.

Phần cuối cùng của tệp PE mà chúng tôi quan tâm tại thời điểm này là các cấu trúc IMAGE_SECTION_HEADER (<http://msdn.microsoft.com/en-us/library/ms680341.aspx>). Cấu trúc IMAGE_FILE_HEADER chứa một giá trị chỉ định số phần nên có trong tệp PE và do đó, số lượng cấu trúc IMAGE_SECTION_HEADER cần được đọc. Các cấu trúc IMAGE_SECTION_HEADER có kích thước 40 byte và chứa tên của phần (dài tám ký tự), thông tin về kích thước của phần cả trên đĩa và trong bộ nhớ (bạn đã thấy tham chiếu đến phần này trong Chương 3) và các đặc điểm của phần (nghĩa

là phần đó có thể được đọc, ghi vào, thực thi, v.v.). Hình 6.7 minh họa cấu trúc của IMAGE_SECTION_HEADER.

pFile	Data	Description	Value
000001B0	2E 74 65 78	Name	.text
000001B4	74 00 00 00		
000001B8	000776EC	Virtual Size	
000001BC	00001000	RVA	
000001C0	00078000	Size of Raw Data	
000001C4	00001000	Pointer to Raw Data	
000001C8	00000000	Pointer to Relocations	
000001CC	00000000	Pointer to Line Numbers	
000001D0	0000	Number of Relocations	
000001D2	0000	Number of Line Numbers	
000001D4	60000020	Characteristics	
	00000020		IMAGE_SCN_CNT_CODE
	20000000		IMAGE_SCN_MEM_EXECUTE
	40000000		IMAGE_SCN_MEM_READ

Hình 6. 7 IMAGE_SECTION_HEADER được xem trong PEView

Một lưu ý khi xem tên các phần là không có yêu cầu khó và nhanh về việc tên phần nên hay có thể là gì. Tên phần không có gì khác ngoài một loạt các ký tự (tối đa tám) có thể là bất cứ thứ gì. Thay vì trên mạng “.text”, tên của phần có thể là “kiểu thời gian”. Thay đổi tên không ảnh hưởng đến chức năng của tệp PE. Trên thực tế, một số tác giả phần mềm độc hại sẽ chỉnh sửa và sửa đổi tên phần, có lẽ để loại bỏ các nhà phân tích phần mềm độc hại thiếu kinh nghiệm. Hầu hết các chương trình “bình thường” có các tên như .code, .data, .rsrc hoặc .text. Các chương trình hệ thống có thể có các tên như PAGE, PAGEDATA, v.v. Mặc dù các tên này là bình thường, một tác giả phần mềm độc hại có thể dễ dàng đổi tên các phần trong chương trình độc hại để chúng có vẻ vô hại. Một số tên phần có thể được liên kết với các nhà đóng gói và mật mã trực tiếp. Ví dụ: bất kỳ chương trình nào có tên phần bắt đầu bằng UPX đã được xử lý bằng một trong những chương trình đó. Chúng tôi sẽ thảo luận về điều này ở độ dài lớn hơn sau này trong chương này.

Tất cả thông tin tệp PE cũng có sẵn thông qua pedump.exe. Thông tin phần trong Hình 6.7 xuất hiện như sau khi được xem qua pedump.exe:

```

01 .text    VirtSize: 000776EC VirtAddr:      00001000
raw data offs: 00001000 raw data size: 00078000
relocation offs: 00000000 relocations: 00000000
line # offs: 00000000 line #'s: 00000000
characteristics: 60000020
CODE EXECUTE      READ ALIGN_DEFAULT(16)

```

Như bạn có thể thấy, không có sự khác biệt đáng kể trong thông tin có sẵn thông qua hai công cụ. Thông tin địa chỉ và kích thước ảo xác định cách tệp hình ảnh thực thi sẽ trông như thế nào khi trong bộ nhớ và thông tin dữ liệu thô của dữ liệu áp dụng cho tệp hình ảnh thực thi khi nó tồn tại trên đĩa. Như bạn đã thấy trong Chương 3, thông tin này cũng cung cấp cho bạn bản đồ đường đi khi trích xuất hình ảnh thực thi từ kết xuất bộ nhớ.

2.3.2. *IMPORT Tables*

Ngày nay, rất hiếm khi một ứng dụng được viết hoàn toàn từ đầu. Hầu hết các chương trình được xây dựng bằng cách truy cập vào giao diện chương trình ứng dụng Windows (API) thông qua các chức năng khác nhau có sẵn trong các thư viện (DLL) trên hệ thống. Microsoft cung cấp một số lượng lớn các DLL cung cấp quyền truy cập vào các chức năng được tạo sẵn để tạo cửa sổ, menu, hộp thoại, ổ cắm và bất kỳ tiện ích, đối tượng và cấu trúc nào trên hệ thống. Không cần phải tạo bất kỳ thứ gì trong số này hoàn toàn bằng tay khi tạo một ứng dụng hoặc chương trình. Đó là trường hợp, khi các chương trình được viết và sau đó được biên dịch và liên kết thành các tệp hình ảnh thực thi, thông tin về các DLL và các chức năng mà chương trình đó truy cập cần phải có sẵn cho hệ điều hành khi ứng dụng đang chạy. Thông tin này được duy trì trong bảng `IMPORT` và bảng `IMPORT ADDRESS` của tệp thực thi.

Trong khi trở lại, tôi đã có cơ hội làm việc trong một dự án liên quan đến việc xác định xem một tập tin thực thi có khả năng mạng hay không. Tôi đã thực hiện một số công việc kiểm tra các ứng dụng để xác định xem chúng có khả năng của một máy chủ mạng (đã nghe các kết nối, như cửa hậu Trojan) hay máy khách (thực hiện kết nối với máy chủ, như IRCbot), nhưng với dự án này, mục tiêu là tự động hóa quá trình. Vì vậy, chúng tôi đã bắt đầu bằng cách kiểm tra các DLL có sẵn để xác định chức năng nào trong số chúng cung cấp chức năng kết nối mạng (tức là `wininet.dll`, `ws2_32.dll`, v.v.), sau đó chúng tôi xác định chức năng nào cung cấp chức năng cốt lõi được đề cập. Khi chúng tôi có thông tin đó, chúng tôi có thể tự động hóa quy trình bằng cách phân tích cấu trúc tệp PE, định vị bảng `IMPORT` và xác định DLL và chức năng nào được sử dụng. Tuy nhiên, một điều cần lưu ý là việc đọc bảng `IMPORT` của tệp thực thi

phần mềm độc hại có thể không dễ dàng nếu tệp bị che giấu theo một cách nào đó.

Công cụ pedump.exe cung cấp quyền truy cập dễ dàng vào thông tin bảng IMPORT, bằng cách định vị thư mục dữ liệu nhập và phân tích cú pháp các cấu trúc để xác định DLL và các chức năng mà ứng dụng sử dụng. Ví dụ đầu ra từ pedump.exe xuất hiện như sau:

Import Table:

```
...
KERNEL32.dll
OrigFirstThunk: 0000D114 (Unbound IAT)
TimeDateStamp: 00000000 -> Wed Dec 31 19:00:00 1969
ForwarderChain: 00000000
First thunk RVA: 0000B000
Ordin Name
448 GetSystemTimeAsFileTime
77 CreateFileA
393 GetNumberOfConsoleInputEvents
643 PeekConsoleInputA
571 LCMaStringW
570 LCMaStringA
443 GetSystemInfo
```

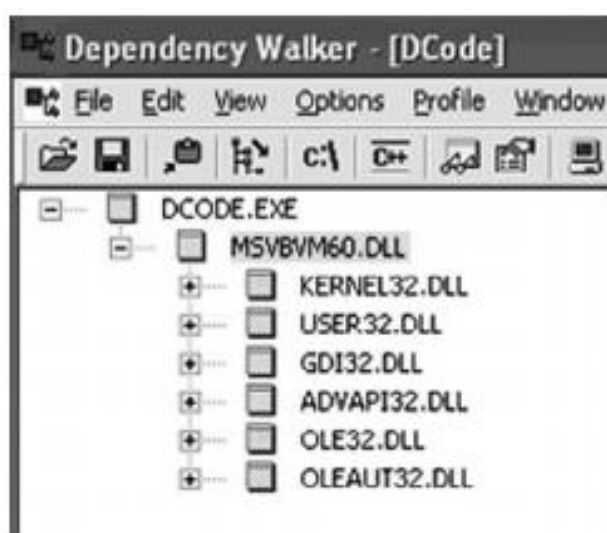
Như bạn có thể thấy, ứng dụng mẫu nhập một số hàm từ kernel32.dll. Mặc dù DLL thực sự cung cấp một số hàm có sẵn để sử dụng (xem phần “bảng EXPORT” trong phần sau của chương này), ví dụ này có thể thực thi các hàm nhập như `GetSystemTimeAsFileTime ()` và `CreateFileA ()` để sử dụng. Microsoft cung cấp rất nhiều thông tin liên quan đến nhiều chức năng có sẵn, vì vậy bạn có thể nghiên cứu trực tuyến để xem các chức năng khác nhau có nghĩa là gì. Ví dụ, hàm `GetSystemTimeAsFileTime ()` lấy thời gian hệ thống hiện tại dưới dạng đối tượng FILETIME 64 bit và giá trị được trả về biểu thị số lượng khoảng 100 nano giây kể từ ngày 1 tháng 1 năm 1601, ở định dạng Thời gian phối hợp phổ biến (UTC).

Bạn có thể tra cứu các chức năng Microsoft API thông qua MSDN. Tôi giữ một liên kết đến trang Tìm kiếm Nâng cao của Microsoft trên thanh công cụ trình duyệt của mình để truy cập nhanh. Nhập tên của hàm mà tôi yêu thích, chẳng hạn như `GetSystemTimeAsFileTime`, cung cấp cho tôi không chỉ thông tin về chức năng API mà còn với thông tin phụ trợ quan trọng.

Xem những chức năng mà một ứng dụng nhập khẩu cung cấp cho bạn một manh mối chung về những gì nó làm (và không làm). Ví dụ: nếu ứng dụng không nhập bất kỳ tệp DLL nào chứa mã mạng, dưới dạng các hàm ổ cắm cấp










thấp hoặc API Internet cấp cao hơn, thì ứng dụng đó không phải là một cửa hậu hoặc có thể được sử dụng để truyền thông tin tắt hệ thống và lên Internet. Đây là một kỹ thuật hữu ích, một kỹ thuật mà tôi đã sử dụng để cung cấp thông tin và trả lời các câu hỏi về một ứng dụng. Tôi đã từng được đưa ra một hình ảnh thực thi và được hỏi liệu nó có hoặc có khả năng trở thành một cửa hậu mạng hay không. Sau khi ghi lại tập tin, tôi đã xem bảng NHẬP và thấy rằng không có tệp DLL nào được nhập cung cấp khả năng kết nối mạng. Tôi đã tiến hành phân tích của mình một bước nữa bằng cách xem xét các hàm được nhập và thấy rằng mặc dù một số chức năng toán học được cung cấp, không có chức năng mạng nào được cung cấp.

Một công cụ hữu ích khác để xem thông tin liên quan đến DLL và các chức năng được yêu cầu bởi một ứng dụng là công cụ Dependency Walker, còn được gọi là Dep.exe, có sẵn từ trang web cùng tên. Hình 6.8 minh họa một đoạn trích của GUI Dependency Walker, với ứng dụng mẫu dcode.exe mở trong Dependency Walker.



Hình 6. 8 Trích từ Dependency Walker GUI

Như được minh họa trong Hình 6.8, ứng dụng dcode.exe dựa vào các chức năng từ MSVBVM60.DLL, lần lượt dựa vào các chức năng từ sáu DLL khác (mỗi DLL có các bản phân phối Windows mới nhất). Hình 6.9 minh họa một phần các hàm được xuất bởi MSVBVM60.DLL, như được báo cáo bởi công cụ Dependency Walker.

E	Ordinal ^	Hint	Function	Entry Point
	100 (0x0064)	60 (0x003C)	ThunRTMain	0x0000DE3E
	101 (0x0065)	73 (0x0049)	VBDllUnRegisterServer	0x00018CFC
	102 (0x0066)	70 (0x0046)	VBDllCanUnloadNow	0x0002C692
	103 (0x0067)	72 (0x0048)	VBDllRegisterServer	0x000A4A8A
	104 (0x0068)	71 (0x0047)	VBDllGetClassObject	0x00028FCA
	105 (0x0069)	69 (0x0045)	UserDllMain	0x00018BA7
	106 (0x006A)	13 (0x000D)	DllRegisterServer	0x000D3AD5
	107 (0x006B)	14 (0x000E)	DllUnregisterServer	0x000D3CB3
	108 (0x006C)	94 (0x005E)	__vbaAryLock	0x000E24D0

Công cụ Dependency Walker cho phép bạn xem không chỉ các DLL và các chức năng mà một tệp nhập khẩu thực thi có thể là một tệp .exe hoặc một tệp DLL mà còn các chức năng được xuất bởi DLL. Chúng ta sẽ thảo luận về bảng XUẤT thêm một chút trong phần tiếp theo.

Công cụ Dependency Walker cũng có chức năng định hình hữu ích, cho phép bạn đặt các tham số cụ thể về cách mô-đun hoặc ứng dụng sẽ được định hình, sau đó khởi chạy ứng dụng để xem mô-đun (DLL) nào sẽ được tải. Điều này cho phép bạn theo dõi các lệnh gọi hàm DLL khác nhau và các giá trị được trả về khi ứng dụng chạy. Điều này có thể hữu ích trong việc phát hiện các mô-đun được tải động nhưng aren được liệt kê trong các bảng NHẬP của các mô-đun khác hoặc để xác định lý do tại sao một ứng dụng không thành công trong việc khởi tạo lỗi chính xác được báo cáo. Tuy nhiên, điều này nằm ngoài phạm vi phân tích tĩnh, vì nó yêu cầu tệp phải được chạy.

2.3.3. *EXPORT Tables*

Vì DLL cung cấp các hàm mà các tệp thực thi khác có thể nhập, nên các DLL tự duy trì một bảng các hàm có sẵn trong bảng EXPORT (bạn đoán nó). Đây là các hàm có sẵn cho các hình ảnh thực thi khác (DLL, EXE, v.v.) để nhập hoặc sử dụng để các tác giả ứng dụng không cần phải viết mã của riêng họ cho mọi thứ họ muốn làm trên hệ thống. Các DLL hoạt động như các thư viện hoặc kho lưu trữ mã được viết sẵn có sẵn để sử dụng trên hệ thống.

Pedump.exe sẽ kết xuất bảng EXPORT từ DLL. Ví dụ: đây là một phần của bảng EXPORT cho ws2_32.dll:

exports table:

Name: WS2_32.dll
Characteristics: 00000000
TimeDateStamp: 41107EDA -> Wed Aug 04 02:14:50 2004
Version: 0.00
Ordinal base: 00000001
of functions: 000001F4
of Names: 00000075

Entry	Pt	Ord	Name
00011028	1		accept
00003E00	2		bind
00009639	3		closesocket
0000406A	4		connect
00010B50	5		getpeername
0000951E	6		getsockname
000046C9	7		getsockopt
00002BC0	8		htonl
00002B66	9		htons
00004519	10		ioctlsocket
00002BF4	11		inet_addr

Nếu bạn có bất kỳ kinh nghiệm nào về lập trình ổ cắm UNIX và / hoặc Perl, bạn sẽ nhận ra các chức năng được xuất là chức năng cốt lõi cho truyền thông dựa trên mạng. Ví dụ, các hàm bind () và accept () được sử dụng bởi các dịch vụ hoặc trình tiện ích lắng nghe các kết nối (backreen, v.v.) và chức năng kết nối () được sử dụng bởi các tiện ích máy khách kết nối với máy chủ, như trình duyệt Web và IRCbots.

Tôi nên chỉ ra rằng các DLL có thể nhập các hàm từ các DLL khác, ngoài việc xuất các hàm riêng của chúng. Ví dụ: sử dụng pedump.exe để xem thông tin PE cho ws2_32.dll, chúng tôi thấy rằng các hàm nhập thực thi từ kernel32.dll, ws2help.dll, ntdll.dll và các thông tin khác. Một số DLL sẽ nhập chức năng từ các DLL khác để xây dựng dựa trên chức năng cơ bản được cung

cấp. Các công cụ như Dependency Walker sẽ hiển thị cho bạn các phụ thuộc DLL được xâu chuỗi hoặc xếp tầng theo định dạng GUI đẹp.

2.3.4. Tài nguyên

Nhiều lần, một tệp PE sẽ có một phần có tên là “.rsrc”, và cũng sẽ có một thư mục dữ liệu tài nguyên được liệt kê. Phần tài nguyên này có thể chứa thông tin về những thứ như hộp thoại và biểu tượng và các thông tin hữu ích khác có thể giúp bạn xác định tệp, nhưng có lẽ điều hữu ích nhất trong quá trình phân tích tệp thực thi là thông tin phiên bản tệp.

Tập lệnh Perl fvi.pl (nằm trên phương tiện đi kèm) sử dụng mô-đun Win32 :: File :: VersionInfo để trích xuất thông tin phiên bản tệp từ tệp PE, nếu thông tin đó có sẵn. Fvi.pl lấy tên tệp (với đường dẫn đầy đủ) làm đối số duy nhất và trả về thông tin mà nó tìm thấy như sau:

```
C:\Perl>fvi.pl c:\windows\system32\svchost.exe
Filename : c:\windows\system32\svchost.exe
Type : Application
OS : NT/Win32
Orig Filename : svchost.exe
File Descriptoin : Generic Host Process for Win32 Services
File Version : 5.1.2600.2180 (xpsp_sp2_rtm.040803-2158)
Internal Name : svchost.exe
Company Name : Microsoft Corporation
Copyright : • Microsoft Corporation. All rights reserved.
Product Name : Microsoft« Windows« Operating System
Product Version : 5.1.2600.2180
Trademarks:
```

Bạn cần ghi nhớ một vài điều khi sử dụng các công cụ như thế này. Đầu tiên, mô-đun Win32 :: File :: VersionInfo dành riêng cho nền tảng Windows. Thứ hai, cả mô-đun và tập lệnh Perl đều không thực hiện bất kỳ nỗ lực nào để xác minh rằng tệp đang đề cập thực sự là tệp PE. Điều này có nghĩa là nếu fvi.pl không trả lại bất kỳ thông tin nào, điều đó không có nghĩa là tệp đang nghi vấn là phần mềm độc hại. Trên thực tế, nhiều tác giả phần mềm độc hại đảm bảo rằng thông tin đó không được biên dịch vào các công cụ của họ, trong khi những người khác sẽ bao gồm thông tin phiên bản tệp giả mạo để loại bỏ các nhà điều tra. Một số thậm chí bao gồm thông tin phiên bản tệp tin chỉ đơn giản là để giải trí cho chính họ và những người khác.

Mặc dù việc sử dụng thông tin phiên bản tệp không phải lúc nào cũng là phương tiện phân tích kết luận, nhưng nó cung cấp thông tin bổ sung sẽ bổ sung vào bức tranh tổng thể về cuộc điều tra của bạn.

2.3.5. *Obfuscation*

Cho đến nay, chúng tôi đã sử dụng các tệp thực thi hợp pháp, bình thường để minh họa các cấu trúc khác nhau của các tệp PE. Mặc dù bạn có thể sử dụng các công cụ và kỹ thuật này để xác định tệp, các tác giả phần mềm độc hại thường nỗ lực để ngụy trang hoặc xóa obfuscate các tệp của họ, không chỉ để tránh bị quản trị viên và điều tra viên phát hiện mà còn để ẩn khỏi chương trình chống vi-rút và các chương trình phần mềm bảo mật khác. Nhiều lần, các tác giả phần mềm độc hại sẽ sử dụng các trình đóng gói và thậm chí các công cụ mã hóa để ngụy trang phần mềm của họ hoặc đơn giản là họ sẽ tạo các phiên bản mới của chương trình của họ.

Bạn có thể sử dụng nhiều tiện ích khác nhau để làm xáo trộn các tệp thực thi, chẳng hạn như chất kết dính, trình đóng gói và mật mã. Chúng tôi sẽ lần lượt xem xét từng thứ một.

2.3.5.1. *Binders*

Chất kết dính là các tiện ích cho phép người dùng liên kết một ứng dụng này với ứng dụng khác, về bản chất là tạo ra một ứng dụng Trojan. Ý tưởng là ứng dụng nhà mạng sẽ lôi kéo người dùng khởi chạy nó; ví dụ bao gồm các trò chơi và các thực thi khác. Khi nạn nhân khởi chạy ứng dụng vận chuyển, anh ta thấy ứng dụng chạy và không có gì có vẻ không ổn. Tuy nhiên, trong suốt thời gian đó, ứng dụng Trojan chạy, thường ở phía sau hậu trường, không biết đến nạn nhân. Một trong những chất kết dính đầu tiên có sẵn là eLiTeWrap (<http://homepage.ntlworld.com/chawmp/elitewrap/>), nhưng Silk Rope và SaranWrap (<http://packetstormsecurity.org/trojans/bo/index3.html>) trở nên phổ biến khi Cult of the Dead Cow đã phát hành tiện ích Back Orifice. Nhìn vào các bài viết và mô tả về phần mềm độc hại có sẵn tại các trang web chống vi-rút (cũng như các phần mềm khác), có vẻ như các chất kết dính không còn là mối đe dọa giữa các tác giả phần mềm độc hại và có lẽ không còn được coi là “cool” nữa. do thực tế là các chất kết dính để lại chữ ký đã được phát hiện từ lâu bởi phần mềm chống vi-rút.

Mặc dù nhiều chất kết dính có sẵn dưới nhiều tên khác nhau, nhưng tất cả chúng đều thực hiện cùng một chức năng cơ bản: để ràng buộc một thực thi này với một thực thi khác. ELiTeWrap có lẽ là duy nhất ở chỗ nó cho phép người dùng cấu hình một tập lệnh của các lệnh sẽ được chạy hoặc các phản hồi sẽ

được cung cấp, cung cấp một số chức năng bổ sung trong các tệp thực thi bị ràng buộc.

Warning

Sau khi tải ELiTeWrap 1.04 về hệ thống Windows XP Pro SP2, tôi đã thử nhiều lần khác nhau để tạo ra một gói hoạt động, ràng buộc và mỗi lần đều thất bại. Tôi đã thử sử dụng ELiTeWrap trong chế độ tương tác, cũng như sử dụng tập lệnh. Mỗi lần, tôi kết thúc với một tệp đầu ra nhỏ hơn nhiều so với bất kỳ tệp đầu vào nào và khi tôi cố chạy tệp đầu ra, tôi nhận được một hộp thoại có ghi “Error #57 reading package”.

2.3.5.2. *Packers*

Các gói Packer là một tên gọi khác của các chương trình cho phép người dùng nén chương trình của họ, tiết kiệm không gian. Một tên gọi khác của các công cụ đó là “compressors”. dựa trên hệ thống chống vi-rút và bảo vệ xâm nhập. Trình đóng gói cũng làm cho việc phân tích thực thi trở nên khó khăn hơn. Một số công ty hợp pháp đóng gói các chương trình của họ để làm cho chúng chạy nhanh hơn (ít hơn để tải từ đĩa vào RAM) hoặc để bảo vệ bí mật thương mại. Mặc dù có nhiều trình đóng gói có sẵn, các chương trình đóng gói phổ biến bao gồm ASPack (www.aspack.com) và UPX (<http://upx.sourceforge.net>).

ASPack hoạt động bằng cách nén hình ảnh thực thi, viết một thói quen giải nén nhỏ ở cuối tệp. Sau đó, điểm nhập cảnh thực thi được thay đổi thành điểm bắt đầu của thói quen giải nén và điểm nhập ban đầu được lưu. Khi thực thi được giải nén vào bộ nhớ, điểm vào được đặt lại về giá trị ban đầu. Một dấu hiệu cho thấy ASPack đã được sử dụng là sự tồn tại của các tên phần như .adata, .udata và .aspack (tuy nhiên, xin lưu ý rằng các tên của phần chỉ là tên, và chúng có thể được thay đổi). Các công cụ được báo cáo là có sẵn sẽ cho phép bạn giải nén các tệp được đóng gói với ASPack.

UPX là một trình đóng gói phổ biến khác và mặc dù bạn có thể sử dụng nó như một trình đóng gói, bạn cũng có thể sử dụng nó để giải nén các tệp đã được đóng gói với UPX; vì vậy, nó cũng là một trình giải nén cho chính nó. Một dấu hiệu cho thấy bạn có tệp được nén bằng UPX là sự tồn tại của tên phần UPX0 và UPX1, nhưng bạn nên nhớ rằng những tên này có thể được thay đổi bằng cách chỉnh sửa tệp PE bằng trình chỉnh sửa hex.

Đây chỉ là một vài ví dụ về các tiện ích nén được sử dụng bởi các tác giả phần mềm độc hại và còn rất nhiều, rất nhiều thứ khác. Tùy thuộc vào tiện ích

nén được sử dụng, bạn có thể tìm thấy một ứng dụng hoặc trình cắm có nghĩa là giải nén thuật toán đó, đảo ngược quá trình. Bạn có thể phải dành một chút thời gian để nghiên cứu trên Internet để xem liệu đảo ngược việc nén có phải là một tùy chọn hay không và liệu có tiện ích nào hỗ trợ bạn hay không. Các công cụ như ProcDump32 (www.fortunecity.com/millennium/firemansam/962/html/procdump.html) bao gồm khả năng giải nén các thuật toán nén phổ biến. Hình 6.10 minh họa hộp thoại Chọn Unpacker cho ProcDump32 mà từ đó người dùng có thể chọn thuật toán được sử dụng để đóng gói thực thi.



Hình 6. 9 Chọn hộp thoại giả nén từ Procdump32

ProcDump32 cũng bao gồm các chức năng khác, chẳng hạn như cho phép người dùng kết xuất một quy trình đang chạy vào đĩa, giải nén hoặc giải mã các tệp PE bằng thuật toán phổ biến và chỉnh sửa tiêu đề PE. Các công cụ khác cho phép bạn làm điều này, nhưng ProcDump32 cung cấp một số chức năng khá hữu ích và nên được đưa vào như một phần của bộ công cụ phân tích phần mềm độc hại của bạn.

2.3.5.3. *Cryptors*

“Cryptors” của người dùng là một ngôn ngữ cho các chương trình cho phép người dùng mã hóa các chương trình khác. Mã hóa một tệp thực thi là một phương pháp khác mà người sử dụng phần mềm độc hại sử dụng để cố gắng tránh sự phát hiện của cả hệ thống chống vi-rút và chống xâm nhập dựa trên máy chủ và mạng. Điều này thực sự có vẻ là một phương pháp khá phổ biến để làm xáo trộn phần mềm độc hại và trong một số trường hợp, thuật toán mã hóa hoặc thói quen có thể được biết đến hoặc ít nhất có thể phát hiện được (dựa trên một chữ ký của một loại nào đó), trong khi trong những trường hợp khác thì nó có thể hoàn toàn không biết.

Như một ví dụ về một phần mềm độc hại bị xâm nhập, chúng ta sẽ xem xét một tệp mà chúng ta biết đã bị xâm nhập theo một cách nào đó. Dự án Honeynet đã cung cấp các thử thách thú vị về (SotM)

(<http://old.honeynet.org/scans/index.html>), cung cấp nhiều loại dữ liệu và kịch bản khác nhau để mọi người dùng thử lúc giải mã. Điều thú vị về SotM là sau một thời gian, các bài nộp được đánh giá và đăng, vì vậy bạn có thể xem các thử thách đã được giải quyết chi tiết như thế nào. Ed Skoudis cung cấp những thách thức tương tự tại trang web của mình, CounterHack.net.

Ví dụ: Honeynet SotM 32 được thiết kế để phân tích nhị phân phần mềm độc hại có tên rada.exe. Hình 6.11 minh họa biểu tượng cho nhị phân phần mềm độc hại



Hình 6. 10 Biểu tượng cho nhị phân phần mềm độc hại rada.exe

Sử dụng pedump.exe và PEView để xem xét rada.exe, chúng tôi thấy rằng nó có một tiêu đề PE khá bình thường và mọi thứ dường như dịch tốt. Điều đó có nghĩa là các công cụ có thể phân tích thông tin tiêu đề PE và từ góc độ phân tích cú pháp dường như có ý nghĩa. Nếu không, thì con trỏ sẽ trỏ đến các phần lạ của tệp hoặc hoàn toàn kết thúc tệp thực thi.

Tệp có ba phần: JDR0, JDR1 và .rsrc. Bây giờ, .rsrc là một phần mà chúng ta quen thuộc, nhưng hai phần khác chúng ta đã thấy trong các tệp PE mà chúng ta đã xem xét cho đến nay. Một điều khác mà chúng ta nhận thấy là bảng IMPORTS chỉ liệt kê hai DLL, KERNEL32.DLL và MSVBVM60.DLL, như được hiển thị ở đây :

```
Imports Table:
  KERNEL32.DLL
  OrigFirstThunk: 00000000 (Unbound IAT)
  TimeDateStamp: 00000000 -> Wed Dec 31 19:00:00 1969
  ForwarderChain: 00000000
  First thunk RVA: 00010BE0
  Ordn Name
    0 LoadLibraryA
    0 GetProcAddress
    0 ExitProcess
  MSVBVM60.DLL
  OrigFirstThunk: 00000000 (Unbound IAT)
  TimeDateStamp: 00000000 -> Wed Dec 31 19:00:00 1969
  ForwarderChain: 00000000
  First thunk RVA: 00010BF0
  Ordn Name
    618
```


Đây là phần mềm độc hại và bất kỳ phần mềm độc hại nào thực sự làm điều gì sẽ nhập nhiều hơn hai DLL và chắc chắn không chỉ có ba chức năng từ KERNEL32.DLL.

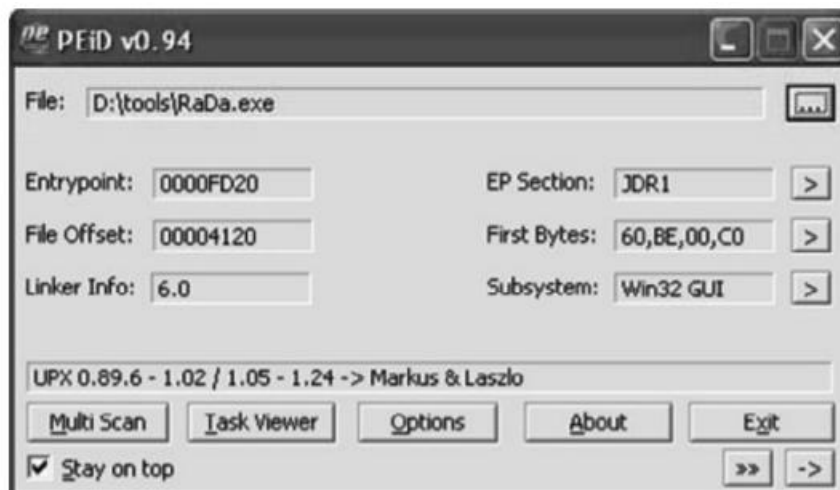
Đây cũng là một cách hay để phát hiện phần mềm độc hại bị chỉnh sửa nhanh chóng. Khi bảng IMPORT chỉ hiển thị Kernel32.DLL (hoặc có thể là DLL và một hoặc hai người khác) và chỉ một số hàm được nhập từ DLL đó bao gồm LoadLibraryA và GetProcAddress, điều này cho thấy rằng tệp đã bị chỉnh sửa theo một cách nào đó.

Mô-đun nhập khác, MSVBVM60.DLL, là thời gian chạy ngôn ngữ lập trình. Đầu ra của fvi.pl cho chúng ta biết rằng mô tả tệp từ phần tài nguyên của DLL đó là “Visual Basic Virtual Machine”. Từ điều này, chúng ta có thể suy luận rằng phần mềm độc hại đã được viết bằng Visual Basic. Phần này cũng được đưa ra trong các bài thử thách để phân tích tệp này được liệt kê tại trang web Honeynet.

rada.exe có một phần tài nguyên, chúng ta có thể chạy fvi.pl dựa vào đó và khi thực hiện, hãy truy xuất như sau:

```
Filename       : d:\tools\rada.exe
Type           : Application
OS            : Unknown/Win32
Orig Filename  : RaDa
File Descriptoin :
File Version   : 1.00
Internal Name  : RaDa
Company Name   : Malware
Copyright      :
Product Name   : RaDa
Product Version : 1.00
Trademarks     :
```

Người tạo đang cho chúng tôi biết rằng, đây là phần mềm độc hại. Bây giờ chúng ta đã thấy các dấu hiệu rõ ràng cho thấy phần mềm độc hại này bị chỉnh sửa (và chúng ta đã lừa một chút bằng cách chọn một chương trình mà chúng ta đã biết bị che giấu), chúng ta muốn biết làm thế nào nó bị chỉnh sửa. Một gói được sử dụng như nào? Được nén sử dụng, hoặc làm thế nào về mã hóa? Chúng ta có thể sử dụng một công cụ tiện dụng có tên PEiD (<http://peid.has.it/>) để kiểm tra tệp này và cố gắng xác định phương thức obfuscation. Hình 6.12 minh họa rada.exe được tải vào PEiD.

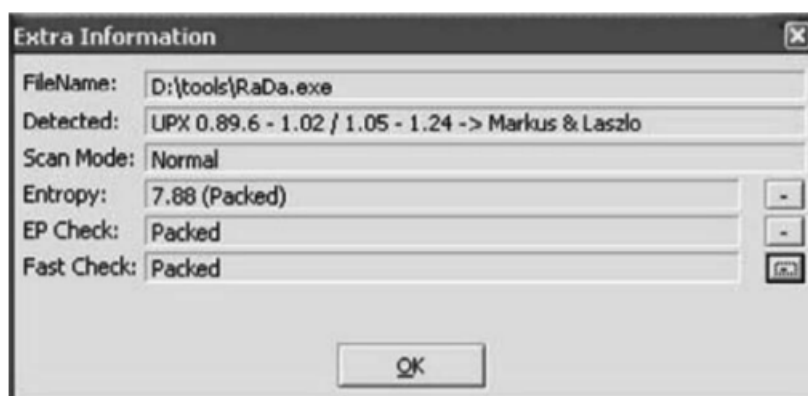


Hình 6. 11 Rada.exe được tải vào PEiD

Lưu ý rằng PEiD đã phát hiện phương thức obfuscation như một phiên bản của tiện ích nén UPX. Vì tên phần được liệt kê bởi PEView là JDR0 và JDR1, thay vì UPX0 và UPX1. Như ta đã đề cập trước đó, tên phần UPX0 và UPX1 chỉ ra tiện ích nén UPX. Điều này cho chúng ta biết rằng nếu thông tin PEiD là chính xác, người tạo đã sử dụng một trình soạn thảo để sửa đổi các tên phần đó.

Tất cả mọi thứ chúng ta đã xem xét liên quan đến các tệp thực thi cho đến nay đã cho chúng tôi một số ý tưởng về các khía cạnh khác nhau của các tệp mà chúng ta có thể điều tra để xác định bản chất của tệp. Điều này đặc biệt hữu ích trong việc hiểu những gì các sản phẩm chống vi-rút của chúng ta đang nói, với chúng ta hoặc những gì họ không nói với chúng ta khi họ không thể xác định biến thể mới nhất của một số phần mềm độc hại. Đây là nơi các công cụ như Yara (<http://code.google.com.vn/p/yara-project/>) có thể hữu ích. Dự án Yara cung cấp một khung để hỗ trợ các nhà nghiên cứu phần mềm độc hại (cũng như người trả lời) trong việc xác định và phân loại các khía cạnh khác nhau của các tệp phần mềm độc hại. Yara chạy trên Windows và Linux, và cũng có sẵn dưới dạng mô-đun Python. Don Weber đã mở rộng mô-đun Yara Python (www.cutawaysecurity.com/blog/archives/422) thành thứ mà anh gọi là Yara-Scout Sniper hoặc yara-ss, thêm một số khả năng hữu ích, chẳng hạn như truy cập các hệ thống từ xa. Bạn cũng có thể sử dụng chữ ký PEiD có sẵn công khai (tìm userdb.txt tại www.peid.info/BobSoft/Tai_xuong.html) như một phần của quy tắc Yara của bạn. Một cái gì đó như thế này có thể cực kỳ hữu ích trong việc nhanh chóng xác định và phân loại các biến thể mới của phần mềm độc hại, đặc biệt khi nó được duy trì và mở rộng như một phần của nỗ lực cộng đồng

PEiD phát hiện các trình đóng gói, mật mã và trình biên dịch phổ biến bằng cách định vị điểm vào của ứng dụng và phân tích các byte tại vị trí đó, cố gắng xác định phương thức obfuscation được sử dụng. Người tạo PEiD đã thu thập chữ ký cho nhiều công cụ obfuscation khác nhau và bao gồm chúng với PEiD. Họ cũng đã bao gồm một số công cụ tiện lợi cùng với PEiD, trong số đó có trình xem tác vụ để xem các quy trình đang chạy và các mô-đun họ sử dụng, một hộp thoại để xem thêm thông tin về tệp (được minh họa trong Hình 6.13), một hộp thoại để xem tiêu đề PE và thậm chí là một hộp thoại để xem nhị phân tháo rời.



Hình 6. 12 Hộp thoại thông tin bổ sung PEiD với rada.exe

Nếu bạn có cơ hội tải xuống cả PEiD và tệp rada.exe, hãy chạy trình dịch ngược bằng cách nhấp vào nút có mũi tên bên phải ở bên phải của trường văn bản Byte đầu tiên. Nếu bạn quen thuộc với lập trình ngôn ngữ lắp, những thứ có thể thu hút sự chú ý ngay lập tức của bạn là các hướng dẫn nhảy, nhiều hướng dẫn mà bạn thấy được liệt kê. Nếu bạn tò mò về các chi tiết phân tích của nhị phân này, hãy xem các bài nộp tại trang Honeynet, đặc biệt là bài của Chris Eagle. Chris là một giảng viên và người dẫn chương trình nổi tiếng tại các hội nghị BlackHat (www.blackhat.com), đồng thời là giảng viên cao cấp và chủ tịch liên kết cho Khoa Khoa học Máy tính tại Trường Sau đại học Hải quân ở Monterey, California.

Công cụ Mandiant's Red (www.mandiant.com/software/redcurtain.htm) có rất nhiều chức năng mà bạn đã thấy cho đến nay trong chương này, bao gồm cả PEiD, một bước nữa. Công cụ này sẽ báo cáo xem xét nội dung của tệp thực thi, tìm kiếm entropy ngẫu nhiên, chỉ dẫn đóng gói hoặc obfuscation, sự hiện diện của chữ ký số, cũng như các đặc điểm khác của tệp thực thi và tạo ra điểm số đe dọa. được dự định để chỉ cho nhà phân tích xem anh ta có nên điều tra tệp

tin thêm một chút không. Hình 6.14 minh họa tệp rada.exe đang mở trong công cụ Mandiant's Red



Hình 6. 13 .exe được tải trong Mandiant's Red Curtain

Phần lớn thông tin được trình bày bởi Red Curtain tương tự như những gì có sẵn trong PEiD. Nhấp vào nút Details button ở phía bên phải của giao diện Red Curtain sẽ mở một hộp thoại khác minh họa các phần PE (JDRO, JDR1, .rsrc) và thông tin của chúng, cũng như sự bất thường được xác định, trong trường hợp này là là “checksum_is_zero”, cho biết có thể giả mạo tệp (xem Hướng dẫn sử dụng rèm đỏ, có thể truy cập từ tùy chọn menu Trợ giúp, để biết thêm thông tin về công cụ và thông tin được trình bày).

Nếu bạn quan tâm đến việc tìm hiểu sâu hơn về hoạt động bên trong của phần mềm độc hại và tệp thực thi nói chung, thì nên đọc qua các bài SotM 32 và 33 tại trang web Thử thách Honeynet. Bạn sẽ không chỉ thấy sự tương đồng giữa tất cả các phân tích, mà bạn cũng sẽ thấy thông tin về các công cụ khác mà bạn có thể sử dụng để đi sâu hơn vào phân tích của mình.

3. Phân Tích Động

Phân tích động bao gồm khởi chạy một tệp thực thi trong môi trường được kiểm soát và giám sát để bạn có thể quan sát và ghi lại các hiệu ứng của nó trên hệ thống. Đây là một cơ chế phân tích cực kỳ hữu ích, trong đó nó cung cấp cho bạn cái nhìn chi tiết hơn về những gì phần mềm độc hại gây ra và cho một hệ thống, và đặc biệt là theo thứ tự nào. Điều này hữu ích nhất trong trường hợp phần mềm độc hại được đóng gói hoặc mã hóa, vì hình ảnh thực thi phải được giải nén hoặc giải mã (hoặc cả hai) trong bộ nhớ trước khi chạy. Vì vậy, không chỉ bạn sẽ nhìn thấy các dấu vết và các thông tin bị gãy, mà còn sử dụng các kỹ thuật để chụp và phân tích nội dung của bộ nhớ (như đã thảo luận trong Chương 3)

3.1. Kiểm Tra Môi Trường

Nếu bạn có ý định thực hiện phân tích động về phần mềm độc hại, một trong những cân nhắc của bạn sẽ là môi trường thử nghiệm hoặc máy chủ lưu trữ. Đó là một ý tưởng tốt để xem một phần mềm độc hại làm gì bằng cách thả nó vào mạng sản xuất và để cho nó chạy.

Một cách để thiết lập môi trường thử nghiệm của bạn là có một hệ thống trên một mạng riêng, không có kết nối điện (lưu ý ở đây là tôi không nói lời kết nối logic logic hay Vlan trên một công tắc mạng) với phần còn lại của mạng. Phải có khoảng trống không khí. Tôi thực sự khuyên bạn không nên chỉnh sửa với việc chuyển đổi để tách phần mềm độc hại của mình. Ngoài ra, nếu bạn đang trải qua một cuộc kiểm toán theo yêu cầu của bất kỳ cơ quan quản lý nào, điều cuối cùng bạn muốn có là một cách để phần mềm độc hại có khả năng đánh cắp dữ liệu cá nhân nhạy cảm để xâm nhập vào mạng nơi dữ liệu cá nhân nhạy cảm tồn tại. Nếu phòng thí nghiệm của bạn được công nhận hoặc chứng nhận bởi một cơ quan thích hợp, bạn có thể gây nguy hiểm nghiêm trọng cho tình trạng đó bằng cách chạy các chương trình không tin cậy trên mạng trực tiếp.

Một trong những nhược điểm của việc có một hoặc hai hệ thống là một phải cài đặt lại hệ điều hành sau mỗi lần kiểm tra; Làm thế nào để đảm bảo rằng bạn đang thu thập dữ liệu sạch và kết quả của bạn không bị nhiễm bởi một phần mềm độc hại khác? Một cách để thực hiện điều này là với ảo hóa.

3.1.1. Ảo Hóa

Nếu bạn không có một hệ thống tốt mà bạn có thể liên tục cài đặt lại và trở về trạng thái ban đầu, Ảo hóa là một tùy chọn khác có sẵn cho bạn. Một số phần mềm miễn phí và các công cụ ảo hóa thương mại có sẵn cho bạn, chẳng hạn như:

- Bochs : Chạy trên Windows, Linux và thậm chí Xbox, là nguồn mở và miễn phí (<http://bochs.sourceforge.net/>).
- Parallels Chạy trên nền tảng Mac cũng như Windows và Linux (www.parallels.com).
- Microsoft Virtual PC Chạy trên Windows với tư cách là hệ điều hành máy chủ; có thể chạy hệ điều hành khách DOS, Windows và OS / 2 và có sẵn miễn phí (www.microsoft.com/windows/products/winfamily/virtualpc/default.mspx)
- Công cụ ảo “Bare metal install” (có nghĩa là nó không được cài đặt trên hệ điều hành máy chủ) và có thể chạy Windows và Linux với tốc độ gần như nguyên bản (www.virtualiron.com/).

- Win4Lin chạy trên Linux; cho phép bạn chạy các ứng dụng Windows (<http://win4lin.net/content/>).
- VMware Chạy trên Windows và Linux và cho phép bạn lưu trữ một số hệ điều hành khác. Các sản phẩm VMware Server và VMware Player có sẵn miễn phí. VMware được nhiều người coi là tiêu chuẩn thực tế cho các sản phẩm ảo hóa và được thảo luận chi tiết hơn trong các phần sau (www.vmware.com).

Tất nhiên, đây không phải là một danh sách đầy đủ. Tùy chọn ảo hóa bạn chọn phụ thuộc phần lớn vào nhu cầu, môi trường của bạn (nghĩa là, hệ thống có sẵn, ngân sách, v.v.) và mức độ thoải mái khi làm việc với các hệ điều hành máy chủ và khách khác nhau. Nếu bạn không chắc chắn về tùy chọn nào phù hợp nhất với mình, hãy xem So sánh so sánh các máy ảo trang Trang (http://en.wikipedia.org/wiki/Comparison_of_virtual_machines) trên Wikipedia. Điều này có thể giúp bạn thu hẹp các lựa chọn của bạn dựa trên môi trường, ngân sách của bạn và mức độ nỗ lực cần thiết để có được một nền tảng ảo hóa và chạy.

Lợi ích của việc sử dụng một hệ thống ảo khi phân tích phần mềm độc hại là bạn có thể tạo ra một snapshot chụp hình của hệ thống đó và sau đó, lây nhiễm vào hệ thống đó và thực hiện tất cả các thử nghiệm và phân tích của bạn. Khi bạn đã thu thập tất cả dữ liệu của mình, bạn có thể quay lại ảnh chụp nhanh, đưa hệ thống trở về trạng thái nguyên sơ, trước khi bị nhiễm. Theo cách này, không chỉ các hệ thống có thể được phục hồi dễ dàng hơn mà nhiều phiên bản phần mềm độc hại tương tự cũng có thể được kiểm tra trên cùng một nền tảng để so sánh đồng đều hơn.

Có lẽ nền tảng ảo hóa được biết đến nhiều nhất là VMware. VMware cung cấp một số sản phẩm ảo hóa miễn phí, chẳng hạn như VMware Player, cho phép bạn chơi các máy ảo (mặc dù không tạo ra chúng) và VMware Server. Ngoài ra, một số máy hoặc thiết bị ảo dựng sẵn có sẵn để tải xuống và sử dụng.

Có một sự cảnh báo khi sử dụng VMware và nó cũng áp dụng cho các môi trường ảo hóa khác. Cách đây không lâu, đã có những cuộc thảo luận về cách phần mềm có thể được sử dụng để phát hiện sự tồn tại của môi trường ảo hóa. Ngay sau đó, các nhà phân tích bắt đầu thấy phần mềm độc hại không chỉ phát hiện sự hiện diện của môi trường ảo hóa mà còn thực sự hoạt động khác đi hoặc đơn giản là không hoạt động. Vào ngày 19 tháng 11 năm 2006, Lenny Zeltser đã đăng một mục nhật ký của bộ xử lý ISC (<http://isc.sans.org/diary.php?storyid=1871>) thảo luận về phát hiện máy ảo trong phần mềm độc hại thông qua việc sử dụng các công cụ thương mại. Đây là điều

bạn nên ghi nhớ và cân nhắc khi thực hiện phân tích phần mềm độc hại động. Hãy chắc chắn phỏng vấn kỹ lưỡng bất kỳ người dùng nào chúng kiến sự cố và xác định càng nhiều các tạo phẩm tiềm năng càng tốt trước khi đưa mẫu phần mềm độc hại của bạn trở lại phòng thí nghiệm. Theo cách đó, nếu bạn thấy hành vi hoàn toàn khác nhau trong phần mềm độc hại khi chạy trong môi trường ảo, bạn có thể đã tìm thấy một ví dụ về phần mềm độc hại có chứa mã này.

3.1.2. *Hệ Thống Lỗi*

Nếu ảo hóa đơn giản không phải là một lựa chọn (do giá cả, kinh nghiệm, mức độ thoải mái, v.v.), bạn có thể chọn sử dụng các hệ thống lỗi có thể nhanh chóng được tạo hình và xây dựng lại. Một số tổ chức doanh nghiệp sử dụng các công cụ như Symantec, Norton Norton Ghost để tạo hình ảnh cho các hệ thống có cùng phần cứng. Bằng cách đó, một bản dựng tiêu chuẩn có thể được sử dụng để thiết lập các hệ thống, giúp chúng dễ quản lý hơn. Các tổ chức khác đã sử dụng một cách tiếp cận tương tự với môi trường đào tạo, cho phép nhân viên CNTT nhanh chóng đưa tất cả các hệ thống về trạng thái đã biết. Ví dụ, khi tôi đang thực hiện đánh giá lỗ hổng, tôi đã thực hiện đánh giá cho một tổ chức có môi trường đào tạo. Họ tự hào nói với tôi rằng sử dụng Norton Ghost, họ hoàn toàn có thể tải lại hệ điều hành trên tất cả 68 máy trạm đào tạo chỉ với một đĩa mềm

Nếu đây là điều bạn chọn để làm, bạn cần đảm bảo các hệ thống không được gắn vào mạng công ty hoặc mạng sản xuất dưới bất kỳ hình thức nào. Bạn có thể nghĩ rằng điều này không cần phải nói, nhưng các mạng kiểm tra và đảm bảo chất lượng đã bị gỡ xuống do một quản trị viên vội vàng hoặc một mạng cục bộ ảo (Vlan) được cấu hình không đúng trên một bộ chuyển mạch. Bạn nên đảm bảo rằng bạn có nhiều hơn một khoảng cách logic giữa nền tảng thử nghiệm của bạn và bất kỳ mạng nào khác.

Khi bạn quyết định bạn sẽ sử dụng trên nền tảng, bạn có thể theo cùng quy trình thu thập và phân tích dữ liệu mà bạn sẽ sử dụng trong môi trường ảo trên các hệ thống vớt bỏ; quá trình thực sự không khác nhau. Tuy nhiên, trên một hệ thống vớt đi, bạn sẽ cần đưa vào một số phương pháp để ghi lại nội dung của bộ nhớ trên nền tảng của bạn (hãy nhớ rằng các phiên của VMware có thể bị đình chỉ), đặc biệt nếu bạn đang phân tích phần mềm độc hại bị giấu.

3.1.3. *CÔNG CỤ*

Có thể sử dụng nhiều công cụ khác nhau để giám sát các hệ thống khi kiểm tra phần mềm độc hại. Đối với hầu hết các phần, bạn muốn tất cả các công cụ của mình trước khi bạn chạy mẫu phần mềm độc hại. Ngoài ra, bạn muốn

làm quen với những gì công cụ của bạn có khả năng cũng như cách sử dụng chúng.

Một trong những khác biệt lớn giữa phân tích phần mềm độc hại và phản hồi sự cố là khi người phân tích phần mềm độc hại, bạn có cơ hội thiết lập và định cấu hình hệ thống kiểm tra trước khi bị nhiễm. Mặc dù về mặt lý thuyết, các quản trị viên hệ thống cũng có cơ hội tương tự, nhưng khá hiếm khi bạn tìm thấy các hệ thống máy chủ lớn được cấu hình mạnh với bảo mật và đặc biệt là phản ứng sự cố

Khi kiểm tra phần mềm độc hại, có một số thách thức mà bạn phải nhận thức được. Ví dụ: bạn không biết phần mềm độc hại sẽ làm gì khi được khởi chạy. Tôi biết điều này nghe có vẻ đơn giản, nhưng hơn một lần tôi đã nói chuyện với những người mà không tính đến điều này. Điều tôi muốn nói là bạn không nên biết phần mềm độc hại sẽ mở ra và ngồi ở đó, chờ đợi để được phân tích, hay liệu nó sẽ thực hiện công việc của mình một cách nhanh chóng và biến mất. Tôi đã thấy một số phần mềm độc hại sẽ mở một cổng chờ kết nối (cửa sau), phần mềm độc hại khác đã cố gắng kết nối với các hệ thống trên Internet (IRCbots) và phần mềm độc hại chỉ mất một phần giây để tiêm mã của nó vào một phần mềm khác. Quá trình chạy rồi biến mất. Khi thực hiện phân tích động, bạn có cơ hội lặp đi lặp lại nhiều lần về tội phạm và cố gắng xem chi tiết. Khi chúng tôi thực hiện các hoạt động ứng phó sự cố, chúng tôi chủ yếu chụp ảnh hiện trường, sử dụng các công cụ để nắm bắt thông tin trạng thái từ hệ thống tại các thời điểm riêng biệt. Điều này giống như cố gắng thực hiện giám sát với máy ảnh Polaroid. Trong quá trình phân tích động, chúng tôi muốn theo dõi cảnh bằng video trực tiếp, nơi chúng tôi có thể ghi lại thông tin trong một khoảng thời gian liên tục thay vì tại các thời điểm riêng biệt. Bằng cách đó, hy vọng chúng tôi sẽ có thể nắm bắt và phân tích những gì diễn ra trong toàn bộ tuổi thọ của phần mềm độc hại.

Để bắt đầu, chúng tôi muốn đăng nhập bất kỳ và tất cả thông tin kết nối mạng, vì phần mềm độc hại có thể cố gắng liên lạc với hệ thống từ xa hoặc mở một cổng để lắng nghe kết nối hoặc cả hai. Một cách chúng ta có thể làm là chạy một trình thám thính mạng như Wireshark (trước đây gọi là Ethereal, được tìm thấy tại www.wireshark.org) trên mạng. Nếu bạn đang sử dụng một hệ thống độc lập, bạn sẽ muốn có công cụ nghe lén trên một hệ thống khác và nếu bạn đang sử dụng VMware, bạn sẽ muốn Wireshark chạy trên hệ điều hành máy chủ, trong khi phần mềm độc hại đang được thực thi một trong những hệ điều hành khác

Một công cụ khác mà bạn sẽ muốn cài đặt trên hệ thống của mình là Port Reporter (<http://support.microsoft.com/kb/837243>), được cung cấp miễn phí từ Microsoft. Port Reporter hoạt động như một dịch vụ trên các hệ thống Windows và ghi lại hoạt động cổng Giao thức điều khiển truyền (TCP) và giao thức gói dữ liệu người dùng (UDP). Trên các hệ thống Windows XP và Windows 2003, Port Reporter sẽ ghi lại các cổng mạng được sử dụng, quy trình hoặc dịch vụ sử dụng các cổng đó, các mô-đun được tải bởi quy trình và tài khoản người dùng chạy quy trình.

Ít thông tin hơn được ghi lại trên các hệ thống Windows 2000. Port Reporter có nhiều tùy chọn cấu hình, chẳng hạn như trong hệ thống tệp, các tệp nhật ký được tạo, cho dù dịch vụ tự động khởi động khi khởi động hệ thống hoặc thủ công (là mặc định), v.v. Bạn có thể kiểm soát các tùy chọn này thông qua các tham số dòng lệnh được thêm vào khởi chạy dịch vụ sau khi cài đặt Port Reporter

Một số phần mềm độc hại có thể ngừng hoạt động và chỉ cần tắt nếu không thể kết nối với hệ thống trên Internet, chẳng hạn như máy chủ chỉ huy và kiểm soát. Một cách để giải quyết vấn đề này là xem xét lưu lượng mạng được tạo bởi quy trình và xem liệu nó có tìm kiếm hệ thống tên miền (DNS) cho một tên máy chủ cụ thể không. Sau đó, bạn có thể sửa đổi tệp máy chủ của mình (nằm trong thư mục % WinDir% \ system32 \ driver \ etc) để trỏ hệ thống của bạn đến một hệ thống cụ thể trên mạng của bạn, thay vì trên Internet. Xem bài viết 172218 của Cơ sở tri thức Microsoft (<http://support.microsoft.com/kb/172218>) để biết thông tin cụ thể về cách hệ thống Windows giải quyết tên máy chủ TCP / Internet Protocol (IP).

Port Reporter tạo ba loại tệp nhật ký: nhật ký khởi tạo (nghĩa là nhật ký PR-INITIAL - *.log, Với dấu hoa thị thay thế ngày và giờ ở định dạng 24 giờ khi nhật ký được tạo) ghi lại thông tin trạng thái về hệ thống khi dịch vụ bắt đầu; một bản ghi cổng (tức là, PR-PORTS - *.log) duy trì thông tin về các kết nối mạng và việc sử dụng cổng, tương tự như netstat.exe; và nhật ký ID quy trình (tức là, PR-PIDS - *.log) duy trì thông tin quy trình.

Microsoft cũng cung cấp một WebCast (<http://support.microsoft.com/kb/840832>) giới thiệu công cụ Trình báo cáo cổng và mô tả chức năng của nó. Microsoft cũng có sẵn công cụ Port Reporter Parser (<http://support.microsoft.com/kb/884289>) để giúp phân tích nhật ký Trình báo cáo cổng có khả năng dễ dàng và thực tế hơn nhiều.

Tại sao tôi có thể chạy nó trên cùng một nền tảng phân tích động với tất cả các công cụ giám sát khác của tôi. Câu trả lời liên quan đến rootkit, chúng ta

sẽ thảo luận trong Chương 7. Tuy nhiên, câu trả lời ngắn gọn là rootkit cho phép phần mềm độc hại che giấu sự hiện diện của nó trên một hệ thống bằng cách ngăn hệ điều hành khởi nhìn thấy quy trình, kết nối mạng. Theo văn bản này, kiểm tra kỹ lưỡng đã không được thực hiện bằng nhiều rootkit khác nhau, vì vậy chúng tôi muốn chắc chắn rằng chúng tôi thu thập càng nhiều thông tin càng tốt. Bằng cách chạy sniffer mạng trên nền tảng khác, tách biệt với nền tảng thử nghiệm, chúng tôi đảm bảo rằng một phần của quy trình giám sát của chúng tôi không bị ảnh hưởng bởi phần mềm độc hại khi nó được khởi chạy và hoạt động. Nó cũng có thể hữu ích trong quá trình phân tích phần mềm độc hại động để quét hệ thống đã bị nhiễm vi-rút từ một hệ thống khác. Quá trình quét này có thể hiển thị một cửa hậu được mở trên hệ thống nhưng được ẩn thông qua một số phương tiện, chẳng hạn như rootkit (chúng tôi sẽ thảo luận chi tiết về rootkit trong Chương 7). Bạn có thể sử dụng các công cụ như Nmap (<http://nmap.org/>) và PortQry (<http://support.microsoft.com/kb/832919>) để quét nhanh hệ thống đã bị nhiễm virus và thậm chí cố gắng xác định bản chất của dịch vụ nghe trên một cổng cụ thể. Mặc dù các vấn đề về kết nối TCP / IP và cổng gõ gõ vượt quá phạm vi của cuốn sách này, luôn có khả năng một số truy vấn (hoặc kết hợp truy vấn) được gửi đến một cổng mở trên hệ thống bị nhiễm virus có thể khiến quá trình bị ràng buộc đến cổng đó để phản ứng theo một cách nào đó.

Hãy nhớ rằng, một trong những điều chúng ta cần hiểu là giám định. Trong bối cảnh phân tích phần mềm độc hại động, điều này có nghĩa là nếu chúng tôi thấy lưu lượng truy cập mạng phát ra từ nền tảng thử nghiệm và đi ra Internet (hoặc tìm kiếm các hệ thống khác trên mạng con cục bộ), nhưng chúng tôi không quan sát bất kỳ dấu hiệu nào của quy trình hoặc lưu lượng mạng được tạo thông qua các công cụ giám sát trên nền tảng thử nghiệm, chúng tôi có thể có rootkit trên tay.

Như một lời cảnh báo, đây là cơ hội tốt để tôi bày tỏ sự cần thiết của một quy trình phân tích phần mềm độc hại kỹ lưỡng và được ghi lại. Tôi đã thấy phần mềm độc hại không có khả năng rootkit, nhưng thay vào đó lại tiêm mã vào một không gian bộ nhớ quy trình khác và chạy từ đó. Đây là điều bạn cần hiểu, vì đưa ra giả định rằng rootkit có liên quan sẽ dẫn đến báo cáo không chính xác, cũng như các hành động không chính xác để đáp ứng với vấn đề. Nếu bạn ghi lại quá trình và các công cụ bạn sử dụng, ý tưởng là người khác sẽ có thể xác minh kết quả của bạn. Rốt cuộc, bằng cách sử dụng cùng các công cụ và cùng một quy trình và cùng một phần mềm độc hại, người khác sẽ có thể thấy kết quả tương tự, phải không? Hoặc người đó sẽ có thể xem xét quy trình

của bạn và hỏi về sự vắng mặt hoặc sử dụng một công cụ cụ thể, điều này sẽ cho phép kiểm tra và phân tích kỹ hơn về phần mềm độc hại

Khi thực hiện phân tích phần mềm độc hại, bạn phải lập kế hoạch nhiều nhất có thể, nhưng đồng thời, bạn không nên quá tải hoặc tải hệ thống của mình xuống với rất nhiều công cụ mà bạn dành quá nhiều thời gian để quản lý các công cụ mà bạn Tôi đã mất dấu những gì bạn đang phân tích. Tôi đã làm việc với sự tham gia của khách hàng một lần khi chúng tôi tìm thấy một tập tin bất thường. Dấu hiệu ban đầu của tệp là trong Sổ đăng ký; Khi được khởi chạy, nó đã thêm một giá trị cho khóa Run Run của người dùng, cũng như cho khóa RunOnce. Thật thú vị, nó đã thêm giá trị vào khóa RunOnce bằng cách đặt trước tên của tệp với (* *; điều này báo cho hệ điều hành phân tích và khởi chạy nội dung của khóa ngay cả khi hệ thống được khởi động ở Chế độ an toàn (khá khó khăn!). Chúng tôi đã phải dùng đến phân tích động, vì phân tích tĩnh nhanh chóng tiết lộ rằng phần mềm độc hại đã được mã hóa và PEiD không thể xác định phương thức mã hóa được sử dụng. Sau khi khởi chạy phần mềm độc hại trên nền tảng của chúng tôi và phân tích dữ liệu bị bắt, chúng tôi có thể thấy phần mềm độc hại sẽ khởi chạy trình duyệt Web một cách vô hình (quá trình trình duyệt đang chạy, nhưng GUI không hiển thị trên máy tính để bàn) và sau đó tự đưa vào quy trình của trình duyệt không gian. Từ đó, chúng tôi có thể xác định rằng một khi phần mềm độc hại đã được khởi chạy, chúng tôi sẽ tìm kiếm quy trình trình duyệt để biết thêm thông tin. Nó cũng giải thích tại sao, trong quá trình phân tích dữ liệu dễ bay hơi, chúng tôi đã thấy rằng quy trình trình duyệt chịu trách nhiệm cho các kết nối mạng bất thường và không có bằng chứng nào về quy trình phần mềm độc hại đang chạy

Event Log có thể giúp bạn theo dõi một số hoạt động khác nhau trên hệ thống, bao gồm việc sử dụng đặc quyền người dùng, đăng nhập, truy cập đối tượng (cài đặt này yêu cầu bạn cũng định cấu hình danh sách điều khiển truy cập [ACL] trên các đối tượng, tệp thư mục, Khóa đăng ký, v.v., mà bạn đặc biệt muốn theo dõi), v.v. Bởi vì chúng tôi rất quan tâm đến các quy trình trong quá trình phân tích phần mềm độc hại động, cho phép kiểm tra Theo dõi quá trình cho cả các sự kiện thành công và thất bại sẽ cung cấp cho chúng tôi một số dữ liệu hữu ích. Sử dụng audpolpol.exe từ Bộ tài nguyên (mà chúng ta đã thảo luận trong Chương 1), chúng ta có thể định cấu hình chính sách kiểm toán của nền tảng phân tích động, cũng như xác nhận rằng nó được đặt đúng trước khi thử nghiệm. Ví dụ: sử dụng dòng lệnh sau để đảm bảo rằng kiểm toán phù hợp được bật:

```
C:\tools>auditpol /enable /process:all
```

Để xác nhận rằng kiểm toán thích hợp vẫn được bật trước khi thử nghiệm, chỉ cần khởi chạy audpolpol.exe từ dòng lệnh không có đối số.

Bạn cũng có thể muốn kích hoạt kiểm toán các sự kiện Hệ thống, nhưng hãy chắc chắn không kích hoạt quá nhiều kiểm toán. Có một thứ như là có quá nhiều dữ liệu và điều này thực sự có thể làm chậm quá trình phân tích của bạn, đặc biệt nếu dữ liệu không được sử dụng nhiều cho bạn. Một số người có thể cảm thấy họ muốn giám sát mọi thứ để họ đảm bảo rằng họ không bỏ lỡ bất cứ điều gì, nhưng có giới hạn về số lượng dữ liệu bạn có thể sử dụng và phân tích một cách hiệu quả. Đánh giá kỹ lưỡng những gì bạn lên kế hoạch thực hiện và thiết lập một cấu hình tiêu chuẩn cho nền tảng thử nghiệm của bạn và tuân thủ nó, trừ khi có một lý do thuyết phục để thay đổi nó. Quá nhiều dữ liệu có thể gây nguy hiểm cho một cuộc điều tra cũng như quá ít dữ liệu.

Như đã đề cập trước đó, một cách để giám sát quyền truy cập vào tệp và khóa Sổ đăng ký là cho phép kiểm tra truy cập đối tượng, đặt ACL trên tất cả các đối tượng bạn quan tâm và khi bạn đã thực hiện phần mềm độc hại, hãy cố gắng hiểu nội dung Event Log. Hoặc bạn có thể xem xét hai cách để giám sát quyền truy cập vào tệp và Registry keys: Một cách là chụp trước và sau khi chụp nhanh và so sánh hai cách, và cách khác là sử dụng snapshots. Khi thực hiện phân tích phần mềm độc hại, bạn sẽ cần một số công cụ. Bạn có thể truy cập trang web của Microsoft và tải xuống các công cụ FileMon và RegMon (cho phép bạn theo dõi hoạt động của hệ thống tệp và đăng ký trong thời gian thực) hoặc bạn có thể tải xuống Trình theo dõi tiến trình. Lợi ích của việc sử dụng các công cụ giám sát thời gian thực thay vì các công cụ chụp nhanh là bạn không chỉ thấy các tệp và khóa Registry được tạo hoặc sửa đổi mà còn có thể xem các tệp và khóa Registry có thể đã được tìm kiếm nhưng không được tìm thấy. Hơn nữa, bạn có thể thấy dòng thời gian của hoạt động, xem thứ tự các tệp hoặc khóa Registry được truy cập. Đây có thể là một phần quan trọng trong phân tích phần mềm độc hại của bạn

FileMon và RegMon là các công cụ giám sát tuyệt vời có sẵn từ trang web Microsoft Sysinternals (<http://technet.microsoft.com/en-us/sysinternals/bb795535.aspx>). Mặc dù mỗi công cụ này vẫn được cung cấp riêng, cả hai đều có chức năng được thêm vào công cụ Giám sát quy trình, cũng có sẵn từ cùng một trang.

3.1.4. *Quá Trình*

Quá trình thiết lập nền tảng thử nghiệm của bạn để phân tích động phần mềm độc hại khá đơn giản và đơn giản, và điều quan trọng là phải thực sự có một quy trình hoặc danh sách kiểm tra. Như với việc thu thập dữ liệu để bay hơi

hoặc phân tích pháp y, bạn không muốn thử thực hiện phân tích động từ bộ nhớ mỗi lần, bạn chỉ đơn giản là quên một bước quan trọng trong quy trình. Tôi đã chia sẻ các kịch bản phân tích của mình, nơi tôi phải bắt đầu lại vì tôi quên bật một trong những công cụ của mình. Tôi phải quay lại và cài đặt hoàn toàn và làm mới hệ thống đã bị xâm nhập, và sau đó đảm bảo rằng các công cụ của tôi đã được cài đặt và cấu hình hệ thống của tôi là chính xác.

Điều đầu tiên bạn muốn làm là đảm bảo rằng bạn đã xác định, tải xuống và cài đặt tất cả các công cụ mà bạn sẽ cần. Tôi đã đề cập đến một số lượng lớn các công cụ trong chương này, nhưng trong tương lai, có thể có các công cụ khác mà bạn sẽ quan tâm khi sử dụng.

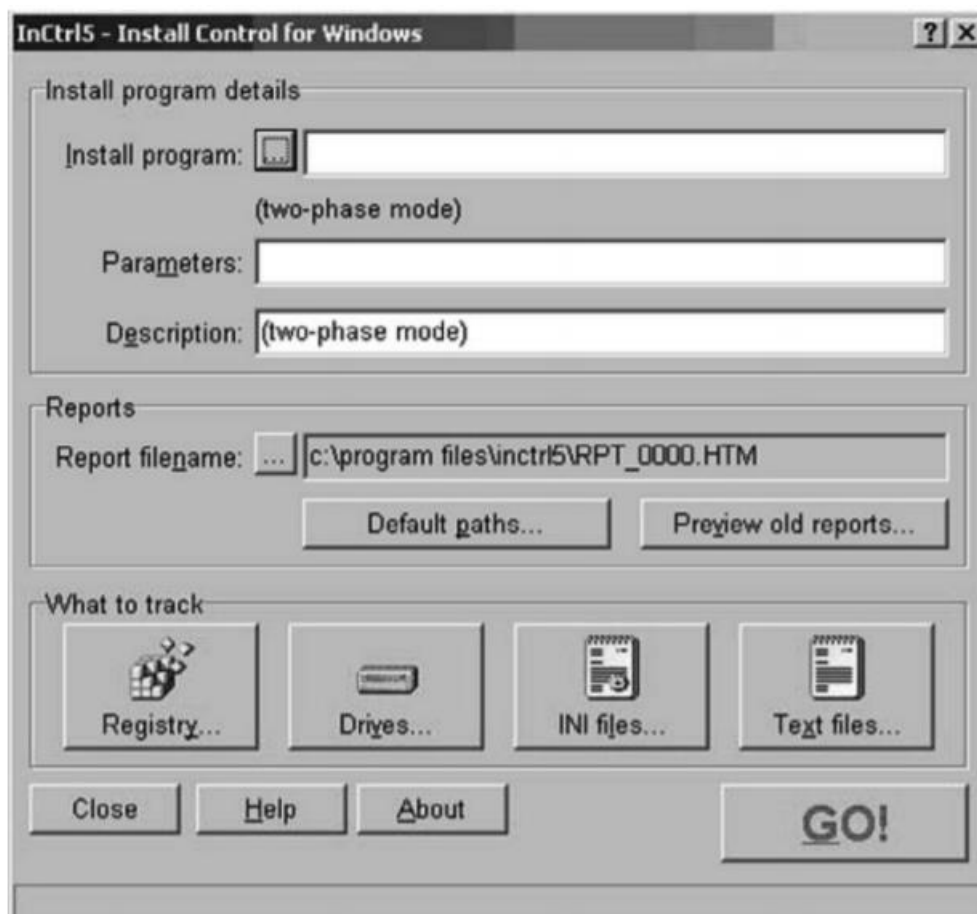
Khi bạn đã có tất cả các công cụ của mình, hãy chắc chắn rằng bạn hiểu cách chúng được sử dụng và đảm bảo rằng bạn biết và hiểu các tùy chọn cấu hình cần thiết. Hầu hết các công cụ sẽ được khởi động thủ công và bạn cần có một danh sách kiểm tra thứ tự mà bạn sẽ bắt đầu các công cụ của mình. Ví dụ: các công cụ như Regshot (<http://sourceforge.net/projects/regshot/>), được minh họa trong Hình 6.15 và InControl5, được minh họa trong Hình 6.16, chụp ảnh nhanh của hệ thống để so sánh, vì vậy bạn muốn khởi chạy cái đầu tiên giai đoạn (thu thập ảnh chụp đường cơ sở) trước, sau đó bắt đầu các công cụ giám sát thời gian thực.



Hình 6. 14 Regshot GUI

Regshot lưu đầu ra của nó ở dạng văn bản thuần hoặc định dạng HTML. Khi sử dụng các công cụ chụp nhanh và giám sát như Regshot, bạn nên nhớ rằng hầu hết các công cụ sẽ chỉ có thể theo dõi các thay đổi trong bối cảnh

người dùng của riêng họ hoặc bên dưới. Điều này có nghĩa là chạy các công cụ trong tài khoản Quản trị viên sẽ cho phép bạn theo dõi các thay đổi được thực hiện ở bối cảnh người dùng đó và bên dưới, nhưng không thay đổi được thực hiện bởi các tài khoản cấp hệ thống



Hình 6. 15 InControl5 GUI

InControl5 cung cấp cho bạn một báo cáo hay (HTML, bảng tính hoặc văn bản) về các tệp và khóa Sổ đăng ký đã được thêm, sửa đổi hoặc xóa. InControl5 cũng sẽ giám sát các tệp cụ thể để thay đổi, mặc dù danh sách các tệp được theo dõi khá hạn chế. Bạn cũng có thể chọn chương trình cài đặt, chẳng hạn như tệp MSI, để InControl5 giám sát.

Khi bạn đã khởi chạy phần mềm độc hại và thu thập dữ liệu bạn cần, bạn muốn tạm dừng các công cụ giám sát thời gian thực và sau đó chạy giai đoạn thứ hai của các công cụ chụp nhanh để so sánh. Tại thời điểm này, bạn sẽ quyết định xem bạn muốn lưu nhật ký từ các công cụ giám sát thời gian thực trước hay sau khi bạn chạy giai đoạn thứ hai của các công cụ chụp nhanh. Nền tảng thử nghiệm của bạn là dành cho mục đích sử dụng của bạn và nó sẽ không được sử dụng làm bằng chứng, vì vậy, nó sẽ quyết định theo thứ tự các bước cuối

cùng này. Trước tiên tôi lưu dữ liệu được thu thập bởi các công cụ giám sát thời gian thực và sau đó hoàn tất các quy trình của công cụ chụp nhanh. Tôi biết tôi sẽ thấy các tệp mới được tạo từ các công cụ giám sát thời gian thực trong đầu ra của các công cụ chụp nhanh và tôi biết khi nào và làm thế nào các tệp đó được tạo. Do đó, tôi có thể dễ dàng tách dữ liệu đó khỏi dữ liệu do phần mềm độc hại tạo ra.

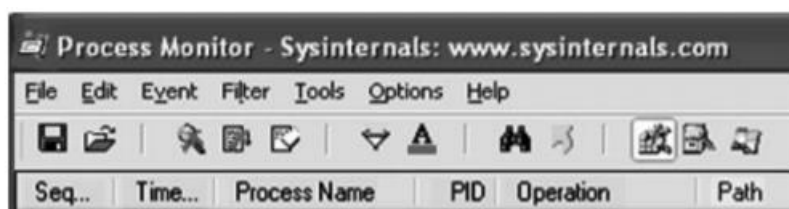
Bạn nên tạo một thư mục riêng cho tất cả các tệp nhật ký của mình. Điều này làm cho việc tách dữ liệu trong quá trình phân tích dễ dàng hơn cũng như giúp thu thập dữ liệu ra khỏi hệ thống dễ dàng hơn khi bạn đã hoàn thành việc giám sát. Trên thực tế, bạn thậm chí có thể xem xét thêm thiết bị lưu trữ di động USB vào hệ thống và gửi tất cả các tệp nhật ký của mình đến thiết bị đó.

Quá trình sẽ diễn ra như sau:

- Đảm bảo rằng tất cả các công cụ giám sát được cập nhật / cài đặt; tham khảo danh sách công cụ.
- Đảm bảo rằng tất cả các công cụ giám sát được cấu hình đúng.
- Tạo một vị trí lưu trữ nhật ký (ổ cứng cục bộ, bộ lưu trữ di động USB, v.v.).
- Chuẩn bị phần mềm độc hại cần phân tích (sao chép tệp phần mềm độc hại vào hệ thống phân tích, ghi lại vị trí bằng hệ thống tệp).
- Khởi chạy giai đoạn cơ bản của snapshot
- Cho phép các công cụ giám sát real-time
- Khởi chạy phần mềm độc hại (ghi lại phương thức khởi chạy; tác vụ theo lịch trình, nhấp đúp shell, khởi chạy từ dấu nhắc lệnh, v.v.).
- Dừng các công cụ giám sát real-time và lưu dữ liệu của chúng vào vị trí đã chỉ định.
- Khởi chạy giai đoạn thứ hai của các công cụ snapshot ; lưu dữ liệu của họ đến vị trí được chỉ định.

Bạn đã ngạc nhiên khi biết bao nhiêu dữ liệu quan trọng và hữu ích bị bỏ lỡ khi một quá trình như thế này được thực hiện. Hy vọng rằng, bạn có một quy trình mà bạn có thể làm theo, và từ đó bạn có thể đi sâu và cung cấp tên của các công cụ mà bạn sẽ sử dụng. Những công cụ này có thể thay đổi theo thời gian. Ví dụ, trong một thời gian dài, RegMon và FileMon từ Sysinternals.com là các công cụ được lựa chọn để theo dõi các truy cập hệ thống tệp và đăng ký, tương ứng, theo quy trình.

6.17 Thanh công cụ Process Monitor, hiển thị biểu tượng RegMon và FileMon



Hình 6. 16 minh họa thanh công cụ Process Monitor.

Nếu trước đây bạn đã sử dụng RegMon hoặc FileMon, thanh công cụ Process Monitor được minh họa trong Hình 6.17 có vẻ quen thuộc; hầu hết các biểu tượng là cùng một biểu tượng và theo cùng thứ tự với hai ứng dụng cũ.

Khi sử dụng Process Monitor để nắm bắt thông tin truy cập hệ thống tệp và đăng ký, bạn cần lưu ý rằng tất cả các truy cập đều bị bắt và điều này có thể tạo ra khá nhiều dữ liệu để lọc qua. Ví dụ: nhấp vào kính lúp có chữ X màu đỏ xuyên qua nó và chỉ cần ngồi và xem, mà không cần chạm vào bàn phím hoặc chuột. Các sự kiện sẽ ngay lập tức bắt đầu xuất hiện trong cửa sổ Process Monitor, mặc dù bạn đã làm điều gì đó. Khá nhiều điều xảy ra trên một hệ thống Windows mỗi giây mà bạn không bao giờ thấy. Khi xem thông tin được thu thập trong Process Monitor, bạn có thể nhấp vào một mục và chọn loại trừ. Tên quy trình để lọc ra các quy trình không cần thiết và xóa dữ liệu không liên quan. Ghi nhớ Khóa đăng ký tùy chọn thực thi tệp hình ảnh mà chúng ta đã thảo luận trong Chương 4. Process Monitor rất tốt để hiển thị cách hệ thống Windows truy cập khóa này. Để kiểm tra, hãy mở một dấu nhắc lệnh và nhập lệnh sử dụng mạng, nhưng không nhấn Enter. Mở Process Monitor và bắt đầu nắm bắt thông tin truy cập Registry. Quay trở lại dấu nhắc lệnh và nhấn Enter, và khi bạn thấy lệnh hoàn tất, hãy tạm dừng quá trình chụp Màn hình quy trình bằng cách nhấp vào kính lúp để X màu đỏ xuất hiện. Hình 6.18 minh họa một phần thông tin được ghi lại, cho thấy quá trình net.exe cố gắng xác định liệu có bất kỳ Tùy chọn thực thi tệp hình ảnh nào cho các DLL được liệt kê hay không.

\\Windows NT\\CurrentVersion\\Image File Execution Options\\ntdll.dll	NAME NOT FOUND
\\Windows NT\\CurrentVersion\\Image File Execution Options\\kernel32.dll	NAME NOT FOUND
\\Windows NT\\CurrentVersion\\Image File Execution Options\\msvcrt.dll	NAME NOT FOUND
\\Windows NT\\CurrentVersion\\Image File Execution Options\\RPCRT4.dll	NAME NOT FOUND
\\Windows NT\\CurrentVersion\\Image File Execution Options\\ADVAPI32.dll	NAME NOT FOUND
\\Windows NT\\CurrentVersion\\Image File Execution Options\\NETAPI32.dll	NAME NOT FOUND

Hình 6. 17 Process Monitor hiển thị quyền truy cập vào tệp tùy chọn thực thi tệp hình ảnh

Bạn cũng có thể cân nhắc sử dụng một số công cụ khác. Ví dụ, phiên bản tháng 7 năm 2007 của công cụ công cụ (được viết bởi Russ McRee và có sẵn từ <http://holisticinfosec.org/toolsmith/docs/july2007.pdf>), có tiêu đề Công cụ phân mềm phân tích phần mềm Mal, phân tích SysAnalyzer từ iDefense (<http://labs.iddefense.com/software/malcode.php>). SysAnalyzer cho phép bạn theo dõi trạng thái thời gian chạy hệ thống trực tiếp trong khi thực thi phần mềm độc hại trong quá trình phân tích động. SysAnalyzer sẽ giám sát các khía cạnh khác nhau của hệ thống trong khi phần mềm độc hại đang thực thi, do đó, không cần phải nói rằng hệ thống sẽ bị nhiễm; tuy nhiên, việc sử dụng một hệ thống ảo làm cho việc trở lại trạng thái nguyên sơ trước đây trở nên cực kỳ đơn giản.

Một bước cuối cùng cần lưu ý là bạn có thể muốn loại bỏ nội dung của bộ nhớ vật lý (RAM) bằng một trong các phương pháp được thảo luận trong Chương 3. Không chỉ bạn sẽ có tất cả dữ liệu từ phân tích động sẽ cho bạn biết những thay đổi nào phần mềm độc hại được tạo trên hệ thống, nhưng trong trường hợp phần mềm độc hại bị che khuất, bạn cũng sẽ có tùy chọn trích xuất hình ảnh thực thi từ kết xuất RAM, cho bạn cái nhìn về phần mềm độc hại thực sự trông như thế nào, tăng cường phân tích của bạn.

Chương này đã trình bày rất nhiều thông tin rất hữu ích để giúp bạn hiểu các tệp Windows PE, bao gồm thông tin về cấu trúc của chúng. Tuy nhiên, vào mùa hè 2008, Syngress Publishing đã xuất bản Forensics Malware: Điều tra và phân tích mã độc hại và các tác giả (Cameron Malin, Eoghan Casey và James Aquilina) nên được ghi nhận là có thể hướng dẫn toàn diện và hữu ích nhất về chủ đề này hiện nay, giải quyết cả phần mềm độc hại Windows và Linux từ một số quan điểm. Một trong nhiều khía cạnh có giá trị của cuốn sách là số lượng các công cụ có sẵn miễn phí được liệt kê có thể được sử dụng trong một loạt các kịch bản phân tích.

4. Tóm Tắt

Trong chương này, chúng tôi đã xem xét hai phương pháp bạn có thể sử dụng để thu thập thông tin về các tệp thực thi. Bằng cách hiểu các cấu trúc cụ thể của một tệp thực thi, bạn biết phải tìm gì cũng như những gì trông lạ, đặc biệt khi các hành động cụ thể đã được thực hiện để cố gắng bảo vệ tệp khỏi phân tích. Các phương pháp phân tích mà chúng tôi đã thảo luận trong chương này cho phép bạn xác định những tác động của một phần mềm (hoặc phần mềm độc hại) đối với hệ thống, cũng như các tạo phẩm mà nó để lại sẽ cho thấy sự hiện diện của nó. Đôi khi điều này hữu ích cho một nhà điều tra, vì phần mềm chống vi-rút có thể không phát hiện ra hoặc phần mô tả và mô tả của nhà cung cấp phần mềm chống vi-rút không cung cấp đủ chi tiết. Là người phản hồi đầu

tiên, những tạo tác này sẽ giúp bạn định vị các hệ thống khác trong cơ sở hạ tầng mạng có thể đã bị xâm phạm. Là một điều tra viên, những cổ vật này sẽ cung cấp cho bạn cái nhìn toàn diện hơn về sự lây nhiễm, cũng như những gì phần mềm độc hại đã làm trên hệ thống. Trong trường hợp Trojan backdoor và phần mềm điều khiển / truy cập từ xa, các tạo phẩm sẽ giúp bạn thiết lập dòng thời gian của các hoạt động trên hệ thống

Mỗi kỹ thuật phân tích được trình bày đều có những lợi ích và hạn chế của nó, và giống như bất kỳ công cụ nào, mỗi công cụ nên được chứng minh và ghi chép kỹ lưỡng. Phân tích tĩnh cho phép bạn xem những loại điều có thể xảy ra với phần mềm độc hại và nó sẽ cung cấp cho bạn manh mối về những gì bạn có thể mong đợi khi bạn thực hiện phân tích động. Tuy nhiên, phân tích tĩnh nhiều lần chỉ cung cấp một cái nhìn hạn chế vào phần mềm độc hại. Phân tích động cũng có thể được gọi là phân tích hành vi, và khi bạn thực thi phần mềm độc hại trong môi trường được kiểm soát, giám sát, bạn sẽ thấy phần mềm độc hại có ảnh hưởng gì đến hệ thống nạn nhân của vùng trộm và theo thứ tự nào. Tuy nhiên, phân tích động phải được sử dụng hết sức cẩn thận, vì bạn thực sự đang chạy phần mềm độc hại và nếu bạn không cẩn thận, bạn có thể sẽ lây nhiễm toàn bộ cơ sở hạ tầng

Ngay cả khi bạn sẽ không thực sự thực hiện bất kỳ phân tích về phần mềm độc hại nào, hãy đảm bảo ghi lại đầy đủ tài liệu đó, nơi bạn tìm thấy nó trong hệ thống tệp, bất kỳ tệp nào khác được liên kết với nó, tính toán băm mật mã, v.v. Các tác giả phần mềm độc hại luôn đặt tên cho các ứng dụng của họ với một cái gì đó nổi bật, chẳng hạn như syskiller.exe. Nhiều lần, tên của phần mềm độc hại là vô hại, hoặc thậm chí có ý định đánh lừa nhà điều tra, vì vậy việc ghi chép đầy đủ phần mềm độc hại sẽ vô cùng quan trọng.

5. *Giải pháp theo dõi*

Phân tích tĩnh

- Ghi lại bất kỳ ứng dụng hoặc tệp đáng ngờ nào bạn tìm thấy trong quá trình điều tra là bước đầu tiên để xác định những gì nó làm với hệ thống và mục đích của nó.
- Nội dung của một thực thi đáng ngờ có thể khó hiểu đối với hầu hết mọi người, nhưng nếu bạn hiểu các cấu trúc được sử dụng để tạo các tệp thực thi, bạn sẽ bắt đầu thấy cách sử dụng thông tin nhị phân trong tệp trong quá trình điều tra.
- Đừng chỉ dựa vào tên tập tin khi điều tra một tập tin đáng ngờ. Ngay cả các nhà phân tích phần mềm độc hại có kinh nghiệm đã được biết là trở thành con mồi của một kẻ xâm nhập, người chỉ mất

vài phút để cố gắng che giấu phần mềm độc hại của mình bằng cách đặt cho nó một cái tên vô hại.

Phân tích động

- Một quy trình phân tích động sẽ cho phép bạn xem phần mềm độc hại có ảnh hưởng gì đến hệ thống
- Sử dụng kết hợp các công cụ snapshot-based và real-time sẽ cho bạn thấy không chỉ các tạo phẩm do nhiễm phần mềm độc hại mà còn cả thứ tự (dựa trên thời gian) mà chúng xảy ra.
- Khi thực hiện phân tích động, nên sử dụng các công cụ giám sát không nằm trên nền tảng thử nghiệm để thông tin có thể được thu thập theo cách không bị ảnh hưởng bởi phần mềm độc hại.
- Khi phân tích phần mềm độc hại động đã được hoàn thành, nền tảng thử nghiệm có thể phải chịu phản ứng sự cố cũng như phân tích pháp y máy tính sau khi chết. Điều này không chỉ cho phép một nhà phân tích tra cứu kỹ năng của cô ấy, mà nó còn cung cấp xác minh bổ sung về các phần mềm độc hại.

6. Các câu hỏi thường gặp

Q: Khi thực hiện phản hồi sự cố, tôi thấy rằng một tệp có tên là Svchost.exe chịu trách nhiệm cho một số kết nối trên hệ thống. Hệ thống này có bị nhiễm phần mềm độc hại không?

A: câu hỏi đặt ra là hệ thống có thực sự bị nhiễm hay không, mà là liệu có phải là Svchost hay không. exe là một phần mềm độc hại. Lý do thông qua điều này, câu hỏi đầu tiên tôi sẽ hỏi là, bạn đã làm gì để xem các kết nối mạng. Cụ thể, trạng thái của các kết nối là gì? Họ có lắng nghe, chờ kết nối hoặc kết nối được thiết lập với các hệ thống khác không? Thứ hai, những cổng nào có liên quan đến các kết nối mạng? Có phải chúng thường được thấy trong sự liên kết với svchost.exe? Cuối cùng, bạn đã tìm thấy tập tin ở đâu trong hệ thống tập tin? Tệp svchost.exe thường được tìm thấy trong thư mục system32 và được bảo vệ bởi Windows File Protection (WFP), tự động chạy trong nền. Nếu không có dấu hiệu nào cho thấy WFP đã bị xâm phạm, bạn đã tính toán một hàm băm mật mã cho svchost.exe và so sánh nó với một ví dụ nổi tiếng chưa? Nhiều lần trong quá trình ứng phó sự cố, sự thiếu quen thuộc với hệ điều hành dẫn đến người phản hồi đi xuống con đường sai dẫn đến kết luận sai.

