|  |
| --- |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  Logo HvKTMM |
| ĐỀ TÀI PHÒNG CHỐNG VÀ ĐIỀU TRA TỘI PHẠM MÁY TÍNH  **Windows Forensic Analysis**  **Chương 4** |
| *Sinh viên thực hiện:*   |  |  | | --- | --- | | Nguyễn Như Ngọc | AT12E | | Hoàng Hữu Tú | AT12E | | Phạm Huy Công | AT12E | |  |  | |
|  |

**Chương 4**

Phân tích Registry

**Các giải pháp trong chương này:**

■ ■ **Bên trong Registry**

■ ■ **Phân tích Registry**

**Tóm lược**

**Giải pháp theo dõi nhanh**

**Các câu hỏi thường gặp**

Mục Lục

[**1** **Giới thiệu** 4](#_Toc25827561)

[**2** **Bên trong Registry** 5](#_Toc25827562)

[2.1 Cấu trúc Registry trong một tệp Hive 10](#_Toc25827563)

[**2.2** **Registry dưới dạng tệp nhật ký** 17](#_Toc25827564)

[**2.3** **Giám sát các thay đổi đối với Registry** 19](#_Toc25827565)

[**3** **Phân tích Registry** 21](#_Toc25827566)

[3.1 RegRipper 23](#_Toc25827567)

[**3.2** **Rip** 27](#_Toc25827568)

[3.3 RipXP 32](#_Toc25827569)

[3.4 System Information 33](#_Toc25827570)

[**3.5** **ComputerName** 34](#_Toc25827571)

[**3.6** **Network Interfaces** 37](#_Toc25827572)

[**3.7** **Địa chỉ MAC** 39](#_Toc25827573)

[**3.8** **Shares** 40](#_Toc25827574)

[**3.9** **Audit Policy và Event Logs** 42](#_Toc25827575)

[**3.10** **Wireless SSIDs** 47](#_Toc25827576)

[**3.11** **Autostart Locations** 48](#_Toc25827577)

[**3.11.1** **System Boot** 51](#_Toc25827578)

[**3.11.2** **User Login** 54](#_Toc25827579)

[**3.11.3** **User Activity** 54](#_Toc25827580)

[**3.11.4** **Enumerating Autostart Registry Locations** 59](#_Toc25827581)

[**3.11.5** **Autoruns Functionality** 61](#_Toc25827582)

[**3.11.6** **NtfsDisableLastAccessUpdate** 62](#_Toc25827583)

[**3.11.7** **NukeOnDelete** 63](#_Toc25827584)

[**3.12** **USB Removable Storage Devices** 64](#_Toc25827585)

[3.12.1 **USB Device Issues** 70](#_Toc25827586)

[**3.13** **Mounted Devices** 71](#_Toc25827587)

[**3.14** **Thiết bị cầm tay** 77](#_Toc25827588)

[**3.15** **Tìm Kiếm Người Dùng** 79](#_Toc25827589)

[**3.16** **Theo Dõi Hoạt Động Người Dùng** 85](#_Toc25827590)

[**3.16.1** **Các khóa UserAssist** 85](#_Toc25827591)

[**3.16.2** **MUICache** 91](#_Toc25827592)

[**3.16.3** **Dánh sách MRU** 94](#_Toc25827593)

[**3.16.4** **Trợ lý tìm kiếm** 101](#_Toc25827594)

[**3.16.5** **Connecting to Other Systems** 103](#_Toc25827595)

[**3.16.6** **Ghi đĩa CD** 104](#_Toc25827596)

[**3.16.7** **IM và P2P** 105](#_Toc25827597)

[**3.17** **Registry hệ thống Windows XP** 107](#_Toc25827598)

[**3.17.1** **Chuyển hướng** 119](#_Toc25827599)

[**3.17.2** **Ảo hóa** 119](#_Toc25827600)

[**3.17.3** **Deleted Registry Keys** 120](#_Toc25827601)

[**4** **Summary** 123](#_Toc25827602)

[**5** **DVD Contents** 124](#_Toc25827603)

[**6** **Giải pháp theo dõi nhanh** 125](#_Toc25827604)

[**6.1** **Bên trong Registry** 125](#_Toc25827605)

[**6.2** **Phân tích Registry** 126](#_Toc25827606)

**Chương 4 . Phân tích Registry**

# **Giới thiệu**

Đối với hầu hết các nhà quản lý và những nhà phân tích điều tra số, Registry giống như là 1 lối vào nơi bí mật, 1 cảnh quan hang động cấm của hệ điều hành Windows. Những người khác có thể thấy Registry như 1 cánh cửa tối tăm ở cuối 1 hành lang dài, với dòng chữ “ Từ bỏ hy vong, tới tất cả những người đã vào đây” viết nghuệch ngoạc trên đó. Sự thật là Registry thực sự là 1 mỏ vàng thông tin cho cả nhà quản lý và các nhà phân tích điều tra số. Trong nhiều trường hợp, phần mềm đã được sử dụng bởi những kẻ tấn công sẽ tạo ra những dấu chân trong Registry, để lại cho các điều tra viên manh mối về vụ việc. Biết được cách để tìm trong Registry, biết cách diễn giải những gì bạn đã thấy tìm thấy, sẽ tạo ra 1 con đường dài hướng tới việc cung cấp cho bạn cái nhìn sâu sắc có giá trị về hoạt động mà đã xảy ra trên hệ thống.

Mục đích của chương này là cung cấp cho bạn sự hiểu biết sâu sắc hơn về Registry và sự phong phú của thông tin mà nó nắm giữ. Bên cạnh thông tin cấu hình, Windows Registry lưu giữ thông tin liên quan đến các tệp được truy cập gần đây và có nhiều thông tin về hoạt động của người dùng. Tất cả những điều này có thể cực kỳ có giá trị, tùy thuộc vào bản chất của vụ việc mà bạn đang làm việc và các câu hỏi bạn cần trả lời. Hầu hết các phân tích về Registry mà tôi đề cập đến trong chương này sẽ là postmortem, nói cách khác, sau khi bạn đã có được một hình ảnh của hệ thống. Tuy nhiên, trong một số trường hợp, tôi sẽ mô tả phân tích từ một hệ thống trực tiếp cũng như cung cấp các ví dụ về các khóa và giá trị trông như thế nào trên một hệ thống trực tiếp.   
Có một vài lưu ý nhỏ cần lưu ý khi bạn thực hiện phân tích trực tiếp so với phân tích hậu hiện đại; Tôi sẽ chỉ ra những điều đó khi tôi thảo luận về chủ đề này.

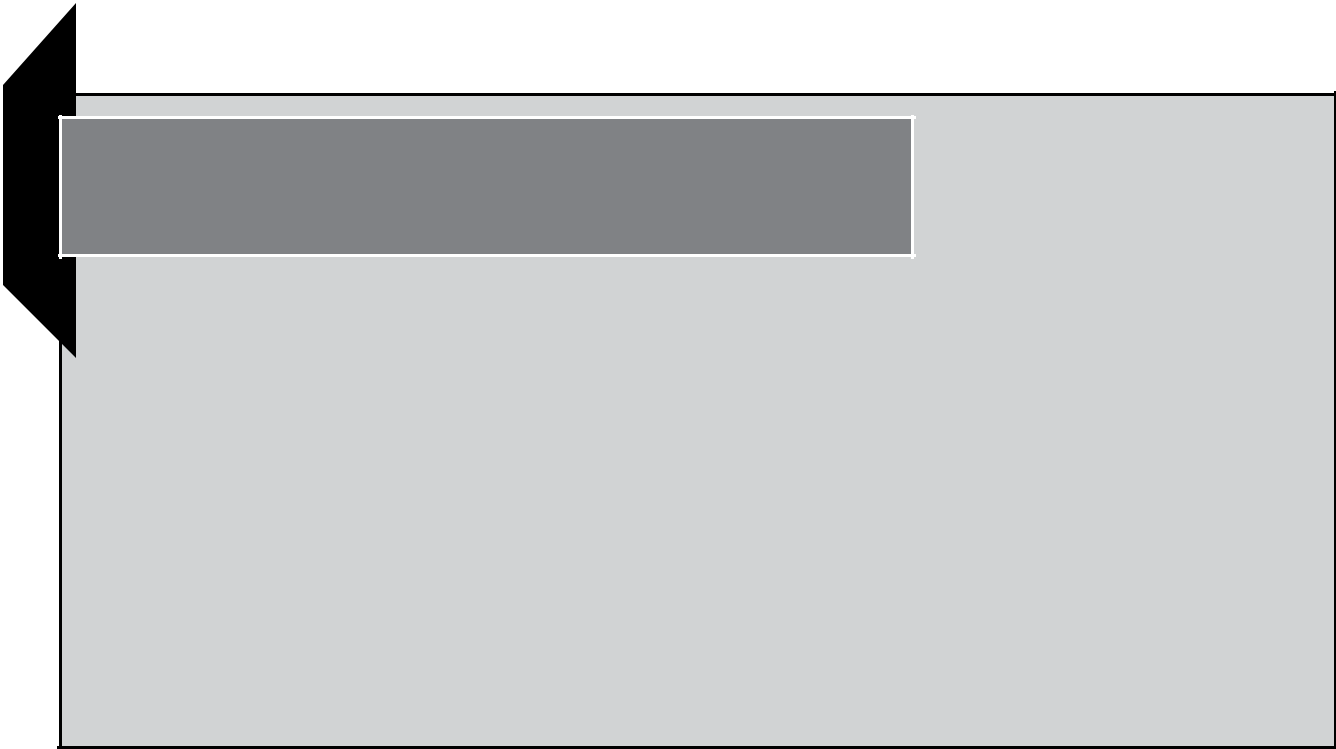
Trong suốt chương này, chúng tôi sẽ thảo luận về các khóa và giá trị Registry khác nhau có thể được quan tâm trong quá trình điều tra. Tuy nhiên, bạn không nên coi chương này là một danh sách toàn diện về tất cả các giá trị Registry có thể có thể quan tâm. Mặc dù các khóa Registry được liên kết trực tiếp với hệ điều hành có thể khá trì trệ một khi hệ thống đã được cài đặt, chúng có thể thay đổi khi Gói dịch vụ và bản vá được cài đặt, cũng như trên các phiên bản của chính hệ điều hành Windows. Ngoài ra, rất nhiều ứng dụng có sẵn, tất cả đều có các mục Registry duy nhất của riêng họ và chúng cũng có thể thay đổi giữa các phiên bản ứng dụng. Hãy nghĩ về chương này như một hướng dẫn và một danh sách tham khảo, nhưng cũng nên nhớ rằng nó không có nghĩa là hoàn thành. Mục tiêu của tôi trong chương này là mô tả định dạng của các tệp Registry, cách kiểm tra Registry và một số khóa quan trọng trong Registry và cách phân tích chúng. Vào thời điểm bạn kết thúc chương này, bạn sẽ hiểu những gì mà Registry nắm giữ và có kinh nghiệm trong việc truy xuất và phân tích thông tin từ bên trong Registry.

# **Bên trong Registry**

Vậy, Registry là gì? Nếu bạn nhớ lại DOS và các phiên bản Windows đầu tiên (3.1, 3.11, v.v.), thông tin cấu hình (trình điều khiển, cài đặt) cho hệ thống được quản lý chủ yếu bởi một số tệp tin đặc biệt, autoexec.bat, config.sys, win.ini (trên Windows),và system.ini. Nhiều cài đặt khác nhau trong các tệp này đã xác định chương trình nào đã được tải và cách hệ thống xem và phản hồi với đầu vào của người dùng.

Các phiên bản sau này của Windows đã thay thế các tệp này bằng Registry, cơ sở dữ liệu phân cấp trung tâm (http://support.microsoft.com/kb/256986) duy trì cài đặt cấu hình cho các ứng dụng, thiết bị phần cứng và người dùng. Cơ sở dữ liệu này, dữ liệu (tôi sử dụng thuật ngữ một cách lỏng lẻo) thay thế các tệp cấu hình dựa trên văn bản được sử dụng bởi các phiên bản trước của hệ điều hành Microsoft.

Đối với hầu hết các phần, các quản trị viên tương tác trực tiếp với Registry thông qua một số ứng dụng trung gian, phổ biến nhất trong số đó là các trình soạn thảo Registry giao diện đồ họa (GUI) đi kèm với Windows reg reg.exe.exe hoặc regedt32.exe. Trong nhiều trường hợp, phạm vi tương tác của người dùng và Quản trị viên với Registry là thông qua chương trình cài đặt (ứng dụng phần mềm, bản vá), không cho phép người dùng tương tác trực tiếp với các khóa và giá trị cụ thể trong Sổ đăng ký. Các bản phân phối Windows XP và 2003 bao gồm reg.exe, công cụ giao diện dòng lệnh (CLI) có thể được sử dụng từ dấu nhắc lệnh hoặc trong các tập lệnh. Để biết thêm thông tin về các tiện ích chỉnh sửa GUI Registry, hãy xem thanh bên của Reg Regit và RegEdt



**Công cụ và bẫy**

**RegEdit and RegEdt32**

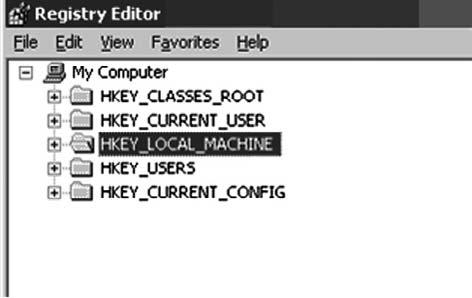
Điều thú vị là hai tiện ích chỉnh sửa Registry được cung cấp cùng với hệ điều hành Windows không tương đương với tất cả các phiên bản Windows (<http://support.microsoft.com/kb/141377>).

Ví dụ: trên Windows NT và 2000, bạn sẽ sử dụng regedit.exe chủ yếu cho các khả năng tìm kiếm của nó nhưng không sửa đổi danh sách kiểm soát truy cập (ACL) trên các khóa Registry. Regedit.exe có một giới hạn bổ sung là không hiểu về các loại dữ liệu REG\_EXPAND\_SZ (chuỗi có độ dài thay đổi) hoặc REG\_MULTI\_SZ (nhiều chuỗi). Nếu các giá trị với một trong hai loại dữ liệu này được chỉnh sửa, dữ liệu sẽ được lưu dưới dạng kiểu dữ liệu REG\_SZ (chuỗi có độ dài cố định) và chức năng của khóa bị mất. Mặt khác, Regedt32.exe không cho phép bạn nhập hoặc xuất các tệp hive (.reg).Trên Windows XP và 2003, regedit.exe cho phép bạn thực hiện tất cả những điều này và regedt32.exe không gì khác hơn là một chương trình nhỏ chạy regedit.exe.

Khi quản trị viên mở regedit.exe, anh ta thấy một cấu trúc giống như với cấu trúc

năm thư mục gốc cho phép người quản trị điều hướng dễ dàng thông qua Registry, giống như một hệ thống tệp.Các thư mục, hoặc “hives”, trong khu vực điều hướng của GUI, như Hình 4.1 minh họa. Điều này

**Hình 4.1** Regedit.exe View Showing Five Root Hives



Mỗi Hives này đóng một vai trò quan trọng trong chức năng của hệ thống. Hives HKEY\_ USERS chứa tất cả các hồ sơ người dùng được tải tích cực cho hệ thống đó. HKEY\_CURRENT\_ USER là hồ sơ người dùng đang hoạt động, được tải cho người dùng hiện đang đăng nhập. Hives HKEY\_LOCAL\_ MACHINE chứa một lượng lớn thông tin cấu hình cho hệ thống, bao gồm cài đặt phần cứng và cài đặt phần mềm. Hives HKEY\_CURRENT\_CONFIG chứa cấu hình phần cứng mà hệ thống sử dụng khi khởi động. Cuối cùng, Hives HKEY\_CLASSES\_ROOT chứa thông tin cấu hình liên quan đến ứng dụng nào được sử dụng để mở các tệp khác nhau trên hệ thống. Hives này được phân lớp thành cả HKEY\_CURRENT\_USER \ Software \ Classes (cài đặt dành riêng cho người dùng) và HKEY\_LOCAL\_MACHINE \ Software \ Classes (cài đặt toàn hệ thống).

Tất cả điều này là tốt, nhưng nó giúp biết được Hives đến từ đâu và chúng tồn tại trên ổ cứng trong hệ thống tập tin. Nội dung của phần lớn Registry hiển thị trong trình chỉnh sửa Registry có sẵn trong một số tệp, như được liệt kê trong Bảng 4.1.

**Bảng 4.1** Đường dẫn Registry và tập tin tương ứng

**Registry Path** **File Path**

HKEY\_LOCAL\_MACHINE\System HKEY\_LOCAL\_MACHINE\SAM HKEY\_LOCAL\_MACHINE\Security HKEY\_LOCAL\_MACHINE\Software HKEY\_LOCAL\_MACHINE\Hardware HKEY\_LOCAL\_MACHINE\System\Clone HKEY\_USERS\*User SID*

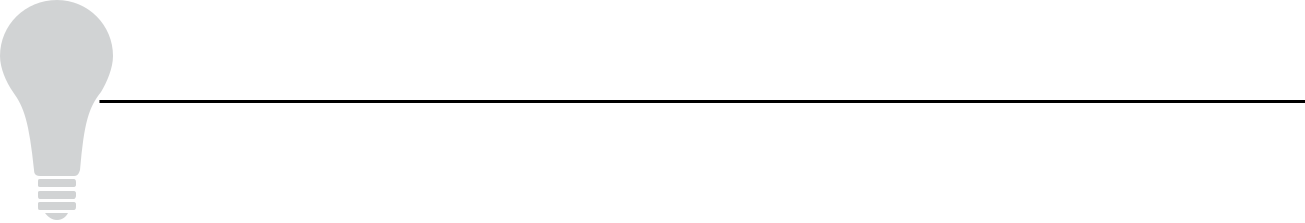
HKEY\_USERS\Default

%WINDIR%\system32\config\System %WINDIR%\system32\config\Sam %WINDIR%\system32\config\Security%WINDIR%\system32\config\SoftwareVolatile hive

Volatile hive

User profile (NTUSER.DAT); “Documents and Settings\*User* (changed to “Users\*User*” on Vista)

%WINDIR%\system32\config\defau

****

**Tip**

Vista và Windows 7 bao gồm các tệp hive Registry bổ sung, cụ thể là tệp hive Thành phần (được tìm

thấy trong thư mục system32 \ config) và tệp usrgroup.dat, nằm trong thư mục C: \ Users \ username \ AppData \ Local \ Microsoft \ Windows. Chúng ta sẽ thảo luận về những hive này, đặc biệt là hive usr class.dat, ở phần sau của chương này.

Bạn sẽ nhận thấy rằng một số đường dẫn Registry không ổn định và không tồn tại trong các tệp trên ổ cứng. Những hive này được tạo ra trong quá trình khởi động hệ thống và không khả dụng khi hệ thống tắt. Điều này rất quan trọng để nhớ khi bạn thực hiện phân tích điều tra số sau khi thao tác cũng như phản hồi trực tiếp trên một hệ thống đang chạy. Nếu dữ liệu trong hive dễ biến mất , để khắc phục sự cố hệ thống hoặc tiến hành các hoạt động ứng phó sự cố, bạn nên xem xét việc xuất toàn bộ hive dễ biến mất sang tệp .reg qua regedit.exe hoặc sử dụng một số cơ chế khác để thu thập dữ liệu cụ thể từ hive dễ biến mất trước đó tắt hệ thống.

Ngoài các phần hoặc hive khác nhau, Registry hỗ trợ một số loại dữ liệu khác nhau cho các giá trị khác nhau mà nó chứa. Bảng 4.2 liệt kê các loại dữ liệu khác nhau và mô tả của chúng.

**Bảng 4.2** Kiểu dữ liệu Registry và mô tả

|  |  |
| --- | --- |
| **Data Type** | **Description** |
|  |  |
| *REG\_BINARY* | Raw binary data |
| *REG\_DWORD* | Data represented as a 32-bit (4-byte) integer |
| *REG\_SZ* | A fixed-length text string |
| *REG\_EXPAND\_SZ* | A variable-length data string |
| *REG\_MULTI\_SZ* | Multiple strings, separated by a space, comma, |
|  | or other delimiter |
| *REG\_NONE* | No data type |
| *REG\_QWORD* | Data represented by a 64-bit (8-byte) integer |
| *REG\_LINK* | A Unicode string naming a symbolic link |
| *REG\_RESOURCE\_LIST* | A series of nested arrays designed to store |
|  | a resource list |
| *REG\_RESOURCE\_* | A series of nested arrays designed to store a |
| *REQUIREMENTS\_LIST* | device driver’s list of possible hardware resources |
| *REG\_FULL\_RESOURCE\_* | A series of nested arrays designed to store a |
| *DESCRIPTOR* | resource list used by a physical hardware device |
|  |  |

Như bạn có thể thấy, một loạt các loại dữ liệu được tìm thấy trong Registry. Dường như không có bất kỳ quy tắc hoặc tính nhất quán giữa các giá trị được tìm thấy trong các khóa khác nhau; các giá trị phục vụ tương tự mục đích có thể có các loại dữ liệu khác nhau, cho phép dữ liệu của chúng được định dạng và lưu trữ khác nhau. Điều này có thể trở thành vấn đề khi bạn thực hiện tìm kiếm văn bản cho dữ liệu trong Registry. Trường hợp một ứng dụng có thể lưu trữ danh sách các tài liệu được truy cập gần đây như Các chuỗi văn bản ASCII, một chuỗi khác có thể lưu trữ một danh sách tương tự như các chuỗi Unicode trong một kiểu dữ liệu nhị phân trong trường hợp đó, tìm kiếm văn bản ASCII sẽ bỏ lỡ dữ liệu đó. Trong thực tế, một kiến ​​thức Microsoft Bài viết cơ bản (http://support.microsoft.com/kb/161678) nói rõ rằng bạn có thể sử dụng công cụ tìm kiếm trong RegEdit chỉ để tìm dữ liệu chuỗi ASCII chứ không phải dữ liệu DWORD hoặc nhị phân.

## Cấu trúc Registry trong một tệp Hive

Bây giờ bạn đã thấy các tệp hive của Registry nằm ở đâu, hãy để bên trong xem xét

những tập tin đó và xem cấu trúc của chính Registry, ở mức độ thấp hơn nhiều. Bạn thì có lẽ tự hỏi tại thời điểm này tại sao chúng ta muốn làm điều này. Vâng, bằng cách hiểu cơ bản các thành phần Registry, chúng tôi có thể có thể thu thập thêm một số thông tin thông qua tìm kiếm từ khóa của các vị trí và nguồn khác, chẳng hạn như tệp trang, bộ nhớ vật lý,hoặc thậm chí không phân bổ không gian. Nếu chúng ta biết những gì cần tìm hoặc những gì chúng ta đang tìm kiếm, chúng ta có thể có thể trích xuất thêm một chút thông tin. Ngoài ra, bằng cách biết thêm về thông tin có sẵn trong Registry, chúng tôi sẽ hiểu rõ hơn về những gì có thể và những gì cần tìm kiếm.

Mark Russinovich đã viết một bài viết xuất sắc cho Tạp chí Windows NT có tên là Bên trong Registry (http://technet.microsoft.com/en-us/l Library

cc750583.aspx), mô tả các thành phần hoặc ô khác nhau của Registry. Thông tin tương tự này được đề cập trong Windows Internals, được Mark đồng sáng tác với David Solomon. Mỗi loại ô có cấu trúc cụ thể và chứa các loại thông tin cụ thể.Các loại ô khác nhau là :

**■■**  **Ô chính** : *Ô này chứa thông tin khóa Sổ đăng ký và bao gồm các giá trị bù cho*

*các ô khác cũng như thời gian LastWrite cho khóa (chữ ký: kn).*

**■**■ **Ô giá trị** : *Ô này chứa một giá trị và dữ liệu của nó (chữ ký: kv).*

**■**■ **Ô danh sách khóa con** : *Đây là một ô được tạo thành từ một loạt các chỉ mục (hoặc offset) trỏ đến các ô chính; đây là tất cả các khóa con cho ô khóa cha.*

**■**■ **Ô danh sách giá trị** : *Đây là một ô được tạo thành từ một loạt các chỉ mục (hoặc offset) trỏ đến các ô giá trị; đây là tất cả các giá trị của một ô khóa chung.*

**■**■ **Ô mô tả bảo mật :** *Đây là một ô chứa thông tin mô tả bảo mật cho một ô chính (chữ ký: ks).*

Hình 4.2 minh họa các loại ô khác nhau (ngoại trừ ô mô tả bảo mật) khi chúng xuất hiện trong Registry Editor.

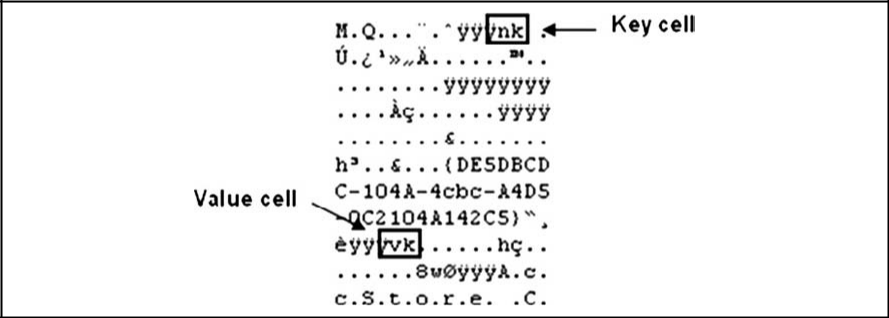
**Hình 4.2** Trích dẫn Windows Registry Hiển thị Khóa, Giá trị và Dữ liệu



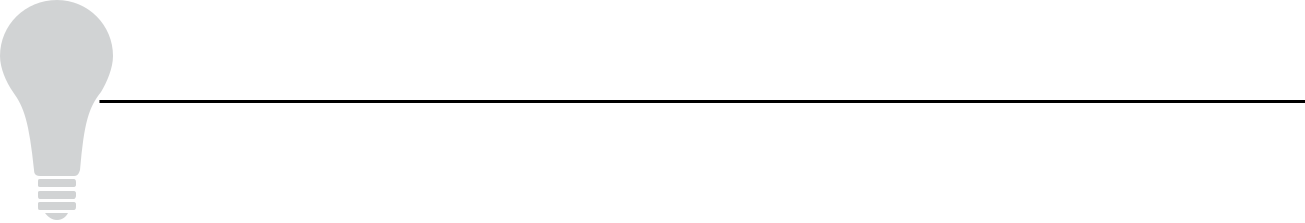
Hình 4.3 minh họa một đoạn trích của tệp Registry được mở trong trình soạn thảo hex. Hình hiển thị các chữ ký cho một ô chính cũng như cho một ô giá trị. Lưu ý rằng do các vấn đề về cuối, chữ ký cho một ô chính (i.e, kn) xuất hiện dưới dạng nk và cho một ô giá trị (i.e, kv) xuất hiện dưới dạng vk.

**Hình 4.3** Excerpt of a Raw Registry

Tệp hiển thị chữ ký khóa và giá trị



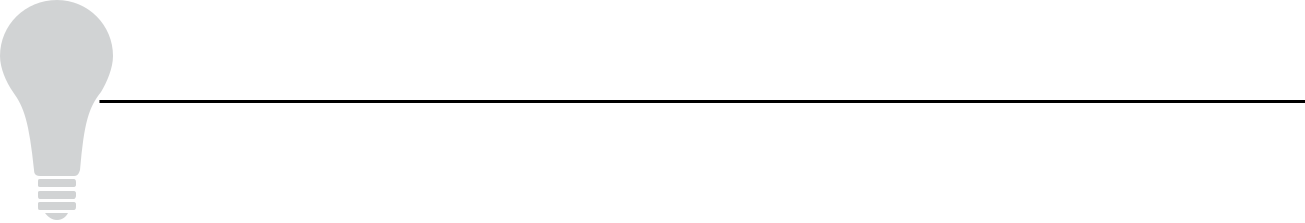
Như đã đề cập trước đó, những chữ ký này có thể cung cấp cho chúng tôi thông tin cực kỳ có giá trị trong quá trình điều tra. Sử dụng các chữ ký này, chúng tôi có khả năng có thể khắc thông tin khóa và giá trị thông tin ra khỏi các cụm chưa được phân bổ của một hình ảnh thu được hoặc thậm chí ra khỏi các bãi RAM. (Xem Chương 3 để biết thêm thông tin về các bãi RAM.)

****

**Tip**

Hiểu cấu trúc nhị phân của các khóa và giá trị Registry có thể cực kỳ hữu ích trong quá trình điều tra. Hầu hết các cuộc điều tra liên quan đến một số mức độ phân tích Registry sẽ tập trung vào chính các tệp Registry, sử dụng các công cụ được trình bày trong chương này. Tuy nhiên, khi kiểm tra không gian chưa phân bổ hoặc nội dung của kết xuất RAM, bạn có thể xác định vị trí các chữ ký như được minh họa trong Hình 4.3. Biết định dạng của khóa Registry và cấu trúc giá trị cho phép bạn, nếu cần, trích xuất và phân tích dữ liệu thành một cái gì đó dễ hiểu. Ví dụ, bạn có thể tìm thấy một khóa Registry trong bộ nhớ và có thể trích xuất thời gian LastWrite. Hiểu cấu trúc của các khóa và giá trị của Registry cung cấp ngữ cảnh cho thông tin được truy xuất từ ​​kết xuất bộ nhớ hoặc từ không gian chưa được phân bổ trong một hình ảnh thu được, cũng như cho phép bạn phân tích các khóa bị xóa từ trong không gian mật mã không được phân bổ trong các tệp hive. Chúng tôi sẽ giải quyết chủ đề định vị các khóa Registry trong không gian chưa phân bổ của các tệp hive sau trong chương này.

Tài liệu tham khảo tốt nhất cho cấu trúc nhị phân lập trình thực tế của các ô khác nhau trong các tệp hive của Registry không có sẵn từ nhà cung cấp; Thật kỳ lạ, nó có sẵn từ một người đã viết một tiện ích Linux. Peter Nordahl-Hagen đã tạo Trình chỉnh sửa mật khẩu và đăng ký ngoại tuyến (http://home.eunet.no/pnordahl/ntpasswd/), một tiện ích cho phép bạn khởi động hệ thống Windows (giả sử bạn có quyền truy cập vật lý vào hệ thống) Linux đĩa và chỉnh sửa Registry để bao gồm thay đổi mật khẩu. Peter cung cấp nguồn cho tiện ích của anh ấy và nếu bạn chọn tải xuống, bạn sẽ quan tâm nhất đến tệp có tên ntreg.h. Đây là tệp tiêu đề C xác định cấu trúc cho các loại ô khác nhau tạo nên Registry.



**Tip**

Tim Morgan đã viết một bài báo xuất sắc mô tả cấu trúc của Windows Registry, mô tả các loại ô khác nhau và cách chúng được kết nối với nhau. Tim cung cấp giấy có sẵn tại trang web Sentinelchicken.com (www.sentinelchicken.com/data/TheWindowsNTRegistryFileFormat.pdf).

Từ mã nguồn Peter, chúng ta có thể thấy rằng ô chính dài 76 byte, cộng với tên có độ dài thay đổi. Do đó, nếu chúng ta xác định vị trí chữ ký cho một ô chính (như minh họa trong Hình 4.3) trong một bãi chứa RAM hoặc trong các cụm không được phân bổ, chúng ta rất có thể có nội dung của một ô chính. Từ đó, chúng ta có thể phân tích 74 byte tiếp theo (chữ ký là hai byte dài ở định dạng hex cuối nhỏ, 0x6B6E) và xem những thứ như tên của khóa và thời gian LastWrite của nó.

Ví dụ: giả sử bạn có một số cụm chưa được phân bổ hoặc kết xuất RAM từ hệ thống và bạn đã xác định vị trí một ô chính. Đưa ra một biến gọi là $ offset là phần bù trong tệp cho ô chính mà bạn đã định vị, mã Perl để phân tích ô chính đó và trích xuất dữ liệu từ nó như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| seek(FH,$offset,0); | |  |  |
| read(FH,$data,76); | |  |  |
| my | %nk; |  |  |
| my | (@recs) | = | unpack("vvV17vv",$record); |
| $nk{id} | | = | $recs[0]; |
| $nk{type} | | = | $recs[1]; |
| $nk{time1} | | = | $recs[2]; |
| $nk{time2} | | = | $recs[3]; |
| $nk{time3} | | = | $recs[4]; |
| $nk{no\_subkeys} | | = | $recs[6]; |
| $nk{ofs\_lf} | | = | $recs[8]; |
| $nk{no\_values} | | = | $recs[10]; |
| $nk{ofs\_vallist} | | = | $recs[11]; |
| $nk{ofs\_sk} | | = | $recs[12]; |
| $nk{ofs\_classname} | | = | $recs[13]; |
| $nk{len\_name} | | = | $recs[19]; |
| $nk{len\_classname} = $recs[20]; | | | |
| # | Get the name |  |  |

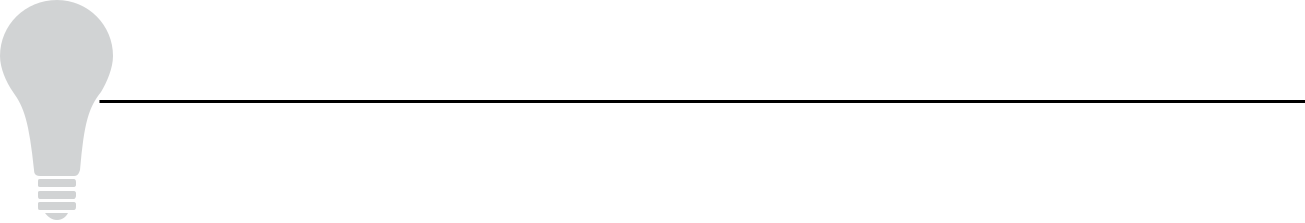
seek(FH,$offset + 76,0);

read(FH,$data,$nk{len\_name});

Trong mã này, FH là tệp xử lý Perl đối với tệp chứa dữ liệu mà chúng tôi đang kiểm tra,và $*data* là một biến vô hướng có nghĩa là chứa nội dung của những gì chúng ta vừa đọc từ tệp.Mã này cung cấp một phương pháp đơn giản, đơn giản để phân tích thông tin cần thiết từ một ô chính: giá trị định danh ô chính (i.e.,*$nk* {id}), như Hình 4.3 minh họa. Cũng trong Hình 4.3, biểu diễn thập lục phân của khóa Thời gian LastWrite được hiển thị trong lần đầu tiên một nửa dòng ngay bên dưới định danh *nk*. Mã trước đọc trong nhị phân dữ liệu và phân tích nó thành các

thành phần khác nhau dựa trên cấu trúc dữ liệu cho một ô chính.

Chúng ta có thể thấy các giá trị cung cấp cho chúng ta thời gian LastWrite cũng như số lượng khóa con và giá trị được liên kết với ô chính. Chúng tôi cũng có thể nhận được các giá trị bù cho các cấu trúc dữ liệu bổ sung, chẳng hạn như danh sách các ô chính khác (i.e., the subkeys), cũng như danh sách giá trị (i.e., values for this key).

** Tip**

Giá trị $*offset* rất quan trọng khi làm việc với các khóa Registry, đặc biệt là các tệp trong tệp hive của Registry. Giá trị 4 byte (DWORD) ngay trước chữ ký khóa được tìm thấy tại $ offset là kích thước của khóa. Khi được đọc dưới dạng giá trị dài không dấu, trong nhiều trường hợp, giá trị này sẽ là số âm. Các nhà nghiên cứu đã tìm thấy rằng một giá trị kích thước âm cho thấy rằng chính khóa đang được sử dụng; một giá trị kích thước dương cho biết khóa đã bị xóa. Chúng ta sẽ thảo luận điều này với độ dài lớn hơn trong chương sau.

Các ô giá trị chỉ dài 18 byte với tên có độ dài thay đổi và dữ liệu có độ dài thay đổi. Kiểu dữ liệu được lưu trữ trong giá trị 4 byte (DWORD) nằm trong chính ô giá trị. Sử dụng kỹ thuật tương tự như với ô chính, chúng ta có thể phân tích thông tin về ô giá trị từ bất kỳ nguồn nào mà chúng tôi đang xem và xem các giá trị thực tế. Mã Perl để phân tích ô giá trị trông như thế này:

seek(FH,$offset,0);

|  |  |  |  |
| --- | --- | --- | --- |
| my | $bytes | = | read(FH,$data,20); |
| my | (@recs) | = | unpack("vvVVVvv",$data); |
| $vk{id} | | = | $recs[0]; |
| $vk{len\_name} = $recs[1]; | | | |
| $vk{len\_data} = $recs[2]; | | | |
| $vk{ofs\_data} = $recs[3]; | | | |
| $vk{val\_type} = $recs[4]; | | | |
| $vk{flag} | | = | $recs[5]; |
| if ($vk{len\_name} == 0) { | | | |
| $vk{valname} | | = "Default"; | |

}

else {

seek(FH,$offset + 20,0);

read(FH,$data,$vk{len\_name});

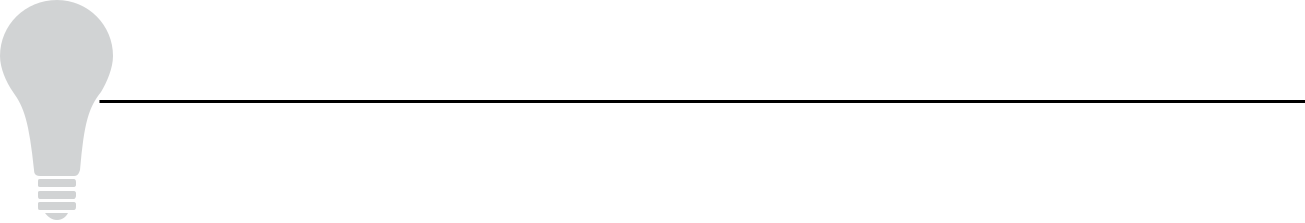
$vk{valname} = $data;

}

Như với ô chính, mã định danh ô giá trị (i.e., $*vk* {id}) đã được minh họa trong Hình 4.3. Mặc dù chúng tôi có thể phân tích dữ liệu giá trị của Registry từ

nhiều nguồn khác nhau, nhưng có thể có thông tin không cần thiết để thêm ngữ cảnh vào những gì chúng tôi khám phá, chẳng hạn như Registry nào ban đầu thuộc về khóa giá trị, khi nó được tạo ra, hoặc nó được sử dụng như thế nào. Hãy nhớ rằng, các khóa Registry có dấu thời gian được liên kết với chúng (i.e., the *LastWrite* time); Giá trị Registry thì không. Trích xuất thông tin về các khóa và giá trị từ các bãi RAM hoặc không gian chưa phân bổ có thể có giá trị giới hạn, tùy thuộc vào nhu cầu điều tra.

Trước đây, tôi đã viết một tập lệnh Perl mà tôi gọi là trình phân tích Registry ngoại tuyến, hay gọi tắt là regp.pl, dựa trên các cấu trúc mà Peter cung cấp. Nó kết hợp mã Perl mà bạn vừa thấy. Kịch bản cho phép bạn phân tích tệp Registry chưa xử lý, trích xuất thông tin ô chính và giá trị và in thông tin ra bàn điều khiển ở định dạng ASCII. Đầu ra có thể được chuyển hướng đến một tệp để lưu trữ, tìm kiếm hoặc so sánh với các tệp khác. Có một số lợi thế khi sử dụng tập lệnh này và mã giống như nó. Đầu tiên, nó cung cấp một ví dụ mã nguồn mở về cách phân tích cú pháp các tệp Registry chưa xử lý; ai đó có hiểu biết về lập trình Perl có thể dễ dàng mở rộng khả năng của tập lệnh, chẳng hạn như ghi lại đầu ra trong bảng tính hoặc cơ sở dữ liệu để xử lý dễ dàng hơn. Thứ hai, tập lệnh chỉ dựa trên chức năng Perl cơ bản; nó không sử dụng bất kỳ mô-đun cụ thể nền tảng ưa thích nào, cũng không dựa vào giao diện chương trình ứng dụng (API) của Microsoft. Điều này có nghĩa là bạn có thể chạy tập lệnh trên bất kỳ hệ thống nào hỗ trợ Perl, bao gồm cả Linux. Nếu một điều tra viên đang sử dụng Bộ công cụ Sleuth (www.sleuthkit.org) hoặc PyFlag (www.pyflag. Net / cgi-bin / moin.cgi) trên hệ thống Linux làm nền tảng phân tích, cô ấy không bị ngăn không cho phân tích cú pháp, xem, và phân tích nội dung của các tập tin Registry. Mặc dù điều này không bổ sung bất kỳ chức năng mới nào vào bất kỳ công cụ phân tích pháp y có sẵn nào, nhưng nó cung cấp các tùy chọn để giảm dữ liệu và xử lý nhanh hơn, hiệu quả hơn. Ngay khi tôi đang cân nhắc việc mở rộng tập lệnh này thành một thứ linh hoạt và linh hoạt hơn, James MacFarlane đã phát hành mô-đun Parse :: Win32Registry Perl của mình, cung cấp quyền truy cập hướng đối tượng vào các tệp hive của Registry, bao gồm cả các hệ thống Windows 98 và ME. Cũng như regp.pl, mã mô-đun James, một phần dựa trên các mô tả của Peter, về các thành phần khác nhau của Registry.

****

**Tip**

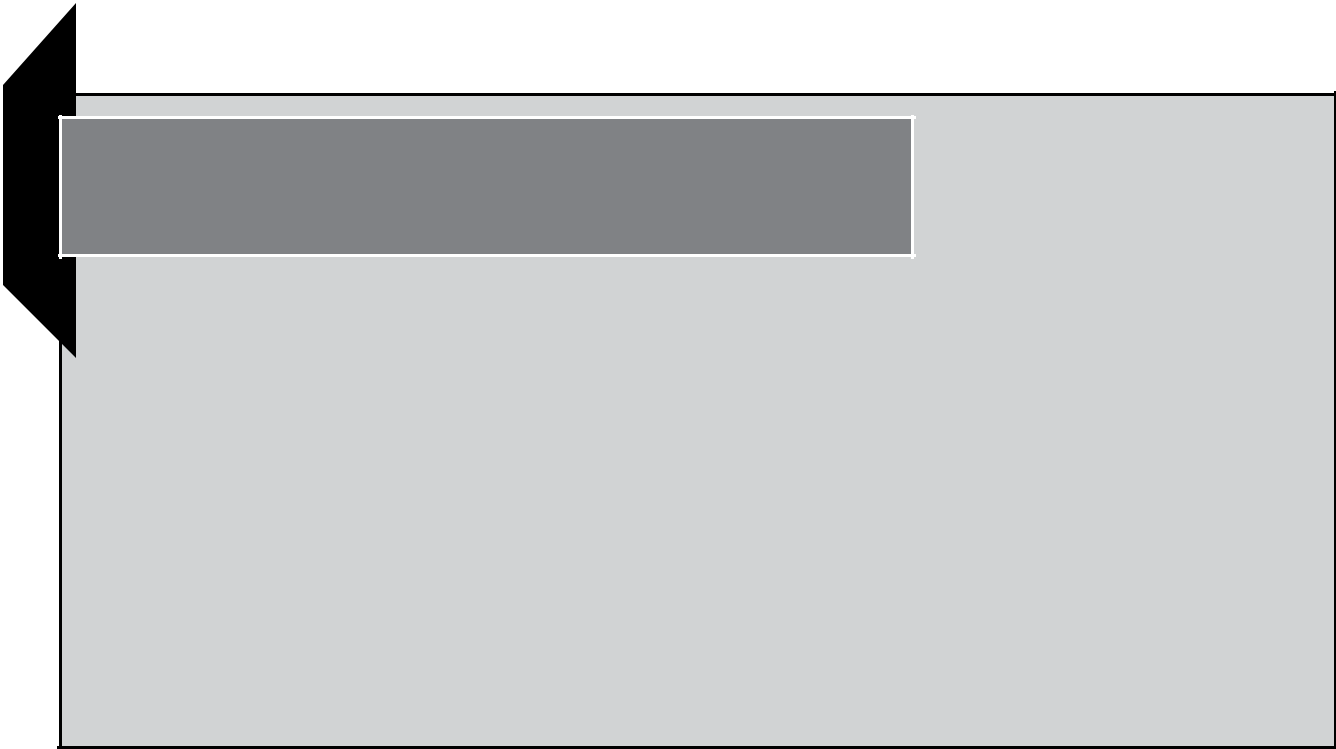
Cả PlainSight (www.plainsight.info/) và Workstation Forensic Toolkit (SIFT) Workstation (có sẵn từ http://forensics.sans.org/community/ download /) đều dựa trên Linux và bao gồm RegRipper (www.regripper.net),một công cụ dựa trên Perl để phân tích dữ liệu tệp hive của Registry trong quá trình phân tích postmortem. RegRipper dựa trên mô-đun Parse :: Win32Registry của James MacFarlane và cài đặt dễ dàng trên Windows, Linux và Mac OS X.

Tập lệnh reg.pl Perl (cũng như tập lệnh Windows thực thi Windows độc lập được biên dịch từ tập lệnh sử dụng Perl2Exe) có sẵn trên phương tiện đi kèm với cuốn sách này.Một tệp thú vị khác mà Peter cung cấp trong mã nguồn cho tiện ích của mình là sam.h. Tệp này cung cấp một số thông tin rất có giá trị liên quan đến các cấu trúc trong phần SAM của Registry. Chúng tôi sẽ thảo luận về các cấu trúc này và cách bạn có thể sử dụng chúng sau này trong chương này, trong phần Tìm kiếm người dùng trên mạng.

## **Registry dưới dạng tệp nhật ký**

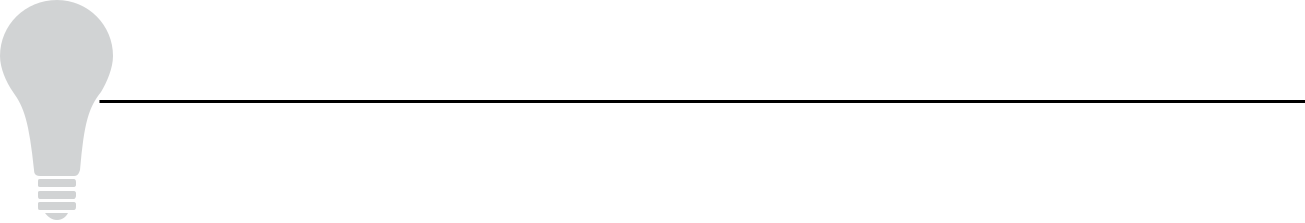
Các ô chính trong Registry tạo thành các khóa hoặc thư mục bạn nhìn thấy khi bạn mở Registry biên tập. Đây là cấu trúc duy nhất chứa giá trị thời gian, được gọi là lần cuối viết. Thời gian LastWrite là một đối tượng FILETIME 64 bit, tương tự như thời gian sửa đổi lần cuối cùng trên một tệp (xem phần bên cạnh Registry khóa của LastWrite TimeW để biết thêm thông tin). Thông tin này không chỉ cung cấp tham chiếu khung thời gian cho một số người dùng nhất định các hoạt động trên hệ thống, nhưng trong một số trường hợp, nó cũng sẽ cho bạn biết khi nào một giá trị cụ thể là thêm vào một khóa hoặc đã được sửa đổi.

Notes from the Underground…

**Registry Key LastWrite Time**

Khóa Registry có một số thuộc tính được liên kết với chúng, một trong số đó là thời gian LastWrite, tương tự như thời gian sửa đổi cuối cùng được liên kết với các tệp và thư mục. Thời gian LastWrite được lưu trữ dưới dạng đối tượng FILETIME (hoặc cấu trúc) đại diện cho số lượng khoảng thời gian 100 nano giây kể từ ngày 1 tháng 1 năm 1601 (dựa trên lịch Gregorian). Để chuyển đổi giá trị 64 bit này thành thứ gì đó dễ đọc, bạn phải dịch (http://msdn.microsoft.com/en-us/l Library / ms724280.aspx) thành cấu trúc HỆ THỐNG. Dịch trực tiếp cấu trúc FILETIME sang cấu trúc HỆ THỐNG bằng API FileTimeToSystemTime () sẽ cho phép bạn hiển thị ngày ở định dạng UTC, được định nghĩa một cách lỏng lẻo là Giờ chuẩn Greenwich (GMT). Để dịch các giá trị này thành thứ gì đó thể hiện chính xác thời gian hệ thống hiện tại, trước tiên bạn phải vượt qua cấu trúc FILETIME thông qua API FileTimeToLocalFileTime (). API này tính đến thời gian tiết kiệm ánh sáng ban ngày và thông tin múi giờ của hệ thống địa phương. Đối tượng FILETIME có thể được đọc thành Perl dưới dạng hai số 32 bit và sau đó được dịch thành giá trị thời gian UNIX 32 bit (http://en.wikipedia.org/wiki/Unix\_time) và được hiển thị bằng các hàm Perl tích hợp.

Một trường hợp cụ thể mà tôi đã thấy điều này hữu ích là trong các cuộc điều tra xâm nhập sau khi thao tác. Tôi muốn mở trình xem Registry trong ProDiscover và định vị khu vực nơi các khóa Registry dành riêng cho các dịch vụ Windows được duy trì. Từ đó, tôi sẽ sắp xếp cấp độ khóa đầu tiên dựa trên thời gian LastWrite của chúng và tôi luôn có thể dễ dàng xác định vị trí các dịch vụ được cài đặt trong khi xâm nhập, chẳng hạn như backtime truy cập từ xa hoặc thậm chí cả rootkit.

****

**Tip**

Chúng tôi sẽ thảo luận về rootkit chi tiết hơn trong Chương 7. Tuy nhiên, W32 / Opanki. worm! MS06-040, như được xác định bởi Network Associates, đóng vai trò là một ví dụ hữu ích (http://vil.nai.com/vil/Content/v\_140546.htm) cho cuộc thảo luận của chúng tôi ở đây. Sâu máy tính cài đặt một thành phần rootkit, tạo ra hai dịch vụ Windows (ngoài một số tạo phẩm khác). Bởi vì hầu hết các thỏa hiệp xảy ra sau khi hệ điều hành được cài đặt, việc sắp xếp các khóa Dịch vụ dựa trên thời gian LastWrite của chúng thường sẽ nhanh chóng tiết lộ sự cố. Đây là một kỹ thuật hữu ích để định vị bất kỳ phần mềm độc hại nào tạo ra Windows Services khi chúng lây nhiễm vào hệ thống.

Ngoài thời gian LastWrite, Registry duy trì thông tin có dấu thời gian trong một số dữ liệu được liên kết với các giá trị cụ thể. Chúng ta sẽ thảo luận về một số các giá trị này sau trong chương này, nhưng bây giờ, nó đủ để biết rằng có các khóa Registry có giá trị chứa tám byte dữ liệu nhị phân bao gồm một đối tượng FILETIME.

**Warning**

Một điều phù hợp với Windows Registry là sự thiếu nhất quán của nó. Tôi biết bạn có lẽ đang đọc cái này và nghĩ rằng có lẽ tôi đã dậy quá muộn, hoặc tôi đã uống quá nhiều cà phê. Chà, nếu bạn đã từng dành thời gian duyệt qua Registry, bạn sẽ hiểu ý tôi là gì. Một số giá trị trong Registry chứa dữ liệu nhị phân, khi được phân tích cú pháp hoặc dịch, có thể được đọc dưới dạng đối tượng FILETIME 64 bit. Hai ví dụ về điều này nằm trong một số giá trị khóa UserAssist và giá trị ShutdownTime. Tuy nhiên, trong các trường hợp khác, giá trị dấu thời gian được duy trì dưới dạng thời gian UNIX 32 bit.



Vậy, tại sao tôi lại xem Registry như một tệp nhật ký? Vì các tệp nhật ký thường có một số loại hành động hoặc sự kiện liên quan đến thời gian và trong nhiều trường hợp, điều này đúng với Registry. Mặc dù Registry có thể chứa hàng ngàn mục nhập, nhưng không phải tất cả chúng đều được quan tâm trong quá trình điều tra. Bạn sẽ quan tâm đến các phím cụ thể trong thời gian khác nhau các loại điều tra và nếu bạn hiểu cách hệ thống tạo, sửa đổi và sử dụng các khóa và giá trị đó, bạn sẽ bắt đầu thấy cách Registry thực sự ghi lại sự xuất hiện của các sự kiện khác nhau, cùng với dấu thời gian liên quan.

## **Giám sát các thay đổi đối với Registry**

Thực sự không có tài nguyên duy nhất, hợp nhất các khóa Registry sẽ hữu ích trong mọi tình huống cụ thể. Một bảng tính chứa nhiều khóa mà tôi và những người khác thấy hữu ích trong các loại điều tra khác nhau được bao gồm trong thư mục ch4 trên phương tiện đi kèm. Tuy nhiên, đây không phải là một giải pháp tất cả và cuối cùng, bởi vì đơn giản chỉ là một giải pháp.Trong một số trường hợp, các khóa Registry được tạo trong khi cài đặt hoặc được sửa đổi trong quá trình sử dụng có thể thay đổi giữa các phiên bản của một ứng dụng cụ thể. Ngay sau khi cuốn sách này được in, được xuất bản và sẵn sàng để mua, bạn có thể chắc chắn rằng một ứng dụng mới sẽ có sẵn để ghi lại cấu hình và thông tin cài đặt trong Registry.

Vì vậy, làm thế nào để bạn đi về việc xác định các khóa và giá trị Registry quan trọng đối với bạn? Một cách là chụp nhanh Registry, thực hiện hành động nguyên tử (nghĩa là chỉ làm một việc), chụp lại Registry một lần nữa và so sánh hai ảnh chụp nhanh để tìm sự khác biệt. Một công cụ cụ thể hữu ích cho nhiệm vụ này là InControl5, được Tạp chí PC (nhà phân phối ứng dụng) gọi là "Inc Incrlrl5".

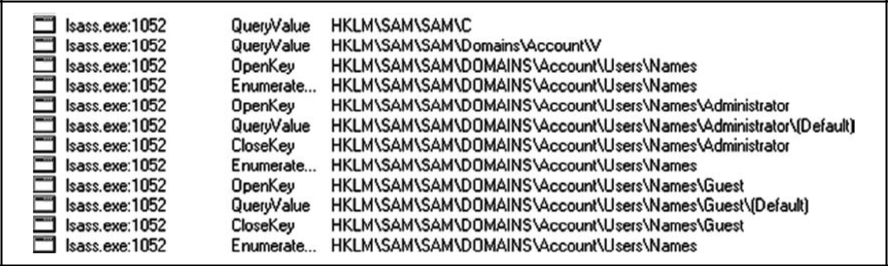
**Warning**

* Inctrl5 có sẵn từ www.pcmag.com/article2/0,4149,9882,00.asp; tuy nhiên, nó không phải là mộtphần mềm được tải về miễn phí. Nhà xuất bản tính một khoản phí nhỏ để tải xuống công cụ và thỏa thuận cấp phép đặc biệt cấm phân phối lại phần mềm.

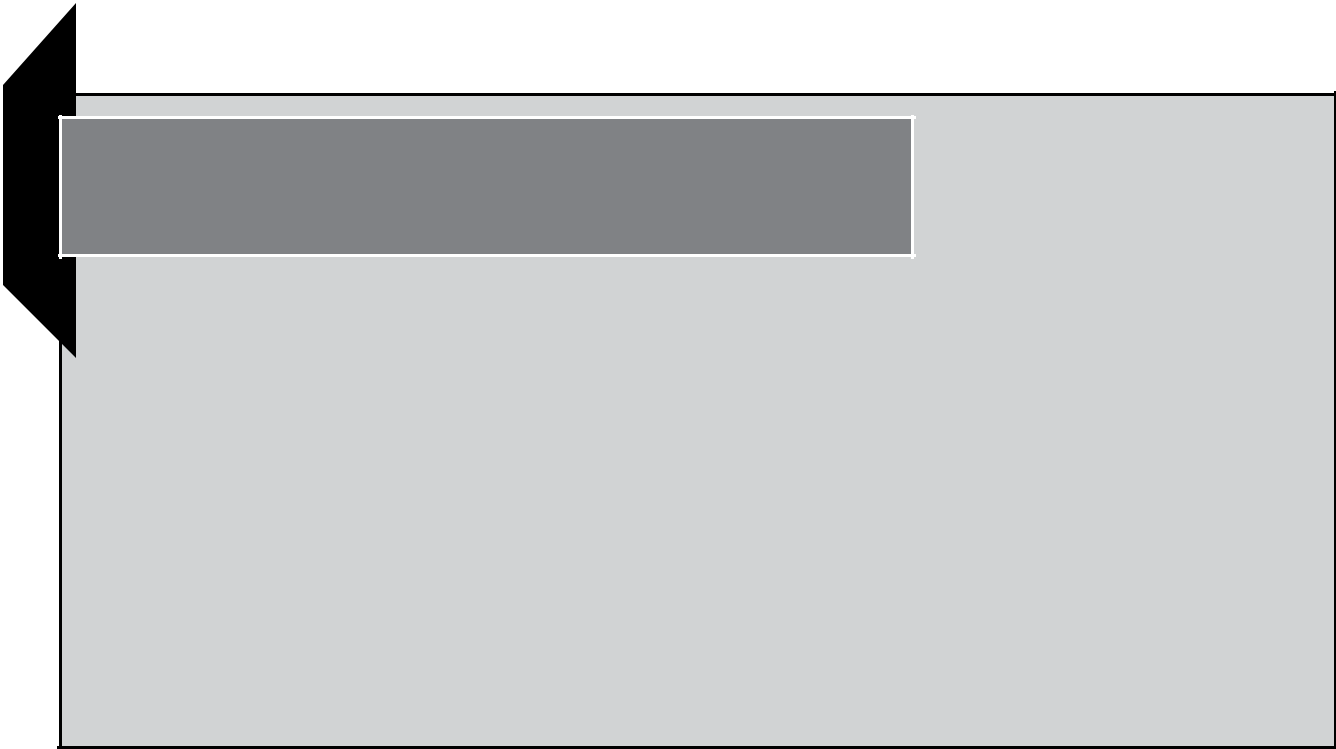
InControl5 là một tiện ích tuyệt vời để thực hiện tất cả các loại phân tích. Tôi thường có xu hướng chạy InControl5 ở chế độ hai pha, nơi tôi mở ứng dụng, để nó chụp nhanh hệ thống của tôi, làm bất cứ điều gì tôi sẽ làm (cài đặt ứng dụng chia sẻ tệp ngang hàng [P2P] hoặc một chút của phần mềm độc hại), sau đó mở lại InControl5 và hoàn tất quy trình. Đầu ra HTML thường đủ cho mục đích của tôi và tôi có thể thấy rõ các khóa Registry (và tệp) đã được thêm, sửa đổi hoặc xóa trong quá trình này.

Một cách khác để khám phá các khóa hữu ích hoặc quan trọng là sử dụng RegMon từ Microsoft (RegMon có sẵn từ http://technet.microsoft.com/en-us/sysi INTERNals / bb896652.aspx, nhưng cũng được bao gồm như một phần của tiện ích Process Explorer của Microsoft ) để theo dõi Registry trong thời gian thực. Khi tôi đang cố xác định nơi lưu giữ thông tin người dùng trong Registry, tôi thấy rằng bằng cách chạy RegMon trong khi thực hiện lệnh người dùng mạng tôi có thể xác định các khóa và giá trị nào đã được truy cập khi truy xuất thông tin người dùng, như Hình 4.4 minh họa.

**Hình 4.4** Trích đoạn đầu ra RegMon



Hình 4.4 cho thấy một đoạn trích của đầu ra RegMon được sử dụng lệnh để giám sát người dùng mạng. Tôi bắt đầu chụp, chạy lệnh và sau đó tạm dừng bắt và tìm kiếm tất cả đầu ra cho SAM. Bằng cách đó, tôi có thể thấy rằng mặc dù tôi đã sử dụng net.exe để yêu cầu thông tin, net.exe đã gửi yêu cầu đến lsass.exe, truy cập vào Registry có được thông tin liên quan đến tên người dùng trên hệ thống.



Test Drive…

**Giám sát Registry**

Ở đây, một cái gì đó bạn có thể thử. Tải xuống một bản sao của RegMon về hệ thống của bạn, mở nó và sau đó tạm dừng chụp (nhấp vào biểu tượng kính lúp để nó có dấu X màu đỏ trên nó). Sau đó mở dấu nhắc lệnh và nhập tài khoản mạng, nhưng không nhấn Enter. Quay trở lại RegMon và bắt đầu chụp, sau đó quay lại dấu nhắc lệnh và ngay lập tức nhấn Enter. Ngay khi bạn thấy các chính sách tài khoản xuất hiện trong dấu nhắc lệnh, hãy quay lại RegMon và dừng việc chụp. Để làm cho mọi thứ dễ dàng hơn, hãy lọc trên lsass.exe và xem các khóa Registry nào đã được truy cập để cung cấp thông tin mà bạn thấy trong dấu nhắc lệnh. Bây giờ khi bạn thực hiện một cuộc điều tra sau khi kết thúc, bạn sẽ biết những khóa nào cần tìm thông tin liên quan đến các chính sách tài khoản.

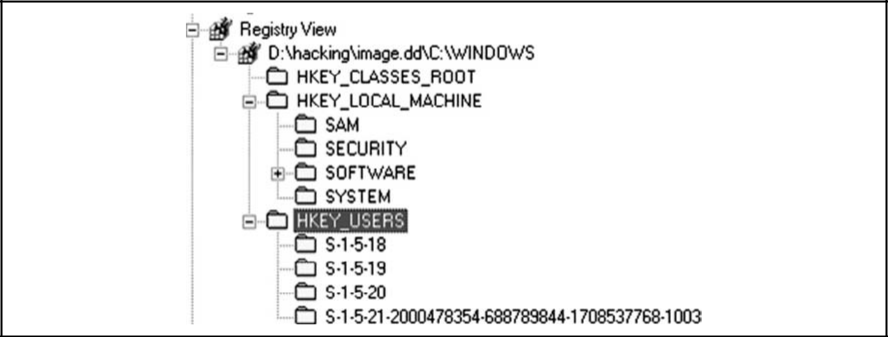
Trong suốt phần còn lại của chương, bạn sẽ thấy một số tài liệu tham khảo về việc sử dụng InControl5 cũng như RegMon. Sẽ là một ý tưởng tốt khi có các bản sao của chúng trong tay (do các điều khoản của giấy phép cho từng công cụ, chúng không thể được phân phối trên phương tiện đi kèm theo cuốn sách này), đặc biệt nếu bạn muốn theo dõi và thử một số kỹ thuật khác nhau được đề cập.

# **Phân tích Registry**

Bây giờ bạn đã biết cấu trúc của Registry, hãy để có một cái nhìn về việc truy xuất và phân tích thông tin từ bên trong Registry. Phần lớn thông tin này sẽ có sẵn cho bạn trong quá trình phản hồi trực tiếp (ngoại trừ các khóa bạn không thể truy cập do quyền) và tất cả thông tin đó (ngoại trừ các phần dễ biến mất Registry) sẽ có sẵn trong quá trình điều tra sau khi kết thúc.

ProDiscover cung cấp một giao diện đơn giản (thực ra là API) để truy cập vào Registry trong quá trình phân tích hậu kỳ. Khi một trường hợp được tải vào ProDiscover, tất cả những gì bạn cần làm là nhấp chuột phải vào thư mục Windows trong Chế độ xem nội dung và chọn Thêm vào Trình xem Registry. ProDiscover sau đó định vị các tệp cần thiết để điền vào Trình xem Registry, như Hình 4.5 cho thấy.

**Hình 4.5** Trình xem Registry trong ProDiscover IR

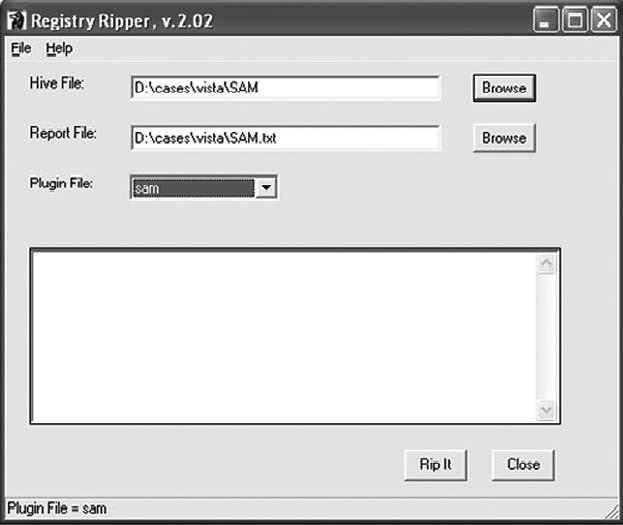


ProScript, ngôn ngữ kịch bản ProDiscover, ProScript, dựa trên Perl và cung cấp một phương tiện tuyệt vời để trích xuất thông tin từ Registry theo cách tự động. Các ProScript. mô-đun pm tạo điều kiện cho API viết các tập lệnh Perl tương tác gần như hoàn toàn với ProDiscover; bất cứ điều gì bạn có thể làm thông qua giao diện người dùng ProDiscover bạn cũng có thể làm với ProScript. Đối với Windows Registry và phân tích điều tra số sau khi kết thúc, bạn có thể sử dụng ProScripts để tự động hóa hoàn toàn rất nhiều bộ sưu tập và định dạng dữ liệu dài dòng.   
Trong suốt chương này, tôi sẽ mô tả các ProScripts để thực hiện các nhiệm vụ khác nhau như trích xuất và phân tích thông tin từ bên trong Registry trong quá trình phân tích sau khi kết thúc. Bạn có thể tìm thấy từng ProScript trong thư mục ch4 \ code \ Proscripts trên phương tiện đi kèm với cuốn sách này.

## RegRipper

Trong phiên bản đầu tiên của cuốn sách này, tôi đã bao gồm một số tập lệnh Perl có thể được sử dụng để trích xuất dữ liệu cụ thể từ các tệp hive của Registry (tất cả các tập lệnh này đã được chuyển đến thư mục ch4 \ code \ old trên phương tiện đi kèm với cuốn sách này ). Khi tôi tiếp tục thực hiện phản ứng sự cố và phân tích điều tra số, tôi thấy rằng tôi tiếp tục tìm kiếm các khóa Registry giống nhau, đồng thời thêm các khóa bổ sung vào danh sách các tạo phẩm Registry đang phát triển nhanh chóng của tôi. Tôi bắt đầu thấy rằng tôi cần một phương tiện ghi lại rất nhiều thông tin; ví dụ, đó là một điều để trích xuất một giá trị cụ thể từ khóa Registry, nhưng bao gồm thông tin tham chiếu (ví dụ: các bài viết của Cơ sở tri thức Microsoft, v.v.) đã cung cấp thêm một chút thông tin, đặc biệt là khi thêm thông tin đó vào báo cáo. Sau đó tôi nghĩ rằng tôi có thể phát triển một framework giúp dễ dàng thêm tất cả các khía cạnh này khi các khóa và giá trị Registry mới được tìm thấy phù hợp để kiểm tra để tôi có thể cung cấp framework đó cho người khác sử dụng. Ngoài ra, framework phải mạnh mẽ, nhưng đủ dễ sử dụng để ai đó có thể dễ dàng học cách cập nhật và mở rộng nó. Các công cụ dựa trên plugin như Nessus (www.nessus.org) dường như hoạt động tốt, vì vậy tôi đã viết GUI và bắt đầu phát triển những gì tôi sẽ gọi RegRipper. Hình 4.6 minh họa GUI cho RegRipper Phiên bản 2.02.

**Hình 4.6** RegRipper v.2.02 GUI



Một bản sao hoàn chỉnh, cập nhật của RegRipper v2.02 có sẵn trong thư mục ch4 \ RegRipper trên phương tiện đi kèm với cuốn sách này.

RegRipper được viết bằng Perl, vì vậy bạn có thể truy cập mã nguồn để xem những gì nó làm. GUI dựa trên mô-đun Win32 :: GUI (có nghĩa là GUI sẽ chỉ chạy trên các hệ thống Windows) và RegRipper sử dụng mô-đun Parse :: Win32Registry của James MacFarlane để thực hiện hầu hết các công cụ nâng cấp nặng nề đối với truy cập các tệp hive của Registry . Tuy nhiên, tôi cũng cung cấp RegRipper và các công cụ khác được liên kết với nó như các tệp thực thi được biên dịch với Perl2Exe (tìm thấy tại www.indigostar.com/perl2exe.htmlm; các công cụ khác như PerlApp từ ActiveState hoặc PAR, Perl Archiver, có thể được sử dụng để đạt được kết quả tương tự) để bạn có thể chạy chúng trên hệ thống Windows của mình mà không phải cài đặt Perl.

Để chạy trên các nền tảng khác, chẳng hạn như Linux, RegRipper (rr.exe) sẽ cần phải được chạy trong Wine (www.winehq.org/), với các sửa đổi nhỏ đối với mã Perl để giải quyết các vấn đề phân tách đường dẫn. Các phiên bản CLI của các công cụ RegRipper có thể được chạy từ Linux hoặc Mac OS X. Các bản phát hành trong tương lai của các công cụ này sẽ bao gồm sự quan tâm và chú ý nhiều hơn đến khả năng tương thích đa nền tảng.

Ý tưởng cơ bản đằng sau RegRipper khá đơn giản. Tôi bắt đầu nghĩ rằng tôi đã không muốn để giữ một danh sách tất cả các khóa Registry có thể tôi muốn kiểm tra dưới vô số hoàn cảnh khác nhau. Tôi thấy mình giữ danh sách các khóa và giá trị và tham chiếu đến làm thế nào các khóa được sửa đổi hoặc tại sao chúng quan trọng trong các trường hợp khác nhau, và tôi thậm chí còn có một số lưu ý về cách tương quan thông tin từ nhiều khóa Registry để xây dựng một bức tranh đầy đủ hơn về hoạt động. Tôi bắt đầu viết các đoạn mã cụ thể để làm các nhiệm vụ cụ thể và mã đó đã tạo nên rất nhiều chương này trong phiên bản đầu tiên của điều này

sách. Nhưng tôi bắt đầu thấy sự cần thiết phải có mức độ linh hoạt và khả năng sử dụng cao hơn. Suy nghĩ của tôi là tạo ra một framework công tác, giống như Nessus, sử dụng các plugin dựa trên văn bản để thực hiện tất cả các công việc nặng nâng để trích xuất dữ liệu Registry, và thậm chí thực hiện một số mức độ phân tích. Bởi vì plugin dựa trên văn bản, bất kỳ ai cũng có thể mở chúng trong trình chỉnh sửa như Notepad và đọc hoặc sửa đổi chúng; tất cả những gì cần thiết là một chút kiến ​​thức lập trình Perl. tôi đã tìm thấy rằng tôi có thể dễ dàng thêm những thứ như tài liệu tham khảo vào các bài viết của Microsoft Knowledge Base lời khuyên phân tích trên máy tính để giải thích cách giải thích thông tin được trích xuất và thông tin tương tự.Với đủ các plugin khác nhau được sử dụng làm mẫu mã, hy vọng của tôi là cuối cùng các nhà phân tích sẽ bắt đầu mở rộng các plugin và thậm chí viết các plugin của riêng họ. Đó là chính xác chuyện gì đã xảy ra.

Bạn cần hiểu một vài khái niệm đằng sau RegRipper (những khái niệm này áp dụng cho các công cụ khác trong hệ thống RegRipper, chúng tôi sẽ thảo luận ngay) để chạy công cụ đúng cách và hiệu quả. Đầu tiên, có thể truy cập các tệp hive của Registry theo một số cách: thông qua ổ cứng của hệ thống từ xa, được gắn kết qua F-Feedback, bằng cách trích xuất tệp hive từ một hình ảnh thu được (sử dụng FTK Imager, v.v.) hoặc bằng cách gắn kết Một hình ảnh thu được dưới dạng hệ thống tệp chỉ đọc thông qua các ứng dụng như Smart Mount hoặc Mount Image Pro. Một số người thậm chí đã thêm các công cụ RegRipper vào ứng dụng phân tích điều tra số phần mềm hướng dẫn của En Enase như một trình xem bên ngoài.

**Warning**

Bạn không nên chạy RegRipper đối với các tệp trung tâm Registry hệ thống trực tiếp. Chạy RegRipper và chọn bất kỳ tệp trung tâm nào trên hệ thống trực tiếp của bạn không phải là cách sử dụng công cụ này. Ngoài ra, một số người dùng đã chạy RegRipper chống lại các tệp hive được trích xuất từ ​​thư mục systemprofile hoặc Default Users và không tìm thấy bất kỳ thông tin hữu ích nào; trong nhiều trường hợp, các tệp này là các phần giữ chỗ và chứa đầy số không.Như đã đề cập trước đó, bạn có thể chạy RegRipper với các hệ thống trực tiếp nếu bạn cần, nhưng bạn cần phải có hệ thống từ xa Ổ cứng gắn trên hệ thống phân tích của bạn dưới dạng một ký tự ổ đĩa chỉ đọc thông qua F-Feedback để làm như vậy. Chương 3 đề cập đến một tài liệu PDF trên phương tiện đi kèm với cuốn sách này mô tả các bước chi tiết để cài đặt (và xóa) F-Feedback Enterprise Edition từ xa qua mạng.

RegRipper hoạt động như một framework hoặc một phương tiện để chạy các plugin chống lại các tệp hive của Registry. Plugin là các tập lệnh Perl dựa trên văn bản được duy trì trong / plugin / thư mục con trong thư mục chứa RegRipper. Các plugin có phần mở rộng .pl và chứa tất cả các hướng dẫn mã thực tế để truy cập, trích xuất và hiển thị dữ liệu tệp trung tâm của Registry.

RegRipper cũng có thể phân tích danh sách các plugin từ các tệp bổ trợ. Tệp plugin chỉ đơn giản là một tệp văn bản chứa danh sách các plugin. Trong khi một plugin có phần mở rộng .pl thì tệp plugin không có phần mở rộng. Trong tệp plugin, mỗi plugin mà nhà phân tích muốn chạy được liệt kê theo thứ tự sẽ chạy, không có phần mở rộng. Ví dụ: plugin appinitdlls.pl được viết để chạy với tệp hive Phần mềm để trích xuất dữ liệu từ giá trị Registry của AppInit\_DLLs. Trong bản phân phối RegRipper, trong thư mục / plugins /, bạn cũng sẽ thấy một tệp có tên là phần mềm trực tuyến. Tệp này chứa danh sách các plugin mà bạn có thể chạy với tệp hive Phần mềm để trích xuất nhiều thông tin. Trong tệp phần mềm, bạn sẽ thấy các appinitdll được liệt kê không có phần mở rộng; khi bạn chọn tệp này trong giao diện người dùng RegRipper hoặc thông qua dòng lệnh, công cụ sẽ chạy tất cả các plugin được liệt kê đối với tệp hive đã chọn.

Vì vậy, thay vì chạy một plugin với tệp hive cùng một lúc, bạn có thể tập hợp danh sách bổ trợ của riêng mình vào tệp plugin, đặt chúng theo thứ tự mà bạn muốn chúng chạy, và thậm chí tạo riêng bài kiểm tra của riêng bạn tập tin hive. Phân phối RegRipper có một số tệp bổ trợ mặc định dựa trên tên của các tệp trung tâm mà chúng được dự định sử dụng; các plugin chứa thông tin tương tự. Tuy nhiên, cả RegRipper và plugin đều không kiểm tra xem liệu chúng có đang chạy với loại tệp hive chính xác hay không; nếu một nhà phân tích chọn chạy một plugin dành cho tệp hive System đối với tệp hive SAM, anh ta không nên ngạc nhiên khi anh ta không tìm thấy bất kỳ thông tin hữu ích nào.

RegRipper vận chuyển cả rr.pl và rr.exe, đó là tập lệnh RegRipper Perl, được biên dịch với Perl2Exe để bạn không phải cài đặt Perl để chạy nó. Để khởi chạy RegRipper, chỉ cần nhấp đúp vào tệp rr.exe. GUI được minh họa trong Hình 4.6 mở ra và từ đó bạn có thể duyệt để chọn tệp trung tâm bạn muốn phân tích, chọn vị trí của tệp báo cáo đầu ra, sau đó chọn tệp bổ trợ mà bạn muốn chạy với hive tệp từ hộp thả xuống (RegRipper tự động điền danh sách này khi nó bắt đầu). Nhấp vào nút Rip It sẽ khởi chạy các plugin trong tệp plugin đã chọn, mỗi lần một tệp, đối với tệp hive đã chọn.

Một khía cạnh thú vị khác của RegRipper là nó tự động tạo ra một bản ghi hoạt động của nó, ghi lại các sự kiện cụ thể được bao gồm trong các plugin. Khi bạn chọn vị trí của tệp báo cáo, RegRipper sẽ tự động tạo tệp nhật ký của nó bằng cùng một tên tệp (có phần mở rộng khác nhau), trong cùng thư mục với tệp báo cáo. Ít nhất, tệp nhật ký sẽ chứa đường dẫn đến tệp hive được kiểm tra, tên và phiên bản của mỗi plugin chạy (theo thứ tự) và thời gian mà mỗi plugin được chạy. RegRipper rất nhanh, do đó thời gian trong tệp nhật ký sẽ rất gần nhau; tuy nhiên, tệp nhật ký này đóng vai trò là tài liệu kỹ lưỡng mà bạn có thể thêm vào ghi chú trường hợp của mình.

Bản phân phối RegRipper có sẵn trên các phương tiện đi kèm theo cuốn sách này, cũng như tại trang web RegRipper.net được thiết lập và duy trì bởi Brett Shavers. Các diễn đàn và trang tải xuống RegRipper.net phục vụ như một cách tuyệt vời để yêu cầu, kiểm tra và phân phối các plugin mới, điều mà nhiều người đã làm; một số đã đăng các plugin của họ lên trang web và những người khác đã gửi email công việc của họ trực tiếp cho tôi. Jason Koppe thậm chí đã viết một trình tạo plugin RegRipper (http://nssadoc.blogspot.com/2008/10/regripper-regview-and-bluetooth.html) để tạo điều kiện cho việc tạo các plugin RegRipper đơn giản.

## **Rip**

Khi làm việc với RegRipper, tôi thấy rằng trong một số trường hợp tôi chỉ muốn thực hiện kiểm tra nhanh, có thể chỉ cần chạy một plugin chống lại tệp hive. Ví dụ: khi phân tích Nhật ký sự kiện Windows, một trong những điều đầu tiên tôi làm là trích xuất chính sách kiểm toán từ tệp trung tâm bảo mật. Tôi có thể không muốn làm điều này bằng cách mở GUI, nhưng chỉ muốn xem nhanh thông tin trên bàn điều khiển để tôi có thể thực hiện phân tích của mình và sau đó dán thông tin đó vào ghi chú trường hợp của tôi. Vì vậy, tôi đã viết một phiên bản CLI của RegRipper, được gọi là bản rip, bản hay rip.pl. Giống như RegRipper, rip.pl cũng được cung cấp dưới dạng tệp thực thi của Windows để bạn không phải cài đặt Perl trên hệ thống của mình để sử dụng. Rip dựa trên cùng một khung được sử dụng bởi RegRipper; nó sử dụng các plugin tương tự.

Thông tin cú pháp cho rip.pl/.exe xuất hiện như sau:

C:\Perl\forensics\rr>rip -h

Rip v.20080419 - CLI RegRipper tool

Rip [-r Reg hive file] [-f plugin file] [-p plugin module] [-l] [-h]

Parse Windows Registry files, using either a single module, or a plugins file.

All plugins must be located in the "plugins" directory; default plugins file used if no other filename given is "plugins\plugins".

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| -r Reg hive file.... | | | Registry hive file to parse | | |
| -g | ................ | | Guess the hive file (experimental) | | |
| -f | [plugin | file].... | use the plugin file (default: plugins\plugins) | | |
| -p | plugin | module... | use | only | this module |
| -l | ................ | | list | all | plugins |
| -c | ................ | | Output list in CSV format (use with -l) | | |
| -h | ................ | | Help | (print this information) | |

Ex: C:\>rr -r c:\case\system -f system C:\>rr -r c:\case\NTUSER.DAT -p userassist C:\>rr -l -c

All output goes to STDOUT; use redirection (ie, > or >>) to output to a file.

Như bạn có thể thấy, rip.pl có rất nhiều tiện ích. Tôi đã thêm khả năng liệt kê các plugin có sẵn để tôi có thể theo dõi những gì có sẵn và sau đó tôi đã thêm khả năng liệt kê các plugin theo định dạng giá trị được phân tách bằng dấu phẩy để tôi có thể mở danh sách trong Excel và sắp xếp nó dựa trên những thứ như ngày của plugin hoặc tệp hive mà plugin dự định sử dụng. Khả năng này có thể truy cập thông qua các thiết bị chuyển mạch rip.pl, các ứng dụng Ripl Công cụ chuyển đổi trên mạng cho phép rip.pl phân tích cú pháp thông qua thư mục plugin (một lần nữa, mã hóa cứng thành / plugin) và liệt kê tất cả các plugin. Chính nó, khóa chuyển đổi liệt kê các plugin khác nhau theo thứ tự bảng chữ cái, với mỗi plugin được đánh số, như được minh họa trong đoạn trích đầu ra sau:

1. auditpol v.20080327 [Security]

Get audit policy from the Security hive file

autoendtasks v.20081128 [NTUSER.DAT]

* 1. Automatically end a non-responsive task

autorun v.20081212 [NTUSER.DAT]

* 1. Gets autorun settings

1. banner v.20081119 [Software]
   1. Get HKLM\SOFTWARE.. Logon Banner Values
2. bho v.20080418 [Software]
   1. Gets Browser Helper Objects from Software hive
3. bitbucket v.20080418 [Software]

Get HKLM\..\BitBucket keys\values

Như bạn có thể thấy, mỗi plugin được liệt kê theo thứ tự bảng chữ cái, cùng với phiên bản của plugin (đó là ngày ở định dạng YYYYMMDD), hive mà plugin dự định sẽ chạy và nội dung của trường mô tả ngắn trong chính plugin.

Thêm khóa chuyển đổi, in cùng một danh sách các plugin, chỉ không có số và trong một định dạng giá trị được phân tách bằng dấu phẩy (.csv). Bạn có thể lọc các plugin tại dòng lệnh bằng cách sử dụng lệnh find (ví dụ: C: \ Perl> rip.pl lấyl , C: \ Perl> rip.plittlọc> plugins.csv) và mở và sắp xếp danh sách trong Excel. Tôi nên chỉ ra rằng plugin banner.pl được viết bởi một thành viên của cơ quan thực thi pháp luật để đáp ứng nhu cầu của cô ấy trong quá trình phân tích; cô ấy đã đủ tốt để cung cấp cho tôi một bản sao của plugin để phân phối với RegRipper.

Một đoạn mã hữu ích khác, mặc dù đã được thử nghiệm, mà tôi đã thêm là khả năng đoán loại tệp hive (System, SAM, Software, NTUSER.DAT, Security) mà công cụ này đang chạy. Tôi đã thực hiện chuyển đổi có thể truy cập thông qua dòng lệnh, nhưng ý định của tôi là kết hợp trực tiếp mã đó vào mã và làm cho nó trong suốt cho người dùng, vì chức năng này có thể sẽ được đưa vào các phiên bản RegRipper trong tương lai.

Là một công cụ, rip.pl cực kỳ linh hoạt và mạnh mẽ. Cách cơ bản và trực tiếp nhất để sử dụng rip.pl là chạy một plugin duy nhất đối với tệp hive:

C:\Perl>rip.pl –r d:\case1\Security –p auditpol Plugins Dir = C:\Perl\forensics\rr\plugins/ Launching auditpol v.20080327 auditpol

Policy\PolAdtEv

LastWrite Time Fri Sep 9 01:11:43 2005 (UTC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Auditing is enabled. | |  |  |  |
| Audit | System | Events | = | S/F |
| Audit | Logon Events | | = | S/F |
| Audit | Object | Access | = | N |
| Audit | Privilege Use | | = | N |
| Audit | Process | Tracking | = | N |
| Audit | Policy | Change | = | S/F |
| Audit | Account | Management | = | S/F |
| Audit | Dir Service Access | | = | N |
| Audit | Account | Logon Events | = | S/F |

Trong ví dụ này, tôi đã chạy plugin Audpol dựa trên tệp hive Security, điều này (như bạn có thể thấy) cho phép tôi xem chính sách kiểm toán từ hệ thống đó. Plugin in một số thông tin về các hành động và khóa được phân tích cú pháp có thể hữu ích cho nhà phân tích (chẳng hạn như thời gian LastWrite chính), sau đó in ra chính sách kiểm toán. Bài viết 246120 của Cơ sở tri thức Microsoft (http://support.microsoft.com/kb/246120) cung cấp thông tin về cách dịch dữ liệu nhị phân trong giá trị Sổ đăng ký thành thứ dễ dàng cho nhà phân tích đọc và hiểu; plugin audpolpol.pl đảm nhiệm việc dịch thuật đó.

Các phiên bản của hệ điều hành Windows ngoài XP không có sự khác biệt của chúng, nhưng khi nói đến các tệp hive của Registry thì cấu trúc của các tệp cũng như nội dung thực tế đã thay đổi đáng kể. Ví dụ, tôi đã chạy plugin winver.pl chống lại các tệp hive Phần mềm từ cả hệ thống Vista và hệ thống Windows 7 (beta tại thời điểm viết bài này) và thấy các kết quả sau (nhấn mạnh thêm):

C:\Perl\forensics\rr>rip.pl -p winver -r d:\cases\vista\software

Plugins Dir = C:\Perl\forensics\rr\plugins/

Launching winver v.20081210 ProductName = Windows Vista (TM) Home Premium

CSDVersion = Service Pack 1 InstallDate = Fri Aug 31 15:21:10 2007 C:\Perl\forensics\rr>rip.pl -p winver -r d:\cases\win7\software Plugins Dir = C:\Perl\forensics\rr\plugins/ Launching winver v.20081210 ProductName = Windows 7 Ultimate InstallDate = Fri Dec 12 20:52:50 2008

Theo như nội dung thực tế của các tệp hive của Registry trên các phiên bản khác nhau của hệ điều hành Windows có liên quan, thì có những khác biệt trong những gì mà Lưu trữ được lưu trữ trong Registry và cách thức lưu trữ của nó; Tôi sẽ giải quyết những khác biệt trong suốt chương này.

Sử dụng khóa chuyển đổi rip.plTHERf, bạn có thể chạy toàn bộ tệp plugin đối với tệp trung tâm, thay vì chỉ một plugin tại một thời điểm. Hoặc, như rip.pl (và rip.exe được liên kết) là một công cụ CLI, bạn có thể tạo các tệp bó chỉ chạy các plugin cụ thể đối với các tệp hive. Ví dụ, audpolpol.bat được cung cấp như một phần của bản phân phối RegRipper và chứa các phần sau:

@echo off

REM Batch file to automate running of auditpol.pl plugin file

REM in order to enumerate the audit policy from the Security hive

REM file.

REM REM Usage: auditpol <path\_to\_Security\_hive\_file> REM REM copyright 2008 H. Carvey, [keydet89@yahoo.com](mailto:keydet89@yahoo.com)

**rip.exe -r %1 -p auditpol**

Như bạn có thể thấy, khối tệp chỉ là một khối tệp DOS tiêu chuẩn, hàng ngày bao gồm các nhận xét về tính dễ đọc (và như một phương tiện tài liệu). Mã thực tế được chạy trong dòng in đậm ở cuối danh sách.

Chạy plugin chống lại tệp hive được xuất từ ​​hình ảnh thu được bằng FTK Imager hoặc ProDiscover là cách duy nhất để sử dụng các công cụ này. Gắn hình ảnh thu được dưới dạng hệ thống tệp chỉ đọc bằng Smart Mount hoặc ImDisk (xem Chương 5 để biết mô tả về các công cụ này và việc sử dụng chúng), bạn có thể sử dụng rip.pl (hoặc tệp bó bằng rip.pl hoặc rip.exe) như theo sau (hình ảnh được gắn chỉ đọc là H: \):

C:\Perl>auditpol h:\windows\system32\config\security Launching auditpol v.20080327 auditpol

Policy\PolAdtEv

LastWrite Time Wed Jan 30 05:31:12 2008 (UTC)

Audit System Events = S/F

Audit Logon Events = S/F

Audit Object Access = N

Audit Privilege Use = N

Audit Process Tracking = N

Audit Policy Change = S/F

Audit Account Management = S/F

Audit Dir Service Access = N

Audit Account Logon Events = S/F

Như bạn có thể thấy, RegRipper, và đặc biệt, rip.pl, là những công cụ rất linh hoạt và linh hoạt. Thay vì mở tệp hive Registry trong trình xem (như RegEdit), các công cụ như cung cấp cho nhà phân tích một phương tiện tự động để trích xuất (và trong nhiều trường hợp dịch) dữ liệu cụ thể từ các tệp hive của Registry. Bằng cách này, một kiểm tra đầy đủ và kỹ lưỡng hơn có thể được thực hiện, theo cách nhanh hơn nhiều, giảm sai lầm và cho phép nhà phân tích tập trung vào phân tích.

Như đã đề cập trong phần giới thiệu của chương này, chúng tôi sẽ xem xét một số khóa và giá trị Registry thích hợp được tìm thấy trong các hive khác nhau có thể được các giám khảo quan tâm. Khi làm như vậy, chúng tôi sẽ xem xét một số plugin được sử dụng bởi các công cụ RegRipper (RegRipper, công cụ rip.pl CLI, cũng như ripxp.pl, được mô tả trong phần tiếp theo và được trình bày sau trong chương này). Tuy nhiên, chúng tôi sẽ không bao gồm tất cả các plugin và chúng tôi cũng không thể thảo luận về tất cả các khóa Registry có sẵn.

## RipXP

Nhóm công cụ RegRipper cũng bao gồm một công cụ hữu ích khác là biến thể của rip.pl, được gọi là ripxp.pl. Tôi đã chọn tên này vì đây là công cụ CLI dựa trên rip.pl dành riêng cho Windows XP. Windows XP duy trì Điểm khôi phục hệ thống (xem Chương 5 để thảo luận chi tiết hơn về Điểm khôi phục hệ thống XP) có chứa các phần của tệp hive Registry. Để chạy ripxp.pl, bạn cần xác định đường dẫn đến tệp hive mà bạn quan tâm để phân tích cú pháp, đường dẫn đến thư mục Điểm khôi phục và sau đó bạn muốn chạy plugin nào.

**Note**

Ripxp.pl không được cung cấp trên phương tiện truyền thông đi kèm với cuốn sách này. Tại thời điểm viết bài này, công cụ này còn quá thô để phát hành cho công chúng. Tôi đã trình diễn việc sử dụng công cụ tại Hội nghị điều tra số Sans ở Las Vegas vào tháng 10 năm 2008 và cung cấp một bản sao đánh giá công cụ này cho một người bạn, người đã thành công lớn với nó. Một phần lý do cho điều này là bạn cần

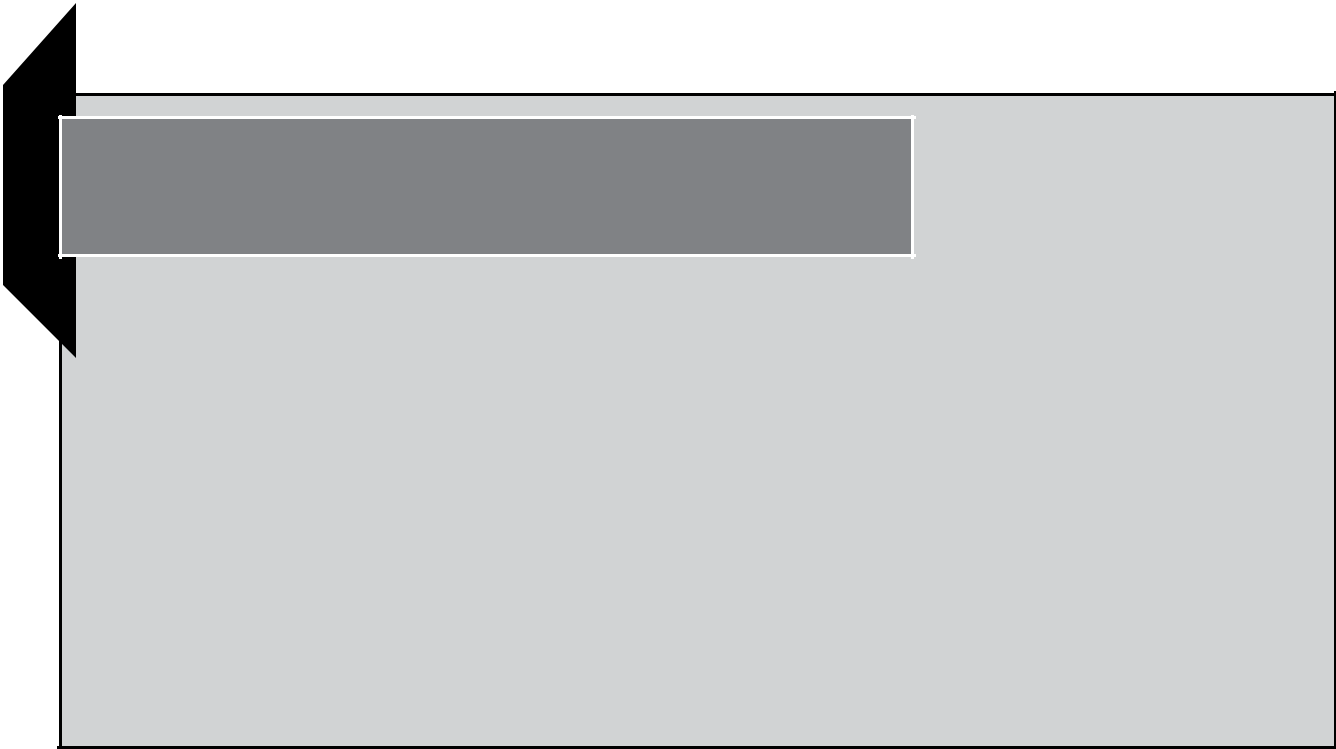
để làm theo một số bước rất cụ thể để chạy ripxp.pl, do đó, không chỉ cần phát triển các tài liệu đào tạo bổ sung, mà còn cần phải thực hiện thêm công cụ để làm cho công cụ trở nên hữu dụng hơn và tha thứ.

Ripxp.pl sẽ có thể đoán kiểu của tệp hive và sau đó xác định xem plugin có thể chạy được với tệp hive hay không. Nó sẽ chạy plugin dựa vào tệp hive, sau đó điều hướng đến thư mục Điểm khôi phục, xác định vị trí từng điểm khôi phục, xác định và hiển thị ngày tạo của điểm khôi phục, sau đó chạy plugin đối với tệp hive cụ thể. Nó làm tất cả điều này tự động.

Khả năng của các công cụ như ripxp.pl để cung cấp dữ liệu lịch sử từ bên trong Điểm khôi phục hệ thống Windows XP là vô cùng hữu ích. Như với hầu hết các công cụ mà tôi đã phát triển, nhu cầu về chức năng đó nảy sinh từ nhu cầu của riêng tôi hoặc từ người khác nói Hey, sẽ thực sự tuyệt vời nếu bạn có thể lồng vào. Trên thực tế, tín dụng cho ý tưởng về ripxp.pl dành cho Rob Lee, bởi vì vào mùa xuân năm 2008, anh ta đã nói rằng: Sẽ không hay nếu bạn có thể tự động chạy một công cụ như RegRipper chống lại XP System Restore Điểm? Chỉ tôi nghĩ, vâng, nó sẽ được. Tôi sẽ trình bày việc sử dụng ripxp.pl sau trong chương này.

## System Information

Khi làm việc với hình ảnh thu được của một hệ thống Windows trong quá trình phân tích hậu hiện đại, có rất nhiều thông tin cơ bản về hệ thống mà bạn có thể quan tâm. Phần lớn thông tin này tương đối dễ dàng có được trong quá trình phản hồi trực tiếp; ví dụ, trong nhiều trường hợp, bạn có thể xác định phiên bản của hệ điều hành (như Windows 2000, XP, 2003 hoặc Vista) bằng cách chỉ cần nhìn vào trình bao. Hoặc bạn có thể bấm chuột phải vào Máy tính của tôi và chọn Thuộc tính để xem nhiều thông tin cơ bản, chẳng hạn như phiên bản của hệ điều hành, cấp Gói dịch vụ và tên của máy tính. Thông tin này cũng có sẵn trong Registry, nơi nó có thể dễ dàng truy cập trong quá trình phân tích sau khi kết thúc. Quan trọng hơn, thông tin về chính hệ thống được duy trì chủ yếu trong các tệp hive của Hệ thống và Phần mềm (có một số trong tệp hive Bảo mật) và thông tin về người dùng được duy trì trong tệp hive SAM. Thông tin cụ thể cho người dùng và hoạt động của họ được duy trì trong tệp người dùng NTUSER.DAT. Trong phần này của chương, chúng tôi sẽ tập trung chủ yếu vào các tệp trung tâm của Hệ thống, Phần mềm và Bảo mật.

Nếu bạn còn nhớ, CurrentControlset là một phần không ổn định của Registry và bạn đã giành chiến thắng tìm thấy nó trong một hình ảnh thu được. Thanh bên tìm kiếm hiện tại thanh bên hiện tại minh họa các cách mà bạn có thể xác định bộ điều khiển nào được đánh dấu là dòng hiện tại trên hệ thống trực tiếp. Khi bạn xác định bộ điều khiển nào là hiện tại, bạn muốn tập trung kiểm tra vào các khóa trong bộ điều khiển cụ thể đó.

Forensic Feats …

**Tìm kiếm hiện tại**

Hầu hết khi bạn truy cập vào một hình ảnh thu được, bạn sẽ quan tâm để biết cái nào trong hai ControlSets có thể nhìn thấy trong Trình xem Registry (trong ProDiscover), sys-tem hoạt động được sử dụng làm CurrentControlSet. Để làm như vậy, điều hướng đến khóa HKEY\_LOCAL\_MACHINE \ System \ Chọn và bạn sẽ tìm thấy một số giá trị, như được minh họa trong Hình 4.7.

Phân phối RegRipper có một tệp mẫu mà bạn có thể sử dụng để nhanh chóng bắt đầu tạo plugin; mẫu này bao gồm mã để xác định bộ điều khiển hiện tại từ tệp hive hệ thống. Ngoài ra, bạn có thể sử dụng bất kỳ plugin nào trong thư mục / plugin truy cập vào hive System làm cơ sở cho các plugin của riêng bạn.

**Hình 4.7** Locating the*CurrentControlSet*in an Image

Như được hiển thị trong Hình 4.7, bộ điều khiển mà hệ điều hành được sử dụng như là bộ điều khiển hiện tại khi nó đang hoạt động được đánh số 1. Trong Trình xem Registry có hai ControlSets: ControlSet001 và ControlSet002. (Trong một số trường hợp, bạn sẽ tìm thấy các số khác nhau, bao gồm ControlSet03 và ControlSet04; tuy nhiên, nhìn chung bạn sẽ chỉ thấy hai ControlSets.)

## **ComputerName**

Bạn có thể sử dụng nhiều thông tin có sẵn trong hive hệ thống và phần mềm để xác định thông tin cơ bản xác định hệ thống. Các tệp hive này chứa rất nhiều thông tin mà bạn có thể sử dụng để thực hiện nhận dạng hệ thống cơ bản (nghĩa là tên hệ thống, múi giờ hệ thống được định cấu hình để sử dụng và thông tin về các thiết bị đính kèm và cổ phần có sẵn), cũng như hướng dẫn phân tích của bạn, chẳng hạn như cấu hình kiểm toán của hệ thống, như đã trình bày trước đó.

Bạn có thể tìm thấy tên máy tính trong khóa sau, trong giá trị *ComputerName*:

SYSTEM\CurrentControlSet\Control\ComputerName\ActiveComputerName

Plugin compname.pl trả về thông tin sau:

C:\Perl\forensics\rr>rip.pl -p compname -r d:\cases\system Plugins Dir = C:\Perl\forensics\rr\plugins/ Launching compname v.20080324 ComputerName = PETER

C:\Perl\forensics\rr>rip.pl -p compname -r d:\cases\win7\system

Plugins Dir = C:\Perl\forensics\rr\plugins/

Launching compname v.20080324

ComputerName = TUXDISTRO-PC

Bạn có thể tìm thấy thời gian tắt hệ thống lần cuối trong khóa sau:

SYSTEM\ControlSet00x\Control\Windows

Giá trị ShutdownTime bên dưới khóa này là một đối tượng FILETIME và có thể tương quan với các thời điểm khác trên hệ thống, chẳng hạn như các mục Nhật ký sự kiện (được thảo luận trong Chương 6) và muốn hỗ trợ phát triển dòng thời gian của hoạt động và sử dụng hệ thống. Khóa sau cũng có thể có giá trị trong quá trình điều tra:

SOFTWARE\Microsoft\Windows NT\CurrentVersion

Khóa này chứa một số giá trị cung cấp thông tin về hệ thống. Các giá trị ProductName, CurrentBuildNumber và CSDVersion sẽ cho bạn biết hệ điều hành và phiên bản nào (bao gồm Gói dịch vụ) mà bạn đang làm việc với. Các giá trị của Tổ chức đã đăng ký và Chủ sở hữu đã đăng ký, mặc dù không phải lúc nào cũng được điền vào, có thể được sử dụng để xác định thêm hệ thống. Các giá trị ProductId và InstallDate cũng có thể được sử dụng. Plugin winnt\_cv.pl trích xuất tất cả các giá trị và dữ liệu của chúng từ khóa:

C:\Perl\forensics\rr>rip.pl -p winnt\_cv -r d:\cases\vista\software

Plugins Dir = C:\Perl\forensics\rr\plugins/

Launching winnt\_cv v.20080609

WinNT\_CV

Microsoft\Windows NT\CurrentVersion

LastWrite Time Fri Dec 12 18:26:31 2008 (UTC)

RegisteredOrganization :

CurrentVersion : 6.0

CurrentBuildNumber : 6001

CurrentBuild : 6001

CSDBuildNumber : 1616

SoftwareType : System

RegisteredOwner : Harlan

SystemRoot : C:\Windows

PathName : C:\Windows

EditionID : HomePremium

CSDVersion : Service Pack 1

CurrentType : Multiprocessor Free

ProductId : 89578-OEM-7332157-00204

BuildLab : 6001.vistasp1\_gdr.080917-1612

InstallDate : Fri Aug 31 15:21:10 2007 (UTC)

ProductName : Windows Vista (TM) Home Premium

BuildGUID : 4c600e9b-ab0a-4f8e-ac60-b42c6428d3e9

BuildLabEx : 6001.18145.x86fre.vistasp1\_gdr.080917-1612

Plugin winnt\_cv.pl được viết để sắp xếp và hiển thị dữ liệu giá trị khác nhau dựa trên độ dài của dữ liệu. Tất cả các thông tin này cung cấp nhận dạng cơ bản của hệ thống đang được kiểm tra và bạn có thể sử dụng nó để ghi lại hệ thống, cũng như hướng dẫn phân tích tiếp theo.

TimeZoneInformation

Bạn có thể tìm thông tin về cài đặt múi giờ trong khóa sau:

SYSTEM\CurrentControlSet\Control\TimeZoneInformation

Thông tin này có thể cực kỳ quan trọng để thiết lập dòng thời gian hoạt động trên hệ thống. Trong suốt phần còn lại của chương này, chúng tôi sẽ thảo luận về các tập lệnh khác nhau mà bạn có thể sử dụng để lấy thông tin từ Registry; trong các chương khác, chúng tôi sẽ thảo luận về các tệp, mục Nhật ký sự kiện và những thứ tương tự. Nhiều công cụ có sẵn sẽ trích xuất thông tin về thời gian và ngày theo giờ UTC / GMT và bạn có thể sử dụng giá trị ActiveTimeBias (được liệt kê trong vài phút) từ khóa TimeZoneInform để dịch hoặc chuẩn hóa thời gian sang các nguồn khác từ hệ thống, chẳng hạn như các mục nhập trong tệp nhật ký. Plugin Regzone của timeR.pl RegRipper hiển thị thông tin cho bạn.

## **Network Interfaces**

Thông tin về giao diện mạng hoặc thẻ giao diện mạng (NIC) được duy trì trong cả tệp trung tâm Phần mềm và Hệ thống. Trong tệp hive Phần mềm, khóa Registry sau chứa thông tin về thẻ mạng:

Microsoft\Windows NT\CurrentVersion\NetworkCards

Trong tổ hợp hệ thống, thông tin về giao diện mạng được duy trì trong

 khoá Registry sau:

ControlSet00n\Services\Tcpip\Parameters\Interfaces

Cả RegRipper và rip.pl đều có thể sử dụng các plugin networkcards.pl và nic\_mst2.pl để lấy thông tin giao diện mạng từ các tệp hive tương ứng. Ví dụ: từ hệ thống Vista, plugin Networkcards.pl lấy thông tin sau:

C:\Perl\forensics\rr>rip.pl -r d:\cases\vista\software -p networkcards

Launching networkcards v.20080325

NetworkCards

Microsoft\Windows NT\CurrentVersion\NetworkCards

Broadcom 440x 10/100 Integrated Controller [Fri Aug 31 15:19:33 2007]

Intel(R) PRO/Wireless 3945ABG Network Connection [Fri Aug 31 15:19:37 2007]

Từ cùng một hệ thống Vista, plugin nic\_mst2.pl lấy ra như sau:

C:\Perl\forensics\rr>rip.pl -r d:\cases\vista\system -p nic\_mst2

Launching nic\_mst2 v.20080324

**Network key**

ControlSet001\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318} ControlSet001\Services\Tcpip\Parameters\Interfaces

LastWrite time Sun Dec 28 04:44:51 2008 (UTC)

Interface {EE564486-60AE-4868-BB0B-E1A906CC2B44}

Name: Local Area Connection

Control\Network key LastWrite time Tue Sep 11 14:33:17 2007 (UTC)

Services\Tcpip key LastWrite time Sun Dec 28 04:44:51 2008 (UTC)

DhcpDomain = us.dell.com

DhcpIPAddress = 10.12.138.119

DhcpSubnetMask = 255.255.248.0

DhcpNameServer = 143.166.95.37 143.166.99.14

DhcpServer = 143.166.165.254

Interface {D2B6079F-D864-4E0A-A852-4EB72B87E87B}

Name: Wireless Network Connection

Control\Network key LastWrite time Tue Sep 11 14:33:17 2007 (UTC)

Services\Tcpip key LastWrite time Mon Jan 12 12:46:24 2009 (UTC)

DhcpDomain =

DhcpIPAddress = 192.168.2.102

DhcpSubnetMask = 255.255.255.0

DhcpNameServer = 192.168.0.1

DhcpServer = 192.168.2.1

Thông tin trước có thể cực kỳ hữu ích trong việc không chỉ xác định hệ thống mà còn tương quan hệ thống Windows với dữ liệu khác, chẳng hạn như tệp nhật ký thiết bị mạng hoặc chụp gói tin mạng.

## **Địa chỉ MAC**

Như bạn có thể thấy, thông tin đáng kể về thiết lập mạng của hệ thống Windows (tức là, giao diện mạng và cấu hình của chúng) được duy trì trong Registry. Tuy nhiên, địa chỉ MAC Media Control Control (MAC) của hệ thống, là địa chỉ được mã hóa cứng hoặc bị đốt cháy thành một NIC, thường không được duy trì như một phần của thông tin cấu hình đó. Khi hệ điều hành Windows cần xác định địa chỉ MAC, trước tiên, nó sẽ kiểm tra khóa Registry và nếu không thể tìm thấy địa chỉ, nó sẽ truy vấn chính NIC. Windows tìm giá trị NetworkAddress trong khóa sau:

HKLM\SYSTEM\ControlSet00x\Control\Class\{4D36E972-E325-11CE-BFC1- 08002bE10318}\000n

Trong khóa Registry này, ngay bây giờ, “ 000n” là số lượng bộ điều hợp. Một lần nữa, địa chỉ MAC không được duy trì trong khóa này theo mặc định; những người viết công cụ cho phép người dùng thay đổi hoặc giả mạo địa chỉ MAC của họ (đôi khi vì mục đích bất chính), chẳng hạn như Công cụ thay đổi địa chỉ MAC Technitium (http://tmac.technitium.com/tmac/index.html), đều biết điều này. Một vị trí khác mà bạn có thể tìm thấy địa chỉ MAC đó (và tôi nhấn mạnh rằng may may ở đây vì tôi không thấy đây là hành vi nhất quán) nằm trong khóa sau:

HKLM\SOFTWARE\Microsoft\Windows Genuine Advantage

Đôi khi, tôi đã tìm thấy một giá trị trong khóa này, được đặt tên là MAC, có chứa một hoặc nhiều địa chỉ MAC được liệt kê dưới dạng chuỗi (trái ngược với dữ liệu nhị phân cần được phân tích và dịch). Ví dụ: trên một trong các hệ thống của tôi, giá trị MAC hiển thị trong khóa trước và chứa dữ liệu sau (dữ liệu này đã được sao chép và dán ra khỏi Registry):

00-15-C5-1B-97-12;00-16-CE-74-2C-B3;00-50-56-C0-00-01;00-50-56-C0-00-08;

Mỗi địa chỉ MAC được liệt kê tương ứng với các giá trị mà tôi có thể thấy rõ trong đầu ra của lệnh ipconfig / all; những người bắt đầu với phiên bản “00 00-5-56-C0”, tương ứng với các giao diện cho cài đặt ứng dụng VMware của tôi.

Plugin macaddr.pl đi kèm với bản phân phối RegRipper trên phương tiện đi kèm là một thứ gì đó của plugin lai, trong đó thay vì chỉ dành riêng cho một tệp hive duy nhất (ví dụ: tệp hive Phần mềm hoặc Hệ thống), nó bao gồm mã cho phép nó để xác định tệp hive nào (Phần mềm hoặc Hệ thống) mà nó đang được chạy, và sau đó nó truy vấn khóa Registry thích hợp để biết thông tin cần thiết. Nếu giá trị phù hợp nằm bên dưới các khóa đã nói ở trên trong các tệp hive tương ứng, plugin macaddr.pl sẽ trích xuất thông tin. Điều thú vị là trong quá trình thử nghiệm Vista, Vista 64 bit và Windows 7 các hệ thống, một số khóa trong tệp hive của Hệ thống sẽ có giá trị NetworkAddress, mặc dù không có dữ liệu.

Định vị thông tin địa chỉ MAC trong một hình ảnh có thể hỗ trợ nhà phân tích theo một số cách. Các tệp lối tắt Windows (\* .lnk) có thể có các địa chỉ MAC được nhúng trong chúng (xem Chương 5) hoặc nhà phân tích cũng có thể có dữ liệu thu thập lưu lượng mạng để làm việc và có thể cần xác định duy nhất một hệ thống dựa trên các địa chỉ MAC được tìm thấy trong thu thập dữ liệu. Ít nhất, thông tin địa chỉ MAC có thể giúp xác định duy nhất một hệ thống, ngay cả khi Giao thức cấu hình máy chủ động (DHCP) được sử dụng để gán địa chỉ Giao thức Internet (IP) khác và người dùng thay đổi tên hệ thống.

## **Shares**

Các hệ thống Windows sẽ có sẵn các chia sẻ để người dùng có thể truy cập hệ thống

từ xa. Trong hầu hết các trường hợp, điều này đúng với máy chủ tệp, nhưng nó cũng có thể đúng với máy trạm của người dùng, máy tính xách tay, v.v…Theo mặc định, các hệ thống Windows 2000, XP, 2003 và Vista sẽ tạo ra các chia sẻ quản trị ẩn trên một hệ thống. Có các chia sẻ IPC$ (truyền thông liên tiến trình), ADMIN $, các chia sẻ đề cập đến các ổ đĩa cứng trên hệ thống (C $, D $, v.v.), trong số những người khác. Nếu người dùng tạo một chia sẻ bổ sung, chẳng hạn như thông qua lệnh chia sẻ mạng, chia sẻ đó sẽ xuất hiện trong khóa sau (trừ khi có quy định khác, tất cả các khóa Sổ đăng ký trong phần này được đặt trong tổ hợp HKEY\_LOCAL\_MACHINE):

SYSTEM\CurrentControlSet\Services\lanmanserver\Shares

Bạn có thể sử dụng Shares.pl plugin để lấy thông tin về các chia sẻ có sẵn từ

một tệp từ hệ thống, như được minh họa ở đây:

C:\Perl\forensics\rr>rip.pl -p shares -r d:\cases\local\system Plugins Dir = C:\Perl\forensics\rr\plugins/

Launching shares v.20090112

print$

Path=C:\WINDOWS\system32\spool\drivers

Remark=Printer Drivers

Type=0

SharedDocs

Path=C:\DOCUMENTS AND SETTINGS\ALL USERS\DOCUMENTS

Remark=

Type=0

Printer2

Path=hp deskjet 5550 series, LocalsplOnly

Remark=hp deskjet 5550 series

Type=1

Như bạn có thể thấy, plugin lấy thông tin về chia sẻ, bao gồm tên, đường dẫn chia sẻ, mọi nhận xét và loại chia sẻ. Theo Win32\_share lớp công cụ quản lý Windows (WMI) (<http://msdn.microsoft.com/en-us/library/aa394435(VS,85).aspx)> , *Type=0* có nghĩa là chia sẻ cho ổ đĩa và *Type=1* có nghĩa là chia sẻ là cho một hàng đợi in. Bây giờ, theo mặc định, bạn có thể không thấy bất kỳ cổ phiếu nào được liệt kê dưới khóa Shares. Điều này đơn giản có nghĩa là người dùng chưa tạo bất kỳ cổ phiếu mới nào. Tuy nhiên, các hệ thống Windows sẽ tự động tạo một số chia sẻ quản trị nhất định. Nếu người dùng hoặc quản trị viên thực hiện các bước để vô hiệu hóa việc tạo các chia sẻ quản trị bị ẩn đó, bạn sẽ muốn xem bên dưới khóa sau:

SYSTEM\CurrentControlSet\Services\lanmanserver\parameters

Nếu bạn thấy một giá trị có tên *AutoShareServer* (http://[support.microsoft.com/kb/288164](https://l.facebook.com/l.php?u=http%3A%2F%2Fsupport.microsoft.com%2Fkb%2F288164%3Ffbclid%3DIwAR3fo5rlQH1B8dwbm411rKFnVKOqSSICscTGKBlwXAaMiQ3dN3cYN-Hm5jc&h=AT35NrqpSrKzojKfWMZNrSTVaD55IrkMVLByEunQNfDyHZ9gTollU_Hrc-isMKM9zD_vlIO7yoB_6I6MLHDQnNLhUszhyJAYrAVpAXDWC598r5KtqbzYBuLYwMrxeHySHoQbrPvCtleg-sGmdNuMlA)) bên dưới khóa này và dữ liệu là 0, điều này cho thấy hệ thống đã được sửa đổi cụ thể để ngăn chặn việc tạo các chia sẻ quản trị ẩn. Shares.pl plugin kiểm tra sự hiện diện của giá trị *AutoShareServer*.

**Warring**

Các phiên bản khác nhau của hệ điều hành Windows có thể đánh vần một số khóa con khác nhau. Ví dụ, trên Windows XP, khóa con Lanmanserver được đánh vần là "lanmanserver", trong khi trên Vista, nó được đánh vần là "LanmanServer". Trên Windows, khóa con Tham số (bên dưới khóa Lanmanserver) được đánh vần là "tham số" và trên Vista, nó được đánh vần là "Tham số". shares.pl plugin bao gồm mã bù cho sự chênh lệch này.

## **Audit Policy và Event Logs**

Trên các hệ thống Windows 2000, XP và 2003, chính sách kiểm toán của hệ thống ([http://support.microsoft](https://l.facebook.com/l.php?u=http%3A%2F%2Fsupport.microsoft%2F%3Ffbclid%3DIwAR3xs8ICs3ly-Wrq9vNbp0vNfIYwyFAIRoF0dInEXNXW5DfdDY7xKaBUSzI&h=AT35NrqpSrKzojKfWMZNrSTVaD55IrkMVLByEunQNfDyHZ9gTollU_Hrc-isMKM9zD_vlIO7yoB_6I6MLHDQnNLhUszhyJAYrAVpAXDWC598r5KtqbzYBuLYwMrxeHySHoQbrPvCtleg-sGmdNuMlA). Com / kb / 246120) được duy trì trong tổ hợp bảo mật, bên dưới khóa Policy\PolAdtEv. Giá trị (Default) là loại dữ liệu *REG\_NONE* và chứa thông tin nhị phân mà chính sách kiểm toán được mã hóa. Chính sách kiểm toán được trích xuất từ ​​một hình ảnh mẫu bằng cách sử dụng Trình phân tích đăng ký ngoại tuyến cho thấy:

\SECURITY\Policy\PolAdtEv

LastWrite time: Fri Sep 9 01:11:43 2005

--> Default;REG\_NONE;01 17 f5 77 03 00 00 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 00 03 00 00 00 00 00 00 00 03 00 00 00 09 00 00 00

DWCORD đầu tiên (4 byte) của dữ liệu nhị phân (thực ra là byte đầu tiên) cho bạn biết liệu kiểm toán có được bật hay không. Trong trường hợp này, giá trị là 01, do đó kiểm toán đã được bật (00 cho biết rằng nó đã bị tắt). Các hệ thống Windows 2000 và XP có chín loại sự kiện có thể được kiểm tra và mỗi khu vực đó được biểu thị bằng một giá trị DWORD trong chuỗi byte. Giá trị DWORD cuối cùng không được sử dụng.

**Chú ý**

Thông tin trong phần này chỉ áp dụng cho Windows 2000,XP và 2003. Cơ chế kiểm toán và ghi lại cho các hệ điều hành Windows đã được thay đổi với các sự ra đời của Vista, bao gồm số lượng ghi lớn hơn và định dạng ghi khác nhau ( XML, hơn là nhị phân).

Để giải mã thông tin này chúng ta cần hiểu một chút về định dạng. Ánh xạ

mẫu sau trên dữ liệu được lấy từ khóa PolAdtEv :

0Z XX XX XX AA 00 00 00 BB 00 00 00 CC 00 00 00 DD 00 00 00 EE 00 00 00 FF 00 00 00 GG 00 00 00 HH 00 00 00 II 00 00 00 XX 00 00 00

Giá trị cho Z xác định xem có bật tính năng kiểm tra hay không (1 cho bật, 0 cho tắt).Phần còn lại của các giá trị tương ứng với danh sách sau (chúng tôi không quan tâm đến các giá trị X):

AA Audit System Events

BB Audit Logon Events

CC Audit Object Access

DD Audit Privilege Use

EE Audit Process Tracking

FF Audit Policy Change

GG Audit Account Management

HH Audit Directory Service Access

II Audit Account Logon Events

Đối với mỗi cặp chữ cái, 00 có nghĩa là không có kiểm toán, 01 có nghĩa là các sự kiện thành công được kiểm toán, 02 có nghĩa là các sự kiện thất bại được kiểm toán và 03 có nghĩa là cả sự kiện thành công và thất bại đều được kiểm toán.

Chúng ta có thể thấy rằng cả kiểm tra thành công và thất bại đã được bật trên hình ảnh mẫu cho Sự kiện hệ thống, sự kiện đăng nhập, sự kiện thay đổi chính sách, sự kiện quản lý tài khoản và Sự kiện đăng nhập tài khoản.

Thông tin này có thể hữu ích khi kiểm tra Windows 2000, XP và 2003 bởi vì nó sẽ cho chúng ta biết những loại sự kiện nào chúng ta sẽ thấy trong Nhật ký sự kiện. (Chúng tôi sẽ thảo luận về phân tích Nhật ký sự kiện trong Chương 5.). RegRipper Audpol.pl sẽ trích xuất và giải thích dữ liệu từ tệp Security, như được minh họa ở đây:

C:\Perl\forensics\rr>rip.pl -p auditpol -r d:\cases\local\security

Plugins Dir = C:\Perl\forensics\rr\plugins/

Launching auditpol v.20080327

Auditpol

Policy\PolAdtEv

LastWrite Time Mon Aug 7 16:14:22 2006 (UTC)

\*\*Auditing is NOT enabled.

C:\Perl\forensics\rr>rip.pl -p auditpol -r d:\cases\security

Plugins Dir = C:\Perl\forensics\rr\plugins/

Launching auditpol v.20080327

auditpol Policy\PolAdtEv

LastWrite Time Fri Sep 9 01:11:43 2005 (UTC)

Auditing is enabled.

Audit System Events = S/F

Audit Logon Events = S/F

Audit Object Access = N

Audit Privilege Use = N

Audit Process Tracking = N

Audit Policy Change = S/F

Audit Account Management = S/F

Audit Dir Service Access = N

Audit Account Logon Events = S/F

Như bạn có thể thấy, trong một trường hợp, plugin Audpol.pl chỉ ra rằng kiểm toán không phải là được bật và mặt khác, kiểm toán đã được bật và các cài đặt khác nhau dành cho các sự kiện khác nhau (N = không kiểm toán, S = kiểm toán các sự kiện thành công, F = kiểm toán thất bại sự kiện và S / F = kiểm toán cho cả sự kiện thành công và thất bại).

Thông tin về các tệp Event Log được duy trì trong khóa HKLM\System\ ControlSet00x\Services\EventLog. Nó đúng trên tất cả các hệ thống Windows gồm Windows 2000,XP,2003,Vista và Windows 7. Tuy nhiên, sự khác biệt cho mỗi là nhiều khóa con được tìm thấy bên dưới khóa EventLog trên mỗi phiên bản Windows, như số lượng khóa con tương đương với số lượng Event Log khác nhau có sẵn. Dành cho ví dụ, trên các hệ thống Windows XP, bạn thường sẽ tìm thấy Ứng dụng, Hệ thống, Bảo mật, và thậm chí có thể là Internet Explorer (nếu hệ thống đã được nâng cấp lên Internet Explorer version 7) khóa con. Trong trường hợp Windows 2003, bạn cũng có thể tìm thấy một hệ thống tên miền (DNS) Event Log máy chủ và tôi đã thấy các phiên bản của Event Log cho các ứng dụng cụ thể. Trong hầu hết các trường hợp (đặc biệt là của Microsoft Event Log mặc định), bạn sẽ tìm thấy các giá trị liên quan đến tên và vị trí của tệp Event Log, kích thước tối đa của nó và cách nhiều ngày các mục Event Log sẽ được giữ lại (mặc dù bài viết Cơ sở tri thức Microsoft 102998, http://support.microsoft.com/kb/102998, cũ hơn, thông tin vẫn còn liên quan). Plugin eventlog.pl trích xuất thông tin này, xử lý một số cài đặt khác nhau vào một cái gì đó dễ đọc hơn một chút và hiển thị nó như được minh họa ở đây:

C:\Perl\forensics\rr>rip.pl -p eventlog -r d:\cases\local\system

Launching eventlog v.20090112

Application \ Fri Dec 12 12:06:13 2008Z

File = %SystemRoot%\system32\config\AppEvent.Evt

DisplayNameFile = %SystemRoot%\system32\els.dll

MaxSize = 512.00KB

Retention = 7.00 days

AutoBackupLogFiles = 0

Internet Explorer \ Fri Aug 29 00:36:11 2008Z

Security \ Fri Aug 29 00:36:11 2008Z

File = %SystemRoot%\System32\config\SecEvent.Evt

DisplayNameFile = %SystemRoot%\System32\els.dll

MaxSize = 512.00KB

Retention = 7.00 days

System \ Mon Oct 6 23:15:13 2008Z

File = %SystemRoot%\system32\config\SysEvent.Evt

DisplayNameFile = %SystemRoot%\system32\els.dll

MaxSize = 512.00KB

Retention = 7.00 days

Như bạn thấy , plugin eventlog.pl trích xuất thông tin đáng kể về cài đặt Event Log. Mỗi tên Event Log được hiển thị cùng với thời gian khóa *LastWirte* cho biết lần cuối cùng nội dung của khóa được thay đổi. Kích thước tệp tối đa và thời gian lưu

(được liệt kê trong Sổ đăng ký dưới dạng giây, nhưng được hiển thị bởi plugin dưới dạng ngày), ngoài tệp đường dẫn đến tệp nhật ký, được hiển thị. Như bạn có thể thấy từ trước đăng nhập tài khoản Event Log. Giá trị *AutoBackupLogFiles* (giá trị được mô tả trong Microsoft Knowledge Base article 312571, http://support.microsoft.com/kb/312571/) được hiển thị, nếu nó có sẵn. Một giá trị khác cụ thể cho bảo mật Event Log là *WarningLevel* ( được mô tả trong Microsoft Knowledge Base article 945463, <http://support.microsoft.com/kb/945463>). Khi đó sẽ khiên bảo mật Event Log tạo ra một sự kiện có ID 523 khi nó đạt đến mức chỉ định. Event log plugin cũng sẽ hiển thị giá trị này nếu có sẵn.

**TIP**

Cũng trong khoảng thời gian tôi đang viết eventlog.pl plugin, Don Weber của Cutaway Security (www.cutawaysecurity.com/) đã viết plugin riêng của mình được gọi là eventlogs.pl và đăng nó lên diễn đàn RegRipper.net.

Ngoài ra, là một phần của Windows 2000 Resource Kit, Microsoft đã phát hành một Quản lý Event Log tập lệnh Perl có tên eventlog.pl, được mô tả trong Microsoft Knowledge Base article 318763 (http://support.microsoft.com/kb/318763). Kịch bản này được dự định để chạy trên một hệ thống trực tiếp và cho phép quản trị viên để có được nhiều thông tin giống như có sẵn từ đặc biệt, RegRipper eventlog.pl plugin, thuộc tính của Event Log.

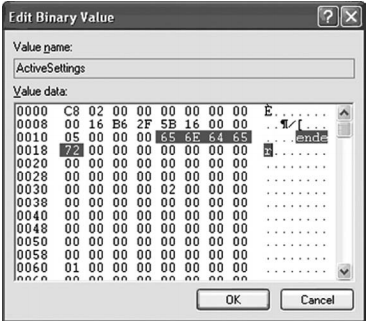
## **Wireless SSIDs**

Trên các hệ thống trực tiếp (thường là máy tính xách tay), Windows sẽ duy trì một danh sách các mã định danh bộ dịch vụ (SSID) mà nó đã kết nối. Nếu các kết nối không dây được quản lý bởi Wireless Zero Configuration Service (WZCSVC), danh sách này được duy trì trong khóa Sổ đăng ký sau:

SOFTWARE\Microsoft\WZCSVC\Parameters\Interfaces\{GUID}

GUID trong trường hợp này là mã định danh duy nhất toàn cầu (GUID) cho giao diện không dây. Bên dưới khóa này, bạn có thể thấy giá trị *ActiveSettings* và sau đó là một vài giá trị khác được gọi là *Static#000x*. trong đó x là một số nguyên, bắt đầu từ 0. Các giá trị này đều là nhị phân và SSID cho mọi điểm truy cập không dây đã được truy cập sẽ được bao gồm trong dữ liệu nhị phân. Trong dữ liệu nhị phân, tại offset 0x10, là giá trị DWORD có chứa chiều dài của SSID. Tên SSID ngay sau giá trị DWORD này cho số byte / ký tự được liệt kê. Hình 4.8 minh họa nội dung nhị phân của *Activesettings* giá trị, được lấy từ một hệ thống sống. Lưu ý rằng SSID đã được tô sáng.

Hình 4.8 Giá trị cài đặt ActiveS Settings từ hệ thống trực tiếp, hiển thị SSID



Bây giờ và một lần nữa, một câu hỏi xuất hiện trên một danh sách công khai liên quan đến việc xác định SSID mà hệ thống đã được kết nối. Điều này có thể hữu ích trong các tình huống trái phép truy cập là một vấn đề hoặc trong trường hợp điều đó rất quan trọng để theo dõi địa chỉ IP mà cá nhân đó Hình 4.8 Giá trị cài đặt *Activesettings* từ hệ thống trực tiếp, hiển thị SSID phân tích đăng ký đã sử dụng. Do đó, ssid.pl plugin sẽ trích xuất dữ liệu từ tổ hợp phần mềm, bao gồm cả SSID của điểm truy cập không dây được kết nối với, cũng như ngày cuối cùng kết nối với SSID đó, như được minh họa ở đây:

C:\Perl\forensics\rr>rip.pl -p ssid -r d:\cases\test\software

Plugins Dir = C:\Perl\forensics\rr\plugins/

Launching ssid v.20080327

SSID

Microsoft\WZCSVC\Parameters\Interfaces

NIC: 11a/b/g Wireless LAN Mini PCI Express Adapter

Static#0000 SSID : tmobile [Wed Oct 3 16:44:25 2007]

Static#0001 SSID : ender [Mon Oct 8 10:12:46 2007]

Từ thông tin này, chúng ta có thể thấy rằng hệ thống (trong trường hợp này là máy tính xách tay) được kết nối lần cuối đến một điểm truy cập không dây với SSID của “tmobile” (thường được cung cấp qua Starbucks, trong số các địa điểm khác) vào tháng 10 năm 2007 và “ender” SSID trong cùng tháng. Thông tin này có thể cực kỳ có giá trị trong việc thiết lập dòng thời gian hoạt động trên hệ thống, đặc biệt khi tương quan với dữ liệu khác. Tuy nhiên, hãy nhớ rằng thông tin dựa trên thời gian từ plugin này (và cả những thứ khác nữa) được trình bày trong Universal Coordinated Time, hoặc định dạng UTC, và có thể cần phải được sửa cho phù hợp múi giờ.

## **Autostart Locations**

Tự động khởi chạy các vị trí trong Registry là các vị trí cho phép các ứng dụng được khởi chạy không có bất kỳ sự tương tác nào từ người dùng và rất thường không biết đến người dùng như làm một cái gì đó trên hệ thống. Những vị trí này ban đầu được cung cấp để thuận tiện cho người dùng. Một số ứng dụng, như trình điều khiển cảm ứng và ứng dụng trên máy tính xách tay, cũng như phần mềm chống vi-rút và các ứng dụng tường lửa, rất hữu ích khi chúng khởi động tự động. Trong những trường hợp khác, Người dùng muốn có các ứng dụng như ứng dụng khách nhắn tin tức thời (IM) bắt đầu tự động khi họ đăng nhập vào hệ thống của họ, như một vấn đề thuận tiện.

Vào năm 2004, các thành viên của nhóm nghiên cứu quản lý hệ thống và quản lý hệ thống của Microsoft tại Trung tâm nghiên cứu của Microsoft đã trình bày một bài báo, “Gatekeeper: Giám sát các điểm mở rộng tự động bắt đầu để quản lý hệ thống”, tại một hội nghị USENIX. Trong bài báo đó, các tác giả đề cập đến các vị trí *auto-start extensibility points* hoặc ASEP. Bài viết phân loại ASEP theo cách khác với tôi và cung cấp một biểu đồ cho thấy có lẽ ASEP phổ biến nhất được sử dụng bởi phần mềm gián điệp (tại thời điểm dữ liệu được biên dịch) để cài đặt làm đối tượng trợ giúp trình duyệt (BHO). BHO về cơ bản là một thư viện liên kết động (DLL) mà Internet Explorer tự động tải khi nó khởi động và từ đó nó có thể giám sát các hoạt động của người dùng. Xem nghiên cứu điển hình của (BHOs) để biết ví dụ về cách bạn có thể sử dụng autostart / ASEP.

***Are You Owned?***

*Nghiên cứu điển hình: BHOs*

*Tôi là quản trị viên bảo mật tại một công ty dịch vụ tài chính khi tôi sử dụng BHOs ​​thú vị. Chủ nhân của tôi đã cung cấp dịch vụ giám sát tín dụng và bảo vệ chống trộm danh tính cho khách hàng của mình, cho phép nhiều cấp độ giám sát tín dụng của họ, từ báo cáo hàng quý đến các trang gần như ngay lập tức và / hoặc e-mail mỗi khi có truy vấn xuất hiện. Và giống như hầu hết các doanh nghiệp, chủ nhân của tôi đã có sự cạnh tranh. Tại một thời điểm, tôi nhận được một cuộc gọi từ những người ở phía phát triển kinh doanh của ngôi nhà với một vấn đề an ninh tiềm ẩn. Một số người dùng đã truy cập trang Web của công ty và khi họ tải trang Web chính, mỗi phiên bản tên công ty của chúng tôi được tô sáng và nhấp vào đó sẽ đưa người dùng đến một trang web của đối thủ cạnh tranh!*

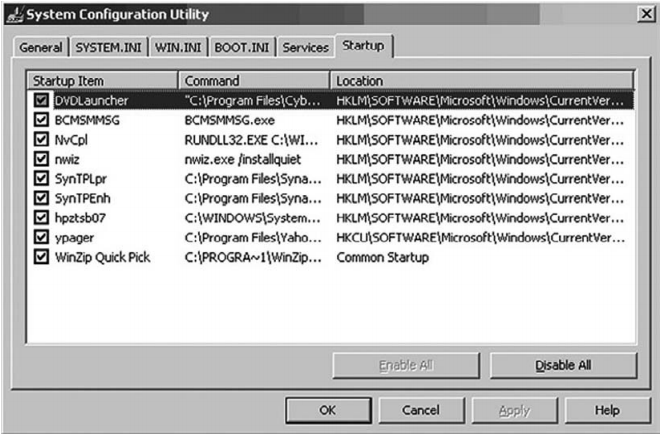
*May mắn thay, Microsoft rất tốt bụng khi cung cấp bài viết Knowledge Base article 298931 (http://support.microsoft.com/kb/298931) để cho người dùng biết cách vô hiệu hóa BHOs. Bài viết trong Knowledge Base cung cấp các liên kết đến thông tin chi tiết hơn về BHO, được liệt kê bên dưới khóa Registry key HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects. RegRipper’s bho.pl plugin trích xuất và hiển thị thông tin này cho bạn.*

*Lưu ý rằng khóa này nằm trong tệp HKEY\_LOCAL\_MACHINE, có nghĩa là trình duyệt tải BHOs ​​bất cứ khi nào nó được mở, bất kể người dùng.*

*Tập lệnh bho.pl Perl nằm trên phương tiện đi kèm (trong thư mục ch4\code\old) cho phép bạn xem các BHO đã được cài đặt trên hệ thống trực tiếp.*

Một trường phái có vẻ là hầu hết người dùng và quản trị viên đều không quen với các vị trí tự khởi động trong Registry và tin rằng chỉ có một vài khóa (đặc biệt là các khóa Run và RunOnce phổ biến) có thể được sử dụng cho các mục đích đó. Cách suy nghĩ này được hỗ trợ, một phần, bằng tài liệu và ứng dụng được cung cấp bởi nhà cung cấp hệ điều hành. Ví dụ:trực tiếp trên hệ thống Windows XP, một lệnh được gọi là MSConfig ra mắt tiện ích cấu hình hệ thống (<http://support.microsoft.com/kb/310560>) Bạn có thể chạy lệnh này bằng cách nhấp vào nút Bắt đầu trên Thanh tác vụ, chọn Chạy và nhập msconfig vào hộp văn bản, sau đó nhấn Enter. Hình 4.9 minh họa tiện ích Cấu hình hệ thống được mở để hiển thị nội dung của Startup tab.

Hình 4.9 Tab khởi động tiện ích cấu hình hệ thống



Chạy lệnh này trên hệ thống XP của bạn và xem xét kỹ hơn về cột Vị trí. Bạn sẽ thấy rằng các khóa Registry được kiểm tra là khóa Run từ cả hai thư mục HKEY\_CURRENT\_USER và HKEY\_LOCAL\_MACHINE.

Thật không may, đây dường như là chìa khóa duy nhất được kiểm tra. Có một lớp WMI có tên Win32\_StartupCommand sẽ cho phép lập trình viên tự động truy xuất nội dung của các khóa này, nhưng với tiện ích Cấu hình hệ thống, nó chỉ kiểm tra một số lượng rất hạn chế các vị trí khởi động trong Registry.

Một trường phái khác là số lượng vị trí tự khởi động trong Registry rất lớn nên việc kiểm tra các vị trí này là tốt nhất để các chuyên gia and/or phần mềm được tạo bởi các chuyên gia, như các tiện ích chống vi-rút và phần mềm chống phần mềm thương mại.

Sự thật là một nơi nào đó ở giữa. Có, có một số vị trí tự khởi động trong Registry, nhưng đối với hầu hết các phần, chúng là hữu hạn và hạn chế. Trong các phần sau, tôi sẽ chia các vị trí này thành ba khu vực và mô tả một số khóa Sổ đăng ký được truy cập khi hệ thống khởi động, những vị trí được truy cập khi người dùng đăng nhập và những vị trí được truy cập khi người dùng thực hiện một số hoạt động trên hệ thống . Sau đó, chúng tôi sẽ xem xét các cách để liệt kê các mục trong các vị trí này. Tuy nhiên, các khóa được liệt kê không nên được coi là bao gồm tất cả hoặc đại diện cho một danh sách đầy đủ và toàn diện của tất cả các khóa. Phương tiện đi kèm với cuốn sách này bao gồm một bảng tính có tên regref.xls có chứa một số bảng tính. Mỗi bảng tính bao gồm các Registry keys khác nhau thuộc danh mục bảng tính đó, cũng như mô tả ngắn gọn về khóa đó và, nếu có thể, một tài liệu tham khảo đáng tin cậy mô tả chức năng của khóa đó. Bảng tính này nên được coi là điểm khởi đầu để phân tích Registry, nhưng vì có rất nhiều ứng dụng ngoài kia và các phiên bản mới được sản xuất mọi lúc, nên nó không được coi là hoàn chỉnh.

### **System Boot**

Tự động khởi động các vị trí trong Registry được truy cập khi hệ thống khởi động là mục ưa thích của các tác giả phần mềm độc hại vì chúng cho phép phần mềm độc hại được khởi chạy mà không có sự tương tác của người dùng nào mà thậm chí không đăng nhập vào hệ thống. Một địa điểm là Windows Services, hoặc cụ thể hơn:

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services

Khi hệ thống khởi động, giá trị cho *ControlSet* hiện tại sẽ được sử dụng được xác định và cài đặt cho *ControlSet* đó được sử dụng. Các dịch vụ được liệt kê trong *ControlSet* được quét và các dịch vụ được đặt tự động bắt đầu (Giá trị bắt đầu là 0x02) được khởi chạy.

Khi thực hiện phân tích xâm nhập, thường thì tất cả những gì bạn phải làm là hình ảnh thu được của một hệ thống và báo cáo sự cố (và thường thì báo cáo sự cố không khác gì nhiều so với “chúng tôi thấy điều này xảy ra vào ngày này”). Khi bạn phải đối mặt với điều gì đó như thế này, một cách dễ dàng để bắt đầu là mở hình ảnh trong ProDiscover, điền vào Registry Viewer, xác định vị trí *Controlset* được đánh dấu hiện tại, sau đó sắp xếp các khóa con bên dưới Services key dựa trên thời gian *LastWrite* của chúng . Nhiều thời gian *LastWrite* này thường xếp hàng theo thời gian nhanh nhất mà hệ thống đã được khởi động, vì nhiều Services key đã được sửa đổi, thay vì đọc quá trình, trong quá trình khởi động. Tuy nhiên, trong một số trường hợp, tôi đã sử dụng kỹ thuật này để dung nạp các dịch vụ và trình điều khiển được cài đặt bởi kẻ tấn công, bao gồm rootkit và trình điều khiển có tên là ndriv.sys (là tệp trình điều khiển rootkit được sử dụng bởi một số Trojans và Internet Relay Chat, hoặc IRC, bots). Trong một số trường hợp, tôi đã định vị các dịch vụ này và cả tên của các dịch vụ cũng như thời gian *LastWrite* không liên quan đến báo cáo sự cố mà tôi đã nhận được. Về cơ bản, tôi đã tìm thấy một sự xâm nhập tách biệt với người mà tôi đang điều tra!

Bạn có thể tìm thấy các đồ tạo tác thú vị khác bằng cách sử dụng kỹ thuật tương tự này. Ví dụ: Tôi đã định vị npf.sys, trình điều khiển được cài đặt với các tiện ích WinPcap cho phép bạn thực hiện việc đánh hơi gói tin trên hệ thống của bạn. Trình điều khiển này được cài đặt bởi các công cụ như Wireshark và các công cụ sniffer có sẵn với các tiện ích WinPcap, nhưng chúng cũng có thể được cài đặt bởi kẻ tấn công.

RegRipper có thể sử dụng hai plugin, svc.pl và svc2.pl, để phân tích thông tin có sẵn trong Services key. Trong cả hai trường hợp, thông tin được trích xuất và sắp xếp dựa trên mỗi thời gian subkey’s LastWrite. svc.pl plugin trình bày một cái nhìn ngắn gọn, nhanh chóng về thông tin, trong khi svc2.pl trình bày thông tin chi tiết hơn (bao gồm tên hiển thị của dịch vụ hoặc ổ đĩa, đường dẫn, loại dịch vụ / trình điều khiển, v.v.) ở định dạng giá trị được phân tách bằng dấu phẩy (.csv). Khi chạy với rip.pl, đầu ra plugin svc2.pl có thể được chuyển hướng đến một tệp và được mở trong Excel để phân tích nhanh.

Trong một số trường hợp, người phản hồi sự cố có thể cần truy cập vào các hệ thống thông tin chính về dịch vụ. Chẳng hạn, có phần mềm độc hại sử dụng Windows Services làm sự kiên trì của nó, cơ chế và cài đặt như một DLL được chạy theo quy trình svchost.exe (Microsoft Knowledge Base article 314056, http://support.microsoft.com/kb/314056, cung cấp mô tả về quy trình svchost.exe trên Windows XP Professional). Vì phần mềm độc hại sử dụng cơ chế tồn tại này không dễ nhìn thấy (ví dụ, sâu Conficker viết một mục ngẫu nhiên, khiến nó khó tìm hơn), tôi đã viết một công cụ CLI để chạy trên các hệ thống trực tiếp, được gọi là regscan.pl (công cụ này đứng - độc lập và không phải là một phần của RegRipper, nó có sẵn trên phương tiện đi kèm với cuốn sách này trong thư mục ch4\code\Regscan) phân tích cú pháp thông qua khóa Dịch vụ và liệt kê thông tin cơ bản về từng dịch vụ, bao gồm cả thời gian *LastWrite* của khóa, giá trị *ImagePath* (nếu có) và giá trị *Parameters\ServiceDll* (nếu có). Thông tin này được phân định bằng đường ống và được gửi đến STDOUT, để dễ sử dụng. Người trả lời có thể chạy regscan.pl (hoặc chương trình thực thi được biên dịch) trên hệ thống cục bộ hoặc cung cấp địa chỉ IP hoặc tên hệ thống mà cô ấy có quyền truy cập Quản trị viên và trích xuất cùng một dữ liệu. Định vị phần mềm độc hại được cài đặt bằng phương pháp này cũng đơn giản như gõ lệnh sau:

C:\Perl\regscan>regscan.pl | find "svchost.exe -k netsvcs"

Lệnh này trích xuất tất cả thông tin Dịch vụ từ hệ thống cục bộ, nhưng thực hiện giảm dữ liệu bằng cách chuyển đầu ra thông qua lệnh find để chỉ hiển thị các mục được khởi chạy thông qua quy trình svchost.exe. Một đoạn trích của đầu ra xuất hiện như sau:

Wed Jan 14 01:16:37 2009Z|BITS|%SystemRoot%\system32\svchost.exe -k netsvcs|C:\WINDOWS\system32\qmgr.dll

Wed Jan 14 01:16:37 2009Z|Browser|%SystemRoot%\system32\svchost.exe -k netsvcs|%SystemRoot%\System32\browser.dll

Wed Jan 14 01:16:37 2009Z|CryptSvc|%SystemRoot%\system32\svchost.exe -k netsvcs|%SystemRoot%\System32\cryptsvc.dll

Wed Jan 14 01:16:37 2009Z|Dhcp|%SystemRoot%\system32\svchost.exe -k netsvcs|%SystemRoot%\System32\dhcpcsvc.dll

Các dấu (“|”) giữa giữa mỗi phân đoạn đầu ra giúp dễ dàng phân tích và phân tích, và cho phép đầu ra cho mỗi phím nằm trên một dòng để có thể sử dụng các công cụ khác như lệnh find để giảm lượng dữ liệu hiển thị mà người trả lời cần phải xử lý với. Dữ liệu được liệt kê theo từng service’s *LastWrite* time, do đó, nếu người phản hồi có cửa sổ thời gian gần đúng hoặc thời gian xảy ra sự cố, cô ấy có thể thu hẹp dữ liệu hơn nữa.

### **User Login**

Theo tài liệu của Microsoft, quy trình khởi động cho một hệ thống không được coi là hoàn tất cho đến khi người dùng đăng nhập. Khi người dùng đăng nhập vào hệ thống, một số khóa Registry nhất định được truy cập và phân tích cú pháp để các ứng dụng được liệt kê có thể chạy. Các phím đó là (theo thứ tự):

1. HKLM\Software\Microsoft\Windows\CurrentVersion\Runonce

2. HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

3. HKLM\Software\Microsoft\Windows\CurrentVersion\Run

4. HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run

5. HKCU\Software\Microsoft\Windows\CurrentVersion\Run

6. HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce

Vì lợi ích ngắn ngọn, *HKLM* đề cập đến thư mục HKEY\_LOCAL\_MACHINE và HKCU đề cập đến thư mục HKEY\_CURRENT\_USER.

Mỗi khi người dùng mới đăng nhập vào hệ thống, các phím 1, 3, 5 và 6 được phân tích cú pháp và các chương trình được liệt kê sẽ được chạy (http://support.microsoft.com/kb/137367). Theo mặc định, các phím Run này bị bỏ qua nếu hệ thống được khởi động ở Chế độ an toàn. Tuy nhiên, trên các hệ thống Windows XP và 2003, nếu bạn mở đầu các giá trị RunOnce (phím 1 và 6) bằng dấu hoa thị (\*), bạn có thể buộc chương trình được liên kết chạy ngay cả khi hệ thống được khởi động ở Chế độ an toàn (http: / / support.microsoft.com/kb/314866). Hơn nữa, trên Windows XP, các phím 1, 3, 5 và 6 được cung cấp cho các chương trình cũ và khả năng tương thích ngược để các chương trình được viết cho các phiên bản Windows trước đó (hoặc trước khi Windows XP được phát hành) vẫn có thể được sử dụng.

### **User Activity**

Tự động khởi động các vị trí đăng ký thuộc danh mục này là những vị trí được truy cập khi người dùng thực hiện một hành động, chẳng hạn như mở một ứng dụng như Internet Explorer hoặc Outlook. Nếu bạn chạy RegMon trên một hệ thống và chỉ cần di chuyển chuột hoặc mở một ứng dụng (hoặc không làm gì cả), bạn sẽ thấy rằng có khá nhiều quyền truy cập vào Registry, ngay cả khi rõ ràng không có gì liên quan đến người dùng tương tác với hệ thống. Cũng như các vị trí tự khởi động khác, các tác giả phần mềm độc hại (vi rút, Trojan, worm, v.v.) thấy các khóa Registry này cực kỳ hữu ích trong việc duy trì sự bền bỉ của các sản phẩm của họ, đảm bảo rằng chúng có thể chạy và chạy.

Một điểm đáng chú ý như vậy là:

HKEY\_LOCAL\_MACHINE\Software\Classes\Exefile\Shell\Open\command

Registry cũng như các khóa cho các lớp tệp khác (batfile, comfile, v.v.) điều khiển xảy ra khi lớp tệp đó được mở. Ví dụ: trong Windows Explorer, nhấp chuột phải vào bất kỳ tệp nào và menu ngữ cảnh sẽ xuất hiện với từ Mở ở đầu trình đơn, in đậm. Hành động in đậm là, trong hầu hết các trường hợp, điều gì xảy ra khi tệp đó được nhấp đúp. Khi bạn bấm đúp vào một tệp, Windows sẽ quét Sổ đăng ký cho lớp tệp đó và sau đó xác định những hành động cần thực hiện, dựa trên cài đặt Sổ đăng ký cho lớp tệp. Phần mềm độc hại như SirCam (http://support.microsoft.com/kb/311446) và Pretty Park (http://support.microsoft.com/kb/310585) sâu đã sử dụng vị trí Đăng ký này để duy trì sự bền bỉ trên hệ thống bị nhiễm.

Ví dụ, hãy để nói rằng bạn muốn chơi Solitaire trên hệ thống của bạn. Bạn vào dấu nhắc lệnh và gõ lệnh:

C:\>dir /s sol.exe

Đầu ra của lệnh này cho bạn biết vị trí thực thi của Solitaire nằm trong hệ thống tệp. (Trên hệ thống Windows XP Home của tôi, sol.exe nằm trong thư mục C: \ Windows \ system32.) Vì tò mò, bạn tự hỏi điều gì xảy ra khi bạn bấm đúp vào biểu tượng tệp cho Solitaire, vì vậy bạn nhập lệnh sau:

C:\>ftype exefile

Đầu ra của lệnh cho bạn thấy *exefile=“%1” %\*.* Điều này về cơ bản cho hệ thống khởi chạy tệp với đối số đầu tiên (tên tệp) và bất kỳ đối số kế tiếp. Tuy nhiên, bổ sung có thể được thực hiện để *shell\open\command* Registry mục nhập để các tệp khác được khởi chạy bất cứ khi nào một lớp tệp cụ thể được mở. Các mục nhập trong khóa này (và các mục khác tương tự như khóa này, như được mô tả trong một khoảnh khắc) chỉ nên chứa *‘%1’ %\**. và không có gì khác, theo mặc định. Bất kỳ dữ liệu nào khác trong giá trị này nên được xem là đáng ngờ và được điều tra ngay lập tức.

Một mục khác để kiểm tra thông tin tương tự là:

HKEY\_CLASSES\_ROOT\Exefile\Shell\Open\Command

Chức năng này không áp dụng cho chỉ mục Exefile bên dưới HKEY\_CLASSES\_ROOT. Một số phần mềm độc hại sẽ sửa đổi các mục khác cùng loại để đảm bảo tính bền bỉ của nó trên hệ thống. Ví dụ: một số backtime sửa đổi các mục thành khóa sau:

HKEY\_CLASSES\_ROOT\Word.Document.x\shell\open\command

Trong trường hợp này, x là số phiên bản (8, 9, v.v.) cho Word. Điều này cho hệ thống biết rằng bất cứ khi nào lệnh Open command for Microsoft Word được chạy qua trình bao (Windows Explorer), chẳng hạn như khi người dùng nhấp đúp vào tài liệu, phần mềm độc hại sẽ được thực thi.

Một vị trí khác có thể được sử dụng theo cách tương tự là khóa sau:

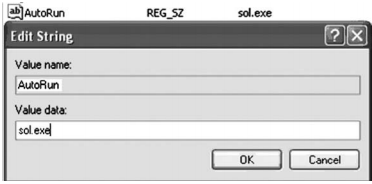
HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\AutoRun

Bạn có thể tìm thấy một tài liệu tham khảo tuyệt vời cho khóa AutoRun tại http://technet.microsoft. com/en-us/library/bb490880.aspx.

Giá trị Registry này liệt kê các lệnh được chạy bất cứ khi nào bộ xử lý lệnh (cmd.exe) được chạy. Ví dụ, nó sẽ chạy một ứng dụng bất cứ khi nào một dấu nhắc lệnh được mở. Giá trị trống theo mặc định. Các mục cũng có thể được thực hiện trong cùng một giá trị trong thư mục HKEY\_CURRENT\_USER và nếu có một mục ở đó, thì nó được ưu tiên hơn mục trong thư mục HKEY\_LOCAL\_MACHINE.

Để minh họa cách đơn giản này, hãy mở RegEdit và điều hướng đến phím Bộ xử lý lệnh. Trên hệ thống Windows XP Pro SP2 của tôi, giá trị đó hiển thị nhưng không có dữ liệu liên quan đến nó. Bấm chuột phải vào giá trị và chọn **Modify**, rồi nhấn **Enter**. Hình 4.10 minh họa một ví dụ về dữ liệu có thể được thêm vào giá trị Registry.

Hình 4.10 Thêm dữ liệu vào Command Processor\AutoRun Value



Khi bạn đã thay đổi giá trị, bấm **OK**. Sau đó bấm **Start |Run**, nhập **cmd** và nhấn **Enter**. Dấu nhắc lệnh sẽ mở, cũng như ứng dụng mà bạn đã chọn. Trong Hình 4.10, tôi đã chọn một ứng dụng có giao diện GUI đẹp để khi tôi chạy dấu nhắc lệnh, rõ ràng có một thứ khác cũng được mở. Một số hành động khác có thể được thực hiện, chẳng hạn như cài đặt âm thầm người dùng hoặc dịch vụ hoặc bắt đầu một cửa hậu Trojan không có GUI có GUI.

Nói về GUI, có một khóa Registry ít được biết đến có thể được sử dụng để tải DLL vào bộ nhớ mỗi khi khởi động ứng dụng GUI. Chìa khóa này là:

HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit\_DLLs

Bài viết Microsoft Knowledge Base 197571 (http://support.microsoft.com/kb/197571) chỉ ra rằng các DLL được liệt kê trong giá trị này được tải vào bộ nhớ bằng cách sử dụng *LoadLLibrary ()* chức năng trong quá trình *DLL\_PROCESS\_ATTACH* của user32.dll; đây là nơi hướng dẫn cho các thành phần GUI khác nhau, như cửa sổ và hộp thoại, được lưu trữ.

Vào năm 2000, J. D. Glaser (trước đây là Foundstone, hiện đang điều hành NT OBJECTives) đã thuyết trình tại các hội nghị BlackHat và USENIX về việc theo dõi sự thỏa hiệp của máy chủ và tìm một mục trong khóa AppInit\_DLLs.

Khóa AppInit\_DLLs cực kỳ hiệu quả làm nơi ẩn náu cho phần mềm độc hại. Phần mềm gián điệp CoolWebSearch được biết là sử dụng khóa này, ví dụ. Tại sao chìa khóa này rất hiệu quả? Khi tôi đã dạy huấn luyện ứng phó sự cố thực hành, một trong những bài tập đầu tiên tôi sẽ chạy là một bài tập đơn giản về “lây nhiễm” có nghĩa là nhìn vào quá trình người tham dự, thay vì xác định ai có thể tìm thấy “lây nhiễm” đầu tiên. Tôi đã dạy khóa học cho các quản trị viên Windows mới và có kinh nghiệm cũng như các quản trị viên UNIX và Linux có kinh nghiệm, những người cũng có trách nhiệm đối với các hệ thống Windows. Lúc nào cũng vậy, trên toàn hội đồng, 100% thời gian, bước đầu tiên mà mọi người tham dự thực hiện là mở một công cụ GUI trên máy tính để bàn. Nó có thể là Trình xem sự kiện; nó có thể là Trình quản lý tác vụ; hoặc thậm chí Windows Explorer. Tuy nhiên, đầu tiên của họ bản năng là luôn luôn với tới một ứng dụng GUI.

Hệ điều hành Windows cung cấp khả năng cảnh báo các chức năng bên ngoài khi một số sự kiện nhất định xảy ra trên hệ thống, chẳng hạn như khi người dùng đăng nhập hoặc tắt hoặc khi trình bảo vệ màn hình khởi động. Các thông báo này được xử lý bởi khóa Registry sau:

HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\

Các mục bên dưới khóa này trỏ đến DLL nhận thông báo về các sự kiện nhất định. Googling cho *Winlogon \ Notify* sẽ cung cấp cho bạn một danh sách dài các liên kết đến phần mềm độc hại sử dụng chức năng khóa chính này. Khi bạn thực hiện phân tích pháp y về một hệ thống, đó sẽ là một ý tưởng tốt để sắp xếp các khóa con bên dưới *Notify* dựa trên thời gian *LastWrite* của chúng và đặc biệt chú ý đến bất kỳ mục nào gần ngày xảy ra sự cố đáng ngờ, cũng như bất kỳ mục nào liệt kê DLL trong giá trị DLL*Name* có thông tin phiên bản tệp đáng ngờ hoặc không có phiên bản tệp đáng ngờ thông tin nào cả. (Chapter 6 bao gồm chủ đề lấy thông tin phiên bản tệp từ tệp thực thi.)

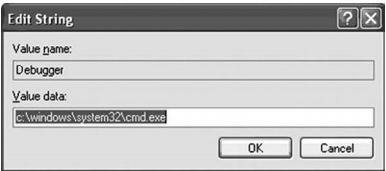
Bên dưới khóa WinLogon (được liệt kê trước đây) là một giá trị có tên là *TaskMan* có thể được các nhà điều tra quan tâm vì nó cho phép người dùng chọn một ứng dụng để thay thế Task Manager. Giá trị này không tồn tại theo mặc định nhưng có thể được thêm vào. Trong thực tế, việc cài đặt Process Explorer từ Sysiternals cho phép bạn chọn Process Explorer để thay thế Task Manager thông thường. Nếu giá trị của *TaskMan* tồn tại bên dưới khóa WinLogon, bạn nên xem xét điều đáng nghi ngờ này trong hầu hết các trường hợp thông thường và điều tra kỹ lưỡng ứng dụng được liệt kê trong dữ liệu.

Có một khóa Registry thú vị cho phép người dùng (thường là nhà phát triển ứng dụng hoặc ai đó gỡ lỗi ứng dụng) để chỉ định trình gỡ lỗi sẽ được khởi chạy khi chạy ứng dụng. Khóa Registry là:

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options

Microsoft cung cấp một số bài viết trong Cơ sở Kiến thức thảo luận về việc sử dụng khóa này để gỡ lỗi các ứng dụng Common Gateway Interface (CGI) (<http://support.microsoft.com/kb/238788>) cũng như tắt tính năng Windows Update trong Windows XP (<http://support.microsoft.com/kb/892894>) Ví dụ, tạo một khóa con cho ứng dụng bạn muốn chặn và thêm giá trị *Debugger* bằng *ntsd -* (*ntsd* theo sau là khoảng trắng và hai dấu gạch ngang) sẽ khiến trình gỡ lỗi gắn vào quy trình và sau đó thoát ngay lập tức. Tuy nhiên, Dana Epp đã xác định (<http://silverstr.ufies.org/blog/archives/000809.html>) một phương pháp sử dụng khóa này như một vectơ tấn công, một hoặc một cách thích hợp hơn, như một phương pháp kiên trì cho phần mềm độc hại. Để thấy điều này trong thực tế, trước tiên hãy thêm một khóa con vào phím “Tùy chọn thực thi tệp hình ảnh” là tên của tệp thực thi mà bạn muốn phá vỡ (ví dụ: notepad.exe). Bạn không cần cung cấp một đường dẫn, chỉ cần tên của tệp thực thi. Sau đó thêm vào khóa một giá trị chuỗi được gọi là *Debugger* và trỏ nó đến dấu nhắc lệnh, như Hình 4.11 minh họa.

Hình 4.11 Thêm giá trị trình gỡ lỗi vào khóa tùy chọn thực thi tệp hình ảnh



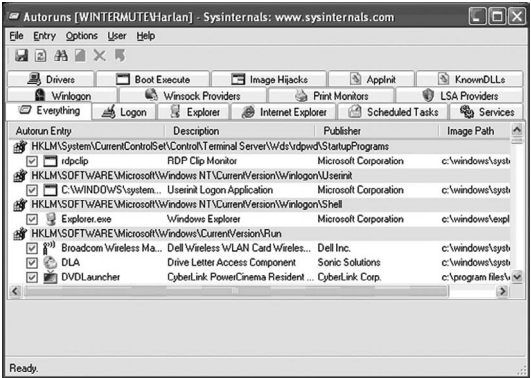
Nhấn **OK** và sau đó chọn **Start | Run** và gõ **notepad**. Bạn sẽ thấy lệnh được mở thay vì Notepad. Không cần khởi động lại để thay đổi có hiệu lực.

Tại thời điểm này, tôi có thể nêu bật một số cách thú vị để tận dụng lợi thế của loại công cụ này, chẳng hạn như trỏ giá trị của *Debugger* vào bản sao notepad.exe của Trojan không chỉ mở Notepad mà còn khởi chạy backdoor hoặc IRCbot hoặc sâu của một số loại. Tuy nhiên, hiện có đủ các ví dụ về phần mềm độc hại đã thiết lập chỗ đứng trong khóa Sổ đăng ký này để làm rõ rằng đây chắc chắn là một khóa đáng để kiểm tra. Trên thực tế, tôi đã thực sự thấy phương pháp kiên trì này được sử dụng bởi phần mềm độc hại trong một số lần kiểm tra, bao gồm cả các trường hợp xâm nhập. Chỉ cần điều hướng qua tất cả các khóa con và kiểm tra tệp thực thi được chỉ ra bởi giá trị *Debugger* (nếu có, không phải tất cả các khóa con sẽ có giá trị *Debugger*) hoặc sử dụng imagefile.pl plugin để tự động tìm kiếm trong thư mục Software.

### **Enumerating Autostart Registry Locations**

Một trong những công cụ tốt nhất hiện có để lấy thông tin từ một số lượng lớn các địa điểm tự động trên hệ thống trực tiếp là Autorun, từ Microsoft (Version 9.39 có sẵn tại thời điểm viết bài này từ http://technet.microsoft.com/en-us/sysiternals/bb963902.aspx). Đây là một công cụ cập nhật có cả phiên bản GUI và CLI. Hình 4.12 cho thấy phiên bản GUI của Autorun.

Hình 4.12 Autoruns(GUI)



Như được hiển thị trong Hình 4.12, Autoruns sẽ lấy các mục từ một số khóa Registry và hiển thị những gì nó tìm thấy. Autoruns cũng sẽ truy xuất mô tả và nhà xuất bản từ tệp thực thi được trỏ đến theo từng giá trị Sổ đăng ký và được liệt kê trong cột Image Path (chỉ hiển thị một phần). Thông tin này cung cấp một chỉ báo nhanh cho điều tra viên về việc liệu có bất kỳ điều gì có thể đáng ngờ đang chạy ở một trong những địa điểm này hay không và cần được điều tra.

Các công cụ khác tồn tại để liệt kê nội dung của các vị trí đăng ký tự động. Một trong số đó là kịch bản Visual Basic có tên là Silent Runners, bạn có thể tìm thấy tại www.silentrunners.org. Trang web bao gồm một danh sách đầy đủ các điểm khởi động được liệt kê bởi tập lệnh, lần đầu tiên được cung cấp cho người dùng thông qua NTBugTraq (www.ntormstraq.com) vào ngày 12 tháng 5 năm 2004. Tập lệnh này dự định chạy trên hầu hết các phiên bản Windows, bao gồm cả Windows 98, nằm ngoài phạm vi của cuốn sách này. Hãy ghi nhớ điều này khi chạy tập lệnh, bởi vì một số vị trí được liệt kê (trong Registry và trong hệ thống tệp) chỉ áp dụng cho các phiên bản Windows đó. Những địa điểm này được chỉ ra trong danh sách các điểm phóng. Đối với các cuộc điều tra sau khi chết, các nhà phân tích yêu cầu các công cụ cho phép không chỉ xem mà còn liệt kê một Sổ đăng ký đã được xây dựng lại từ các tệp thành phần với hình ảnh hệ thống. Khả năng ProScript của ProDiscover cho phép nhà phân tích sử dụng các tập lệnh Perl tương tự như các tập lệnh được viết cho các hệ thống trực tiếp để tìm kiếm Registry trên một hình ảnh trong quá trình phân tích hậu kỳ. Các công cụ như Autorun được cập nhật và cung cấp danh sách toàn diện nhất các khóa Registry có một số loại chức năng tự khởi động.

### **Autoruns Functionality**

Ngoài các vị trí tự khởi động trong Registry, các hệ thống Windows cũng cung cấp một cái gì đó được gọi là khả năng của “Auto Run”hoặc“AutoPlay”. Bạn có thể quen thuộc nhất với điều này khi bạn đặt CD nhạc vào ổ đĩa CD-ROM và khi bạn nghe thấy ổ đĩa quay lên, ứng dụng thích hợp (như Windows Media Player hoặc RealPlayer) sẽ tự động khởi động và bản nhạc đầu tiên của CD sẽ phát. Một ví dụ khác về khả năng AutoRun là khi bạn đặt CD cài đặt phần mềm vào ổ đĩa CD-ROM và quy trình cài đặt sẽ tự động bắt đầu.

Cách thức hoạt động của khả năng này là trên các loại ổ đĩa được chỉ định, Windows sẽ tìm tệp autorun.inf trong thư mục gốc của ổ đĩa và nếu tìm thấy một ổ đĩa, nó sẽ phân tích và thực thi các dòng load = hoặc run = trong tệp. Microsoft từ lâu đã tuyên bố rằng hành vi này được kiểm soát hoặc quy định bởi Registry key sau:

HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\

Trong khóa này, giá trị *NoDriveTypeAutoRun* cho phép bạn tắt chức năng AutoPlay trên các loại ổ đĩa cụ thể. Giá trị là một giá trị bitmapped, nghĩa là mỗi bit trong một byte tương ứng với một loại ổ đĩa cụ thể và một “1” ở bất kỳ vị trí nào đều vô hiệu hóa chức năng AutoPlay trên loại ổ đĩa tương ứng. Do đó, giá trị 0x95 (ký hiệu thập phân 149) sẽ vô hiệu hóa chức năng AutoPlay trên các ổ đĩa di động và mạng và giá trị 0xFF sẽ vô hiệu hóa chức năng trên tất cả các loại ổ đĩa.

**Warring:** Bạn có thể tìm thấy giá trị NoDriveTypeAutoRun trong thư mục HKLM hoặc HKCU. Theo Microsoft, nếu giá trị tồn tại trong thư mục HKLM, giá trị trong thư mục HKCU sẽ bị bỏ qua.

Vào ngày 11 tháng 9 năm 2008, Microsoft lần đầu tiên xuất bản bài viết Knowledge Base 953252 (<http://support.microsoft.com/kb/953252>) , trong đó tuyên bố rằng đối với Windows 2000, XP, 2003 và Vista (Windows 2008 không bị ảnh hưởng), để khả năng *NoDriveTypeAutoRun* thực sự để người dùng có hiệu quả phải cài đặt một bản vá bổ sung sẽ thêm giá trị *HonorAutorunSetting* vào\ Chính sách\ Explorer key trong tổ hợp HKLM, đặt nó thành 0x1. Bài viết trong Cơ sở Kiến thức chứa một cách xử lý hoàn chỉnh cả hai giá trị *NoDriveTypeAutoRun* và *HonorAutorunSetting*. Đây là một vấn đề quan trọng trong khoảng thời gian bài báo của Knowledge Base được xuất bản, vì các biến thể của phần mềm độc hại Sality (và sau này là Conficker) sẽ lan truyền, một phần, bằng cách viết phần mềm độc hại và tệp autorun.inf đề cập đến phần mềm độc hại vào thư mục gốc của mọi ổ đĩa được tìm thấy trên một hệ thống bị nhiễm bệnh. Trên thực tế, một số biến thể của Sality đã được tìm thấy để lây nhiễm các thiết bị lưu trữ di động được kết nối với hệ thống rất lâu sau khi nó bị nhiễm.

**Tools and Traps**

**CD-ROMs**

Ngoài giá trị *NoDriveTypeAutoRun*, bạn có thể bật khả năng AutoPlay cho các ổ đĩa CD-ROM bằng cách đặt giá trị *Autorun* trong khóa HKLM\System\CurrentControlSet\ Services\CDRom thành 1 (bạn có thể tắt nó bằng 0). Theo bài viết 319287 của Microsoft Knowledge Base (<http://support.microsoft.com/kb/319287>) , chức năng thực sự của giá trị *Autorun* là bật hoặc tắt thông báo thay đổi phương tiện. Nếu giá trị *NoDriveTypeAutoRun* được đặt thành 0xb5, khả năng AutoPlay cũng sẽ bị tắt cho CD-ROM. Để biết chi tiết về các cài đặt này, cũng như thông tin bổ sung, hãy xem bài viết 330135 của Microsoft Knowledge Base (<http://support.microsoft.com/kb/330135>) .

### **NtfsDisableLastAccessUpdate**

Hầu hết các nhà phân tích đều biết rằng các hệ thống Windows (cũng như nhiều hệ điều hành) sẽ duy trì thời gian trên các tệp được gọi là “MAC time” (Tôi sẽ đề cập đến chủ đề này chi tiết hơn trong Chương 5). Điều mà nhiều người có thể không nhận ra là các hệ thống Windows có thể được cấu hình, thông qua một giá trị Registry duy nhất, để không cập nhật dữ liệu đóng dấu thời gian của tệp khi tệp được truy cập. Bạn có thể sử dụng giá trị *NtfsDisableLastAccessUpdate* Registry để vô hiệu hóa chức năng này, như một phương tiện để tăng hiệu suất của các máy chủ tệp có khối lượng lớn, như được mô tả trong bài viết 894372 của Cơ sở tri thức Microsoft (http://support.microsoft.com/kb/894372). Tuy nhiên, giá trị này được bật theo mặc định trên các hệ thống Windows Vista và Windows 7, như được minh họa trong đầu ra của disablelastaccess.pl RegRipper plugin được hiển thị ở đây:

C:\Perl\forensics\rr>rip.pl-r d:\cases\vista\system -p disablelastaccess

Launching disablelastaccess v.20090118

DisableLastAccess

ControlSet001\Control\FileSystem

NtfsDisableLastAccessUpdate = 1

C:\Perl\forensics\rr>rip.pl -r d:\cases\win7\system -p disablelastaccess

Launching disablelastaccess v.20090118

NtfsDisableLastAccessUpdate

ControlSet001\Control\FileSystem

NtfsDisableLastAccessUpdate = 1

Đây có thể là một yếu tố quan trọng cần tính đến, đặc biệt là khi thực hiện phân tích hệ thống Windows Vista và Windows 7, vì việc thiếu tệp tem thời gian truy cập lần cuối có thể là một trở ngại đáng kể trong quá trình kiểm tra.

Giá trị này không được bật theo mặc định trên các hệ thống Windows XP và 2003, do đó, sự tồn tại của nó, ngoài giá trị 1, có thể rất đáng nói trong quá trình kiểm tra. Tuy nhiên, ngay cả với giá trị được đặt này, như bạn sẽ thấy sau trong chương này, có một số vị trí trong Registry nơi bạn có thể tìm thấy các dấu hiệu truy cập vào tệp cũng như loại quyền truy cập.

### **NukeOnDelete**

Khi người dùng xóa các tệp trên hệ thống Windows XP hoặc 2003, trong hầu hết các trường hợp, hành vi mặc định là trình bao Explorer sẽ gửi tệp đó đến Thùng rác. Người dùng có thể bỏ qua Thùng rác bằng cách giữ phím Shift khi xóa tệp hoặc bằng cách thêm giá trị NukeOnDelete có thể tìm thấy trong khóa Registry sau trong thư mục Software:

Microsoft\Windows\CurrentVersion\Explorer\BitBucket

Thêm giá trị *NukeOnDelete* và đặt thành 1 sẽ vô hiệu hóa Recycle Bin một cách hiệu quả, vì các tệp bị xóa sẽ bỏ qua Recycle Bin. Do đó, nếu trong quá trình kiểm tra bạn thấy rằng không có tệp nào trong Recycle Bin, hãy chắc chắn kiểm tra giá trị Registry này để xem có lẽ nó đã được thêm và đặt thành 1 hay không.

Như tôi đã đề cập trước đây, mặc dù cấu trúc nhị phân của Registry vẫn giữ nguyên từ Windows NT cho đến Windows 7 beta, các khóa và giá trị đã được tìm thấy để thay đổi giữa các phiên bản của hệ điều hành. Ví dụ: vị trí của giá trị *NukeOnDelete* vừa thảo luận áp dụng cho Windows XP và 2003 và đã thay đổi với sự ra đời của Windows Vista. Trên các hệ thống Windows XP và 2003, khóa BitBucket nằm trong thư mục Software, do đó cài đặt được áp dụng cho tất cả người dùng trên hệ thống. Trên Vista và Windows 7, khóa BitBucket đã di chuyển vào thư mục của người dùng (NTUSER.DAT) theo đường dẫn sau:

Software\Microsoft\Windows\CurrentVersion\Explorer\BitBucket\Volume\{GUID}

Trong khi Windows XP và 2003 duy trì giá trị *NukeOnDelete* trên toàn hệ thống, Vista và Windows 7 cung cấp chức năng trên cơ sở cho mỗi người dùng, trên mỗi khối lượng.

## **USB Removable Storage Devices**

Một trong những chủ đề phổ biến hơn khi tôi thuyết trình tại các hội nghị là cách theo dõi các thiết bị lưu trữ di động USB trên các hệ thống Windows. Khi tôi lần đầu tiên trình bày về chủ đề này, có rất nhiều người “Hey, tôi đã không biết rằng bạn có thể làm được điều đó”. Kể từ đó, tôi đã trả lời hoặc chứng kiến một loạt các câu hỏi liên quan đến chủ đề này.

Khi thiết bị lưu trữ di động USB, ổ USB được kết nối với hệ thống Windows, dấu vết hoặc hiện vật được để lại trong Registry. (Các tạo phẩm cũng được để lại trong tệp setupapi.log.) Khi thiết bị được cắm, Plug and Play (PnP)Manager nhận sự kiện và truy vấn bộ mô tả thiết bị trong firmware (thông tin này không nằm trong vùng nhớ của thiết bị) để biết thông tin về thiết bị, chẳng hạn như nhà sản xuất. PnP Manager sau đó sử dụng thông tin này để định vị trình điều khiển thích hợp cho thiết bị (dựa trên nội dung của tệp .inf) và nếu cần tải trình điều khiển đó.

Thông tin này được ghi lại trong tệp setupapi.log.) Khi thiết bị đã được xác định, Registry key sẽ được tạo bên dưới khóa sau:

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Enum\USBSTOR

Bên dưới khóa này, bạn sẽ thấy các khóa con trông như thế này:

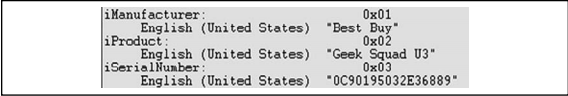
Disk&Ven\_###&Prod\_###&Rev\_###

Khóa con này đại diện cho định danh lớp thiết bị, bởi vì nó xác định một lớp thiết bị cụ thể. Các trường được đại diện bởi ### được điền bởi PnP Manager dựa trên thông tin được tìm thấy trong bộ mô tả thiết bị. Ví dụ: tôi có ổ Geek Squad 1GB mà tôi đã mua từ Best Buy; ID lớp cho thiết bị trông như thế này:

Disk&Ven\_Best\_Buy&Prod\_Geek\_Squad\_U3&Rev\_6.15

Bạn có thể sử dụng UVCView để xem nội dung của bộ mô tả thiết bị. Hình 4.13 minh họa một phần của bộ mô tả thiết bị cho ổ Geek Squad đã đề cập trước đó.

Hình 4.13 Phần mô tả thiết bị qua UVCView



Như bạn có thể thấy trong Hình 4.13, iManufacturer và iProduct sinh sản từ bộ mô tả thiết bị được ánh xạ tới ID lớp thiết bị.

**NOTE:** Microsoft đã từng cung cấp tiện ích UVCView dưới dạng tải xuống miễn phí, nhưng sau đó đã chuyển nó sang Windows Driver Kit (WDK), theo <http://msdn.microsoft.com/en-us/library/aa469207.aspx.UVCView> là một ứng dụng cho phép nhà phân tích xem thông tin mô tả thiết bị từ thiết bị USB; thông tin này không phải là một phần của vùng nhớ của ổ USB, do đó, và do đó không được lấy khi hình ảnh được lấy từ một thiết bị như vậy.

Khi ID lớp thiết bị đã được tạo, một mã định danh cá thể duy nhất cần được tạo cho thiết bị cụ thể. Lưu ý rằng trong Hình 4.13 có một giá trị gọi là *iSerialNumber*. Đây là một định danh cá thể duy nhất cho thiết bị, tương tự như địa chỉ MAC của một NIC. Giá trị này được sử dụng làm ID cá thể duy nhất cho thiết bị sao cho nhiều thiết bị cùng loại( Ví dụ : 2 ổ 1GB Geek Squad) có thể được xác định duy nhất trên

hệ thống. Từ Câu hỏi thường gặp về USB: Trung cấp ([www.microsoft.com/whdc/connect/usb/usbfaq\_intermed.mspx](http://www.microsoft.com/whdc/connect/usb/usbfaq_intermed.mspx)):

“Nếu thiết bị có số sê-ri, Microsoft yêu cầu số sê-ri này xác định duy nhất từng phiên bản của cùng một thiết bị. Ví dụ: nếu hai bộ mô tả thiết bị có các giá trị giống hệt nhau cho các trường idVendor, idProductvà bcdDevice, thì iSerialNumber sẽ phân biệt cái này với cái kia.”

Hình 4.14 minh họa ID lớp thiết bị và ID cá thể duy nhất cấp dưới như xuất hiện trong RegEdit.

Hình 4.14 Phần RegEdit Hiển thị ID lớp thiết bị và ID trường hợp duy nhất



Mặc dù số sê-ri duy nhất là bắt buộc đối với các thiết bị mà nhà sản xuất muốn đủ điều kiện cho chương trình Logo Windows ([www.microsoft.com/whdc/archive/usbfaq.mspx#ERCAC](http://www.microsoft.com/whdc/archive/usbfaq.mspx#ERCAC)) , không phải tất cả các thiết bị đều có số sê-ri. Đối với các thiết bị không có số sê-ri, PnP Manager sẽ tạo ID cá thể duy nhất cho thiết bị đó, trông sẽ giống như sau:

6&26c97b61&0

Lưu ý rằng ký tự thứ hai là ký hiệu (&). Nếu bạn thấy một ID cá thể duy nhất bên dưới khóa USBSTOR trông như thế này, bạn biết thiết bị được cắm vào hệ thống không có số sê-ri trong bộ mô tả thiết bị.

Vì vậy, nếu ID cá thể duy nhất không có & là ký tự thứ hai, bạn có thể xác định thiết bị duy nhất được kết nối với hệ thống. Trong các trường hợp liên quan đến nhiều hệ thống và phương tiện lưu trữ, các nhà điều tra nên chắc chắn bao gồm việc sử dụng UVCView trong phương pháp của họ để các thiết bị sau đó có thể được gắn vào hệ thống, không chỉ thông qua các tạo phẩm của Registry trong khóa USBSTOR mà còn trong khóa MountedDevices như cũng như trong Phím tắt và các tham chiếu khác đến các tệp nằm trên hệ thống.

Khi khóa định danh cá thể duy nhất đã được tạo, khóa đó sẽ được điền với một số giá trị, bao gồm cả *FriendlyName*. Giá trị quan tâm đối với các nhà điều tra sẽ là giá trị *ParentIdPrefix*. Microsoft không cung cấp bất kỳ thông tin nào liên quan đến cách tạo ra giá trị này hoặc liệu giá trị này là duy nhất trên các hệ thống Windows. Tuy nhiên, giá trị *ParentIdPrefix* có thể được sử dụng để tương quan thông tin bổ sung từ bên trong Registry.

Ví dụ: sử dụng cả mã định danh cá thể duy nhất và *ParentIdPrefix*, bạn có thể xác định lần cuối cùng thiết bị USB được kết nối với hệ thống Windows. Trên hệ thống trực tiếp, bạn cần điều hướng đến khóa sau:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\DeviceClasses

Bạn sẽ thấy một số khóa con bên dưới khóa này, tất cả đều là GUIDs. Các lớp thiết bị cụ thể mà chúng tôi quan tâm là {53f56307-b6bf-11d0-94f2-00a0c91efb8b} và {53f5630d-b6bf-11d0-94f2-00a0c91efb8b}. Hai lớp này được định nghĩa trong ntddstor.h tệp tiêu đề (bạn có thể xem ví dụ về tệp tiêu đề tại www.reactos.org/generated/doxygen/ d1/d09/ntddstor\_8h.html) tương ứng là GUIDs cho giao diện thiết bị ổ đĩa và âm lượng. Điều hướng đến thiết bị đĩa GUID, chúng tôi thấy một số khóa con với một số tên thực sự dài; tham khảo lại thiết bị được minh họa trong hình 4.13, chúng ta thấy một khóa con có tên sau:

USBSTOR#Disk&Ven\_Best\_Buy&Prod\_Geek\_Squad\_U3&Rev\_6.15#0C90195032E36889&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

Với mục đích của ví dụ này, tôi nhấn mạnh đã đánh dấu định danh cá thể duy nhất (trong trường hợp này là số sê-ri của thiết bị) để minh họa vị trí trong tên khóa của ID. *LastWrite* time của khóa này tương ứng với lần cuối cùng thiết bị đĩa được kết nối với hệ thống. Điều này là do khi một thiết bị được kết nối với hệ thống Windows, một khóa con có tên là “Control” được tạo ra; khi thiết bị bị ngắt kết nối với hệ thống, khóa con Control sẽ bị xóa. Cả hai hành động này đều gây ra *LastWrite* time của khóa DeviceClass cho thiết bị cụ thể được sửa đổi.

Chúng tôi cũng có thể tiến hành tương quan tương tự với giao diện thiết bị âm lượng GUID, sử dụng *ParentIdPrefix* cho thiết bị, như sau:

##?#STORAGE#RemovableMedia#7&326659cd&0&RM#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}

Một lần nữa, tôi đã nhấn mạnh *ParentIdPrefix* trong khóa con thiết bị để minh họa nơi có thể tìm thấy nó. *LastWrite* time của phím này tương ứng với lần cuối cùng âm lượng được kết nối với hệ thống. Như với thiết bị đĩa GUID, một khóa con Control được tạo bên dưới GUID của thiết bị âm lượng và sau đó bị xóa khi âm lượng (i.e., thiết bị) bị ngắt kết nối, khiến cho khóa LastWrite time được cập nhật tương ứng.

Chúng tôi sẽ đề cập thêm thông tin liên quan đến việc sử dụng *ParentIdPrefix* để tương quan thông tin từ Registry trong phần “Mounted Devices” ở phần sau của chương này.

RegRipper bao gồm một số plugins để trích xuất dữ liệu lưu trữ di động USB từ tệp trung tâm hệ thống. Đầu tiên là usbstor.pl plugin phân tích nội dung của các khóa USBSTOR, liệt kê từng loại thiết bị classes’ keys và tất cả các phiên bản duy nhất của thiết bị trong những phím đó, như được minh họa ở đây:

CdRom&Ven\_SanDisk&Prod\_U3\_Cruzer\_Micro&Rev\_3.27 [Fri Aug 29 00:36:11 2008]

S/N: 0000161511737EFB&1 [Fri Aug 29 00:36:11 2008]

FriendlyName : SanDisk U3 Cruzer Micro USB Device

Disk&Ven\_Best\_Buy&Prod\_Geek\_Squad\_U3&Rev\_6.15 [Fri Aug 29 00:36:11 2008]

  S/N: 0C90195032E36889&0 [Fri Dec 5 14:02:08 2008]

FriendlyName : Best Buy Geek Squad U3 USB Device

ParentIdPrefix: 7&326659cd&0

Disk&Ven\_Generic&Prod\_STORAGE\_DEVICE&Rev\_0026 [Fri Aug 29 00:36:11 2008]

  S/N: 0000082509&0 [Fri Aug 29 00:36:11 2008]

   FriendlyName : Generic STORAGE DEVICE USB Device

   ParentIdPrefix: 7&2a5b4c66&0

Disk&Ven\_Hitachi&Prod\_Easy\_Device&Rev\_ [Fri Aug 29 00:36:11 2008]

  S/N: 200718900AEA&0 [Fri Aug 29 00:36:11 2008]

  FriendlyName : Hitachi Easy Device USB Device

  S/N: 200718900CDC&0 [Fri Aug 29 00:36:11 2008]

  FriendlyName : Hitachi Easy Device USB Device

Disk&Ven\_Maxtor&Prod\_OneTouch&Rev\_0121 [Fri Aug 29 00:36:11 2008]

  S/N: 2HAA06KR\_\_\_\_&0 [Fri Aug 29 00:36:11 2008]

  FriendlyName : Maxtor OneTouch USB Device

  S/N: 2HAA07XG\_\_\_\_&0 [Fri Aug 29 00:36:11 2008]

  FriendlyName : Maxtor OneTouch USB Device

Trong đoạn trích trước của đầu ra từ usbstor.pl plugin, chúng tôi thấy một số thiết bị được liệt kê, đầu tiên là thiết bị U3 (chúng tôi sẽ thảo luận chi tiết về các thiết bị này trong phần tiếp theo). Chúng tôi cũng thấy hai thiết bị Maxtor OneTouch được liệt kê và chúng tôi có thể phỏng đoán rằng vì cả hai thiết bị này đều có giá trị *ParentIdPrefix* được liệt kê, chúng là ổ cứng ngoài thay vì ổ đĩa (và thực tế, chúng là). Plugin dev class.pl lấy dữ liệu mà chúng tôi đã mô tả trước đây từ khóa DeviceClass, cho cả các phím GUID của đĩa và ổ đĩa; một đoạn trích của đầu ra từ khóa GUID đĩa được hiển thị ở đây:

Wed Dec 17 13:18:41 2008 (UTC)

Disk&Ven\_SanDisk&Prod\_U3\_Cruzer\_Micro&Rev\_3.27,0000161511737EFB&0 Wed Dec 17 13:16:36 2008 (UTC)

Disk&Ven\_USB\_2.0&Prod\_USB\_Flash\_Drive&Rev\_0.00,3de63180700745&0 Fri Dec 5 14:02:11 2008 (UTC)

Disk&Ven\_Best\_Buy&Prod\_Geek\_Squad\_U3&Rev\_6.15,0C90195032E36889&0

Lưu ý rằng trong đầu ra từ plugin dev class.pl, các thiết bị khác nhau được liệt kê được sắp xếp theo *LastWrite* time của các khóa tương ứng. Một lần nữa, khi một thiết bị được kết nối với hệ thống Windows, các phím thiết bị bên dưới các phím DeviceClass có khóa con Control đã thêm và khi thiết bị bị ngắt kết nối, khóa con Control sẽ bị xóa. Những hành động này khiến thời gian *LastWrite* cho chính khóa thiết bị bị sửa đổi.

Một plugin khác mà tôi đã viết do một yêu cầu là plugin usbstor2.pl. Giống như tất cả các plugin khác, plugin này có thể được chạy như một phần của RegRipper, nhưng nó được dự định chạy với rip.pl như một phần của tệp bó. Tôi đã nhận được yêu cầu từ một nhà phân tích nói rằng anh ta đã thu được hình ảnh từ 30 hệ thống và nhiệm vụ của anh ta là tương quan các thiết bị lưu trữ di động USB trên tất cả 30 hệ thống, tìm kiếm mô hình sử dụng các thiết bị. Để tạo điều kiện cho điều đó, tôi đã viết usbstor2. plugin pl, trích xuất thông tin tương tự như usbstor.pl plugin, cộng với thông tin bổ sung và trình bày nó ở định dạng giá trị được phân tách bằng dấu phẩy (.csv). Đây là một đoạn trích về đầu ra của usbstor2.pl:

WINTERMUTE,CdRom&Ven\_SanDisk&Prod\_U3\_Cruzer\_Micro&Rev\_3.27,0000161511737EFB&1, 1219970171,SanDisk U3 Cruzer Micro USB Device

WINTERMUTE,Disk&Ven\_&Prod\_USB\_DISK&Rev\_1.13,0738015025AC&0,1219970171, USB DISK USB Device,7&2713a8a1&0

WINTERMUTE,Disk&Ven\_Best\_Buy&Prod\_Geek\_Squad\_U3&Rev\_6.15,0C90195032E36889&0, 1228485728,Best Buy Geek Squad U3 USB Device,7&326659cd&0

Như bạn có thể thấy, đầu ra liệt kê tên hệ thống, số nhận dạng lớp thiết bị, số nhận dạng cá thể duy nhất (hoặc số sê-ri), *LastWrite* time theo định dạng thời gian UNIX 32bit và tên thân thiện của thiết bị. Giá trị cuối cùng là giá trị *ParentIdPrefix*, nếu có. Vì vậy, tôi đã đề nghị với bạn tôi rằng anh ta chạy plugin usbstor2.pl thông qua rip.pl và xuất mỗi lần lặp lệnh thành một tệp bằng cách sử dụng chuyển hướng:

C:\Perl>rip.pl –p usbstor2.pl –r path\system >> devices.csv

Khi plugin đã được chạy trên tất cả các tệp hive của Hệ thống, anh ta có thể mở tệp thiết bị.csv trong Excel và sắp xếp dựa trên các thiết bị khác nhau hoặc tên hệ thống hoặc tuy nhiên anh ta chọn xem dữ liệu. Nếu anh ta làm việc với một ứng dụng xem Registry, thậm chí một ứng dụng được tích hợp vào ứng dụng phân tích pháp y, có thể anh ta phải mất nhiều ngày để hoàn thành những gì anh ta có thể làm trong vài phút với rip.pl.

### **USB Device Issues**

Các thiết bị lưu trữ di động USB từ lâu đã được biết đến (đặc biệt là bởi các chuyên gia bảo mật) gây ra mối đe dọa đối với bảo mật, đặc biệt là trong cơ sở hạ tầng của công ty. Kể từ thời của đĩa mềm (thậm chí trở lại cho đến khi những thứ này thực sự là đĩa mềm!), Số lượng Để làm cho vấn đề tồi tệ hơn, các thiết bị này có mặt khắp nơi. Nó đã từng là bất cứ ai có một ổ đĩa 64 MB có lẽ là một loại über-admin. Bây giờ mọi người đều có những thứ này và sử dụng chúng để lưu trữ hình ảnh, bài thuyết trình, v.v. Còn máy nghe nhạc iPod và MP3 thì sao? Chúng tôi thấy họ trong phòng tập thể dục, trong văn phòng, trên xe buýt; họ ở khắp mọi nơi. Trên thực tế, chúng tôi đã từng nhìn thấy chúng, vì vậy nhìn thấy một cái trên bàn, cắm vào máy tính xách tay, không có gì khác thường cả. Ngay bây giờ, bạn có thể mua một chiếc iPod Nano 8GB với giá khoảng 200 đô la. Nếu ai đó cắm một trong những thứ này vào máy tính xách tay mà kết nối với một công ty địa phương Mạng khu vực (LAN), người có thể biết người dùng đang nghe nhạc hoặc tải xuống dự báo tài chính, kế hoạch bồi thường, giá thầu hợp đồng và thông tin bí mật khác cho thiết bị lưu trữ không?

Một vấn đề khác liên quan đến chức năng AutoPlay được đề cập trước đó trong chương này. Khi đĩa CD hoặc DVD được đặt trong ổ đĩa trên hệ thống Windows, phương tiện mới được phát hiện và nếu tệp autorun.inf nằm trong thư mục gốc của ổ đĩa, nó sẽ được phân tích cú pháp và dòng run = và load = được thực thi. Đây là một phần của trải nghiệm người dùng nâng cao mà Windows cung cấp. Theo mặc định, chức năng AutoRun không được bật cho các thiết bị lưu trữ di động như ổ USB; tuy nhiên, nó được bật theo mặc định cho những gì hệ thống tin là CD-ROM, đây là phần mà thiết bị U3 xuất hiện, do cách thức mà nó định dạng. Các hướng dẫn có sẵn trên Internet chỉ cho bạn cách tạo dữ liệu và chương trình của riêng bạn để được ghi trên phân vùng U3 hiện có, khiến việc này trở nên cực kỳ nguy hiểm cơ chế để có được quyền truy cập vào hệ thống. Công ty U3 cung cấp một tiện ích để cung cấp cho người dùng tính di động cao hơn với các ứng dụng của họ. Tiện ích này tạo một phân vùng nhỏ ở đầu ổ cắm USB và đánh dấu nó là phân vùng CDFS (hệ thống tệp CD) để các hệ thống Windows nhận ra phân vùng đó là CD thay vì một thiết bị lưu trữ di động. Các tiện ích (trình duyệt và tương tự) sau đó được chạy từ phân vùng CDFS và phần còn lại của thiết bị được sử dụng để lưu trữ. Tuy nhiên, điều này có nghĩa là mặc dù chức năng AutoRun bị tắt (theo mặc định) cho các thiết bị lưu trữ di động, nó được bật (theo mặc định) cho phân vùng CDFS.

 Khi tôi kết nối ổ hỗ trợ U3 với hệ thống Windows XP của mình, tôi thấy rằng hai mục nhập ID lớp thiết bị riêng biệt đã được tạo cho cùng một thiết bị:

CdRom&Ven\_Best\_Buy&Prod\_Geek\_Squad\_U3&Rev\_6.15

and

Disk&Ven\_Best\_Buy&Prod\_Geek\_Squad\_U3&Rev\_6.15

Cả hai IDs lớp thiết bị này đều có cùng một khóa con ID cá thể duy nhất bên dưới chúng. Khi bạn thực hiện phân tích pháp y sau khi chết, đây chắc chắn là điều cần tìm vì nó có thể xác định một vec tơ lây nhiễm hoặc phương pháp thỏa hiệp. Trường hợp điều này xảy ra là kẻ tấn công có thể tạo hình ảnh ISO tùy chỉnh để cài đặt vào phân vùng CDFS và sau đó xóa tất cả dấu hiệu của các tiện ích hoặc logo U3 trên thiết bị. Nếu ai đó cắm thiết bị đó vào hệ thống, chức năng AutoRun cho phân vùng CDFS sẽ hoạt động và bất cứ điều gì kẻ tấn công có thể nghĩ ra (cài đặt backtime Trojan, thu thập thông tin Lưu trữ được bảo vệ và mật khẩu khác) sẽ được thực thi tự động.

Xem thanh bên của “USBDumper” để biết thêm thông tin về các mối đe dọa từ các thiết bị lưu trữ di động. Mặc dù không liên quan cụ thể đến phân tích Registry, những mối đe dọa này gây ra những vấn đề thú vị cho các chuyên gia bảo mật.

**Tools and Traps…**

**USBDumper**

Một tiện ích có tên USBDumper cho thấy một rủi ro bảo mật khác liên quan đến các thiết bị lưu trữ di động USB, nhưng với một sự thay đổi khác (http://wiki.hak5.org/wiki/USB\_Hacksaw). USBDumper được cài đặt trên hệ thống Windows và bất cứ khi nào ổ USB di động được kết nối với hệ thống, nội dung của thiết bị sẽ được sao chép âm thầm khỏi thiết bị. Người ta cũng đã nói về một tiện ích âm thầm thu được hình ảnh của một ổ đĩa ngón tay cái khi nó kết nối với một hệ thống để không chỉ có thể lấy tất cả các tệp đang hoạt động mà còn có thể xóa các tệp bị xóa. Cả hai vấn đề này đều được đề cập trong Bruce Schneier’s “Schneier on Security” blog vào ngày 25 tháng 8 năm 2006).

## **Mounted Devices**

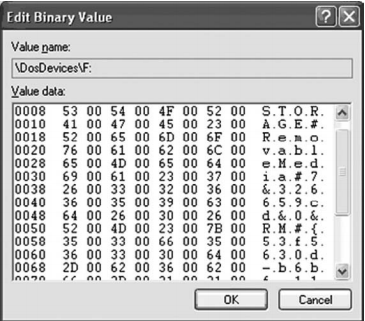
Khóa MountedDevices lưu trữ thông tin về các thiết bị khác nhau và được chia thành hệ thống tệp NTFS. Đường dẫn đầy đủ đến khóa là:

HKEY\_LOCAL\_MACHINE\System\MountedDevices

Ví dụ, khi thiết bị lưu trữ di động USB được kết nối với hệ thống Windows, thiết bị được gán ký tự ổ đĩa; ký tự ổ đĩa đó hiển thị trong khóa MountedDevices. Nếu thiết bị được gán ký tự ổ đĩa F:\, giá trị trong khóa MountedDevices sẽ xuất hiện dưới dạng \DosDevices\F:. Chúng tôi có thể ánh xạ mục nhập từ khóa USBSTOR sang khóa MountedDevices bằng cách sử dụng giá trị *ParentIdPrefix* được tìm thấy trong khóa ID cá thể duy nhất cho thiết bị. Giá trị *ParentIdPrefix* cho thiết bị USB được thảo luận trong phần trước có dữ liệu 7 & 326659cd & 0. Lưu ý rằng đây không phải là ID cá thể duy nhất và do đó không phải là số sê-ri mà chúng ta đã thảo luận trước đó.

Khi chúng tôi có dữ liệu từ giá trị Registry *ParentIdPrefix*, sau đó chúng tôi xác định vị trí ký tự ổ đĩa được gán cho nó bằng cách định vị mục nhập *DosDevices* trong khóa MountedDevices có chứa *ParentIdPrefix* trong dữ liệu của nó. Trên hệ thống trực tiếp, chúng tôi có thể thực hiện việc này một cách dễ dàng bằng cách nhấp chuột phải vào từng giá trị Registry và chọn **Modify**; khi hộp thoại **Edit Binary Value** mở ra, chúng ta có thể xem nội dung của dữ liệu để xem giá trị ParentIdPrefix có ở đó không. Giá trị *ParentIdPrefix* được lưu trữ trong Registry dưới dạng chuỗi, nhưng *DosDevices* Các giá trị trong MountedDevices Regitry key được lưu trữ dưới dạng các kiểu dữ liệu nhị phân, vì vậy một số dịch thuật là cần thiết. Hình 4.15 minh họa hộp thoại Edit Binary Value cho \DosDevices\F: .

Hình 4.15 Dữ liệu cho MountedDevices\DosDevices\F: Value



Chúng ta có thể thấy rõ giá trị *ParentIdPrefix* của 7&326659cd&0 trong Hình 4.15. Sử dụng giá trị *ParentIdPrefix* để ánh xạ giữa các khóa USBSTOR và MountedDevices Registry, chúng ta có thể định vị ký tự trình điều khiển được gán cho thiết bị. Khi chúng tôi tiếp tục điều tra pháp y sau khi chết, chúng tôi có thể tìm thấy các tài liệu tham khảo ở các vị trí khác trong Regestry hoặc trong các tệp Shortcut trỏ đến ổ F:\. Sau đó, chúng ta có thể tương quan *LastWrite* time của khóa ID cá thể duy nhất, khóa MountedDevices và thời gian MAC trên các tệp để phát triển dòng thời gian.

**Warring:** Khi thực hiện tương quan giữa các khóa USBSTOR và MountedDevices bằng cách sử dụng giá trị của *ParentIdPrefix*, hãy nhớ rằng một số thiết bị có thể đã được kết nối với hệ thống theo kiểu nối tiếp và được ánh xạ tới cùng một ký tự ổ đĩa. Tôi có một số ổ USB khác nhau từ các nhà sản xuất khác nhau và có kích cỡ khác nhau và bất cứ khi nào tôi kết nối chúng với máy tính xách tay của mình, tất cả chúng đều được ánh xạ tới ổ F: \. Khi thực hiện loại tương quan này, điều quan trọng là phải ghi nhớ điều này, đặc biệt khi phân tích hệ thống Windows XP hoặc 2003. Như bạn sẽ thấy sau trong chương này, các hệ thống Windows Vista và Windows 7 cung cấp một số cách phân tích bổ sung cho đến khi ánh xạ một thiết bị tới ký tự ổ đĩa.

Nhìn vào khóa MountedDevices, bạn có thể nhận thấy rằng có các mục \ DosDevices (cụ thể là \DosDevices\C:) có dữ liệu chỉ dài 12 byte (ba DWORD). Hình 4.16 minh họa một ví dụ về điều này**.**

Hình 4.16 Dữ liệu cho MountedDevices\DosDevices\C: và D: Values



Các giá trị nhị phân được hiển thị trong Hình 4.16 bao gồm chữ ký ổ đĩa (còn được gọi là *volume* ID) cho ổ đĩa cứng (*DWORD* đầu tiên) và bù phân vùng (*DWORD* thứ hai và thứ ba). Một chữ ký ổ đĩa được đặt ở offset 0x1b8 (440 ở dạng thập phân) trong Master Boot Record (MBR) của ổ cứng.

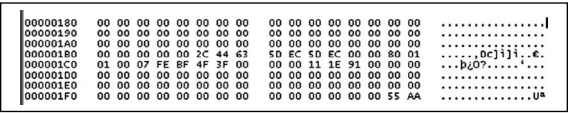
Đối với giá trị \*DosDevices*\C: được hiển thị trong Hình 4.16, các DWORD thứ hai và thứ ba cùng dịch thành giá trị hex 0x7e00, có giá trị thập phân là 32256. Mỗi khu vực trên ổ cứng có kích thước 512 byte; 32256/512 = 63, vì vậy phân vùng C: \ bắt đầu ở khu vực 63.

Khi xem mục nhập cho \*DosDevices*\D: trong Hình 4.15, chúng ta thấy rằng dữ liệu nhị phân chỉ ra rằng ổ D:\ có cùng chữ ký ổ đĩa với ổ C:\, nhưng bù cho phân vùng khác. Điều này là do ổ D:\ là một phân vùng khác trên cùng một đĩa vật lý như ổ C:\. Sử dụng thông tin về độ lệch của phân vùng, chúng ta có thể tính toán phân vùng C:\ có kích thước lớn hơn 29 GB một chút.

Để chỉ ra cách tìm thông tin này trong quá trình điều tra pháp y, tôi đã mở ProDiscover và sau đó mở một trường hợp mẫu. Tôi đã tìm mục nhập \*DosDevices*\C: trong khóa MountedDevices trong Registry Viewer chữ ký ổ đĩa là 5D EC 5D EC ở dạng hex. Sau đó tôi đã nhấp vào chế độ xem cụm cho trường hợp (bắt đầu từ 0). Hình 4.17 minh họa một đoạn trích từ ProDiscover, hiển thị một phần của MBR.

Hình 4.17 Excerpt from ProDiscover Cluster View Showing Drive Signature

(5D EC 5D EC)



Như Hình 4.17 chỉ ra, chữ ký ổ đĩa chúng ta nhận được từ khóa MountedDevices hiển thị rõ ràng ở offset 0x1B8.

Tại sao nó lại quan trọng? Rốt cuộc, nếu chúng tôi đã thu được hình ảnh của ổ cứng, chúng tôi đã có và có thể xác minh chữ ký ổ đĩa từ cả Registry và MBR. Tuy nhiên, nơi thông tin này trở nên hữu ích với chúng ta là nếu có các ổ cứng ngoài liên quan đến vỏ máy. Trong công việc ngày của tôi, tôi là một kỹ sư pháp y và pháp y, tôi sử dụng các ổ cứng ngoài để lưu trữ những thứ như dữ liệu khách hàng, hình ảnh ổ đĩa và tệp nhật ký. Tôi làm điều này bởi vì một khi trường hợp hoàn tất và báo cáo cuối cùng đã được chấp nhận, tôi có thể dễ dàng xóa sạch ổ đĩa ngoài. Nếu tôi duy trì tất cả dữ liệu này trên ổ cứng máy tính xách tay của tôi, tôi sẽ phải xóa sạch ổ đĩa và cài đặt lại hệ điều hành cũng như tất cả các ứng dụng và dữ liệu của tôi, bao gồm các khóa PGP và những thứ tương tự. Ngoài ra, khi tôi viết cuốn sách này, tất cả các tệp được liên kết với cuốn sách được duy trì trên ổ cứng kết nối USB kỹ thuật số Western 120GB. Thiết bị này xuất hiện trong khóa USBSTOR với ID lớp thiết bị sau:

Disk&Ven\_WDC\_WD12&Prod\_00UE-00KVT0&Rev\_0000

Như mọi người có thể mong đợi, thiết bị này có số sê-ri (mà tôi có thể xác minh bằng UVCView), như được chỉ định bởi ID cá thể duy nhất tôi thấy trong Sổ đăng ký của mình. Tuy nhiên, trong khóa Đăng ký ID cá thể duy nhất, không có giá trị có tên ParentIdPrefix. Khi thiết bị được cắm vào máy tính xách tay của tôi, nó sẽ xuất hiện dưới dạng ổ G: \ và thông tin thích hợp, bao gồm cả chữ ký ổ đĩa, được lưu trong phím MountedDevices. Nếu đây là một trường hợp mà tôi đang điều tra, thì tôi có thể kết nối duy nhất ổ cứng ngoài với hệ thống, mặc dù không có giá trị ParentIdPrefix cho ổ đĩa ngoài.

Các giá trị trong khóa MountedDevices bắt đầu bằng cách sử dụng “\??\Volume” có thể được liên kết với một người dùng cụ thể, bằng cách điều hướng đến khóa sau trong tệp user’s NTUSER.DAT:

Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2

Một số loại khóa con khác nằm bên dưới khóa này, nhưng loại mà chúng tôi quan tâm vào thời điểm này là những loại khóa trông giống như GUID; ví dụ, Tiếng {d37df1d3-265b-11db-a216-0015c51b9712}. Bên dưới khóa MountedDevices, bạn cũng sẽ thấy một giá trị \??\Volume {d37df1d3-265b-11db-a216-0015c51b9712} tương ứng. Plugin RegRipper mp2.pl phân tích các khóa con bên dưới khóa người dùng Mount MountPoints2 và liệt kê các mục theo loại điểm gắn kết, như được minh họa ở đây (nhấn mạnh thêm):

C:\Perl\forensics\rr>rip.pl -p mp2 -r d:\cases\NTUSER.DAT

Launching mp2 v.20090115

MountPoints2

Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2

LastWrite Time Mon Sep 26 23:32:34 2005 (UTC)

Drives:

A Mon Sep 26 22:50:40 2005 (UTC)

D Mon Sep 26 22:50:40 2005 (UTC)

C Mon Sep 26 22:50:40 2005 (UTC)

Remote Drives:

##192.168.1.22#c$ Mon Sep 26 23:26:00 2005 (UTC)

##192.168.1.71#c$ Mon Sep 26 23:32:34 2005 (UTC)

Volumes:

{2490fb15-f08b-11d8-958e-806d6172696f} Mon Sep 26 22:50:58 2005 (UTC)

{2490fb13-f08b-11d8-958e-806d6172696f} Mon Sep 26 23:37:22 2005 (UTC)

{082b2da3-6b36-11d9-95d0-000c29960ded} Mon Sep 26 23:14:02 2005 (UTC)

{2490fb12-f08b-11d8-958e-806d6172696f} Mon Sep 26 22:50:58 2005 (UTC)

Analysis Tip: Correlate the Volume entries to those found in the

MountedDevices entries that begin with "\??\Volume".

Các GUIDs được liệt kê trong các Tập tin trên mạng: Sau đó, có thể tương quan với các giá trị được liệt kê trong khóa MountedDevices. Ví dụ: plugin RegRipper mountdev2.pl được chạy với thư mục System được trích xuất từ cùng một hình ảnh hệ thống với tệp NTUSER.DAT được sử dụng trước đó và thông tin sau được hiển thị (nhấn mạnh thêm):

Device: \??\STORAGE#RemovableMedia#7&2713a8a1&0&RM#{53f5630d-b6bf-11d0 -94f2-00a0c91efb8b}

\??\Volume{082b2da3-6b36-11d9-95d0-000c29960ded}

\DosDevices\E:

Trong trường hợp này, thiết bị lưu trữ di động có giá trị *ParentIdPrefix* được liệt kê đã được gắn dưới dạng ổ E:\ và người dùng có thể truy cập được từ hồ sơ mà tệp NTUSER.DAT đã được truy xuất. Bây giờ, theo trực giác điều này có ý nghĩa trên các hệ thống được sử dụng chủ yếu bởi một người dùng; tuy nhiên, trên các hệ thống có nhiều hồ sơ người dùng, nhà phân tích có thể cần xác định người dùng nào có quyền truy cập vào phương tiện lưu trữ di động.

**Tools & Traps…**

**Data Exfiltration:**

Tôi đã nhận được một số câu hỏi thông qua e-mail và từ khách hàng, hỏi về việc hoàn thành dữ liệu. Trong phần lớn các trường hợp, ai đó có một hệ thống (hoặc hình ảnh thu được từ hệ thống) và muốn biết dữ liệu nào được sao chép khỏi hệ thống đó, có thể trên thiết bị lưu trữ di động. Thực tế của vấn đề là có một số phương tiện mà người dùng có thể sao chép dữ liệu khỏi hệ thống, chẳng hạn như bằng cách đính kèm tệp vào e-mail dựa trên Web, sử dụng ứng dụng khách Transfer Protocol (FTP) tích hợp, và kể từ đó trở đi. Khi bạn đang tìm kiếm chỉ dẫn hoặc bằng chứng trực tuyến, các tập tin đã được sao chép từ hệ thống sang phương tiện lưu động (ví dụ: ổ cắm USB, iPod, v.v.), thực tế đơn giản là tại thời điểm này, không có tạo tác rõ ràng nào về điều này và bạn sẽ cần phải thu thập và phân tích cả hai phần của phương tiện (tức là hệ thống là nguồn và phương tiện di động là mục tiêu). Các tạo phẩm của một hoạt động sao chép, chẳng hạn như sử dụng lệnh sao chép hoặc kéo và thả, không được ghi lại trong Registry hoặc trong hệ thống tệp, theo như tôi và những người khác có thể xác định.

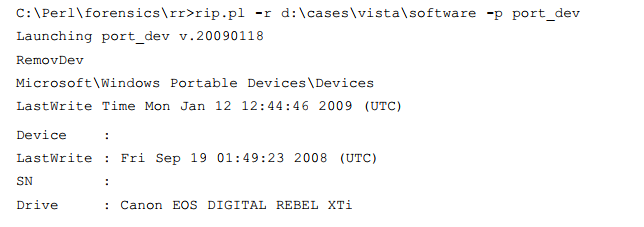
Một lưu ý cho điều này là nếu người dùng sao chép dữ liệu vào ổ ngón tay cái và sau đó nhân đôi tệp đã được sao chép vào ổ ngón tay cái (giả sử, để xác minh rằng nó đã được sao chép chính xác), tệp phím tắt Windows (.Ink) sẽ được tạo trong thư mục user’s Recent Documents . Mặc dù điều này sẽ cung cấp rất nhiều thông tin hữu ích (ví dụ: tem thời gian, đường dẫn tệp, v.v.), thông tin duy nhất có thể chỉ ra rằng tệp đã được sao chép từ hệ thống vào ổ USB có thể là thời gian MAC của tệp, trong đó được nhúng trong tập tin phím tắt. Tuy nhiên, không có tạo tác có sẵn nào chỉ rõ rằng tệp đã được sao chép từ hệ thống vào ổ cắm USB

## **Thiết bị cầm tay**

Vào mùa xuân năm 2008, Rob Lee (danh tiếng của Sans) đã liên lạc với tôi để nói với tôi rằng anh ấy đã phát hiện ra rằng Windows Vista duy trì lịch sử của các thiết bị di động trong thư mục Phần mềm. Mặc dù Trang web Microsoft chứa rất nhiều thông tin liên quan đến các tiêu chuẩn thiết bị di động, nó cung cấp rất ít thông tin (nếu có) đối với khóa Registry sau:

*HKLM\Software\Microsoft\Windows Portable Devices\Devices*

Bên dưới khóa này, có thể có một số khóa con có tên được định dạng theo cách tương tự như bên dưới khóa USBSTOR đã đề cập trước đây trong phần này chương. Trong thực tế, một số tên khóa con có thể bao gồm từ USBSTOR. Trong những khóa con này, bạn có thể tìm thấy giá trị FriendlyName Registry có vẻ như là những gì bạn plugin dev.pl RegRipper trích xuất và hiển thị thông tin từ khóa Thiết bị, như minh họa ở đây:





Như bạn có thể thấy, một số thiết bị đã được kết nối với hệ thống Vista này (đầu ra được hiển thị chỉ là một đoạn trích của những gì có sẵn), bao gồm iPod, máy ảnh kỹ thuật số. Ngoài ra, một số sê-ri (nếu có) được hiển thị, như là giá trị FriendlyName, được liệt kê trong đầu ra cùng với Drive :. Trong bốn thiết bị được liệt kê, bạn có thể thấy rằng tất cả chúng đều được ánh xạ tới ổ F: \. Tôi đã tìm thấy trên một số hệ thống XP khi kết nối nhiều thiết bị lưu trữ di động USB với các hệ thống cùng một lúc một kiểu nối tiếp, tất cả chúng đều được ánh xạ tới cùng một ký tự ổ đĩa. Trên XP và Windows 2003 các hệ thống, đây có thể là một thiếu sót nghiêm trọng khi cố gắng lập bản đồ lưu trữ di động thiết bị cho ký tự ổ đĩa được gán khi nó được cắm vào hệ thống, như bạn đã thấy trước đó trong chương này trong cuộc thảo luận của chúng tôi về khóa MountedDevices. Tuy nhiên, với Vista (và Windows 7), ít nhất một lịch sử một phần của việc gán ký tự ổ đĩa dường như là duy trì trong Phần mềm.

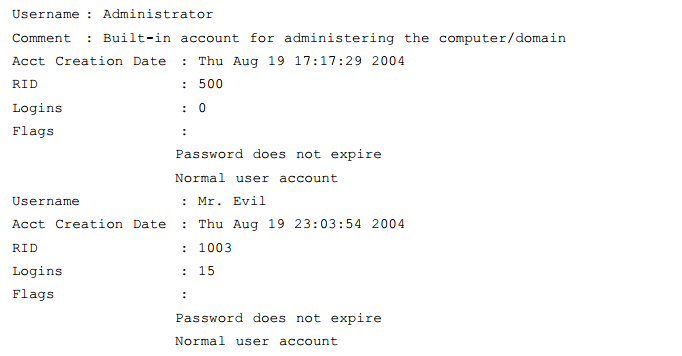
## **Tìm Kiếm Người Dùng**

Thông tin về người dùng được duy trì trong Registry, trong thư mục SAM. Hoàn cảnh bình thường, thư mục này không thể truy cập được, ngay cả với quản trị viên, không phải không các bước đặc biệt để chỉnh sửa thủ công các quyền truy cập trên thư mục. Có một lý do chính đáng cho việc này: Mặc dù phần lớn các Registry có thể bị rối loạn, nhưng có những khu vực của Registry, nơi những thay đổi nhỏ có thể khiến hệ thống không thể sử dụng được. Thư mục SAM là một trong những khu vực đó.

Phần lớn thông tin hữu ích trong tệp SAM được mã hóa ở định dạng nhị phân và may mắn thay, tập tin tiêu đề Peter Nordahl-Hagen’s sam.h C cực kỳ hữu ích trong việc giải mã các cấu trúc và tiết lộ một cái gì đó dễ hiểu.

Bạn có thể sử dụng userdump.pl ProScript (v.0.31, 20060522 được cung cấp trong ch4\code\ProScripts Thư mục trên phương tiện đi kèm với cuốn sách này) để trích xuất người dùng và nhóm thông tin thành viên từ Registry Viewer trong dự án ProDiscover, một khi Trình xem đăng ký đã được điền. Để chạy ProScript, nhấp vào nút Chạy ProScript trên thanh menu và chọn vị trí của ProScript từ hộp thoại Run ProScript cái hộp. (Tùy chọn, bạn cũng có thể nhập bất kỳ đối số nào cho ProScript.) Chọn userdump.pl ProScript và khi quá trình thực thi tập lệnh hoàn tất, thông tin sẽ hoàn tất được phân tích cú pháp từ tệp SAM sẽ hiển thị trong cửa sổ kết quả, nơi nó có thể được chọn, sao chép và dán vào một tập tin hoặc báo cáo

Chạy userdump.pl ProScript dựa trên trường hợp mẫu mà tôi có sẵn, chúng ta có thể xem trích đoạn từ các kết quả để xem chiều rộng của thông tin được trả về bởi tập lệnh. Ví dụ, ProScript phân tích thông tin về tài khoản người dùng trên hệ thống, bao gồm tên người dùng, nhận xét, ngày tạo tài khoản, số lần đăng nhập và cờ người dùng (cung cấp thông tin về tài khoản). ProScript cũng sẽ hiển thị thời gian đăng nhập cuối cùng, nếu nó không khác.



Thông tin người dùng này được duy trì trong giá trị F nằm trong đường dẫn sau:

*SAM\SAM\Domains\Account\Users\{RID}*

{RID}, hoặc Mã định danh tương đối, là một phần của Mã định danh bảo mật (SID) xác định người dùng hoặc nhóm liên quan đến cơ quan cấp SID. Bên cạnh việc cung cấp khá nhiều thông tin về cách tạo SID, Microsoft cũng cung cấp danh sách RID (http://support.microsoft.com/kb/157234) cho cả người dùng và nhóm nổi tiếng cũng như các bí danh nổi tiếng (được thấy trong khóa SAM\SAM\Domains\Builtin\Aliases key)

Giá trị F trong khóa là loại dữ liệu nhị phân và phải được phân tích cú pháp phù hợp (xem tệp sam.h, một phần của mã nguồn cho tiện ích Peter Lau) để trích xuất tất cả thông tin. Một số ngày quan trọng có sẵn trong nội dung của dữ liệu nhị phân cho giá trị F đặc biệt, một số tem thời gian / ngày được biểu thị dưới dạng các đối tượng FILETIME 64 bit. Những giá trị đó và địa điểm như sau:

■ Byte 8-15 thể hiện ngày đăng nhập cuối cùng của tài khoản.

■ Byte 24-31 biểu thị ngày mật khẩu được đặt lại lần cuối (nếu mật khẩu

đã được đặt lại hoặc thay đổi, ngày này sẽ tương quan với ngày tạo tài khoản).

■ Byte 32-39 thể hiện ngày hết hạn tài khoản.

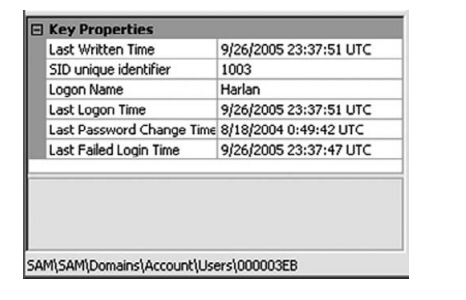
■ Byte 40-47 thể hiện ngày của lần đăng nhập thất bại cuối cùng (vì tài khoản

Tên phải chính xác cho ngày được thay đổi trên một tài khoản cụ thể, ngày này

cũng có thể được gọi là ngày sử dụng mật khẩu không chính xác cuối cùng).

Các công cụ như AccessData’s Registry Viewer sẽ giải mã thông tin này cho bạn

tự động, như Hình 4.18 minh họa..



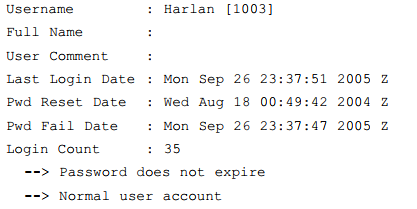
Hình 4.18: Một phần của AccessData, Registry Viewer Showing Decode giá trị F của người dùng

AccessData’s Registry Viewer có sẵn để tải xuống (www.accessdata.com/doads.Html) và sẽ chạy trong chế độ demo nếu bạn không có khóa AccessData. Để sử dụng AccessData’s Registry Viewer để giải mã các giá trị này, trước tiên bạn phải trích xuất tệp Registry thô (trong trường hợp này là tệp SAM) từ hình ảnh, sao chép nó sang vị trí khác.

Mô-đun Perl Parse :: Win32Registry (có sẵn từ <http://search.cpan.org/Jmacfarla/> ) cung cấp một phương thức đa nền tảng có sẵn miễn phí để phân tích nội dung của các tệp Registry thô đã được trích xuất tương tự từ một hình ảnh thu được (hoặc đã được được cung cấp thông qua một số phương tiện khác). Để cài đặt Phiên bản 0.41 của mô-đun này trên các hệ thống Windows, chỉ cần sử dụng tiện ích Trình quản lý gói Perl ActiveState (Perpm) như sau:

**C:\Perl>ppm install parse-win32registry**

Plugin sampude.pl (tập lệnh Perl sam\_parse.pl độc lập nằm trong ch4\code\old thư mục trên phương tiện đi kèm) sử dụng mô-đun này để giải nén và hiển thị nhiều thông tin tương tự có sẵn từ Registry Viewer, cộng với một số thông tin thông tin thêm:



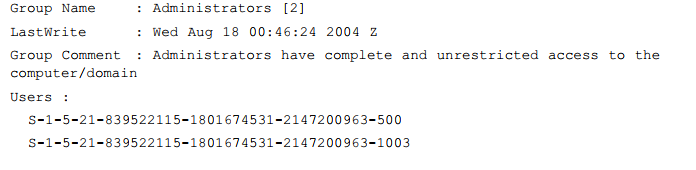
Plugin sampude.pl không chỉ hiển thị dấu thời gian có sẵn từ thông tin người dùng trong tệp SAM mà còn phân tích cờ người dùng và số lần người dùng có đăng nhập vào hệ thống.

Bởi vì mô-đun Parse :: Win32Registry không dựa trên API Windows, tập lệnh Perl sử dụng

mô-đun sẽ độc lập với nền tảng. Điều này có nghĩa là các nhà phân tích và điều tra viên không bị giới hạn ở một nền tảng Windows khi họ muốn phân tích nội dung của các tệp Registry thô. Miễn là các tệp có sẵn (tức là, được trích xuất từ tệp hình ảnh, v.v.), mã được viết bằng cách sử dụng mô-đun có thể chạy trên các hệ thống Linux, Windows hoặc thậm chí Mac OS X hỗ trợ Perl.

Mỗi khóa này cũng có giá trị V cũng là loại dữ liệu nhị phân và có thể được phân tích cú pháp thành nhận cài đặt tài khoản của người dùng, như tên đầy đủ, nhận xét, đường dẫn đến tập lệnh đăng nhập (nếu có), và băm mật khẩu được mã hóa.

Plugin sampude.pl cũng lấy thông tin về các nhóm trên hệ thống, bao gồm tên nhóm, nhận xét nhóm và người dùng được chỉ định cho nhóm:



Thông tin về tư cách thành viên nhóm được duy trì trong SAM\SAM\Domains\Builtin\Aliases key. Mỗi khóa con RID bên dưới khóa Bí danh có giá trị C là nhị phân loại dữ liệu và cần được phân tích cú pháp để xác định người dùng nào là thành viên của nhóm. Các bản đồ đường tốt nhất tôi tìm thấy để phân tích dữ liệu nhị phân này có sẵn từ Andreas Schuster’s blog (<http://computer.forensikblog.de/en/2006/02/list_members_of_a_windows_group.html>). Thông tin này được tích hợp vào userdump.pl ProScript.

Thông tin về người dùng và thành viên nhóm là vô cùng quý giá trong việc tìm hiểu bối cảnh của một cuộc kiểm tra, cụ thể là người dùng nào đã truy cập vào hệ thống và mức độ nào truy cập họ đã có (thông qua thành viên nhóm). Phần lớn thông tin này được trích xuất dễ dàng trong thời gian phản ứng trực tiếp bằng cách sử dụng các công cụ có sẵn. Với một chút thông tin về các cấu trúc khác nhau được duy trì trong Registry, bạn có thể trích xuất thông tin tương tự từ một hình ảnh hậu hiện đại.

## **Theo Dõi Hoạt Động Người Dùng**

Bạn có thể sử dụng một số khóa Registry để theo dõi hoạt động của người dùng. Loại khóa Registry này là khác với các khóa đăng ký hoạt động / người dùng tự động, là các khóa được truy cập khi người dùng thực hiện một hành động cụ thể. Bạn có thể tìm thấy các khóa Registry này trong tệp NTUSER.DAT cho người dùng và họ được cập nhật (tức là, các mục được thêm vào) khi người dùng thực hiện các hành động cụ thể. Khi điều này xảy ra, thời gian khóa LastWrite được cập nhật, điều này đưa chúng ta trở lại với khái niệm của Registry như một tệp nhật ký. Ngoài ra, có các khóa theo dõi hoạt động của người dùng và thêm hoặc sửa đổi thông tin về dấu thời gian được liên kết với các giá trị Registry (ví dụ: các khóa con UserAssist); điều này thông tin dấu thời gian được duy trì trong dữ liệu giá trị.

Thư mục HKEY\_CURRENT\_USER chứa nhiều thông tin nhất về hoạt động của người dùng trong các vị trí được gọi là danh sách được sử dụng gần đây nhất hoặc danh sách MRU. Tên này xuất phát từ thực tế, như bạn sẽ thấy, các phím này duy trì một danh sách các tệp hoặc lệnh cũng như một giá trị được đề cập đến là MRUlist. Mỗi giá trị trong khóa được chỉ định bởi một mã định danh, chẳng hạn như chữ thường chữ cái và giá trị MRUlist hiển thị thứ tự mà chúng được truy cập.

### **Các khóa UserAssist**

Khá nhiều điều đã được viết về các khóa UserAssist, hầu hết chúng đã xuất hiện trong dạng câu hỏi. Các khóa cụ thể mà chúng tôi quan tâm được đặt bên dưới khóa sau đường dẫn trong tập tin người dùng từ NTUSER.DAT:

*Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{GUID}\Count*

GUID là một định danh duy nhất trên toàn cầu; trong trường hợp này, hai khóa như vậy nằm bên dưới Khóa người dùng: 5E6AB780-7743-11CF-A12B-00AA004AE837 và 75048700-EF1F- 11D0-9888-006097DEACF9. Trong thư mục HKEY\_CLASSES\_ROOT, GUID 5E6AB780-7743-11CF-A12B-00AA004AE837 trỏ đến Thanh công cụ Internet (chẳng hạn như % SystemRoot% \ system32 \ Browseui.dll) và GUID 75048700-EF1F-11D0-9888- 006097DEACF9 trỏ đến ActiveDesktop (chẳng hạn như %SystemRoot%\system32\SHELL32.dll). Tầm quan trọng của việc này sẽ trở nên rõ ràng sau khi bạn khám phá ra những gì trong Những chìa khóa này và tại sao nó lại hữu ích.

Bên dưới khóa Count là một vài giá trị; trong thực tế, có thể có nhiều, nhiều giá trị. Khi lần đầu tiên tôi bắt đầu nghiên cứu các khóa này (thường được gọi là các khóa UserAssist hơn so với các khóa Count), tôi đã tìm thấy các trang web trên Internet có báo cáo lên tới 18.000 trở lên các mục bên dưới một khóa và 400 hoặc hơn bên dưới khóa kia. Hệ thống đó là Windows 2000 hệ thống đã hoạt động được khoảng năm năm khi bài đăng được thực hiện trên trang web. Những gì đặc biệt về các khóa này? Tóm lại, họ đăng nhập hoạt động của người dùng, ở một mức độ. Vâng, đúng nó đấy đúng, bạn đọc nó một cách chính xác. Ở một mức độ nào đó, các khóa này thực sự ghi lại hoạt động của người dùng như một tệp nhật ký.

Tuy nhiên, nếu bạn điều hướng đến các khóa này trong RegEdit, bạn sẽ thấy điều đó. Bạn sẽ thấy một cái gì đó như HRZR\_HVGBBYONE, điều này hoàn toàn không có ý nghĩa. Đó là bởi vì các tên giá trị bên dưới hai khóa này được mã hóa ROT-13,. ROT-13 đề cập đến một Mật mã sinh mổ trong đó mỗi chữ cái được thay thế bằng chữ cái 13 khoảng cách xa hơn trong bảng chữ cái. Sử dụng một sự thay thế đơn giản (trong Perl, tr / N-ZA-Mn-za-m / A-Za-z /), chúng ta có thể sau đó thấy rằng HRZR\_HVGBBYONE thực sự là UEME\_UITOOLBAR. Được rồi, đó là một chút dễ đọc hơn, nhưng chúng tôi thực sự không gần gũi hơn với câu trả lời vào thời điểm này, phải không?

Tên giá trị bên dưới cả hai khóa được mã hóa ROT-13 và có thể dễ dàng giải mã.Trên thực tế, tập lệnh Perl uassist.pl (bao gồm trên phương tiện đi kèm) minh họa cách đơn giản bản dịch này là để thực hiện trên một hệ thống sống. Kho báu thực sự trong các khóa này là trong dữ liệu liên quan đến từng giá trị. Trong nhiều trường hợp, tên giá trị được giải mã trỏ đến một ứng dụng hoặc một thực thi. Trong những trường hợp đó, dữ liệu thường là 16 byte (bốn DWORD) dài và không chỉ bao gồm số lần chạy (số lần ứng dụng hoặc thực thi đã được chạy), nhưng cũng là lần chạy cuối cùng (một đối tượng FILETIME 8 byte). Số lượng chạy được lưu trữ trong DWORD thứ hai và bắt đầu từ 5; do đó, số lần chạy là 6 có nghĩa là ứng dụng đã được đưa ra một lần. Đối tượng FILETIME nằm trong DWORD thứ ba và thứ tư.

Khi bạn giải mã các tên giá trị, bạn sẽ thấy rằng nhiều trong số chúng có trước bởi UEME\_, và sau đó RUNPATH, RUNPIDL, RUNCPL, v.v. Các thẻ này có thể tương đối dễ dàng sắp xếp:

■ RUNPATH Đề cập đến một đường dẫn tuyệt đối trong hệ thống tệp; nó xảy ra khi bạn

bấm đúp vào biểu tượng để thực thi trong Windows Explorer hoặc nhập tên

của ứng dụng trong Bắt đầu | Chạy hộp.

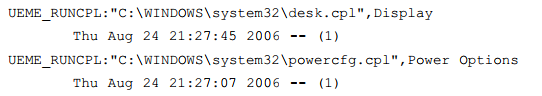
■ RUNCPL Đề cập đến việc khởi chạy một applet Bảng điều khiển.

■ RUNPIDL Một PIDL hoặc con trỏ tới danh sách ID, một phần của Explorer nội bộ không gian tên và được sử dụng để chỉ một đối tượng. Trong trường hợp các phím UserAssist, những khóa này thường là các phím tắt hoặc tệp LNK, như khi bạn chọn Bắt đầu | Các tài liệu và chọn một tập tin.

Ví dụ: hệ thống mà tôi đã viết cuốn sách này là Dell Latitude D820, được mua trong đầu tháng 8 năm 2006. Bất cứ khi nào tôi mua một hệ thống mới, tôi sẽ định dạng lại Driver cài đặt lại hệ điều hành. Đối với các hệ thống Dell, điều này có nghĩa là tôi phải tải xuống và cài đặt một số trình điều khiển (Dell giúp dễ dàng xác định vị trí các trình điều khiển cần thiết). Khi tôi chạy tập lệnh uassist.pl trên hệ thống của mình, tôi đã thấy mục sau đây



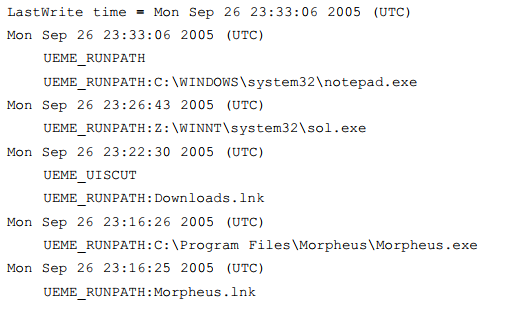
Kịch bản trả về đối tượng FILETIME theo giờ địa phương, vì vậy chúng ta có thể thấy ứng dụng đó D820\_A02\_bios.exe đã được thực thi một lần, vào ngày 7 tháng 8. Các mục khác bao gồm:

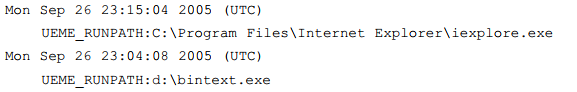


Ở đây chúng ta có thể thấy rằng các applet Bảng điều khiển tùy chọn hiển thị và nguồn điện đều có cả thực hiện vào ngày 24 tháng 8, và đó là lần duy nhất mỗi lần được đưa ra. Chỉ để cho vui thôi Ngày 4 tháng 10, vào khoảng 10:55 tối, tôi đã mở applet Display trong Bảng điều khiển và sau đó chạy lại tập lệnh Perl để thấy rằng ngày đã được đổi thành Thứ tư ngày 4 tháng 10 22:55:59 2006.

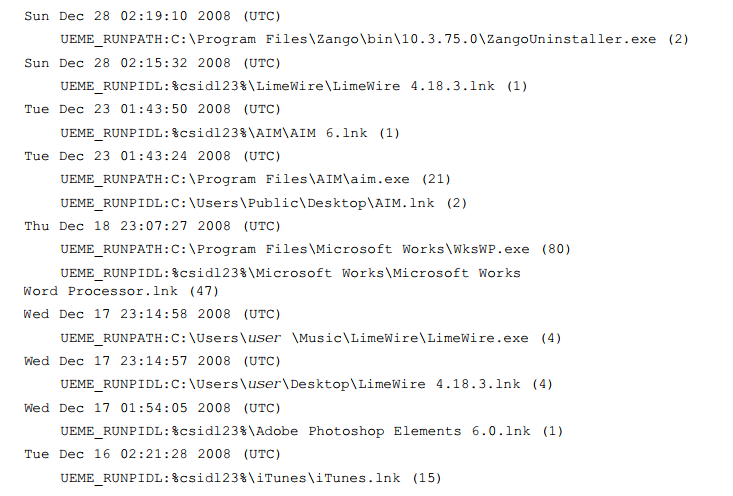
Vì vậy, về cơ bản, các phím này ghi lại số lần các ứng dụng nhất định đã được khởi chạy và lần cuối cùng hành động đó được thực hiện. Thông tin này có thể rất hữu ích khi bạn làm việc trên một trường hợp. Ví dụ: nhìn thấy một cái gì đó như UEME\_RUNCPL: timedate.cpl có thể chỉ ra rằng người dùng đã truy cập vào applet Bảng điều khiển Ngày và Giờ; có thể thay đổi thời gian hệ thống

Userassist.pl sẽ hỗ trợ bạn trong việc thu thập thông tin từ các tệp NTUSER.DAT trong một cuộc điều tra. Plugin sử dụng mô-đun Parse :: Win32Registry để truy cập vào phần thô Tệp NTUSER.DAT (mà bạn đã trích xuất từ hình ảnh thu được của mình) và xác định vị trí Khóa UserAssist chứa GUID trỏ đến ActiveDesktop. Nó nhận được LastWrite thời gian cho khóa và sau đó phân tích cú pháp thông qua khóa, trích xuất và giải mã các tên giá trị của Google. Đầu ra từ tập lệnh, theo sau, được sắp xếp dựa trên dấu thời gian được tìm thấy trong dữ liệu cho từng giá trị:





Một câu hỏi tôi nhận được nhiều lần khi tôi minh họa các công cụ này chạy với Registry Các thư mục từ các hệ thống Windows XP và 2003 là của nhưng nó có hoạt động trên Vista không? vâng, hoạt động. Ở đây, một đoạn trích của đầu ra từ tệp của người dùng từ hệ thống Vista:



Một trong những lợi ích của việc phân tích nội dung của khóa UserAssist là nó không chỉ hiển thị những hành động mà người dùng đã thực hiện (ví dụ: nhấp đúp vào biểu tượng, khởi chạy một ứng dụng thông qua menu Bắt đầu hoặc truy cập các applet Control Panel), nhưng cũng hiển thị khi những hành động này xảy ra.

Mô-đun Parse :: Win32Registry Perl cho phép dễ dàng truy cập vào nội dung nhị phân của thư mục Registry . Một số khía cạnh của Registry đã thay đổi giữa các bản phát hành khác nhau của hệ điều hành; tuy nhiên, cấu trúc nhị phân dường như không thay đổi. Các Cấu trúc và giá trị khóa của UserAssist vẫn giữ nguyên từ Windows 2000 đến Vista, nhưng một cái nhìn ngắn gọn về Windows 7 Beta vào tháng 1 năm 2009 đã cho tôi thấy rằng có là những thay đổi cần được giải quyết. Mặc dù định dạng nhị phân của thư mục Registry không thay đổi, nội dung của các khóa UserAssist có và sẽ yêu cầu phân tích và mã hóa bổ sung sẽ được thêm vào các plugin RegRipper.

Uassist.pl, một ProScript để sử dụng với ProDiscover, có sẵn trên phương tiện đi kèm cuốn sách này, trong thư mục ch4\code\Proscripts. Bạn có thể chạy ProScript này (Phiên bản 0.11, 20060522) đối với Registry khi Registry Viewer đã được điền. Kịch bản phân tích cú pháp thông qua các mục khóa đăng ký UserAssist cho tất cả người dùng và trích xuất thông tin trong một định dạng có thể đọc được, giải mã tên giá trị và phân tích số lần chạy và lần chạy cuối cùng lần từ dữ liệu, nếu áp dụng. Khi điều này được thực hiện, tập lệnh sắp xếp tất cả các giá trị với tem thời gian để thông tin có thể được sử dụng để phân tích dòng thời gian. Phiên bản này của ProScript gửi kết quả của nó tới cửa sổ lệnh trong ProDiscover, nơi điều tra viên sau đó có thể chọn, sao chép và dán các kết quả đó vào một tệp hoặc báo cáo.

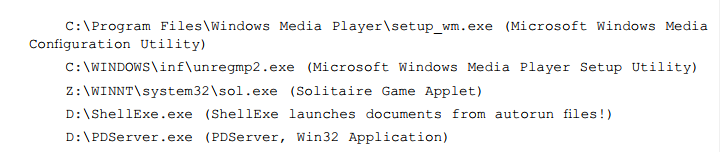
### **MUICache**

Ngoài các giá trị khóa UserAssist, còn có một cách khác để xem phần mềm nào có thể đã được chạy dưới một tài khoản người dùng trên hệ thống. Khóa MUICache nằm ở đường dẫn sau:

*Software\Microsoft\Windows\ShellNoRoam\MUICache*

Không giống như khóa UserAssist, các giá trị trong khóa MUICache không có thời gian và dữ liệu ngày liên quan đến chúng. Trong thực tế, các giá trị trong khóa MUICache có một ý kiến liên quan đến họ. Mặc dù một số giá trị, chẳng hạn như những giá trị bắt đầu bằng Có vẻ như là những mục nhập mặc định cho phần mềm có thể đã được cài đặt trên hệ thống. những cái khác dường như được tạo ra khi phần mềm được chạy trên hệ thống.

Plugin Regicipper muicache.pl trích xuất dữ liệu từ khóa MUICache trong Tập tin hive NTUSER.DAT và bỏ qua các giá trị bắt đầu với @, và hiển thị các giá trị còn lại như được minh họa ở đây:



Như bạn có thể thấy, một loạt các tên và dữ liệu giá trị được hiển thị. Một lần nữa, tài liệu nhỏ (tức là không có), đặc biệt là từ nhà cung cấp, mô tả cách các giá trị được thêm vào khóa này hoặc bất kỳ trường hợp theo đó các giá trị có thể được loại bỏ khỏi khóa. Tuy nhiên, có vẻ như để tạo giá trị trong khóa MUICache, chương trình hoặc ứng dụng phần mềm cần phải được chạy trên hệ thống bởi người dùng; do đó, nội dung của khóa này sẽ cung cấp cho nhà phân tích chỉ báo hoạt động của người dùng, ngay cả khi người dùng đã xóa ứng dụng hoặc (như được chỉ ra bởi đầu ra plugin trước) chạy ứng dụng từ phương tiện riêng biệt.

Diane Barrett thuộc University of Advancing Technology đã thực hiện công việc trong lĩnh vực tạo các ảo hóa; nghĩa là, tìm kiếm các chỉ dẫn về việc sử dụng các công nghệ ảo hóa đã được sử dụng trên các hệ thống. Trong một bài thuyết trình cụ thể, Diane tiết lộ rằng việc sử dụng các ứng dụng ảo hóa như MokaFive (bạn có thể tìm thêm thông tin tại www. mokafive.com/, cũng như tại http://en.wikipedia.org/wiki/Moka5) và MojoPac (www. mojopac.com/) có thể được nhìn thấy trong khóa MUICache. Trong một số trường hợp, các loại ứng dụng có thể được sử dụng để giữ cho các hoạt động của người dùng riêng tư, trong đó các ứng dụng trên máy chủ không được sử dụng và các ứng dụng (Trình duyệt web, v.v.) được cấu hình hoặc thiết kế riêng để không duy trì bộ đệm hoặc lịch sử hoạt động. Thật thú vị, MojoPac vẫn duy trì Đăng ký, vì vậy trong các trường hợp liên quan đến môi trường ảo hóa này, bạn muốn tìm kiếm một Thiết bị MojoPac để có được và phân tích, là tốt. Ngoài ra, hãy nhớ rằng ảo hóa khác các môi trường, chẳng hạn như Máy bảo mật ảo di động miễn phí từ MetroPipe (www.metropipe.net/pvpm.php), tồn tại và có thể được sử dụng.

### **Dánh sách MRU**

Nhiều ứng dụng duy trì danh sách MRU, đây là danh sách các tệp gần đây nhất truy cập. Trong ứng dụng đang chạy, những tên tệp này thường xuất hiện ở phía dưới của menu thả xuống khi bạn chọn Tệp trên thanh menu.

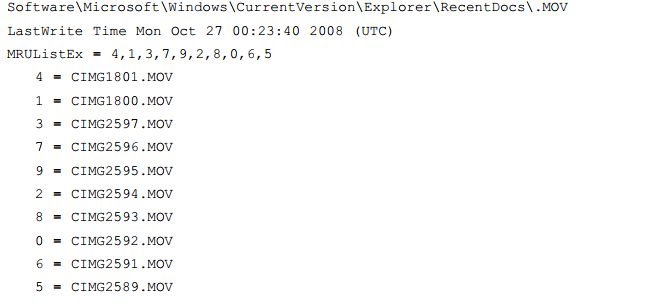
Có lẽ Khóa đăng ký danh sách MRU nổi tiếng nhất (và bao gồm tất cả) là Khóa RecentDocs

*\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs*

Khóa này có thể chứa khá nhiều giá trị, tất cả đều là kiểu dữ liệu nhị phân. Chúng tôi là quan tâm đến các giá trị có số là tên, chứa tên của các tệp được truy cập (bằng Unicode) và giá trị có tên MRUListEx, ghi lại thứ tự các tệp đã được truy cập (dưới dạng DWORD). Ví dụ: trên hệ thống của tôi, DWORD đầu tiên (tức là, giá trị 4 byte) trong dữ liệu giá trị MRUListEx là 0x26 hoặc 38 ở dạng thập phân . Giá trị với tên 38 chỉ vào một thư mục mà tôi đã mở. Cho rằng việc thêm một giá trị và dữ liệu liên quan của nó vào khóa, cũng như sửa đổi giá trị MRUListEx, cấu thành sửa đổi khóa, thời gian LastWrite của khóa RecentDocs sẽ cho chúng tôi biết khi nào tệp đó được truy cập.

Một lần nữa, để rõ ràng, các giá trị bên dưới khóa RecentDocs (và các khóa con của nó) có số như tên; tuy nhiên, chỉ những cái tên này không tương ứng với thứ tự các tập tin được tham chiếu trong dữ liệu được truy cập lần cuối. Thông tin này được duy trì trong Dữ liệu giá trị MRUListEx.

Khóa RecentDocs cũng có một số khóa con, mỗi khóa là phần mở rộng của một tệp đã được mở (.doc, .txt, .html, v.v.). Các giá trị trong các khóa con này được duy trì theo cùng một cách như trong khóa RecentDocs: Tên giá trị được đánh số và dữ liệu của chúng chứa tên của tệp được truy cập dưới dạng kiểu dữ liệu nhị phân (bằng Unicode). Một giá trị khác được gọi là MRUListEx cũng là một kiểu dữ liệu nhị phân và duy trì thứ tự các tệp được truy cập, gần đây nhất đầu tiên, dưới dạng DWORDs. Plugin Recentdocs.pl phân tích cú pháp RecentDocs nội dung chính, cũng như nội dung của từng khoá con và liệt kê các giá trị theo thứ tự MRUListEx, như được minh họa trong đoạn trích đầu ra của plugin như sau:



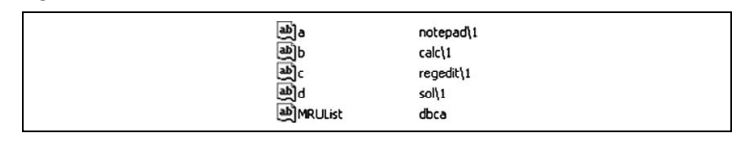


Như bạn có thể thấy, plugin hiển thị tên khóa con RecentDocs, thời gian LastWrite cho khóa, nội dung của giá trị MRUListEx (với giá trị 0xffffffff cuối cùng, chấm dứt đã xóa) và sau đó đặt tên giá trị theo thứ tự MRUListEx. Từ ví dụ này, chúng ta có thể thấy rằng người dùng đã xem tệp phim MOV vào Thứ Hai, 27 tháng 10 và nghe một MP3 cụ thể nộp thứ Sáu, ngày 11 tháng 7.

Bạn có thể tìm thấy một MRUList phổ biến khác trong khóa RunMRU:

*\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU*

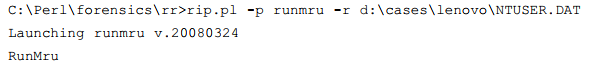
Khóa này duy trì một danh sách tất cả các giá trị được nhập vào hộp Run trên menu Start. Hình 4.19 minh họa nội dung của khóa này có thể trông như thế nào.

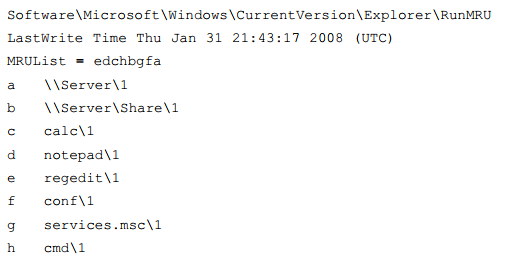


**Hình 4.19:** Trích từ khóa RunMRU

Giá trị RunMRU MRUList được duy trì ở dạng văn bản rõ ràng và dễ đọc hơn so với giá trị RecentULocs MRUListEx. Tuy nhiên, với khóa RecentDocs, hầu hết các mục được nhập gần đây được liệt kê đầu tiên trong giá trị MRUList.

Các mục nhập được thêm vào khóa này khi người dùng nhấp vào nút Bắt đầu, chọn Chạy và gõ một lệnh, tên của một tập tin hoặc tương tự. Plugin runmru.pl hiển thị thông tin trong khóa RunMRU như được hiển thị ở đây:





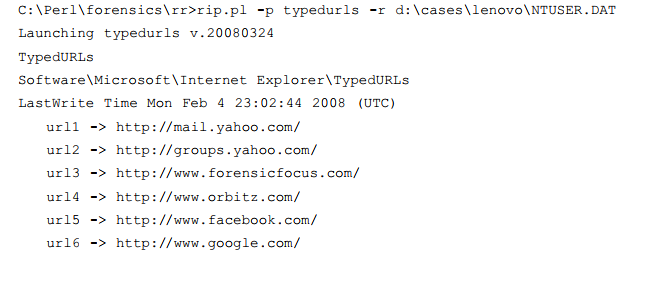
Giá trị MRUList trong khóa RunMRU cho chúng ta biết rằng mục gần đây nhất là được gõ vào hộp Run là vật phẩm E , mà trong trường hợp này là loại regedit. Khóa RunMRU tên giá trị, giống như các giá trị bên dưới khóa RecentDocs, không tương ứng với theo thứ tự các mục khác nhau được nhập vào hộp Run; đúng hơn là giá trị MRUList duy trì thông tin đó Trong ví dụ trước, giá trị của ‘h’ (tức là, cmd) không phải là mục gần đây nhất được gõ vào hộp Run; Chế độ regedit trực tiếp là mục gần đây nhất mà người dùng đã nhập vào hộp Run.

Trong quá trình điều tra xoay quanh hoạt động của người dùng hoặc nhân viên, tôi đã tìm thấy các mục trong Khóa RunMRU hiển thị quyền truy cập vào các hệ thống từ xa (chẳng hạn như các hệ thống được liệt kê ở trước ví dụ), cũng như các ứng dụng và tệp trên phương tiện lưu trữ di động.

Một khóa khác tương tự như khóa RunMRU là khóa TypedURLs:

*\Software\Microsoft\Internet Explorer\TypedURLs*

Tương tự như khóa RunMRU, khóa TypedURLs duy trì danh sách các URL mà người dùng gõ vào thanh Địa chỉ trong Internet Explorer. Tuy nhiên, tên giá trị trong Khóa TypedURL được sắp xếp với URL được nhập gần đây nhất có số thấp nhất; do đó, khóa này không có giá trị MRUList hoặc MRUListEx. Plugin typedurls.pl hiển thị thông tin từ bên trong khóa này như sau:



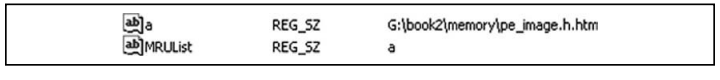


Thông tin này có thể được kết hợp với các tệp Internet tạm thời để hiển thị Web nào các trang web đã được truy cập bằng cách nhấp vào một liên kết và những trang mà người dùng đã nhập bằng tay.

Bạn có thể tìm thấy một vị trí khác cho danh sách MRU trong khóa sau:

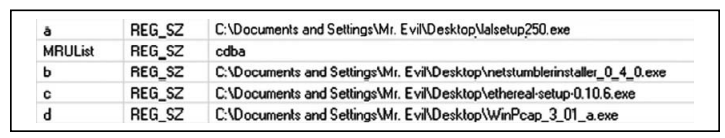
*\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU*

Khóa này (một tài liệu tham khảo tốt có sẵn tại <http://support.microsoft.com/kb/322948>) duy trì danh sách MRU của các tệp được mở thông qua hộp thoại Mở và SaveAs trong Windows vỏ. Tương tự như khóa RecentDocs, khóa OpenSaveMRU cũng duy trì các khóa con của phần mở rộng tập tin cụ thể đã được mở hoặc lưu. Tuy nhiên, giống như khóa RunMRU, dữ liệu trong các khóa này bao gồm các kiểu dữ liệu chuỗi và dễ đọc. Nội dung của nó chìa khóa có thể rất hữu ích theo nhiều cách. Đầu tiên, một số phần mở rộng tệp không xuất hiện thường xuyên trong quá trình sử dụng hệ thống bình thường, do đó, khóa con bên dưới khóa OpenSaveMRU cho tệp đó phần mở rộng có thể chỉ có một mục, được đặt tên là a. Trong trường hợp này, dữ liệu cho giá trị MRUList sẽ chỉ có một danh sách, như trong Hình 4.20. Thời gian LastWrite cho khóa sẽ cho bạn biết khi tập tin đó được mở hoặc lưu.



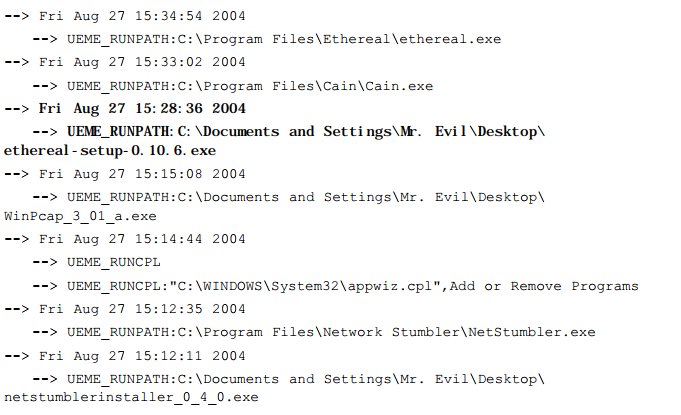
**Hình 4.20**: Trích từ một khóa con của khóa OpenSaveMRU

Mở một hình ảnh thu được từ hệ thống Windows XP trong ProDiscover, tôi đã điều hướng đến khóa OpenSaveMRU cho người dùng. Thời gian LastWrite cho khoá con exe được liệt kê là Ngày 17 tháng 8 năm 2004, lúc 11:18 sáng Hình 4.21 minh họa nội dung của khóa con exe.



**Hình 4.21**: Trích từ exe Khóa con của Khóa OpenSaveMRU (thông qua ProDiscover)

Hình 4.21 cho thấy tệp gần đây nhất được truy cập là tiện ích thiết lập Ethereal (Ethereal là một bộ công cụ phân tích và thu thập lưu lượng mạng, hiện được gọi là Wireshark), được sử dụng để cài đặt ứng dụng. Thông tin này sau đó có thể được tương quan với nội dung của khóa UserAssist (sử dụng uassist.pl ProScript, Phiên bản 0.11, 20060522). Khi ProScript là chạy với một hình ảnh thử nghiệm, chúng tôi thấy như sau:



Như bạn có thể thấy, người dùng của chúng tôi đã khá bận rộn vào ngày 27 tháng 8 năm 2004. (ProScript trích xuất thô Dữ liệu FILETIME và chuyển nó thành thời gian UTC, tương đương với thời gian GMT. Ứng dụng ProDiscover hiển thị tất cả các lần liên quan đến hệ thống examiner và và Cài đặt TimeZoneIn information.) Khóa The TimeZoneInformation Registry hiển thị khi hệ thống đang chạy, nó được đặt cho múi giờ miền Trung, với các điều chỉnh tự động cho tiết kiệm thời gian ban ngày kích hoạt. ActiveTimeBias là 300 phút (năm giờ) và ActiveTimeBias trên hệ thống của tôi (với ProDiscover mở) là 240 phút (bốn giờ). Từ thông tin này, chúng tôi thấy rằng khoảng 10 phút sau khi lưu ứng dụng thiết lập Ethereal vào hệ thống của anh ấy, người tình nghi đã cài đặt ứng dụng đó

Một cách khác thông tin này có thể hữu ích là hiển thị việc sử dụng bộ nhớ ngoài thiết bị. Không chỉ thời gian LastWrite của khóa con sẽ cung cấp ngày và giờ cho thiết bị đã được kết nối với hệ thống, nhưng thông tin cũng có thể tương quan với nội dung của khóa MountedDevices để cung cấp thêm thông tin về thiết bị.

Một danh sách MRU khác có thể được tìm thấy bên dưới khóa sau:

*Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts*

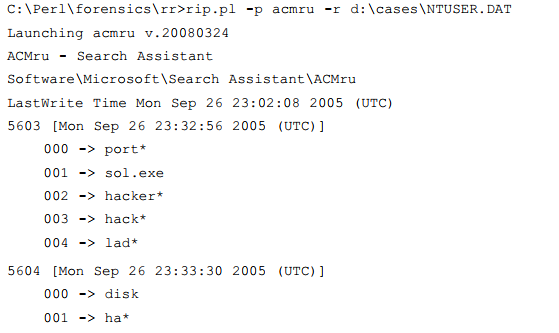
Các khóa con bên dưới khóa này tương ứng với các tiện ích mở rộng cho các tệp đã được mở trên hệ thống. Bên dưới các khóa con mở rộng tệp, bạn sẽ tìm thấy các khóa con có tên OpenWithProgIDs và OpenWithList. Các mục đăng ký này cho hệ thống biết nên sử dụng ứng dụng nào để mở tệp có phần mở rộng đó khi người dùng nhấp đúp vào tệp. Điều này có thể quan trọng trong kiểm tra vì người dùng có thể đã cài đặt và sau đó gỡ cài đặt một ứng dụng sẽ mở các tệp có phần mở rộng cụ thể. Khi ứng dụng bị xóa, thông tin có thể vẫn còn trong khóa FileExts. Các plugin fileexts.pl RegRipper phân tích cú pháp và hiển thị cái này thông tin.

### **Trợ lý tìm kiếm**

Khi người dùng nhấp vào nút Bắt đầu trong Windows XP và chọn Tìm kiếm, sau đó chọn Đối với Tệp và Thư mục, các cụm từ tìm kiếm được nhập vào hộp thoại được lưu trữ trong sau khóa đăng ký

*Software\Microsoft\Search Assistant\ACMru*

Khóa ACMru thường sẽ có một số kết hợp của bốn khóa con: 5001, 5603, 5604 và 5647. Khóa con 5001 chứa danh sách MRU cho Trợ lý tìm kiếm Internet, Khóa con 5603 chứa danh sách MRU cho tìm kiếm tệp và thư mục Windows XP và Khóa con 5604 chứa danh sách MRU tương ứng với từ hoặc cụm từ trong một hộp thoại tập tin cái hộp. Khóa con 5647 duy trì danh sách MRU cho các máy tính được nhập thông qua các máy tính dành cho máy tính hoặc mọi người lựa chọn trong hộp thoại Kết quả tìm kiếm. Tên giá trị trong các khóa con là số có ba chữ số. Số nhỏ nhất (ví dụ: 000) đại diện cho tìm kiếm gần đây nhất và thời gian LastWrite được liên kết với khóa sẽ cho bạn thời gian và ngày tìm kiếm ra mắt. Plugin acmru.pl hiển thị thông tin trong khóa này, như được minh họa ở đây:



Biết được mục đích của các khóa con khác nhau và cách chúng được sinh ra sẽ cho bạn có cái nhìn sâu sắc về các hoạt động của người dùng trên hệ thống. Điều này có thể hữu ích trong quá trình điều tra mối quan tâm đó là những gì người dùng đã làm và khi nào. Trong ví dụ trước, người dùng đã tìm kiếm cho các thuật ngữ khác nhau như cổng \* và sol.exe, tìm kiếm tên tệp và tìm kiếm đĩa dưới dạng từ khóa trong tập tin. Trong một cuộc kiểm tra về sự thỏa hiệp hệ thống mà kẻ xâm nhập đã có truy cập hệ thống thông qua Terminal Services, chúng tôi thấy rằng ông đã tìm kiếm thuật ngữ ngân hàng \*.

Thông tin tìm kiếm cho các hệ thống di sản của các hệ thống khác, như Windows 2000, được duy trì ở các mức khác nhau Khóa Registry và có thể được tìm thấy trên hệ thống nếu được nâng cấp từ Windows 2000 lên XP. Chìa khóa trong câu hỏi là:

*Software\Microsoft\Internet Explorer\Explorer Bars\{C4EE31F3-4768-11D2-BE5C-00A0C9A83DA1}*

Theo nội dung của khóa HKEY\_CLASSES\_ROOT \ CLSID, GUID đó đề cập đến Band Search Explorer Band, có trong shell32.dll. Hai khóa con bên dưới khóa này, FilesNamedMRU và ContainedTextMRU, tương quan với các khóa con 5603 và 5604 (tương ứng) được tìm thấy trên các hệ thống Windows XP.

### **Connecting to Other Systems**

Khi người dùng sử dụng Map Network Drive Wizard (nhấp chuột phải vào biểu tượng Máy tính của tôi và chọn Map Network Drive) để kết nối với hệ thống từ xa, danh sách MRU được tạo bên dưới khóa sau:

*Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU*

Cho dù người dùng sử dụng Trình hướng dẫn ổ đĩa mạng hay lệnh sử dụng mạng, khối lượng người dùng thêm vào hệ thống sẽ xuất hiện trong khóa sau

*Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2*

Như đã đề cập trước đó, các khóa con MountPoints2 xuất hiện dưới dạng GUID có thể được ánh xạ vào các mục \ ?? \ Volume trong phím MountedDevices. Những GUID này cũng có thể được ánh xạ đến khóa con CPC \ Volume bên dưới phím MountPoints2.

Tôi đã sử dụng lệnh sử dụng mạng trên hệ thống của mình để thực hiện kiểm tra và khi tôi kết nối với chia sẻ C $ trên một hệ thống khác, tôi thấy các khóa con như ## 192.168.1.22 # c $ và ## 192.168.1.71 # c $.

Những địa chỉ IP này (Tôi có một mạng kiểm tra phẳng không phải là một miền, vì vậy máy tính tên về cơ bản là các địa chỉ IP) cũng xuất hiện trong khóa Registry sau:

*Software\Microsoft\Windows\CurrentVersion\Explorer\ComputerDescriptions*

Khóa này duy trì các mô tả về các máy tính được trình duyệt mạng nhìn thấy. Đối với các hệ thống là một phần của tên miền, việc thấy một số tên máy tính được liệt kê là bình thường trong khóa này Tuy nhiên, đối với các hệ thống độc lập, chẳng hạn như người dùng gia đình và các hệ thống khác không phải là một phần của tên miền, bạn có thể đã giành được các giá trị được liệt kê cho khóa này. Trên hệ thống nhà của tôi, chỉ các hệ thống mà tôi đã kết nối rõ ràng với việc sử dụng lệnh net use xuất hiện trong khóa này. Tôi sử dụng hệ thống làm việc của mình để kết nối với chủ nhân của mình, mạng nội bộ thông qua mạng riêng ảo (VPN) và một số hệ thống mà tôi đã kết nối sẽ xuất hiện trong Mô tả máy tính Chìa khóa. Một người đặc biệt có mô tả Samba 2.2.7a, chỉ ra rằng đó là một hệ thống Linux chạy Samba. Vì khóa này được tìm thấy trong tệp NTUSER.DAT, nên có thể có các mục khác nhau cho người dùng khác nhau trên cùng một hệ thống.

### **Ghi đĩa CD**

Khi xử lý sự cố trong đó người dùng có thể đã xóa dữ liệu khỏi hệ thống, ở đó là một số con đường để xem xét. Người dùng có sao chép tệp vào bộ lưu trữ di động USB không thiết bị (ví dụ: ổ ngón tay cái, iPod, v.v.) hoặc người dùng đã gửi dữ liệu đó dưới dạng tệp đính kèm tới E-mail dựa trên web (Yahoo!, Gmail)? Một phương tiện khai thác dữ liệu khác cần xem xét là XP tích hợp khả năng ghi đĩa CD. Mặc dù nhiều hệ thống đi kèm với ghi đĩa CD và DVD phần mềm được cài đặt hoặc có sẵn (ví dụ: các hệ thống Dell và Lenovo ThinkPad của tôi đi kèm với Đã cài đặt các sản phẩm Sonic / Roxio), Windows XP và Vista có khả năng tích hợp để ghi Băng đĩa. Đối với Windows XP, khả năng này được mô tả trong bài viết Cơ sở tri thức Microsoft 279157 (http://support.microsoft.com/kb/279157). Khi người dùng chèn một đĩa CD-R trống hoặc CD-RW vào hệ thống, một hộp thoại sẽ mở ra, cho anh ta cơ hội mở thư mục CD có thể ghi bằng Windows Explorer. Khi thư mục đó mở, anh ta có thể kéo tập tin và các thư mục vào thư mục được sao chép vào khu vực tổ chức đặc biệt cho đến khi anh ta sẵn sàng chọn Ghi các tệp này vào CD. Khi người dùng sẵn sàng ghi các tệp vào CD, một khối nguyên khối tập tin hình ảnh đĩa có tên là Cd Cd ghi stash file.bin 'được tạo trong khu vực tổ chức. Điều đó thật đặc biệt khu vực tổ chức nằm trong thư mục sau trên các hệ thống XP (theo mặc định):

*%USERPROFILE%\Local Settings\Application Data\Microsoft\CD Burning*

Trên các hệ thống Vista, khu vực tổ chức mặc định là:

*%USERPROFILE%\AppData\Local\Microsoft\Windows\Burn\Burn*

Vị trí khu vực tổ chức được liệt kê và duy trì trong thư mục người dùng NT NTERER.DAT trong khóa sau:

*Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders*

Plugin shellRers.pl RegRipper trích xuất và liệt kê tất cả các tên giá trị và tên của chúng dữ liệu từ khóa này. Người dùng có thể thay đổi vị trí của thư mục Ghi đĩa CD bằng cách chỉnh sửa giá trị đăng ký và cung cấp một vị trí khác. Nếu đường dẫn thư mục đã được thay đổi, điều này có thể cho nhà phân tích biết rằng chính sách của công ty đã được áp dụng hoặc người dùng có một số trình độ thành thạo kỹ thuật. Điều này cũng có thể chỉ ra rằng các nhà phân tích sẽ cần phải tìm kiếm các tạo phẩm tệp .bin trong thư mục khác.

### **IM và P2P**

Các ứng dụng chia sẻ tệp IM và P2P đang cực kỳ phổ biến, một sự phổ biến dường như xuyên qua mọi thế hệ. Nơi mọi người từng viết thư mất thời gian để viết và để đến người nhận, một email nhanh chóng có thể được gửi và sẽ đợi người đó vào lần tiếp theo Anh ấy đăng nhập. Hoặc bạn có thể cách xa nửa thế giới và nhận được thông báo ngay lập tức bạn bè đăng nhập vào ứng dụng IM của cô ấy. Hoặc, bằng cách sử dụng chia sẻ tệp P2P, bạn có thể tìm thấy bất kỳ số lượng các tập tin hữu ích (hoặc có lẽ không hữu ích) Âm nhạc, phim ảnh, hình ảnh, v.v. Sự phổ biến của các ứng dụng này đã tạo ra sự phổ biến của các khung và ứng dụng khách khác nhau. Yahoo!, AOL và Microsoft đều có các ứng dụng khách IM riêng, mỗi ứng dụng đều có ứng dụng khách riêng. chức năng và dấu chân pháp y độc đáo trên một hệ thống. Trên hết, bạn có thể sử dụng nhiều các ứng dụng của bên thứ ba để thay thế các máy khách đó hoặc thậm chí kết hợp chúng vào một giao diện duy nhất. Ví dụ: Trillian (trillian.cc) cho phép người dùng kết hợp các danh tính IM khác khác thành một đơn ứng dụng, vì vậy họ chỉ phải đăng nhập vào một giao diện duy nhất để truy cập nhiều mạng IM.Meebo.com cung cấp một giao diện dựa trên Web tương tự.

Loại phổ biến này cũng đúng cho các mạng P2P. Mỗi người có một nét độc đáo riêng. những thách thức khi phân tích pháp y. Ví dụ, làm thế nào để một điều tra viên xác định người mà nghi phạm đã trò chuyện (trên IM) nếu ứng dụng không ghi nhật ký cuộc hội thoại theo mặc định? Hoặc làm thế nào một điều tra viên xác định liệu một cuộc trò chuyện đã lưu là kết quả của người dùng đặc biệt lưu cuộc hội thoại hoặc kết quả của tiện ích bổ sung của bên thứ ba để đăng nhập hội thoại? Về P2P, làm thế nào để một điều tra viên xác định thuật ngữ tìm kiếm một nghi phạm được sử dụng, những tập tin được lấy từ mạng chia sẻ và từ các tập tin có nguồn gốc từ đâu?

Sự thay đổi của khách hàng cho cả IM và P2P lớn đến mức họ sẽ yêu cầu cuốn sách riêng để giải quyết đầy đủ các phân tích pháp y của từng. Khi bạn xem xét như thế khác các ứng dụng, ứng dụng khách IM và P2P sẽ thay đổi giữa các phiên bản, bao gồm chức năng mới và tạo các khóa và tệp Registry mới, vấn đề lập danh mục các tài liệu pháp y của các ứng dụng này càng trở nên nan giải hơn. Ví dụ: khi tôi đang sử dụng các phiên bản cũ hơn của ứng dụng khách nhắn tin tức thời AOL (AIM), có một bộ khóa Registry cụ thể trong hồ sơ người dùng mà bạn có thể truy cập và xem mật khẩu được mã hóa của người dùng. Đây là kết quả của việc người dùng chọn tự động đăng nhập vào mạng AIM mà không phải gõ lại mật khẩu của anh ấy. Nếu, như một phần của cuộc điều tra, bạn thấy cần phải thu thập thông tin về hoạt động này của người dùng trên AIM, bạn có thể sử dụng mật khẩu được mã hóa đó để thiết lập hồ sơ tương tự trên một hệ thống khác, sau đó đăng nhập như người dùng đó. Tôi quyết định dùng thử AIM mới Triton client cách đây một thời gian và nó hoạt động rất tốt, mặc dù phải mất một chút để làm quen. Một thay đổi giao diện chính là thay vì một cửa sổ khách khác được mở đối với mỗi cuộc hội thoại, mỗi cửa sổ hiện được gắn thẻ trong một cửa sổ duy nhất. Từ một pháp y tuy nhiên, bây giờ tôi mở RegEdit và không có mục nào cho AOL hoặc AIM bên dưới thư mục HKEY\_CURRENT\_USER \Software .

Để làm cho vấn đề tồi tệ hơn, không có nỗ lực đã được thực hiện để công khai danh mục các hiện vật này. Trong những năm qua, các nhà điều tra pháp y và thực thi pháp luật đã gặp phải tình huống yêu cầu họ phân tích các cổ vật IM và P2P, tuy nhiên, đã có bất kỳ nỗ lực nào để phát triển cơ sở dữ liệu hoặc Wiki trực tuyến cho các mục này. Đây là một lĩnh vực nghiên cứu cần được phát triển.

## **Windows XP System Restore Points**

Windows XP bao gồm một thứ gọi là System Restore, duy trì một loạt khôi phục điểm để hệ thống của bạn trở nên không sử dụng được hoặc bắt đầu hoạt động kỳ quặc, bạn có thể cuộn trở lại hệ thống về cấu hình trước đó, khi nó hoạt động bình thường. Tôi dễ dàng thừa nhận đã phải làm điều này nhiều lần bản thân mình. Thỉnh thoảng tôi lại làm có nghĩa là cài đặt trên máy tính) một cái gì đó kết thúc khiến hệ thống của tôi bắt đầu có sự phù hợp. Hoặc là cài đặt có thể chỉ đơn giản là một sự trùng hợp ngẫu nhiên. Có khả năng để quay trở lại một ngày để tôi biết rằng hệ thống đã hoạt động tốt là rất tốt. Tôi chắc chắn rằng nhiều người dùng khác có tìm thấy điều tương tự là đúng

Đây là một tiện ích cực kỳ hữu ích cho người dùng cũng như cho các nhà điều tra pháp y. Rốt cuộc, Ở đây, một cơ sở hoạt động ở chế độ nền mà không có kiến thức về người dùng, âm thầm tạo ra sao lưu thông tin cấu hình hệ thống quan trọng. Registry được tạo dựa trên một số kích hoạt nhất định, chẳng hạn như khi các ứng dụng hoặc trình điều khiển không dấu được cài đặt hoặc trong khi Cài đặt AutoUpdate. Registry có thể được tạo thủ công và Khôi phục hệ thống dịch vụ cũng tạo các Registry mỗi ngày một lần theo mặc định.

Để hiểu rõ hơn về các Registry hệ thống hữu ích như thế nào đối với phân tích pháp y, bạn cần hiểu một chút về cách thức System Restore hoạt động: cái gì được sao lưu, cái gì không được sao lưu và những gì các khóa Registry kiểm soát cách thức System Restore hoạt động.

Danh sách tiện ích Khôi phục hệ thống khôi phục các mục sau:

■ Registry

■ Hồ sơ địa phương (không chuyển vùng)

■ Cơ sở dữ liệu COM +

■ Bộ đệm DLL bảo vệ tệp Windows

■ Cơ sở dữ liệu WMI

■ Cơ sở dữ liệu máy chủ thông tin Internet (IIS)

■ Các tệp có phần mở rộng được liệt kê trong phần <bao gồm> của Tệp được theo dõi mở rộng trong phần Khôi phục hệ thống của SDK nền tảng

Khôi phục hệ thống không khôi phục như sau:

■ Cài đặt DRM

■ Thư mục SAM

■ Cài đặt WPA (Thông tin xác thực Windows không được khôi phục)

■ Các thư mục / tệp cụ thể được liệt kê trong danh sách Phần mở rộng tệp được giám sát trong Hệ thống phần khôi phục của SDK nền tảng

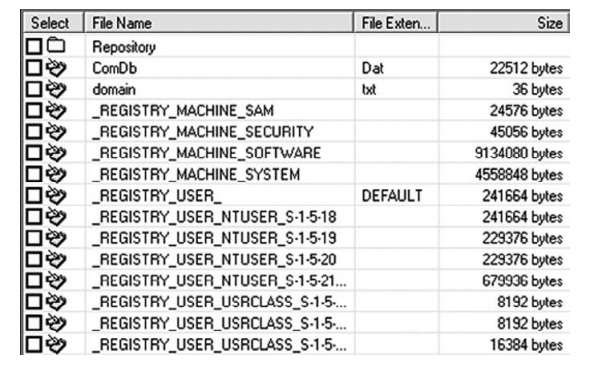
■ Mọi tệp có phần mở rộng không được liệt kê là <bao gồm> trong Tệp được theo dõi danh sách tiện ích mở rộng trong phần Khôi phục hệ thống của SDK nền tảng

■ Dữ liệu do người dùng tạo được lưu trữ trong hồ sơ người dùng

■ Nội dung của các thư mục được chuyển hướng

Điều quan trọng cần lưu ý là mặc dù dịch vụ Khôi phục Hệ thống không khôi phục thư mục SAM, dù sao thì nó cũng hỗ trợ cho nó ít nhất là một phần của nó. Kiểm tra nội dung khôi phục điểm, bạn sẽ thấy các bản sao của thư mục SAM được sao lưu, cùng với các tệp Registry khác.

Đối với mục đích của chương này, chúng tôi quan tâm nhất đến Registry hệ thống bởi vì chúng chứa các bản sao lưu của một số tệp Registry nhất định, chẳng hạn như NTUSER.DAT, SYSTEM,SOFTWARE, and SAM Hình 4.22 minh họa nội dung của thư mục ảnh chụp của một Registry , như thể hiện trong ProDiscover.



**Figure 4.22** Trích từ ProDiscover Hiển thị thư mục ảnh chụp Registry

Như bạn có thể thấy, từ góc độ phân tích Registry, Khôi phục Hệ thống sao lưu khá một chút thông tin rất hữu ích Các tập tin Registry được sao lưu đến các Registry chỉ là một tỷ lệ phần trăm kích thước của những thứ được tìm thấy trong thư mục system32 \ config, nhưng chúng vẫn có thể cung cấp cho một điều tra viên cái nhìn sâu sắc có giá trị về cấu hình của hệ thống tại các điểm trong quá khứ.

Chúng tôi sẽ giải quyết các tệp mà dịch vụ Khôi phục Hệ thống sao lưu trong Chương 5. Trong phần này Chương này, chúng tôi sẽ tập trung chú ý vào các tệp Registry.

Các kỹ thuật phân tích của chúng tôi, đặc biệt là sử dụng các công cụ như Trình phân tích đăng ký ngoại tuyến, là cũng hiệu quả với các tệp Registry nằm trong các Registry như với các tệp thô Các tệp đăng ký mà chúng tôi tìm thấy trong thư mục system32 \ config. Trong thực tế, nhiều chìa khóa và các giá trị chúng ta đã thảo luận trong chương này cũng được tìm thấy trong các bản sao lưu Registry của tệp Registry. Điều này cho phép điều tra viên nhìn vào quá khứ và xem một số cài đặt cấu hình và cài đặt phần mềm trên hệ thống tại thời điểm đó.

Mặc dù vậy, một số cảnh báo về Khôi phục Hệ thống theo thứ tự. Theo mặc định, Khôi phục hệ thống yêu cầu 200 MB dung lượng đĩa trống trên hệ thống. Nếu yêu cầu không gian này không được đáp ứng, dịch vụ Khôi phục Hệ thống sẽ không hoạt động cho đến khi không gian đó khả dụng. Điều này thực tế có thể quan trọng trong quá trình điều tra nếu bạn không thấy các Registry bạn sẽ mong đợi để xem trên hệ thống. Một số nhà điều tra có thể nghi ngờ rằng ai đó có thể truy cập thư mục System Volume Information và cố ý xóa các Registry trong khi thực tế chúng chưa được tạo ra.

Một điều khác cần lưu ý khi làm việc với các Registry là SYSTEM\  
ControlSet00x\Control\BackupRestore cũng đóng vai trò quyết định cái gì là và dịch vụ nào được sao lưu hoặc khôi phục bởi dịch vụ Khôi phục hệ thống. Khóa này có ba khóa con (AsrKeysNotToRestore, FilesNotToBackup và KeysNotToRestore) khá tự lập. Bên dưới mỗi khóa con, bạn sẽ thấy một danh sách các khóa hoặc tệp Registry (trong trường hợp các tệp, bạn sẽ thấy các tiện ích mở rộng được liệt kê bằng ký tự đại diện, nghĩa là tất cả các tệp có tiện ích mở rộng đó). Những các giá trị và dữ liệu của chúng cũng có thể có ảnh hưởng đến những gì điều tra viên nhìn thấy hoặc có quyền truy cập vào trong một cuộc điều tra sau khi chết.

Thông tin cấu hình Khôi phục Hệ thống (http://support.microsoft.com/kb/295659) được duy trì trong khóa Registry sau:

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\SystemRestore*

Một số giá trị quan trọng nằm bên dưới khóa này. Giá trị RPGlobalInterval chỉ định cách Registry thường được tạo ra. Giá trị mặc định là 86400, cho XP biết để tạo khôi phục điểm mỗi ngày theo lịch (60 giây × 60 giây / giờ × 24 giờ / ngày = 86400 hoặc một ngày theo lịch). Nếu giá trị DisableSR được đặt thành 1, chức năng Khôi phục hệ thống đã bị tắt. Theo mặc định, giá trị này được đặt thành 0. Giá trị RPLifeInterval chỉ định thời gian khôi phục điểm sẽ là bao lâu được giữ lại (7776000 giây = 90 ngày).

Một cách đơn giản để truy cập thông tin về Khôi phục hệ thống trên Windows XP trực tiếp hệ thống thông qua SystemRestore ([http://msdn.microsoft.com/en-us/l Library / aa378951.aspx](http://msdn.microsoft.com/en-us/l%20Library%20/%20aa378951.aspx)) và SystemRestoreConfig (http://msdn.microsoft.com/en-us/l Library / aa378955.aspx) WMI các lớp học. Kịch bản sr.pl Perl trên DVD đi kèm cung cấp một ví dụ về cách các các lớp học có thể được sử dụng. Tập lệnh Perl sẽ truy xuất cài đặt cấu hình Khôi phục hệ thống (về cơ bản, các giá trị Registry) có thể truy cập được thông qua lớp WMI SystemRestoreConfig và hiển thị thông tin về từng Registry (nghĩa là số thứ tự, ngày tạo, và chuỗi mô tả lý do tại sao Registry được tạo). Thông tin có sẵn thông qua lớp SystemRestore có thể được truy xuất từ các tệp trong các Registry (nghĩa là, tệp rp.log) và sẽ được giải quyết trong Chương 5.

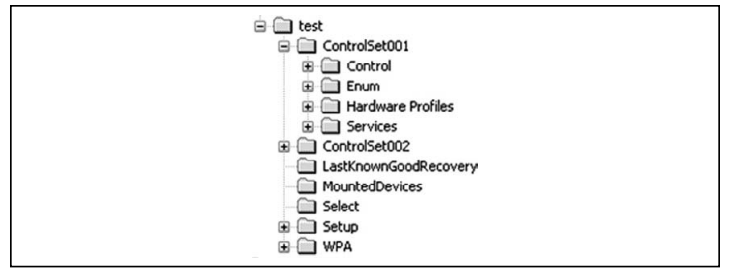
Biết điều này, làm thế nào các tệp Registry trong các Registry hữu ích từ một Quan điểm điều tra? Các thư mục Registry được duy trì trong các Registry chứa nhiều thông tin giống như những gì có trên hệ thống sống. Nếu bạn không có một hệ thống hình ảnh có sẵn và muốn xem những tập tin này trông như thế nào, tải xuống một bản sao của psexec.exe từ Microsoft sang hệ thống Windows XP, sau đó nhập lệnh psexecTHERs cmd. Điều này mở một dấu nhắc lệnh chạy dưới dạng HỆ THỐNG, được yêu cầu để truy cập vào thư mục Thông tin khối lượng hệ thống do quyền NTFS. Thay đổi thư mục thành Thư mục thông tin khối lượng hệ thống bằng cách gõ:

*cd \sys\**

Từ đó, tiến tới thư mục con giữ các Registry :

*cd \_restore\**

Tại thời điểm này, nếu bạn yêu cầu một danh sách thư mục, bạn sẽ thấy một số Registry , được liệt kê dưới dạng tên thư mục bắt đầu bằng RP. Nếu bạn cd vào một trong những thư mục này và sau đó một lần nữa vào thư mục con snapshot, bạn sẽ thấy các tệp Registry. Từ đây, bạn có thể sao chép bất kỳ của các tập tin này để thư mục khác để phân tích. Một cách để xem thông tin trong Các thư mục RegEdit là mở RegEdit và chọn thư mục HKEY\_USERS. Bấm vào Tệp | Tải tệp và sau đó điều hướng đến một trong các tệp bạn đã sao chép từ Registry . Khi được hỏi, hãy đặt tên cho thư mục, chẳng hạn như **test hive** hoặc **test system** (nếu bạn sử dụng Tập tin hệ thống). Hình 4.23 minh họa một hệ thống thư mục được tải theo cách này.



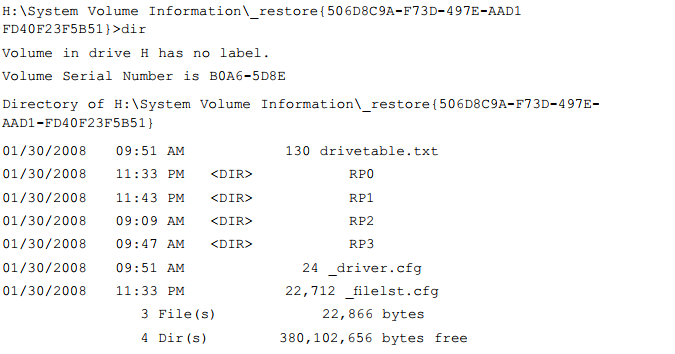
**Hình 4.23** Hệ thống thư mục từ Registry được tải trong RegEdit

Từ đây, bạn có thể xem nội dung của các khóa và thậm chí chạy các công cụ chống lại Registry để trích xuất các giá trị và dữ liệu, giống như bạn làm với hệ thống trực tiếp. Bạn cũng có thể xuất các giá trị từ bên trong thư mục.

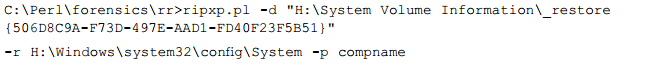
Một cách khác để làm điều này là sử dụng một trong các công cụ RegRipper được đề cập trước đó trong chương, được gọi là ripxp.pl. Đây là phiên bản cụ thể của rip.pl được viết để hoạt động với Windows XP khôi phục điểm tìm thấy trong hình ảnh thu được. Ripxp.pl hoạt động bằng cách kiểm tra một thư mục cụ thể tệp, chạy một plugin chống lại nó (tại thời điểm này, ripxp.pl chỉ chạy một plugin tại một thời điểm, như chạy một số plugin được liệt kê trong tệp plugin có thể dẫn đến việc hiển thị quá nhiều thông tin), sau đó truy cập các Registry và chạy cùng một plugin thư mục tương ứng nằm trong các Registry đó. Nó thực hiện tất cả điều này tự động để giảm khả năng mắc lỗi và tăng hiệu quả của nhà phân tích.

Để chạy ripxp.pl, một số thiết lập được yêu cầu. Đầu tiên, gắn hình ảnh thu được của bạn dưới dạng chỉ đọc hệ thống tệp, sử dụng bất kỳ phần mềm miễn phí nào (VDKWin hoặc ImDisk) hoặc thương mại (Smart Mount hoặc các tiện ích Mount Image Pro) có sẵn để thực hiện việc này; xem Chương 5 để biết thêm thông tin về những công cụ này. Trong ví dụ này, tôi đã gắn một hình ảnh thu được trên hệ thống thử nghiệm của mình là H: \. Tiếp theo, mở một dấu nhắc lệnh và gõ lệnh **psexec –s cmd** như được mô tả trước đó; một lần nữa, điều này là bắt buộc để truy cập các tệp Registry trong Thông tin khối lượng hệ thống danh mục.

Tại thời điểm này, có thể là một ý tưởng tốt để mở một dấu nhắc lệnh khác, sử dụng Lệnh psexec.exe đã được sử dụng, sau đó điều hướng đến thư mục Registry trong hình ảnh được gắn kết, sử dụng các lệnh được liệt kê trước đó trong chương này. Tại đây điểm, đầu ra của lệnh dir trên ảnh thử nghiệm của tôi trông như sau (một số dòng sẽ bọc):

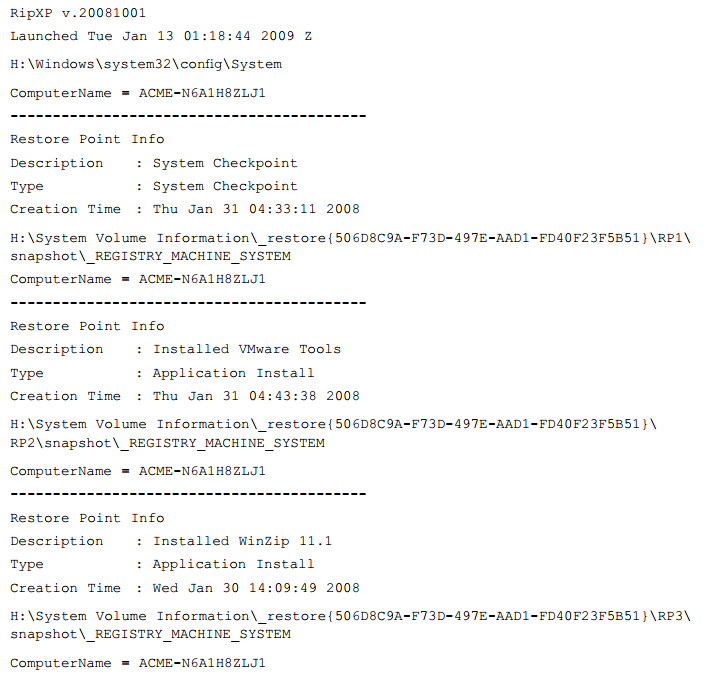


Vì vậy, có vẻ như chúng ta có ba Registry trong ảnh được gắn. Đi ngay quay lại dấu nhắc lệnh đầu tiên mà chúng tôi đã mở và điều hướng đến thư mục nơi bạn lưu trữ ripxp.pl và thư mục plugin. Gõ chỉ ripxp.pl (hoặc ripxp, nếu bạn sử dụng phiên bản dịch được biên dịch trên nền tảng Windows chạy trên Windows) sẽ hiển thị cho bạn cú pháp của công cụ CLI. Chúng tôi sẽ bắt đầu bằng cách chạy một truy vấn đơn giản cho tên máy tính, sử dụng compname.pl cắm vào. Lệnh bạn sẽ cần nhập sẽ xuất hiện như sau:



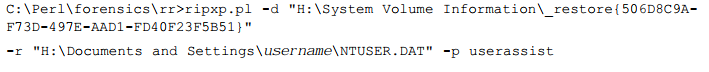
Như bạn có thể thấy, nó có lẽ là một ý tưởng tốt để dán đường dẫn hoàn chỉnh đến Registry thư mục từ cửa sổ nhắc lệnh này sang cửa sổ nhắc lệnh khác hoặc vào tài liệu Notepad cho sử dụng sau. Lệnh này khá dài, thậm chí chỉ có ba yếu tố. Như đường dẫn đến Khôi phục thư mục chứa khoảng trắng, chúng ta phải đặt đường dẫn trong dấu ngoặc kép.

Điều mà thực sự thú vị là đầu ra của lệnh, xuất hiện như sau:

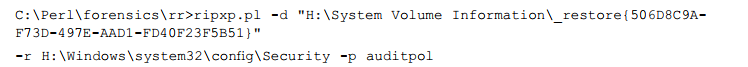


Như đã thảo luận, ripxp.pl bắt đầu bằng cách chạy plugin đã chọn (compname.pl) dựa vào thư mục được chọn (nghĩa là, thư mục hệ thống, nằm trong thư mục mặc định). Sau đó nó truy cập từng điểm khôi phục, phân tích tệp rp.log (có thêm thông tin về định dạng của tệp đó trong Chương 5), hiển thị thông tin về điểm khôi phục (nghĩa là, lý do và ngày cho tạo), và sau đó chạy plugin dựa vào thư mục thích hợp (ripxp.pl chứa mã cho phép nó xác định loại thư mục cụ thể).

Bạn có thể chạy ripxp.pl không chỉ là thư mục Dhệ thống. Ví dụ, bạn có thể sử dụng lệnh sau để chạy plugin userassist.pl đối với tệp NTUSER.DAT, cũng như tất cả các tệp NTUSER.DAT tương ứng trong các điểm khôi phục:



Hoặc bạn có thể sử dụng một lệnh như sau để chạy plugin audpol.pl tất cả các thư mục bảo mật, cả trong thư mục system32 \ config và trong các điểm khôi phục:



Đến bây giờ, rõ ràng là bạn có thể truy xuất rất nhiều dữ liệu lịch sử từ Điểm khôi phục hệ thống Windows XP. Như một ví dụ về lý do tại sao bạn có thể muốn sử dụng một công cụ chẳng hạn như ripxp.pl, giả sử bạn đang điều tra một trường hợp và bạn nghi ngờ rằng đó là một chương trình phần mềm đã bị xóa khỏi hệ thống. Thiết lập hệ thống phân tích của bạn như được mô tả trước đó và sử dụng ripxp.pl, bạn có thể sử dụng plugin compname.pl để xác định khi nào một hệ thống Tên đã được thay đổi hoặc sử dụng plugin muicache.pl để xác định thời điểm thực thi có thể lần đầu tiên được chạy trên một hệ thống. Ripxp.pl là một công cụ cực kỳ mạnh mẽ sử dụng tất cả cùng các plugin được sử dụng bởi RegRipper và rip.pl và có thể được sử dụng để xem dữ liệu lịch sử từ hệ thống, và thậm chí có thể quan sát những thay đổi xảy ra theo thời gian. Nói, ví dụ, rằng bạn nghi ngờ người dùng bị vô hiệu hóa kiểm toán hoặc ngủ đông trong một khoảng thời gian cụ thể, hoặc xóa một số mục quan trọng khỏi thư mục NTUSER.DAT của mình. Bạn có thể nhanh chóng và dễ dàng có được một cái nhìn vào dữ liệu đó, thêm cái nhìn sâu sắc vào bài kiểm tra của bạn.

### **Chuyển hướng**

Khi một nhà phân tích bắt gặp các phiên bản 64 bit của hệ điều hành Windows, một điều cần lưu ý là hệ điều hành sử dụng một thứ gọi là Chuyển hướng trên mạng. Điều này có nghĩa là trên các phiên bản Windows 64 bit, dữ liệu đăng ký cụ thể cho 32 bit các ứng dụng được duy trì ở một vị trí riêng biệt trong thư mục Phần mềm. Theo Microsoft Bài viết cơ sở kiến thức 305097 (http://support.microsoft.com/kb/305097), ứng dụng 32 bit dữ liệu được ghi vào đường dẫn chính sau:

*HKLM\Software\WOW6432Node*

Mặc dù các thư mục của Registry từ các phiên bản Windows 64 bit không khác nhau hoàn toàn từ các phiên bản Windows 32 bit ở cấp độ nhị phân thuần túy, sự khác biệt trong Khóa Registry, chẳng hạn như trong trường hợp chuyển hướng, có thể gây ra sự cố với các nhà phân tích đang tìm kiếm các khóa cụ thể hoặc sử dụng các công cụ như RegRipper và không tính đến nút này.

### **Ảo hóa**

Với sự ra đời của Windows Vista, Microsoft đã giới thiệu ảo hóa trong Registry. Trước lên Vista, nếu quản trị viên cài đặt ứng dụng yêu cầu đặc quyền nâng cao để chạy và một người dùng thông thường đã cố gắng chạy ứng dụng, nó sẽ thất bại. Với Vista, Microsoft đã chọn tham gia sử dụng một cơ chế ảo hóa cho phép Registry ghi các hoạt động có khả năng có tác động toàn cầu (nghĩa là, được viết cho thư mục Phần mềm) để được ghi vào vị trí của mỗi người dùng. Điều này ngăn chặn mọi hoạt động ghi nguy hiểm tiềm tàng cho Registry (thường là tạo hoặc sửa đổi các khóa hoặc giá trị) từ việc có tác động toàn cầu đến hệ thống. Ví dụ: nếu một ứng dụng cố ghi vào khóa HKLM \ Software \ App, nó sẽ tự động được chuyển hướng đến khóa sau:

*HKEY\_USERS\{User SID}\_Classes\VirtualStore\Machine\Software\*

Đến lượt mình, khóa này không tồn tại trong thư mục của người dùng NTUSER.DAT, mà thay vào đó, tập tin sau được ánh xạ tới khóa đó khi người dùng đăng nhập:

*%UserProfile%\AppData\Local\Microsoft\Windows\usrclass.dat*

Theo bài viết của Mark Russinovich từ TechNet về Kiểm soát truy cập người dùng Vista (http://technet.microsoft.com/en-us/magazine/2007.06.uac.aspx), chỉ có các khóa là thường được sửa đổi bởi các ứng dụng cũ, được ảo hóa. Các sửa đổi cho ảo hóa các khóa (các khóa như Software \ Classes, Software \ Microsoft \ Windows và Software \ Microsoft \ Windows NT được miễn ảo hóa) được chuyển hướng đến tệp người dùng usrgroup.dat. Tệp này là tệp trung tâm hợp lệ ở cấp nhị phân và có thể được mở bởi hầu hết Registry Viewers, cũng như được truy cập thông qua mô-đun Parse :: Win32Registry Perl. Tại thời điểm viết bài này, tuy nhiên, không có plugin RegRipper nào trong phân phối được thiết kế riêng để lấy thông tin từ thư mục này.

### **Deleted Registry Keys**

Ở đầu chương này, chúng tôi đã thảo luận về sự cần thiết phải hiểu cấu trúc của Khóa Registry, với tư cách là người kiểm tra có thể tìm thấy chỉ dẫn của khóa trong không gian chưa phân bổ, tệp trang, một bãi chứa bộ nhớ, và như vậy. Biết cấu trúc của các khóa sẽ cho phép bạn phân tích cú pháp dữ liệu bạn tìm thấy vào một cái gì đó có thể đọc được, cung cấp ngữ cảnh cho những gì bạn đã tìm thấy. Khác nơi bạn có thể xác định vị trí các khóa Registry mà RegEdit hoặc các công cụ phân tích khác nhìn thấy với Bản thân các thư mục, trong những gì mà lòng gọi là không gian được phân bổ là tên lửa trong phạm vi thư mục.

Như bạn đã thấy trước đó trong chương trong Hình 4.3, các khóa Registry có thể được xác định bởi chữ ký *nk*. In spring 2008, Jolanta Thomassen (JT) đã xác định rằng bốn byte, hoặc DWORDs, ngay trước chữ ký khóa, khi được đọc dưới dạng giá trị dài không dấu (sử dụng hàm Perl’s unpack() ,bạn sẽ đọc nó dưới dạng L thay vì V được sử dụng trong nhiều trường hợp để giải nén () giá trị DWORD), đối với hầu hết các khóa Registry hợp lệ, sẽ là số âm. Đây là sự thật cho các khóa Registry hiển thị thông qua các công cụ như RegEdit và các công cụ khác. JT cũng tìm thấy chữ ký cho các khóa rõ ràng hợp lệ mà giá trị DWORD trước đó là dương; nghiên cứu của cô ấy trên thực tế, đây là những khóa đã bị xóa, nằm trong thư mục.

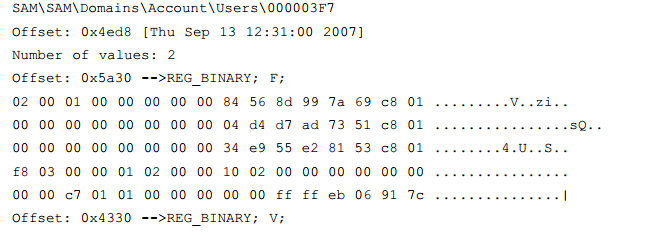
Nghiên cứu của JT, là một phần trong nghiên cứu sau đại học của cô tại Đại học Liverpool. Tôi đã có cơ hội gặp JT tại hội nghị DFRWS 2008 được tổ chức tại Baltimore vào tháng 8 năm 2008. Tôi cũng đã gặp Tim Morgan ở đó, người đang thực hiện nghiên cứu của riêng mình về các khóa bị xóa trong Các thư mục của registry, hoàn toàn độc lập với công việc của JT. Bạn có thể tìm thấy tờ giấy Tim, có tiêu đề Phục hồi dữ liệu đã xóa của Windows từ Windows Registry, và các slide trình bày của anh ấy tại Trang web hội nghị DFRWS 2008 ([www.dfrws.org/2008/program.shtml](http://www.dfrws.org/2008/program.shtml)).

JT đã phát hành mã Perl hoạt động sẽ phân tích cú pháp thư mục (được trích xuất từ một mã thu được hình ảnh hoặc có thể truy cập thông qua hình ảnh được gắn dưới dạng hệ thống tệp chỉ đọc hoặc có thể truy cập qua Matt Sản phẩm F-Feedback của Shannon hạng), trích xuất và hiển thị thông tin về việc đã xóa Các khóa và giá trị đăng ký, cũng như không gian chùng trong thư mục. Kịch bản JT từ Perl, regslack. pl, được bao gồm trong thư mục ch4 \ code \ jt trên phương tiện đi kèm với cuốn sách này, cùng với với một phiên bản Windows biên dịch Windows PE của công cụ.

Để chạy regslack.pl, chỉ cần gõ như sau tại dấu nhắc lệnh:

*C:\perl\jt>regslack.pl d:\cases\SAM > sam.txt*

Regslack.pl doesn đòi hỏi bất kỳ mô-đun Perl bổ sung nào, vì vậy nó có thể được chạy dưới dạng tập lệnh Perl không sử dụng gì nhiều hơn cài đặt Perl cơ bản từ ActiveState.com. Trước đó lệnh chuyển hướng đầu ra của tập lệnh từ bàn điều khiển đến một tệp có tên sam.txt. Tôi khuyên bạn nên làm điều gì đó như thế này vì kịch bản có số lượng đáng kể thông tin mà nó gửi đến bàn điều khiển, do đó việc nắm bắt thông tin đó trong một tệp sẽ tạo ra nó dễ xem hơn, cũng như giữ nó để xem lại sau. Một đoạn trích của tệp sam.txt, minh họa một khóa Registry được phục hồi từ tệp SAM, được hiển thị ở đây:



Trong trường hợp này, regslack.pl đã tìm thấy khóa Registry đã xóa liên quan đến người dùng, cũng như nội dung của cả hai giá trị F và V (tên giá trị V được hiển thị, nhưng vì lợi ích của ngắn gọn, toàn bộ nội dung nhị phân của dữ liệu giá trị không được hiển thị). Đầu ra trước đó hiển thị tên của khóa với đường dẫn đầy đủ được phục hồi, phần bù mà tại đó khóa là tìm thấy, cũng như thời gian LastWrite của khóa. Ngoài tên và dữ liệu giá trị, bù đắp cho các giá trị bản thân cũng được hiển thị, cho phép xác minh trực quan. Regslack.pl cũng hiển thị không gian chùng từ trong thư mục.

# **Summary**

Biết cách điều hướng Registry để biết thông tin cụ thể có thể chứng minh là một kỹ năng vô cùng quý giá cho các quản trị viên, chuyên gia tư vấn và các nhà phân tích pháp y. Cơ quan đăng ký sẽ sẵn sàng tiết lộ bí mật của nó cho những người biết nơi để tìm và cách giải thích thông tin họ tìm thấy. Đi sâu vào Đăng kiểm không khác gì Indiana Jones theo dõi những bí mật cổ xưa trong cát bay của thời gian.

Bạn có thể tìm thấy một cách toàn diện (vì lý do rõ ràng, tôi cực kỳ do dự khi sử dụng từ hoàn thành) danh sách các vị trí tự khởi động và danh sách MRU của người dùng trong Registry trong bảng tính regref.xls có trên phương tiện đi kèm. Danh sách này nên được coi là một điểm khởi đầu, mặc dù là một toàn diện, cho người đọc. Các khóa Registry được liệt kê là lấy từ danh sách trực tuyến, các ứng dụng như Autorun, báo cáo về phần mềm độc hại về phần mềm chống vi-rút Các trang web, và thông qua thử nghiệm cá nhân.

Chương này không nên được coi là một tài liệu tham khảo đầy đủ của tất cả các cơ quan đăng ký khác nhau các khóa có thể có tầm quan trọng đối với một trường hợp cụ thể. Các khóa registry có chức năng tương tự khác nhau về tên và vị trí giữa các ứng dụng và, trong một số trường hợp, giữa các phiên bản của ứng dụng tương tự. Không có cuốn sách nào có thể được coi là một tài liệu tham khảo có thẩm quyền hoàn chỉnh cho Khóa đăng ký quan tâm; nó sẽ quá lớn và đắt đỏ và sau đó gần như ngay lập tức hết hạn ngay khi nó được xuất bản. Mục đích của chương này là để minh họa những gì thông tin có sẵn và làm thế nào để tìm kiếm thêm thông tin.

# **DVD Contents**

Các phương tiện truyền thông đi kèm với cuốn sách này bao gồm một thư mục dành riêng cho chương này. Cái đó thư mục bao gồm một số thư mục con, một thư mục chứa mã được trình bày trong chương này đặc biệt là bản phân phối RegRipper (bao gồm rip.pl và tất cả các plugin và như đã nêu, ripxp.pl không được bao gồm). Các phương tiện truyền thông cũng bao gồm tất cả các tập lệnh Perl từ phiên bản đầu tiên của cuốn sách (trong thư mục con cũ của Cũ), cũng như ProScripts (kịch bản Perl cụ thể được viết để sử dụng với ứng dụng phân tích pháp y ProDiscover) được thảo luận trong chương này. Thư mục con của bảng tính con Thư mục con bao gồm một bản sao của bảng tính tham chiếu Registry Tôi đã biên dịch một chút trước đây. Bảng tính này liệt kê một số khóa Registry và các giá trị quan tâm các nhà phân tích pháp y cũng như một mô tả ngắn gọn về mục đích của họ và, nếu áp dụng, tài liệu tham khảo. Bảng tính được chia thành các bảng tính khác nhau, mỗi bảng bao gồm một cụ thể khu vực chức năng.

Ngoài ra còn có một mẫu thư mục con của Wikipedia có một số tệp Registry từ các hệ thống thực không chỉ những người được tìm thấy trong thư mục system32\config, mà cả những người từ các điểm khôi phục. Tôi khuyến khích bạn xem các tệp này, mở chúng trong trình chỉnh sửa hex và sử dụng bất kỳ công cụ nào được bao gồm trong thư mục con mã cho chương này để bạn có thể phát triển sự quen thuộc với các công cụ cũng như với các tệp Registry thô.

# **Giải pháp theo dõi nhanh**

## **Bên trong Registry**

* Windows Registry là một cơ sở dữ liệu cấu hình nhị phân, phân cấp thông tin không chỉ kiểm soát hệ điều hành và ứng dụng khác nhau cài đặt cấu hình, nhưng cũng duy trì thông tin về các khía cạnh khác nhau của người dùng tương tác với hệ thống.
* Bằng cách hiểu định dạng của các cấu trúc Registry khác nhau (nghĩa là, các khóa và sau đó bạn có thể phân tích cú pháp và xem các phần của Registry được tìm thấy trong bộ nhớ và không gian chưa phân bổ.
* Một số phần của Registry rất dễ bay hơi, được tạo khi hệ thống khởi động và sẽ không được tìm thấy trong một hình ảnh thu được của một hệ thống.
* Khóa Registry (và một số giá trị) có dấu thời gian được liên kết với chúng có thể được sử dụng trong phân tích dòng thời gian. Vì lý do này, Registry có thể được coi là một tệp nhật ký.
* Không có tiêu chuẩn rõ ràng cho cách duy trì thông tin trong Registry. Một số khóa MRU, ví dụ, duy trì dữ liệu giá trị của chúng dưới dạng loại nhị phân, một số khác như Chuỗi ASCII (giúp tìm kiếm chuỗi ASCII có phần dễ dàng). Đăng ký khác Các khóa có tên giá trị của chúng bị xáo trộn với mã hóa ROT-13, hoặc giá trị của chúng dữ liệu được duy trì theo cách cản trở tìm kiếm chuỗi. Bạn cần hiểu cấu trúc của khóa và giá trị cụ thể để phân tích dữ liệu được lưu trữ ở vị trí đó.
* Các công cụ có sẵn để theo dõi truy cập và sửa đổi đối với Registry trực tiếp hệ thống; thông tin này cũng có thể được sử dụng để xác định vị trí các khóa Registry quan tâm như xác định các tạo phẩm do ứng dụng và hoạt động người dùng để lại

## **Phân tích Registry**

* Một số vị trí trong Registry chứa thông tin phù hợp với hầu hết điều tra. Các vị trí khác chứa thông tin phù hợp với các loại cụ thể của điều tra, chẳng hạn như xâm nhập, gian lận hoặc lạm dụng các chính sách sử dụng chấp nhận được.
* Một số khóa và giá trị Registry cụ thể có thể có tầm quan trọng đối với một cuộc điều tra, nhưng thường thì đó là sự tương quan của một số khóa và giá trị Registry có thể cung cấp bức tranh đầy đủ nhất.
* Điểm khôi phục hệ thống Windows XP duy trì các phần của Registry có thể hữu ích trong quá trình điều tra. Ví dụ, kiểm tra nội dung của bảo quản Tệp SAM, một điều tra viên có thể xác định khi nào một nhóm người dùng tư cách thành viên đã thay đổi (ví dụ: đi từ Người dùng sang nhóm Quản trị viên), nếu đó là thích hợp để điều tra. Bạn cũng có thể nói những ứng dụng nào đã được cài đặt trên hệ thống trong thời gian gần đây.