Chương 4:

Các hư hỏng logic trên hệ thống tập tin và cách khắc phục

1.1. CÁC NGUYÊN NHÂN

1.1.1. Do các chương trình mã độc

Có nhiều chương trình được viết ra với mục đích phá hoại dữ liệu, có thể là phá cho hư hẳn (ví dụ như format luôn đĩa của nạn nhân), cũng có thể là mã hóa dữ liệu khiến nạn nhân không thể truy xuất và phải chấp nhận trả tiền cho kẻ xấu theo đúng yêu cầu thì mới được giải mã phục hồi lại. Các chương trình mã hóa dữ liệu và đòi tiền chuộc (ransomwave) này cũng như đa số malware (malicious software – phần mềm độc hại) xâm nhập vào các máy tính có kết nối với Internet khi người dùng thực hiện các thao tác: sử dụng các chương trình crack hoặc các chương trình không rõ nguồn gốc, truy cập các trang web xấu hoặc các trang giả mạo, click vào các link quảng cáo không thuộc các trang có uy tín, chạy các file đính kèm trong spam mail, sử dụng các chương trình có nhiều lỗ hổng nghiêm trọng (như Flash player,…), kết nối với wifi công cộng khi máy không có tường lửa hữu hiệu,…

Một thông báo đòi tiền chuộc từ ransomware

1.1.2. Do các chương trình bình thường bị lỗi

Có nhiều chương trình được viết chưa hoàn hảo dẫn đến đôi khi chạy có trục trặc và việc tạo /ghi dữ liệu có thể bị sai ngay từ nội dung của dữ liệu, hoặc nội dung thì tuy đúng nhưng ghi nhầm vị trí khiến cho các dữ liệu đang có sẵn bị lỗi. Thậm chí nặng nề hơn là việc xóa nhầm hàng loạt dữ liệu hoặc format nhầm volume,… .

1.1.3. Do thiết bị phần cứng

Thiết bị lưu trữ hoạt động không ổn định có thể gây trục trặc dữ liệu. Chẳng hạn khi hoạt động ghi dữ liệu đang được tiến hành thì có sốc điện xảy ra khiến vị trí cần ghi có thể lệch đi và ghi đè lên các dữ liệu cũ – khiến các dữ liệu cũ bị hư.

1.1.4. Do người dùng nhầm lẫn

Đôi khi chính bản thân người sử dụng tự gây ra hư hỏng dữ liệu ngoài ý muốn. Có thể là xóa /format /ghost nhầm vì không cẩn thận hoặc có những sơ suất ngoài ý muốn nào đó (ví dụ như vô tình chạm vào phím delete /enter trên bàn phím chẳng hạn), cũng có thể là thao tác thực hiện chính xác nhưng một thời gian sau mới nhận ra là có cần lại một /một số dữ liệu đã xóa.

1.1.5. Do người xấu phá hoại

Có thể là do mâu thuẫn trong việc gì đó, cũng có thể xuất phát từ động cơ cạnh tranh không lành mạnh, có những người cố tình trực tiếp phá hoại nạn nhân. Và một trong các mục tiêu được nhắm đến là các dữ liệu máy tính – vì thiệt hại gây ra có thể rất to lớn. Việc tạo dựng dữ liệu có thể rất lâu nhưng việc xóa bỏ hoặc format cả volume có thể lại rất nhanh. Nhiều người nghĩ rằng những trường hợp này nếu đã xảy ra rồi thì sẽ khó hoặc không thể phục hồi lại được dữ liệu, tuy nhiên trên thực tế có khi vẫn được.

1.2. CÁC TÌNH HUỐNG CỤ THỂ - CÁCH XỬ LÝ TƯƠNG ỨNG

1.2.1. Sai thông tin mô tả Partition

HDD, SSD, USB disk, Memory card hay nói chung là các thiết bị lưu trữ kích thước lớn ngoại trừ đĩa quang đều có thể tổ chức nhiều Partition. Vì nhiều lý do, bảng Partition

mô tả vị trí và kích thước các Partition trên đĩa có thể bị sai, dẫn đến hệ thống không nhận ra các đĩa (volume) và hệ lụy là mất toàn bộ dữ liệu trên tất cac3 các volume.

Bảng Partition nằm trong Master Boot Record (MBR) và đóng vai trò cực kỳ quan trọng nên thường có ít nhất một bản sao trên 1 sector khác. Cụ thể bảng nằm tại offset 1BE trên MBR, chiếm kích thước là 40h byte và chứa đúng 4 entry (tức mỗi entry có kích thước 10h byte). Ngay sau bảng này là 2 byte cuối của MBR – và cũng là 2 byte dùng để nhận diện BR (luôn là 55, AA)

Cấu trúc của entry như sau:

Offset	**Độ dài (byte)**	**Nội dung**
+0	1	Có active (để khởi động) không (0: không, 80h: có)
+1	1	Head bắt đầu của partition
+2	2	Sector (6bit) và Cylinder (10bit) bắt đầu
+4	1	Kiểu hệ thống tập tin trên partition
+5	1	Head kết thúc của partition
+6	2	Sector (6bit) và Cylinder (10bit) kết thúc
+8	4	Vị trí bắt đầu, tính theo địa chỉ sector logic LBA
+0Ch	4	Tổng số sector trong partition
+10h	*2*	*Bắt đầu của entry kế hoặc AA55h nếu là entry cuối*

Để khắc phục lỗi sai thông tin trong bảng Partition ta sẽ dò tìm bản sao đúng của nó và chép lại vào MBR.

Có rất rất nhiều sector trên đĩa nhưng bản sao của MBR thường hay nằm ở những sector đầu và những sector cuối đĩa. Do đó có thể giới hạn không gian tìm kiếm chỉ trong vài ngàn sector, nếu không thành công mới mở rộng ra toàn bộ đĩa. Để tìm bản sao của MBR ta căn cứ vào dấu hiệu nhận diện (2 byte cuối), sau đó kiểm tra 4 byte đầu của từng entry xem có phải 00 hoặc 80h hay không, nếu đúng thì kiểm lại vị trí cùng kích thước của từng entry có hợp lý hay không (tổng lại không vượt quá dung lượng đĩa. Sau khi tìm được các bản sao hợp lý thì lần lượt thử nghiệm đưa vào MBR để người dùng kiểm chứng.

Trường hợp không tìm được thì có thể dò các vị trí của BR (Boot Sector – sector đầu của volume) để suy ra vị trí và kích thước của từng volume, rồi dựng lại nội dung bảng Partition.

1.2.2. **Sai thông số Volume**

Mỗi volume đều phải có một kiến trúc định dạng cho hệ thống tập tin được lưu trữ trên đó. Và thông tin về vị trí /kích thước các thành phần quan trọng trong kiến trúc định dạng được tổ chức ngay trong Boot Record (BR). Nếu có sai lệch, dù chỉ là 1 thông số, thì volume thường sẽ bị hệ điều hành hiểu sai và không truy xuất được dữ liệu, thậm chí càng cố sử dụng thì có thể lại càng gây ra hư hại nhiều hơn.

Cách phục hồi lại nội dung đúng cho BR trước tiên không phải là dò tìm bản sao đúng của BR và chép lại, mặc dù vì tầm quan trọng của BR rất lớn nên thường sẽ phải có một vài bản sao lưu và bản sao BR cũng thường hay ở đầu volume nên việc tìm thường không mất nhiều thời gian. Việc dò tìm như vậy có thể cho ra nhiều kết quả và có thể có các kết quả nhầm lẫn (với các sector không phải là BR hoặc là các BR không phải của volume đang xét). Trên thực tế ngay trong BR đã có thông tin cho biết bản sao của nó ở đâu, ta chỉ việc đến đúng chỗ đó mà lấy rồi kiểm định thêm có chính xác không. Nếu không hợp lệ thì mới tiến hành dò tìm.

Dấu hiệu nhận diện BR cũng nằm ở 2 byte cuối như MBR, và sau đó ta cũng phải kiểm tra thêm hàng loạt thông tin khác để xác thực. Sau khi tìm được các bản sao hợp lý thì lần lượt thử nghiệm đưa vào BR để người dùng kiểm chứng.

Trường hợp không tìm được thì có thể tham khảo nội dung các vùng lân cận để suy ra vị trí và kích thước của từng thành phần, dựa trên các đặc điểm nhận dạng của chúng cũng như các giá trị mặc định thông dụng (ví dụ kích thước cluster thường là 8 sector, số bảng FAT thường là 2, kích thước vùng thư mục gốc thường là 512 entry hoặc cluster bắt đầu là 2,…), rồi dựng lại nội dung bảng thông số volume. Việc dựng lại nội dung này cũng có thể khá đơn giản nếu chỉ bị sai một vài thông số: ta có thể căn cứ vào giá trị của các thông số khác mà suy ra các giá trị thích hợp cho các thông số sai.

1.2.3. **Sai bảng thư mục và bảng quản lý cluster**

Bảng quản lý cluster theo dạng danh sách liên kết kết hợp chỉ mục thật ra vẫn thường xuyên bị lỗi logic, nhưng nhờ hệ thống thường tổ chức thành 2 bản nên có thể sử dụng bù trừ cho nhau và khắc phục được (điều này được hệ điều hành tự thực hiện).

Bảng thư mục sai logic trong khi bảng quản lý cluster còn đúng thì trước mắt nội dung các tập tin vẫn có thể lấy lại được (nhưng tên và các thuộc tính có thể không còn), và xem như cũng không gây mất mát lớn.

Hiện tượng tất cả các bảng đều bị sai thường xảy ra khi đĩa bị Format hoặc Ghost. Do thông tin mô tả về các thuộc tính và nội dung tập tin đã mất nên việc cứu lại không thể đảm bảo trọn vẹn nếu không may mắn. Trước mắt khi đĩa đang ở tình trạng này thì không nên chép vào đĩa bất cứ nội dung gì, vì càng chép vào sẽ càng làm cho nội dung cũ bị mất đi. Nếu có thể thì cần tạo một bản image cho volume càng sớm càng tốt (và việc cứu lại dữ liệu có thể tiến hành trên chính bản image, không cần truy xuất tới volume).

Cơ sở cho việc cứu lại dữ liệu trong trường hợp này trong kiến trúc FAT là các bảng thư mục con được lưu trữ ở vùng Data đa phần có thể còn nguyên vẹn, và nội dung tập tin thường ít bị phân mảnh. Do đó chỉ với các entry thư mục ta có thể suy ra tên và các thuộc tính của tập tin (2 entry đầu “.” & “..” cũng có thể dùng để khai thác được nhiều thông tin), và 2 thuộc tính quan trọng dùng để định vị nội dung tập tin là <cluster bắt đầu> và <kích thước>. Riêng với các tập tin bị phân mảnh (chiếm tỉ lệ rất ít) thì việc cứu lại chắc chắn rất khó khăn và khả năng thành công khó đạt tỉ lệ 100%, mặc dù phải dùng tới các thuật toán thông minh nhân tạo.

Trên các partition có định dạng NTFS, sau khi format thì các thông tin về hệ thống cũ bị mất, bảng MFT bị thay đổi – nhưng kích thước cluster thì thường vẫn là 8 và một số file record cũ vẫn còn tồn tại và có thể dò tìm được. Khi thông tin về vị trí MFT không còn thì phải dò trên từng cluster để tìm, và nếu trường hợp $MFT file record nằm trên bảng MFT đã không còn thì vẫn có thể tìm $MFT file record trên file $MFTMirror.

1.2.4. **Các tập tin hoặc thư mục bị xóa**

Trước tiên cần kiểm tra trong Recycle Bin xem các dữ liệu đã xóa có ở đó không, vì cơ chế xóa bình thường là nén dữ liệu lại đưa vào đó rồi mới xóa, như vậy nếu có thì việc phục hồi rất đơn giản và nhanh chóng. Đây chính là tính năng nâng cao sự an toàn dữ liệu cho người dùng mà hệ thống đã thiết kế sẵn.

Những năm gần đây điện toán đám mây phát triển mạnh và các dịch vụ sao lưu - lưu trữ dữ liệu trực tuyến với khả năng đồng bộ thời gian thực được nhiều người hoan nghênh và sử dụng (ví dụ như dropbox, google drive, one drive,…). Khi dữ liệu trên máy bị xóa ở cấp độ không thể phục hồi (như trong trường hợp có người phá hoại dùng phần mềm xóa chuyên dụng) nhưng đã được đồng bộ trên đĩa mạng thì chỉ cần dùng các hỗ trợ tương ứng trong các dịch vụ này để lấy lại dữ liệu, thậm chí có thể truy vết để biết được cụ thể thời điểm và thao tác gây ảnh hưởng dữ liệu đã được thực hiện.

Nếu cả hai trường hợp trên không thực hiện được thì có thể tìm đến entry ứng với dữ liệu cần phục hồi, từ đó xác định được chỉ số cluster bắt đầu và kích thước của tập tin (do khi xóa thì hệ thống chỉ đánh dấu vào entry chứ không xóa hết các nội dung). Sau đó qua bảng quản lý cluster để tìm dãy cluster trống liên tiếp tương ứng với số cluster của tập tin. Rất nhiều khả năng nội dung tập tin ứng với dãy cluster này. Còn trong trường hợp có tồn tại một số cluster không đúng thì loại bỏ chúng ra và lấy thêm các cluster trống kế tiếp.

1.2.5. **Các dữ liệu không còn thông tin mô tả**

Trường hợp phức tạp khó khăn nhất: thông tin mô tả về dữ liệu đã mất hẳn (do bị chép đè chẳng hạn). Khi này tên và các thuộc tính tập tin không còn tồn tại nữa, ta phải xác định rõ là cần cứu tập tin theo định dạng gì và dĩ nhiên nội dung ở định dạng đó phải có dấu hiệu nhận biết đặc trưng. Tùy thuộc vào tình trạng volume hiện tại mà ta sẽ phải quét các sector hay cluster nào để dò tìm:

- Với volume chưa bị biến động về định dạng (kể từ khi dữ liệu bị mất): việc dò tìm tiến hành trên các cluster trống.

- Với volume đã bị biến động về định dạng sau khi dữ liệu mất: việc dò tìm tiến hành trên các cluster hoặc sector trên hệ thống cũ, trong đó các cluster /sector này được trích rút ra từ các cluster trống hiện hữu
- Với volume bị lỗi trên cấu trúc định dạng (không còn truy xuất được ở cấp hệ điều hành): cố gắng xác định vị trí và kích thước của cluster và dò tìm trên các cluster, khi không xác định được cluster mới phải dò theo sector rất lâu và kém chính xác hơn.

Để xác định vị trí và kích thước của cluster trên volume dạng FAT /FAT32 đã mất thông tin mô tả ở mức cả vùng hệ thống cũng không còn, ta có thể thực hiện khá đơn giản như sau:

- Dò tìm vị trí bắt đầu của một SDET, bằng cách duyệt qua từng sector cho đến khi gặp sector M có offset 0 là '.' và 10 offtset kế tiếp là ' ' (dãy byte tương ứng: 2E 20 20 20 20 … 20), đồng thời offset 32 có 11 byte là 2E 2E 20 20 20 … 20. Entry "." Đầu tiên chứa thông tin mô tả về chính bản SDET đó nên lấy trường cluster bắt đầu A ra ta sẽ có thông tin cluster A bắt đầu tại sector M (1).
- Tương tự tìm tiếp sector N như trên ta cũng sẽ suy được cluster B bắt đầu tại sector N (2).
- Từ (1) và (2) suy được kích thước cluster Sc=(M-N)/(A-B) (sector) và cluster 2 bắt đầu tại sector M-(A-2)*Sc.

Trên cơ sở định vị được các cluster, ta có thể đọc được các cluster ứng với bảng thư mục, từ đó có thể tìm ra entry ứng với file cần cứu, lấy ra thông tin cluster bắt đầu và kích thước file thì có thể phục hồi lại nội dung file.