Phân Tích Log (1)

Tequila (VietHacker.org Translator Group Leader)

Compose by hieupc

Trong lĩnh vực computer forensics, thì dường như máy tính của bạn là hiện trường nơi xảy ra tội phạm. Nhưng không giống như phân tích về loài người, các nhà phân tích về máy tính thường làm việc với 1 cái máy tính đang làm việc mà có thể đưa ra các dấu hiệu mà những thứ khác có thể sai lầm. Chương này chúng ta sẽ làm việc với việc phân tích log, mà có thể được coi như là 1 nhánh của forensics. Một file log riêng rẽ có thể rất quan trọng và chúng ta phải quyết định đưa chúng vào các phần khác nhau.
Những gì là ví dụ cho những file log? Chúng ta có thể phân loại file log bằng các thiết bị mà làm việc dựa trên chúng, bởi vì những thiết bị thường quyết định loại thông tin chứa trong các file. Ví dụ, các file log của host (như Unix, Linux, Windows, VMS…) là khác nhau từ log của các ứng dụng network khácnhau (vídụ như sản phẩm switchs, routers hoặc các thiết bị mạng khác của Cisco, Nortel, Lucent…). Tương tự như vậy, các log của các ứng dụng an toàn thông tin (như firewall, IDS, thiết bị chống ddos, hệ thống phòng bị…) cũng rất khác nhau trên cả phương diện host và các log mạng. Trong thực tế, các thiết bị an toàn mạng hiển nhiên tạo nên một sự phân bổ không thể tưởng tượng được những gì chúng có thể ghi lại và định dạng mà chúng có thể tạo ra. Sắp xếp từ các địa chỉ IP đơn giản cho tới những giao dịch đầy đủ phức tạp trên mạng, các thiết bị an toàn hệ thống thường tạo nên một giá trị rất to lớn những thông tin rất thú vị, cả những thông tin về những sự việc hợp lệ và không hợp lệ. Làm thế nào để chúng ta có thể tìm được đâu là những sự việc không được cho phép. Làm thế nào để chúng ta học được về những xâm nhập trong quá khứ và thậm chí là tương lai từ logs? Chúng ta hoàn toàn có thể hi vọng vào việc tìm kiếm trong hàng gigabytes file log để tìm ra những hoạt động mà không được phép xảy ra khi mà những hacker đã rất là cẩn thận không để lại một dư thừa nào? Chương này sẽ trả lời cho chúng ta tất cả những câu hỏi đó.

18.1 Cơ bản của việc phân tích Log

Phân tích các log hoặc các chuỗi thống kê là một nghệ thuật của việc trích dẫn đầy đủ ý nghĩa thông tin và đưa ra kết luận về một trạng thái an toàn từ các bản ghi thống kế những sự việc được sản sinh bởi máy tính. Phân tích log không phải là 1 khoa học, nhưng ngày nay, việc tin tưởng vào kỹ năng phân tích độc lập và trực quan cũng như tính chất may mắn trong việc phân tích log chất lượng cũng là một khái niệm khoa học. Định nghĩa việc phân tích log có thể nghe rất khô khan, nhưng quan trọng là rút ra một “Kết luận có ý nghĩa”. Nhìn một cách đơn giản vào các file log không phải là phân tích, bởi vì hiếm có những cái gì ngoài những sự nhàm chán và dường như chẳng liên quan gì đến nhau. Trong trường hợp một thiết bị 1 người sử dụng với rất ít các hoạt động, tất cả những bản ghi log mà chưa được nhìn trước là rất it nghi ngờ, nhưng trong thực tế lại không dễ dàng như vậy.

Hãy thử xem một phân tích log cho những telnet chung. Đầu tiên, hãy nhìn qua toàn bộ log cần phải phân tích(giống như file log của một thiết bị xâm nhập đối với 1 thông báo tấn công thành công) và tạo quan hệ với những nguồn thông tin khác. Việc tạo quan hệ có nghĩa là thực hiện những thao tác bằng tay hoặc tự động để thiết lập nên mối quan hệ giữa các sự kiện tưởng chừng không liên quan xảy ra trên mạng. Các sự kiện xảy ra trên các thiết bị khác nhau trong các thời điểm khác nhau có thể tạo nên những quan hệ tức thời (xuất hiện trong thời gian ngắn). Đây có phải là một lỗ hổng cho kẻ tấn công có thể phát hiện được? Có phải các quy tắc của các hệ thống phát hiện xâm nhập đưa ra 1 dự báo sai. Có phải là một ai đó trong số các nhân viên của bạn đang thử quét các lỗ hổng trong mạng của bạn? Trả lời cho những câu hỏi tương tự như vậy là rất cần thiết trước khi lập kế hoạch phản ứng cho các thông báo của IDS. Các cố gắng kết nối, nắm bắt các dịch vụ và những sai lầm đa dạng của hệ thống thường yêu cầu thực thi rất nhiều những việc tạo mối quan hệ với những nguồn thông tin khác nhau theo nhiều mức để đạt được thông tin có ý nghĩa đầy đủ nhất.

18.2 Những ví dụ về log

Trong phần này chúng ta sẽ lất ví dụ trên các file log đã được tổng hợp trên các hệ thống Unix và sau đó là Windows.

18.2.1 Unix

Việc phổ biến các hệ thống Unix thương mại và miễn phí ngày càng phát triển khiến cho kỹ năng phân tích Unix log cũng là một ưu tiên phát triển hàng đầu. Các hệ thống Unix và Linux tạo ra một loạt các thông báo (giống như các log hệ thống), thường tồn tại dưới các dạng plain text, được định dạng như trong ví dụ sau:
<date / time> <host> <message source> <message>
Ví dụ như :
Oct 10 23:13:02 ns1 named[767]: sysquery: findns error (NXDOMAIN) on ns2.example.edu?

Oct 10 23:17:14 ns1 PAM_unix[8504]: (system-auth) session opened for user anton by (uid=0)

Oct 10 22:17:33 ns1 named[780]: denied update from [10.11.12.13].62052 for "example.edu"

Oct 10 23:24:40 ns1 sshd[8414]: Accepted password for anton from 10.11.12.13 port

2882 ssh2

Ví dụ này rất quen thuộc cho ai quản trị hệ thống Unix trong ít nhất 1 ngày. Định dạng này bao gồm các trường sau:

Timestamp
Giờ hệ thống của thiết bị khi ghi nhận log (trường hợp log 1 đăng nhập từ xa) hoặc của thiết bị tạo log (trong trường hợp tự tạo log).

Hostname or IP address of the log-producing machine
Hostname có thể là một tên domain name chất lượng (FQDN) ví dụ như ns1.example.edu hoặc chỉ là tên máy giống như là ns1 trong ví dụ trên.

Message source
Nguồn có thể là một phần mềm hệ thống (sshd hoặc là named trong ví dụ trên) hoặc là 1 bộ phận (ví vụ như PAM_unix) mà sản sinh ra thông báo log.

Log message
Thông báo log có thể có nhiều định dạng khác nhau, thông thường bao gồm tênứng dụng, các biến tình trạng đa dạng, địa chỉ IP nguồn, giao thức … Thỉnh thoảng định danh tiến trình của một tiến trình có thể tạo ra những bản ghi log và được ghi vào các chỗ trống.
4 thông báo log sau đây được chỉ ra, theo thứ tự:
- Có vấn đề xảy ra đối với DNS server thứ 2
- Một người sử dụng , (anton) đã đăng nhập vào thiết bị
- Một truy cập DNS bị cấm xuất hiện.
- Một người sử dụng (anton) đã được cung cấp mật khẩu an toàn hệ thống đang đăng nhập từ xa từ địa chỉ IP 10.11.12.13.

18.2.1.1Phân tích log hệ thống Unix

Log 1hệ thống Unix được quản lý bởi1 daemon syslog. Thiết bị daemon này đầu tiên xuất hiện trong những hệ thống BSD đầu tiên. Chương trình và các thành phần của hệ điều hành có thể đưa các sự kiện vào syslog thông qua hệ thống các lệnh, một socket (/dev/log), hoặc một kết nối mạng sử dụng UDP cổng 514. Các logging nội bộ thì thường được thực thi thông qua API.

Giống như trong trang hướng dẫn syslogd, “logging hệ thống được cung cấp bởi 1 thiết bị nhận syslogd từ các nguồn BSD,. Các hỗ trợ cho logging kernel được cung cấp bởi tiện ích klogd (trên Linux), cái mà cho phép logging kernel có thể được quản lý trong những mẫu chuẩn riêng hoặc giống như 1 máy trạm của syslogd. Trong mẫu chuẩn riêng, klogd chuyển các thông báo kernel ra 1 file, còn trong mẫu kết hợp, nó đẩy thông báo tới 1 daemon syslogd.

Các kết nối từ xa đòi hỏi daemon syslog phải được thiết lập để lắng nghe trên UDP cổng 514 (cổng chuẩn của syslog) cho các giao tiếp thông tin. Để cho phép 1 đăng nhập từ xa, bạn chạy syslogd –r trong Linux. Chức năng này được mặc định là cho phép trong Solaris và một vài môi trường Unix khác. Các thông báo

tới các mạng dưới dạng plain text và không có liên quan đến thời gian nào (Nó được đánh dấu bởi thiết bị nhận). Các thông báo tới cũng bao gồm các giá trị thực tế và đơn giản, được giải mã bởi daemon syslog.

Các log nhận được hoặc nội bộ được daemon syslog chuyển tới nhiều đích khác nhau (có thể là các file, các thiết bị, các chương trình, điều khiển hệ thống hoặc những hệ thống syslog khác) theo thứ tự và những tiện nghi khác. Những tiện nghi khác bao gồm auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, security (cũng giống như auth), syslog, user, uucp và local0 qua local7. Hướng dẫn syslog cũng đồng thời cung cấp danh sách theo thứ tự của syslog (sắp xếp dựa trên độ quan trọng): debug, info, notice, warning, warn (same cũng giống như warning), err, error (tương tự như err), crit, alert, emerg, và panic (tương tự như emerg). Thứ tự error, warn, and panic hiện nay vẫn được sử dụng cho các hệ thống syslog theo tuân thủ các thứ tự.

File thiết lập syslog thường nằm trong /etc/syslog.conf. Giống như được chỉ ra dưới đay, nó cho phép bạnc ó thể thiết lập các sắp xếp thppng báp theo các file khác nhau và các cấu trúc khác nhau:
. @log host

kern.* /dev/console

*.crit anton,other,root

local2.* |/dev/custom_fifo

*.info;mail.none;authpriv.none;cron.none /var/log/messages

authpriv.* /var/log/secure

mail.* /var/log/maillog

cron.* /var/log/cron

uucp,news.crit /var/log/spooler

local7.* /var/log/boot.log

Các thông báo có thể được trực tiếp đưa đến các file cục bộ (giống như /var/log/messages), gửi tới các thiết bị (như a /dev/console), hoặc được phổ biến tới tất cả hoặc là chỉ những người sử dụng được lựa chọn (anton, other, root) trong các lệnh tương tự hoặc các lệnh wall shell.Thêm vào đó, thông điệp có thể được chuyển tới một remote host (nhìn đoạn log host ở tên) và trực tiếp tới các đường dẫn đã được định danh hoặc những FIFOs khác (trong ví dụ trên là /dev/custom_fifo) được tạo bởi lệnh mknod hoặc mkfifo. Thậm chí những thông điệp mà được tới từ mạng có thể được chuyển tiếp tới những thiết bị khác, được các thiết bị syslog daemon cấu hình để làm nhiệm vụ này (giống như syslogd –h trong Linux). Việc chuyển tiếp được mặc định là không cho phép bởi vì nó có thể gây nên sự tác nghẽn mạng và những vấn đề khác (bởi vì nó nhân đôi lưu lượng trên đường truyền).

Các đăng nhập từ xa được ghi nhận là mối lợi lớn cho những người mà mong muốn tập trung tất cả các bản ghi thu nhận được vào một chỗ. Các thực thi syslog từ các phiên bản Unix khác nhau đều có thể làm việc tốt. Bạn có thể dùng lẫn nhiều box Unix trong 1 nền tảng syslog.
Một vài vấn đề về syslog sẽ xuất hiện một cách hiển nhiên trong khi làm việc. Đây là 1 danh sách ngắn:

1. Định dạng của thông điệp log là mâu thuẫn với nhau ở ứng dụng và hệ điều hành. Một phần là thời gian, host, phần còn lại của thông điệp là một mẫu tự do, điều này có thể tạo ra rất nhiều khó khăn nếu tất cả các thông điệp khác nhau đều hiển thị.

2. Việc lọc các thông điệp theo theo thứ tự và khả năng không thật hiệu quả bởi vì nó có thể dẫn đến một số log file trở thành sọt rác của một mớ hỗn tạp các loại thông điệp. Không có cách nào để lọc các thông điệp theo nội dung của chúng và thậm chí việc điều chỉnh thứ tự hoặc khả năng của một chương trình tạo log cũng thường xuyên chứng tỏ những thử thách đó.

3. Các chuyển dịch trên mạng dựa trên UDP là không thể tin tưởng được, nếu những cái nhận được kết thúc của 1 liên kết UDP (không phải là kết nối, bởi vì UDP còn chưa kết nối) mà giảm xuống, thì thông điệp sẽ bị mất mà không có cơ hội để phục hồi lại.

4. Các chuyển dịch trên mạng dựa trên UDP thường được diễn ra dưới dạng plain text (không được mã hóa), không được xác thực và rất ít được bảo vệ. Đây có thể là một thảm họa về an toàn thông tin. Tuy nhiên thông thường thì đây không phải làm 1 vấn đề trầm trọng bởi syslog được sử dụng trong các mạng nội bộ có thể tin tưởng được hoặc thậm chí là 1 mạng LAN được chỉ định quản lý.

5. Khi chuyển tiếp các thông điệp từ host tới host, chỉ có trạm cuối cùng mới có thể nhìn thấy thông điệp. Bởi vì, nếu 1 thiết bị gửi các thông điệp tới những máy khác - mà có thể chuyển tiếp tới bất kỳ đâu, thì thông điệp nhận được dường như là nguyên bản tại thiết bị thứ 2 này.

6. Việc lưu trữ các log dưới các file plain text có thể làm cho nó trở nên khó khăn hơn khi phân tích một lượng lớn các dữ liệu log. Hãy thử cố gắng để thực thi 1 lệnh grep hoàn chỉnh trên 1 file khoảng 5 GB và bạn sẽ hiểu đang phải đối mặt với vấn đề gì. Trong khi quay vòng log, lưu trữ và giảm bớt tất cả những sự giúp đỡ để giải quyết vấn đề, 1 cơ sở dữ liệu quan hệ là thực sự cần thiết.

7. Các log được lưu trữ là điểm yếu (??) để sửa chữa hoặc xóa đi, đặc biệt là khi lữu trữ nội bộ. Rất là khó để kiểm tra những file log có thiếu một đoạn dữ liệu nào đó hay không, đặc biệt nếu chúng đã được thay đổi bởi một người tấn công có kinh nghiệm với việc truy cập root.
Sự thay thế những syslog của các hệ thống Unix phổ biến xác định những sự thiếu hụt. Chúng ta sẽ xem 2 sự thay thế khá nổi tiếng đó là thay thế syslog-ng bởi BalabIT (http://www.balabit.hu/en/downloads/syslog-ng); và thay thế msyslog bởi CORE SDI (http://www.corest.com). Những chương trình này tạo nên giao tiếp TCP đáng tin cậy với các message bufferring,và nhiều lựa chọn lọc hơn (thêm vào đó với tính và tính thực tế của syslog. Những tài khoản không có quyền root đảm bảo an ninh cho các thao tác trong chroot, cung cấp dữ liệu log và điều khiển truy cập tốt hơn với các dữ liệu được mã hóa và thậm chí cung cấp cả những file log đã được tích hợp.
Hãy thử quan sát cách thiết lập msyslog cho một mạng nhỏ. Không giống như trong ví dụ về cấu hình syslog ở chuyển tất cả các thông điệp tới các thiết bị ở host thông qua UDP, trong trường hợp này, chúng ta sẽ sử dụng TCP với bộ đệm và lưu trữ các log trong dữ liệu và các file dạng plain text. Hơn nữa, chúng ta sẽ cho phép bảo vệ mã hóa cho các log file dạng plain text mà có thể cho phép chúng ta tì ra những thay đổi trong các log đã được lưu giữ.

Trên các máy trạm mà tạo ra hoặc chuyển tiêp các file lò, chúng ta phát triển và cấu hình msyslog. msyslog sử dụng file hợp lệ /etc/syslog.conf với các thay đổi phụ, như ví dụ sau :
. %tcp -a -h log host -p 514 -m 30 -s 8192
Ở ví dụ này, tất cả các thông điệp sẽ được chuyển từ các localhost tới các host log thông qua kết nối TCP cổng 514, ghi vào bộ đệm 8,192 thông điệp trong trường hợp kết nối không thành công và chờ khoảng 30 giây để thiết lập lại kếy nối tới log host . Dòng khác như /etc/syslog.conf có thể có mặt trong những định dạng syslog giống như được miêu tả ở trên, Daemon được kích hoạt chạy thông qua lệnh msyslogd -i linux -i unix hoặc sử dụng những kịch bản mặc định được cung cấp bởi các msyslog package.

Tại server, chúng ta cấu hinh để chạy msyslog như sau:
msyslogd -i linux -i unix -i tcp -a -p 514

Điều này làm cho daemon phải lắng nghe các kết nối qua TCP cổng 514 và cho phép đăng nhập từ tất cả các thiết bị. Các quy ước điều khiển truy cập có thể được ứng dụng để giới hạn các host dựa trên địa chỉ IP (các host có thể chuyển logs). Chúng ta cũng thêm vào bảo vệ crypto nhiều thông điệp quan trọng (chẳng hạn như thứ tự ưu tiên). Để làm được điều này, chúng ta thêm vào dòng lệnh đoạn /etc/syslog.conf như sau::
*.crit %peo -l -k /etc/.var.log.authlog.key %classic /var/log/critical

Tiếp theo, kết thúc msyslog daemon, xóa hoặc quay các logs, và tạo ra các khóa mã sử dụng tiện ích rất quen thuộc:
peochk -g -k /etc/.var.log.authlog.key

Khởi động lại daemon, và bảo vệ log được bật. Sau khi nhận thông điệp mới, msyslog cập nhật lại điều kiện. Và để kiểm tra tính tích hợp của log, chạy lệnh sau:
peochk -f /var/log/messages -k /etc/.var.log.authlog.key

Nếu mọi việc tốt đẹp, bạn sẽ nhìn thấy như sau:
(0) /var/log/critical file is ok
Nếu logfile đã bị thay đổi, bạn sẽ thấy:
(1) /var/log/critical corrupted
Thêm vào đó, để gửi các thông điệp tới cơ sở dữ liệu, một lệnh sau cần phải được thêm vào trong /etc/syslog.conf như sau:
. %mysql -s localhost -u logger -d msyslog -t syslogTB
Lệnh này sẽ lưu một bản copy của thông điệp vào trong cơ sở dữ liệu MySQL . Tuy nhiên, trước khi sự thu thập dữ liệu bắt đầu, bạn cần tạo ra một phác đồ và chèn vào một user được log, Điều này được làm hoàn chỉnh thông qua lệnh sau:
echo "CREATE DATABASE msyslog;" | mysql -u root -p
Lệnh này sẽ tạo ra 1 cơ sở dữ liệu. Nhưng trước đó, MySQL phải được cài đặt và chạy tốt trên hệ thống của bạn. Lệnh tiếp theo sẽ là:
cat syslog-sql.sql | mysql msyslog
Lệnh này định nghĩa 1bảng để lưu trữ log, syslog-sql.sql được chỉ ra như sau:
CREATE TABLE syslogTB (

facility char(10),

priority char(10),

date date,

time time,

host varchar(128),

message text,

seq int unsigned auto_increment primary key

);

Bước cuối cùng là tạo cơ hội cho việc thêm các thông điệp:
echo "grant INSERT,SELECT on msyslog.* to logger@localhost;" | mysql -u root -p

Việc cài đặt cơ sở dữ liệu như ở trên có thể lưu trữ an toàn hàng triệu bản ghi. Dữ liệu có thể được hiển thi thông qua các giao tiếp câu lệnh (mysql) hoặc một trong số nhiều cơ sở dữ liệu GUI database frontends và web frontends (ví dụ như PHPMyAdmin, viết trong PHP).

Để kết luận, msyslog và syslog-ng thao tác lẫn nhau với các thực thi syslog truyền thống nếu log được vận chuyển thông qua UDP. Trong trường hợp này, syslog mới và các syslog truyền thống sẽ được dùng chung để phát triển mạng, và một syslog mới sẽ được phát triển trên log-collection server. Những đặc điểm tiến bộ khác như lọc, kiểm tra tích hợp, sưu tập dữ liệu là có sẵn, và chỉ cách chuyển vận của các log mạng là được làm theo cách cổ điển mà thôi.

18.2.2 Windows

Windows (từ NT/2000/XP trở lên) cũng cung cấp logging hệ thống. Tuy nhiên, nó sử dụng định dạng nhị phân (*.evt) để lưu trữ 3 dạng logfile:hệ thống, ứng dụng và an ninh (system, application, and security). Figure 18-1 là 1 ví dụ của log an toàn của hệ thống windows . Log hệ thống bao gồm rất nhiều các bản ghi có liên quan tới các vận hành thông thường hoặc bất thường của máy tính. Ví dụ này chỉ ra 1 hoạt động thông thường của Windows XP. Xem chi tiết ở hình (Figure 18-2). Để đọc các log của windows, bạn cần sử dụng chương trình hoặc thiết bị có thể đọc được file *.evt . Thiết bị đọc có thể sử dụng để xuất các file ra dưới dạng mỗi giá trị cách nhau 1 dấu phẩy cho việc phân tích hoặc quan sát log qua các text editor.

Figure 18-1. Windows security log showing normal operation
Figure 18-2. Double-clicking to drill down for detail on the Windows security log

18.2.3 Remote Covert Logging

Một chương về logging sẽ không đầy đủ nếu thiếu phần nói về logging chuyển đổi. Trong một vài trường hợp (giống như cho honeypots và cho những kịch bản khác), thật là đáng mong ước che dấu đi sự có mặt của một logging tập trung từ xa khỏi những người khách của bạn. Thông thường, file cấu hình syslog bộc lộ sự hiện diện của logging từ xa và chỉ ra vị trí logging server. Điều này cho phép các hacker có thể tấn công, dò xét các log server và xóa đi những vật chứng. Mặt khác, stealthy logging lại rất khó để cho 1 kẻ tấn công có thể phát hiện ra.

Lựa chọn stealthy logging cơ bản nhất thực sự lại không phải là vụng trộm. Nó chỉ cung cấp 1 site backup cho việc lưu trữ log. Thêm vào việc chỉ định log server (có thể nhìn thấy đối với những kẻ tấn công), 1 sniffer (giống như Snort IDS trong chế độ lắng nghe, tcpdump, hoặc ngrep) được phát triển trên những thiết bị riêng rẽ. Ví dụ như, nếu server có địa chỉ IP là 10.1.1.2 gửi logtới 1 server có địa chỉ 10.1.1.3, một thiết bị đặc biệt khác không có địa chỉ IP sẽ được phát triển trên cùng subnet mà sniffer đang chạy. Tất cả các sniffer đều được cấu hình bằng ngôn ngữ Berkeley Packet Filter (BPF) để nhận những thông tin xác định. Trong trường hợp này, chúng ta sẽ chạy lệnh tương tự như:
ngrep "" src host 10.1.1.2 and dst host 10.1.1.3 and proto UDP and port 514 >

/var/log/stealth-log
Lệnh này cho phép sniffer (trong ví dụ này là ngrep, có sẵn tại địa chỉ http://ngrep.courceforge.net) để lưu lại chỉ những chuyển dịch syslog từ xa giữa 2 host xác định và đổ dữ liệu vào file /var/log/stealth-log.
Rõ ràng rằng, công cụ tcpdump có thể được sử dụng để ghi lại tất cả những syslog dưới các định dạng nhị phân hoặc ASCII , nhưng ngrep dường như làm tốt hơn trong công việc này, bởi vì nó chỉ hiển thị những phần được phép của syslog packet.

Chọn lựa stealthy log thứ 2 gửi file log tới 1 host log mà không chạy syslog (hoặc là bất kỳ một dịch vụ mạng nào khac). Trong trường hợp này, firewall chạy trên log server chỉ đơn giản từ chối mọi đầu vào có gói tin UDP cổng 514. Bạn sẽ thắc mắc nó sẽ thiết lập logging như thế nào? Một sniffer mà sẽ kiểm tra tất cả các gói tin UDP trước khi nó bị firewall đẩy ra được phát triển trên chính log server đó. Sẽ không có một ứng dụng nào trên host có thể nhìn thấy gói tin đó bởi vì nó đã bị firewall đẩy ra, sniffer ghi nó vào 1 file (sử dụng câu lệnh trên).

Nó có thể được thực thi để tránh viễn tưởng hack log server. Thực tế thì chúng ta đã vừa thiết lập nên một cái bẫy honeypot; những thông điệp được chuyển tới router (cái mà hiển nhiên không quan tâm đến việc thông tin nhận được có là một thông điệp syslog hay không). Một người có thể chỉ ra dòng thông điệp ở một nơi nào đó, nhưng sử dụng một host mà không có syslog tạo nên lợi ích trong việc làm cho những kẻ tấn công bị rối ren (và phải cân nhắc xem lỗi cấu hình trên 1 phần của system administrators).
Phần thứ 3, lựa chọn stealthy logging cuối cùng liên quan đến việc chuyển dữ liệu log tới một host không còn tồn tại và sau đó chọn lọc dữ liệu với 1 sniffer giống như trên. Trong trường hợp này, một thiết lập mở rộng nên thay đổi trên thiết bị gửi logfile: stack TCP/IP nên trang trí các gói tin được gửi đi tới thiết bị mà sẽ không bao giờ trả lời (vì nó không tồn tại). Tất cả những cái này được biểu diễn hoàn chỉnh trong câu lệnh sau:
arp -s 10.1.1.4 0A:0B:OC:OD:78:90

Câu lệnh này sẽ trang trí IP stack của thiết bị gửi log sao cho người ta nghĩ rằng có một cái gì đó đang chạy tại địa chỉ 10.1.1.4. Trong trường hợp này, cả địa chỉ IP và địa chỉ MAC đều có thể không có thật, nhưng địa chỉ IP nên là 1 địa chỉ mạng cục bộ. Hãy lưu ý rằng địa chỉ MAC không cần thiết phải thuộc vào một log server thực tế nào đó.

Lựa chọn 1 server không tồn tại là hiệu quả hơn nếu 1 mức độ cao hơn của stealth là cần thiết. Phương pháp này có thể không áp dụng được cho 1 mạng Lan truyền thống, nhưng nó có thể được ứng dụng trong rất nhiều trường hợp đặc biệt khác.

18.2.4 Những kiểu Logging khác

Để kết luận, hãy lần nữa nhìn lại những Unix logfiles khác. Thêm vào các Unix syslogd chuẩn và klogd logging daemons, còn có 1 tiến trình tính toán BSD thường xuyên được nhìn thấy trên các hệ thống Linux, Solaris và BSD khác. Tính toán tiến trình lưu các tiến trình được chạy trên hệ thống Unix và lưu trữ dữ liệu trong các file nhị phân. Một vài tiện ích được cung cấp để kiểm tra dữ liệu, giống như trong ví dụ sau:
lastcomm S X root stdin 3.19 secs Sat Nov 2 22:16

head S root stdin 0.00 secs Sat Nov 2 22:16

egrep root stdin 0.01 secs Sat Nov 2 22:16

grep S root stdin 0.01 secs Sat Nov 2 22:16

bash F root stdin 0.00 secs Sat Nov 2 22:16

bash SF root stdin 0.00 secs Sat Nov 2 22:16

dircolors root stdin 0.00 secs Sat Nov 2 22:16

stty root stdin 0.00 secs Sat Nov 2 22:16

bash SF root stdin 0.00 secs Sat Nov 2 22:16

tput root stdin 0.01 secs Sat Nov 2 22:16

bash SF root stdin 0.00 secs Sat Nov 2 22:16

tput root stdin 0.01 secs Sat Nov 2 22:16

su anton stdin 0.04 secs Sat Nov 2 22:16

head anton stdin 0.01 secs Sat Nov 2 22:16

Những bản ghi trên (được tạo ra bởi lệnhlastcomm | head -20) chỉ ra rằng những lệnh trên bao gồm grep, egrep, bash, và thậm chí cả chính bản thân lệnh lastcom đều chạy trên thiết bị dưới tài khoản root và người sử dụng có tài khoản anton được chuyển đổi thành roor bằng cách sử dụng lệnh su vào lúc 10.16PM ngày 2 tháng 11. Phần nhị phân này của bảng thống kê Unix hoàn thiện bức tranh mà được cung cấp bởi syslog bằng cách thêm và những tiến trình đang chạy một cách chi tiết nhất. Thật không may mắn, không có thiết bị nào cho việc chuyển dịch từ xa những bản ghi đã được liệt kê đó.

Quy trình logging hệ thống Unix cóthể được tích hợp trong những thiết bị chạy trên hệ điều hành Windows bằng các giải pháp như Kiwi Syslog, miễn phí tại http://www.kiwisyslog.com.

Nhìn chung, biên dịch thông điệp Unix trở nên dễ dàng hơn sau khi bạn có được quyền kiểm soát hệ thống. Thử thách đối với việc phân tích log đó là tái tạo lại một bức tranh hoàn chỉnh của việc phát hiện từ các log được thu thập bởi những thiết bị khác nhau trên toàn mạng, khi đưa vào tài khoản đó những sự kiện xuất hiện trong một quá trình trước đó.

18.3 Trạng thái logging

Trong phần này chúng ta sẽ tổng hợp xem những ví dụ ở trên và những log khác trong một bức tranh chung những gì mà bạn có thể trong mong nhìn thấy trong 1 file log. Sự miêu tả này nằm trong 1 phần của đoạn của Tina Bird gửi tới mailing list phân tích log của cô ấy (xem phần reference) và việc thảo luận được đảm bảo, cái mà được thiết lập từ tác giả của cuốn sách này).

Một vài sự kiện mà máy tính có thể đặt vào log:
• Tắt, mỏ, restart hoặc bất cứ 1 hành động liên quan đến đầu cuối của hệ thống hoặc 1 phần mềm.
• Various thresholds được thực thi hoặc các cập tìm kiếm nguy joểi. giống như đầy dung lượng đĩa, exhausted bộ nhớ hoặc bộ xử lý hoạt động quá nhanh.
• Phần cứng thông báo răng hệ thống có thể gặp vấn đề hoặc có thể phát hiện được và ghi log.
• Người dùng truy cập vào hệ thống, có thể là đăng nhập từ xa (telnet, SSH,,,) và các đăng nhập nội bộ hoặc truy cập network (FTP) tới hoặc từ 1 hệ thống khác kể cả thành công hay không thành công.
• Người dùng truy cập đến một thay đổi đáng kể (privilege) giống như lệnh su – kể cả thành công hay thất bại.

• Thay đổi credential người dùng hoặc quyền truy cập, giống như cập nhật tài khoản, tạo mới hoặc xóa bỏ , kể cả thất bại hay thành công.
• Thay đổi thiết lập hệ thống và update phần mềm, kể cả thành công hay không thành công.
• Truy cập vào log của hệ thống để chỉnh sửa, xóa hoặc thậm chí là chỉ đọc.

Danh sách các sự kiện nêu trên có thể đầy đủ cho log của 1 hệ thống và sẵn sàng cho việc phân tích. Công việc cả bạn là cố gắng trả lời câu hỏi “Chuyện gì đã xảy ra” sử dụng tất cả các bản ghi tiềm năng, phức tạp đó.

18.4 Khi nào cần phải quan sát các Log

Một người mới bắt đầu nên bắt đầu từ việc quan sát chung một lượt tất cả những thông tin nhận được để đưa ra sự chú ý thích hợp. Có thể, chỉ là có thể thôi, liệu bạn có thể bỏ qua tất cả mà không cần phân tích dữ liệu hay không? Câu trả lời dường như là KHÔNG. Một quy ước đơn giản nhất của việc phân tích log đó là bạn không ghi nhận những gì mà bạn không có kế hoạch tìm kiếm trên đó. Hoặc là như quy ước Murphy "Chỉ tìm kiếm những vấn đề mà bạn có thể biết cách giải quyết”. Trong lĩnh vực an toàn thông tin, đó có nghĩa là bạn chỉ tìm kiến những gì bạn đã có kế hoạch để trả lời và chỉ ghi nhận những gì mà bạn cần tìm kiếm trên nó. Ví dụ như, 1 hệ thống phát hiện xâm nhập (đề cập ở chương 19) chỉ làm việc tốt khi mà có người phân tích xem xét những đầu ra của nó. Bởi vậy, nếu bạn không có hiểu gì về "WEB-CGI webdist.cgi access" bạn sẽ không thể chạy được Snort với các quy ước được cho phép. Tạo nên một hoạt động được đánh giá cao dựa trên kết quả sẽ là không thể nếu bạn không hiểu rõ chuyện gì đang xảy ra và những hành động mà được đánh giá cao đó có thể trở thành circumstances.

Thiết bị này không negate rằng việc logging tất cả mọi thứ đều là cần thiết cho 1 động thái điều tra và tìm kiếm. Thực tế, nếu log có thể sử dụng cho tất cả các hồi đáp đối với các sự kiện, thì rule giống như "dont log what you wont look at" sẽ không bao giờ được thực hiện. Trong nhiều trường hợp, logging tất cả mọi thứ là 1 router tốt nhất, bởi vì nó dường như ghi nhận tất cả các bit tín hiệu mà cho phép bạn giải quyết vấn đề. Chúng tôi chỉ muốn nói rằng, nếu logfile không bao giờ được nhìn vào (hoặc đơn giản là quay lại bởi 1 chương trình log nào đó) thì nó sẽ chẳng có tác dụng gì..
Hãy cân nhắc trường hợp một hệ thống máy gia đình hoặc máy văn phòng. Trong trường hợp này, log chỉ có tác dụng chính trong những vấn đề của hệ thống chính (ví dụ như phần cứng hoặc là lỗi của hệ điều hành) hoặc là các vấn đề an ninh hệ thống (những vấn đề mà rất dễ có thể ngăn ngừa bởi vì bạn chỉ phải xem xét trên một hệ thống riêng lẽ hoạc chỉ 1 số lượng rất nhỏ các hệ thống. Thậm chí trong những trường hợp này, bạn bắt buộc phải nhìn vào log nếu nó có hi vọng giải quyết được các vấn đề hoặc là ngăn ngừa tác hại của nó. Tuy nhiên, bạn sẽ tốn ít thời gian hơn nếu ngồi cài lại hệ điều hành Windows của bạn, hoặc là thay thế nó bởi Unix. Chúng tôi không khuyên bạn cứ chăm chú vào các file log để tìm các dấu hiệu tiềm năng của 1 vụ xâm nhập ngoại trừ khi bạn thích thú đối với công việc đó hoặc là bạn đang chuẩn bị để lấy 1 chứng chỉ cho việc phân tích xâm nhập nào đó. Chỉ nên cho phép logging một lượng nhỏ cần thiết nào đó.

Tiếp theo, chúng ta sẽ xem xét một business cỡ vừa và nhỏ, mà được chỉ ra rằng sẽ không có nhân viên an ninh. Các hành động để đảm bảo an toàn hệ thống được giới hạn trong “gỡ bỏ các vấn đề”. Trong trường hợp này, nó giống như hệ thống gia đình với những khác biệt không mấy quan trọng. Môi trường này cũng thường xuyên có mặt những người mà (atonish) việc chuyên nghiệp hpá các hành động bảo vệ an toàn hệ thống bởi những câu bình luận kểi như “Tại sao lại có những người muốn hack chúng ta? Chúng ta không làm gì hấp dẫn các hacker”. Ngày nay, tất cả mọi người hiểu rằng bộ nhớ hệ thống, vòng CPU và một kết nối mạng tốc độ cao thì có rất nhiều mối đe dọa về an toàn hệ thống cao. Và bởi vì những mối hiểm nguy có mức đe dọa thấp lại được nhiều người biết đến (chẳng hạn như một người nào đó thực hiện việc scan các cổng) lại có thể được cảnh báo như một cuộc tấn công nghiêm trọng (như là cố gắng xâm nhập hệ thống), do đó, một công ty nhỏ hiếm khi có nguồn nhân lực đủ mạnh và có kỹ năng để khai thác chúng

Một công ty lớn hơn sẽ có nhiều yêu cầu quản trị hơn là 1 cá nhân riêng rẽ. Do vậy mà mức độ an toàn và khả năng accountability được nâng cao hơn. Tất cả các tổ chức kết nối đến Internet ngày nay đều có ít nhất 1 firewall và 1 vài bộ DMZ được cài đặt cho các server public như web, email, FTP, đăng nhập từ xa. Rất nhiều tổ chức đã phát triển những hệ thống phát hiện xâm nhập và các mạng riêng ảo (VPNs). Tất cả những công nghệ tiên tiến đó làm gia tăng những mối quan tâm mới như sẽ phải làm gì với tất cả những tín hiệu thu được từ chúng, và các công ty hiếm khi thuê những nhân viên an ninh hệ thống mới chỉ để giải quyết những tín hiệu đó. Các logs biểu diễn một trong những các phát hiện ra các mối đe dọa từ các hostile Internet.
T
óm lại, trả lời cho câu hỏi “Tôi có phải làm như thế này không” được thay đổi từ “Có thể không” đối với các giao dịch nhỉ cho đến “Vâng, bạn phải làm như vậy” đối với những giao dịch lớn..

18.5 Log Overflow and Aggregation

Thông tin từ các log file là rất đa dạng và phong phú, tuy nhiên thật không may mắn là rất nhiều những thông tin là rất phức tạp để phân tích. Lượng dữ liệu hàng gigabite thông tin được thu thập là không bất thường đối với một công ty lớn, đặc biệt nếu lượng thông tin chuyển dịch trên mạng được log lại. Trong khi tồn tại nhiều phương pháp để lưu trữ lượng thông tin đó, thì việc làm cho chúng trở nên có thể phân tích được và có thể ứng dụng trong những thiết bị giám sát lại là một câu chuyện khác. Có được những log nhờ những thiết bị thu thập tại cùng 1 địa điểm làm cho gia tăng tổng thể những thông tin thu thập được, tuy nhiên lại đơn giản hóa việc tồn tại hàng ngày và những phản hồi đối với các sự kiện đột xuất nhờ vào tốc đô truy cập log nhanh chóng. Việc thống kê hiệu quả, lưu trữ an toàn và có khả năng phân tích là một trong những sự thuận tiện của việc tập trung các log thu được. Thêm vào đó, việc lưu trữ log một cách an toàn và ít bị thay đổi rất có ích nếu một kẻ xâm nhập bị phát hiện ra dựa trên những chứng cớ log. Trong trường hợp này, những tài liệu minh chứng cẩn thận của 1 chương trình ghi log là có thể rất cần thiết

Trong khi việc tập trung log của hệ thống Unix có thể đạt được dễ dàng nhờ syslog chuẩn, sự thay thế syslog cũng có thể làm việc một cách tốt hơn. Việc tập trung log giúp hỗ trợ cho rất nhiều mục đích trong quá trình biên dịch, mặt khác nó làm cho hệ thống trở nên an toàn hơn. Một kẻ xâm nhập cần phải tấn công một hoặc nhiều server hơn mới có thể xóa được những dấu vết của anh ta. Mặt khác, nó cũng làm cho hệ thống trở nên thuận tiện hơn, người quản trị mạng chỉ cần đơn giản kết nối với một thiết bị để xem tất cả những logfile từ mạng. Tuy vậy, có rất nhiều vấn đề xảy ra đối với việc tập trung các log, quan trọng nhất đó là phải giải quyết 1 lượng rất lớn những thông tin log.

18.6 Những thử thách đối với việc phân tích log

Sau khi bỏ rất nhiều thời gian và công sức để tổng hợp và phân tích log, hãy thử đóng vai trò biện hộ và đưa ra những chứng cứ để cố gắng chứng minh một vài lợi ích của nó.

Chúng ta cho rằng những sự việc về an ninh thông tin được điều tra bằng các logfile, tuy nhiên giả thiết đó có thể chỉ là việc đặt ra những câu hỏi. Một vài nguồn cho thấy rằng tất cả mọi hacker đáng giá như Mountain Dew không bao giờ để lại dấu vết trong các log và dễ dàng bỏ qua những hệ thống phát hiện xâm nhập. Nếu những hành động không bị ghi nhận lại thì bạn không thể phân tích chúng. Thêm vào đó, thiết kế hạ tầng cho logging được những kẻ tấn công biết đến để có thể thao tác trên các logfile và có thể chúng đã bị xóa bởi tất cả những kẻ tấn công muốn xóa bỏ dấu vết sau khi thâm nhập hệ thống. Một lần nữa, nếu bạn cho phép kẻ xâm nhập xóa log thì bạn cũng không thể phân tích chúng.

Những chuyện đó thường xuyên xảy ra (trong thực tế, nó đã từng xảy ra đối với chính tác giả) và một người điều tra xuất sắc nhạy cảm với các sự kiện máy tính, thì hành động đầu tiên của ông ta là: “Đầu tiên, hãy nhìn vào log hệ thống”. Tuy nhiên, cho dù là ông ta tìm kiếm đến đâu thì cũng không thể tìm thấy. Việc logging cũng không được mặc định là cho phép hoặc là bị điều chỉnh trực tiếp /dev/null bởi con người không muốn nhìn thấy bộ nhớ bị chiếm dụng. Vậy giải pháp là gì? Thực tế là không chỉ có 1. Nếu việc ghi nhậnlog không được sẵn sàng cho đến khi bạn cần nó thì bạn cũng không thể phân tích được nó.

Thậm chí tồi tệ hơn, thỉnh thoảng 1 số dấu hiệu của kẻ xâm nhập trong những file hệ thống,ví dụ như, 1 địa chỉ IP của một người đã kết nối vào hệ thống có quyền khai thác trong thời điểm mà sự việc xảy ra. Tuy nhiên, nếu tất cả bạn có chỉ là 1 địa chỉ IP thì liệu bạn có thể chứng minh được điều gì? Rất dễ để thuyết phục 1 sự việc xảy ra đáp trả lại khi họ thực hiện việc chặn bắt đường truyền bằng một phiên của thiết bị ghi nhận 1 công cụ thâm nhập. Nhưng trong thực tế, log không phải lúc nào cũng có được thông tin chi tiết. Nếu log không đủ chi tiết để rút ra kết luận về dữ liệu thì bạn cũng không thể phân tích chúng.

Việc phân tích log thường xuyên phải thực hiện cho dẫu những khó khăn đó luôn xảy ra. Tuy nhiên, nó dường như buộc chúng ta phải luôn suy nghĩ về chúng. Nếu logging tất cả mọi thứ không phải là 1 lựa chọn (do giới hạn bộ nhớ, đường truyền hoặc ứng dụng) thì chúng ta chỉ phân tích được trên những gì có được và cố gắng để có được một kết luận đầy đủ dù cho luôn có những khó khăn đó.

Như chúng ta đã đề cập, có rất nhiều công cụ để có thể phân tích các log. Tuy nhiên, trong chương này chúng tôi chỉ giới thiệu giải pháp SIM (Quản lý thông tin an toàn)

18.7 Quản lý thông tin an toàn - SIM

Những công cụ SIM tập hợp, làm bình thường hóa, giảm thiểu, phân tích và liên kết rất nhiều log từ bộ biên dịch. Các sự kiện an toàn thông tin được tập hợp từ tất cả các thiết bị sản xuất ra logfile như firewall, thiết bị phát hiện xâm nhập, hệ thống bảo vệ, các công cụ ngăn chặn virus cũng như các server và các ứng dụng .

Đầu tiên, các bản ghi log được chuyển đổi sang 1 định dạng thông thường, thường là sử dụng định dạng XML, Thứ 2, nó sẽ được giảm đi một cách thông minh kích thước, đóng gói vào những loại khác nhau và chuyển dịch tới 1 điểm thu thập trung tâm (thường là một cơ sở dữ liệu quan hệ) để cho những lưu trữ và phân tích khác. Thêm vào đó, các sự kiện có thể được liên kết bằng các quy ước và phương pháp thống kê liên kết.

Cuối cùng, các sự kiện được biểu diễn sử dụng một giao diện đồ họa thời gian thực. Các công cụ như netForensics (http://www.netForensics.com) có thể thực hiện hàng ngàn sự kiện an toàn thông tin trong 1 giây và liên kết chúng lại trong thời gian thực cũng như ciung cấp cho chúng khả năng phân tích và long term trending.

Một số công cụ cho phép phân tích thời gian thực và phúc hồi một lượng lớn những sự kiện. Chúng có thể biên dịch để tránh việc phải cảnh báo rằng những gì đang diễn ra trong môi trường IP của chúng, cũng như bị cảnh báo bởi các mối đe dọa mà nó đang phải đối mặt.

Tuy nhiên, việc thu thập các sự kiện từ hàng ngàn thiết bị phát triển trên toàn thế giới có thể dẫn đến việc làm quá tải 1 công cụ rất mạnh. Vẫn còn những chuyên gia tin rằng, có nhiều cuộc tấn công mới có thể phòng ngừa được nếu các thiết bị từ nhiều nơi trên thế giới có thể được logging vào một hệ thống trung tâm nào đó. Bởi vậy, một sự tích hợp log toàn cầu là cần thiết.

18.8 Tích hợp log toàn cầu (Global Log Aggregation)

Một chương về việc phân tích log sẽ không hoàn thiện nếu thiếu đề cập đến vấn đề tích hợp log toàn cầu. Rất nhiều tổ chức và công ty đã thu thập các logfile và sẵn sàng chia sẻ chúng, và sau đó họ phân tích toàn thể dữ liệu.SANSs Dshield.org (http://www.dshield.org), MyNetWatchMans Watchman (http://www.mynetwatchman.com), and Symantecs DeepSight Analyzer (https://analyzer.securityfocus.com) thu thập rất nhiều logs từ các firewall cá nhân đến các firewall của các công ty lớn và các hệ thống phát hiện xâm nhập. Các dịch vụ được cung cấp đa dạng trên giao diện web cho việc phân tích và quan sát log. Thêm vào đó, nếu phát hiện thất có 1 hành động đáng nghi, tất cả chúng sẽ thông báo tới người phụ trách ISP của bạn, và điều đó có thể làm cho kẻ tấn công bị mất tài khoản của mình.

Lợi ích của dịch vụ kiểu này là cho 1 tập thể không phải cho những cá nhân người sử dụng. Việc giải quyết một lượng rất lớn dữ liệu log cho phép tổ chức đó có thể phát hiện ra những mối đe dọa trên mạng đối với hệ thống của họ từ rất sớm. Chúng ta có thể nhìn thấy điều này trong thực tế khi Dshield folks phát hiện ra sự phân tán của CodeRed năm 2001 và một loại MSSQL worm vào năm 2002. Con số phát triển về mặt số học của sự truy cập đến cổng (ví dụ như cổng 80 đối với CodeRed và cổng 1433 đối với SQL worm) đã đưa ra gợi ý rằng tất cả những sự tấn công tụ động đều bị thất bại. Một hệ thống cảnh báo sớm cho phép các nhà phân tích an ninh có thể bắt được, nghiên cứu được về loại worm đó và đưa ra giải pháp trước khi chúng có thể vượt ra ngoài tầm kiểm soát. Chúng tôi lưu ý rằng bạn nên cân nhác 1 trong những dịch vụ này để có thể quen thuộc hơn với dữ liệu log của bạn và để xây dựng 1 mạng internet an toàn hơn.