Hệ thống phát hiện xâm nhập (1)

Tequila (VietHacker.org Translator Group Leader)

Compose by hieupc (PDF)

Hệ thống phát hiện xâm nhập (IDSs) cung cấp thêm cho việc bảo vệ an toàn thông tin mạng một mức độ cao hơn. Nó được đánh giá giá trị không giống như firewall và VPN là ngăn ngừa các cuộc tấn công mà IDSs cung cấp sự bảo vệ bằng cách trang bị cho bạn thông tin về cuộc tấn công. Bởi vậy, 1 IDS có thể thoả mãn nhu cầu về an toàn hệ thống của bạn bằng cách cảnh báo cho bạn về khả năng các cuộc tấn công (và thỉnh thoảng thì ngoài những thông báo chính xác thì chúng cũng đưa ra một số cảnh báo chưa đúng.

Nhìn chung, IDSs không tự động cấm các cuộc tấn công hoặc là ngăn chặn những kẻ khai thác 1 cách thành công, tuy nhiên, một sự phát triển mới nhất của IDS đó là hệ thống ngăn chặn xâm nhập (the intrusion prevention systems) đã có để thực hiện nhiều vai trò hơn và có thể ngăn chặn các cuộc tấn công khi nó xảy ra.

Định nghĩa 1 IDS khó hơn là chúng ta tưởng. Đầu tiên, IDS được nhìn nhận như là một cái chuông báo trộm mà có thể thông báo cho bạn biết khi nào thì bạn bị tấn công. Tuy nhiên, những hệ thống IDS hiện đại thì phức tạp hơn nhiều và ít người có thể đồng ý rằng nó có mức độ giống như một cái chuông báo trộm truyền thống đáng tin cậy. Nếu sự giống nhau là cùng được sử dụng, thì một hệ thống IDS trông giống như những chiếc camera chống trộm hơn là 1 cái chuông, những người có trách nhiệm có thể quan sát chúng và đáp trả cho những đe doạ xâm nhập.

Thực tế thì dường như IDS chỉ nói cho chúng ta biết rằng mạng đang bị nguy hiểm. Và điều quan trọng để nhận ra đó là một vài cuộc tấn công vào mạng đã thành công nếu hệ thống không có IDS. Và như chúng ta đã thấy, một mạng có thể trở thành thiên đường cho các hacker trong hàng năm mà chủ nhân của nó vẫn không hề hay biết.

Giá trị chính của 1 hệ thống phát hiện xâm nhập theo quan điểm của chúng tôi đó là nó biết được chuyện gì sẽ xảy ra. Phải, 1 hệ thống IDS có thể giúp chúng ta ngăn ngừa các sự kiện khi nó chưa xảy ra, cung cấp các giải pháp cho mạng và host, và thậm chí cũng có thể hoạt động như một cái chuông báo động (với những giới hạn tương ứng). Tuy nhiên, chức năng chính của nó là thông báo cho bạn biết về các sự kiện có liên quan đến an ninh hệ thống đang sắp sửa xảy ra bên trong mạng và hệ thống mà bạn kiểm soát.

Trong chương này sẽ cho chúng ta cái nhìn tổng quan về IDS bao gồm cả những điểm mạnh và điểm yếu của chúng. Chúng ta sẽ đề cập đến cả network IDS (nhiều khi được đề cập đến như 1 sniffer) và cả host IDS (phân tích log, kiểm tra tích hợp và nhiều thứ khác) .

Sự khác nhau chủ yếu giữa network IDS và host IDS đó là dữ liệu mà nó tìm kiếm. NIDS nhìn vào toàn cảnh các chuyển dịch trên mạng, trong khi host IDS thì quan sát các host, hệ điều hành và các ứng dụng. Trong thực tế, nó được chia cắt ra nhiều lĩnh vực khác nhau, chẳng hạn như host IDS ngăn chặn các truy cập có hại cho mạng, còn NIDS thì cố gắng đoán xem cái gì xảy ra bên trong host.Có một vài giới hạn không rõ nét lắm như công nghệ để phát triển tiếp theo.

Vậy những thuận tiện của Host-base IDS là gì? Sự khác nhau cơ bản giữa chúng đó là trong khi NIDS phát hiện ra các cuộc tấn công tiềm năng (những thứ sẽ được chuyển tới đích) thì host IDS lại phát hiện ra những cuộc tấn công mà đã thành công, có kết quả. Bởi vậy có thể nói rằng NIDS mang tính tiền phong hơn. Tuy nhiên, 1 host IDS sẽ hiệu quả hơn đối với trong các môi trường có tốc độ chuyển dịch lớn, mã hoá và có chuyển mạch - đây là những môi trường mà NIDS rất khó hoạt động. HIDS được thử thách bởi rất nhiều những hành động có mức độ phơi bày cao của kẻ tấn công và đã thực sự nâng tầm xử lý của chúng.

Mặt khác thì NIDS lại là 1 phần rất tuyệt cho môi trường tổng hợp như toàn bộ mạng. Vì thế, NIDS có thể tạo nên một sự quan sát có ý nghĩa đến các phần của vụ tấn công có liên quan đến nhiều host. Nó được thử thách trong môi trường mạng có chuyển mạch tốc độ cao, môi trường mã hoá và các giao thức ứng dụng hiện đại phức tạp, bởi vậy nên các kết quả báo sai cũng hất có khả năng xảy ra.

Bởi vậy, chúng tôi khuyên các bạn nên lựa chọn công nghệ IDS và bởi vậy cung cấp cho chúng ta lựa chọn bổ sung chúng vào mạng của bạn như phân tích Bayesian. Chúng tôi cũng quan tâm đến việc những thay đổi tương lai trong công nghệ IDS có thể mang lại. Cuối cùng chúng tôi sẽ miêu tả một cách đầy đủ việc bổ sung mã nguồn trên Linux.

19.1 Ví dụ về IDS

Phần này sẽ miêu tả một vài hệ thống IDS bao gồm giám sát logfile, quét các dấu hiệu và phát hiện các dấu hiệu bất thường.

19.1.1 Host IDSs

Host-based network IDSs có thể được phân chia lỏng lẻo thành các kiểm soát log, kiểm tra độ tích hợp và các module nhân của hệ thống. Những phần sao sẽ miêu tả từng phần của chúng với các ví dụ cụ thể.

19.1.1.1 Giám sát Logfile :

Một IDS đơn giản nhất là thiết bị giám sát logfile (logfile monitors), thiết bị này sẽ cố gắng phát hiện những sự xâm nhập bằng cách phân tích các log sự kiện của hệ thống. Ví dụ như, một thiết bị giám sát logfile sẽ tìm kiếm những logfile ghi nhận những truy cập Apache để truy cập tới Apache để tìm ra đặc điểm của yêu cầu /cgi-bin/ . Công nghệ này bị giới hạn bởi vì nó chỉ tìm kiếm trong các sự kiện đã được log - là những thứ mà kẻ tấn công rất dễ để thay thế. Thêm vào đó, hệ thống có thể bỏ qua các sự kiện hệ thống cấp thấp mà chỉ ghi lại các hoạt động cấp cao.Ví dụ như, HIDS sẽ bỏ qua nếu kẻ tấn công chỉ đọc nội dung file như file /etc/passwd chẳng hạn. Điều này sẽ xảy ra nếu bạn không đặt file vào chế độ bảo vệ của hệ thống. Giám sát Logfile là một ví dụ chính cho các hệ thống IDS dựa trên host bởi chúng thực hiện chức năng giám sát của chúng trên chỉ 1 máy. Tuy nhiên, một hệ thống giám sát host logfile hoàn toàn có thể giám sát trên nhiều host, thậm chí trên 1 loggging server tích hợp. Sự phát triển của nền tảng host đưa lại một số thuận tiện cho việc giám sát với các công cụ hệ thống được xây dựng, bởi vì host IDSs có kênh chuyển dịch tổng hợp an toàn tới 1 server trung tâm, không giống như những syslog thông thường khác. Nó cũng cho phép tích hợp những logs mà không bình thường để tích hợp trong 1 máy đơn (chẳng hạn như log sự kiện của Windows.

Mặt khác, NIDS thường quét toàn mạng trên mức độ gói tin, trực tiếp từ đường truyền giống như những sniffer. Bởi vậy NIDS có thể phối hợp với rất nhiều host có dữ liệu chuyển qua. Giống như những gì chúng ta thấy trong chương này, mỗi một loại đều có tác dụng và thuận tiện của chúng trong những trường hợp khác nhau.

Thiết bị giám sát logfile nổi tiếng đó là swatch (http://www.oit.ucsb.edu/~eta/swatch/), là nói tắt của "Simple Watcher." Trong khi hầu hết các phần mềm phân tích log chỉ quét log theo định kỳ, thì swatch quét tất cả các đầu vào log và tạo báo cáo cảnh báo theo thời gian thực. Những công cụ khác như logwatch (được tích hợp cùng với Red Hat Linux thì rất tốt cho các thao tác ngoài. Tuy nhiên, mặc dù swatch đi cùng với nhiều bước có liên quan thì nó vẫn đòi hỏi nhiều tính năng động và cấu hình khác với những công cụ khác.

Sau đây chúng ta sẽ miêu tả việc cài đặt swatch. Công cụ này khá là ổn định, do đó mà dường như không thay đổi nhiều trong tương lai. Trước khi cài đặt swatch, bạn cần download và cài đặt Perl modules cần thiết cho nó. Để cài đặt những module này, đầu tiên hãy download phiên bản mới nhất của swatch và chạy các bước sau:

```
perl Makefile.PL

make

make test

make install

make realclean
```

swatch sử dụng diễn dịch thông thường để tìm đến những dòng lệnh thích hợp. Một khi mà nó tìm đúng phần cần thiết, nó liền hành động, chẳng hạn như biểu diễn ra màn hình, email 1 cảnh báo hoặc là làm theo hành động đã được người sử dụng định ra từ trước.

Tiếp sau là 1 ví dụ script cấu hình swatch đơn giản:
watchfor /[dD]enied|/DEN.*ED/

echo bold

bell 3

mail

exec "/etc/call_pager 5551234 08"

Trong ví dụ này, swatch tìm đến dòng có chứa từ “denied”, có thể là “Denied” hoặc bất cứ từ nào có bắt đầu bằng “den” và kết thức với “ed”. Khi mà tìm thấy, nó bôi đen dòng tìm thấy và chuyển tới thiết bị đầu cuối đồng thời rung chuông 3 lần. Sau đó, swatch gửi mail tới người sử dụng swatch (là người có quyền truy cập tới các logfile được giám sát – thông thường được giới hạn cho root) với 1 cảnh báo và thực thi chương trình /etc/call_pager với các lựa chọn đã đươc đưa ra .

Giám sát logfile có thể được coi như 1 hệ thống phát hiện xâm nhập theo một cách đặc biệt. Logs cũng chứa rất nhiều thông tin không trực tiếp lên quan đến sự xâm nhập (chỉ là những thông tin mà NIDS nghe trộm được trên đường truyền). Logs có thể được coi như một cái bể lớn chứa thông tin, một số thông tin bình thường (như thông tin về kết nối của người chịu trách nhiệm, thông tin cấu hình lại daemon…) và những thông tin đáng ngờ chẳng hạn như thông tin về đăng nhập từ 1 IP động, truy cập root một cách kỳ lạ… và rất nhiều những thông tin (malicious) chẳng hạn như RPC buffer overflow được ghi nhận từ rpc.statd. Xem xét và chọn lọc toàn bộ những thông tin đó chỉ dễ hơn 1 chút so với lắng nghe trên mạng và tìm kiếm những cuộc tấn công vào web hoặc là các gói tin dị hình.

Nếu tất cả các ứng dụng đều có một hệ thống log an toàn mà tất cả các sự kiện xấu đều được ghi nhận và đóng gói, những người phân tích log có thể không cần đến 1 hệ thống phát hiện xâm nhập. Trong thực tế, nếu một sự kiện có thể được chỉ ra trong 1 file log hoàn chỉnh thì nó có thể là 1 sự xâm nhập. Tuy nhiên, trong đời thực thì việc tìm kiếm từng phần trong logs đôi khi cũng giá trị như việc tìm kiếm từng phần trên đường dẫn.

Thực tế thì việc đi kèm phân tích log hệ thống với NIDS log là một đặt điểm rất có ích đối với người phân tích log. Người phân tích sẽ nhìn thấy được nhiều hơn là chỉ nhìn trên đường dẫn và tạo ra các chức năng của meta IDS. Ví dụ như, giải pháp quản lý như netForensics cho phép phân tích log qua các thiết bị, bình thường hóa và liên kết chúng (bằng các phần dựa trên rule) sau đó phân tích các sự kiện đã được tổng hợp.

19.1.1.2 Giám sát tính toàn vẹn :

Một công cụ giám sát tính toàn vẹn sẽ nhin vào các cấu trúc chủ yếu của hệ thống để tìm sự thay đổi. Ví dụ như, 1 giám sát toàn vẹn đó sẽ sử dụng 1file hệ thống hoặc một khóa registry như "bait" để ghi lại các thay đổi bởi 1 kẻ xâm nhập. Mặc dù chúng có giới hạn, giám sát toàn vẹn có thể thêm vào các lớp bảo vệ cho 1 hệ thống phát hiện xâm nhập.

Giám sát toàn vẹn phổ biến nhất đó là Tripwire (http://www.tripwire.com). Tripwire có sẵn cho Windows và Unix, và nó chỉ có thể giám sát 1 số các thuộc tính như:
- Việc thêm, xóa, sửa đổi File
- Cờ File (i.e., hidden, read-only, archive, etc.)
- Thời gian truy cập cuối cùng
- Thời gian ghi cuối cùng
- Thời gian thay đổi
- Kích thước File
- Kiểm tra Hash

Khả năng của Tripwire là rất lớn trên Unix và Windows bởi vì các thuộc tính khác nhau của các hệ thống file. Tripwire có thể được thay đổi để phù hợp với các đặc điểm riêng biệt của mạng của bạn, và nhiều Tripwire agents có thể tập trung một cách an toàn các dữ liệu. Trong thực tế, bạn có thể sử dụng Tripwire để giám sát bất kì 1 thay đổi nào trên hệ thống của bạn. Bởi vậy, nó là một công cụ rất mạnh trong IDS

arsenal của bạn. Rất nhiều những công cụ khác (tất cả đều là miến phí và là các phần mềm mã nguồn mở) được viết để đáp ứng những công việc tương tự như thế. AIDE là 1 ví dụ . AIDE (http://www.cs.tut.fi/~rammer/aide.html) là một clone nổi tiếng của Tripwire.

Mấu chốt để sử dụng kiểm tra toàn vẹn hệ thống cho 1 thiết bị phát hiện xâm nhập đó là xác định ranh giới an toàn. Được thiết lập giống như 1 base line chỉ có thể được thiết lập trước khi hệ thống được kết nối với mạng. Nếu không có trạng thái an toàn thì công cụ này sẽ bị giới hạn rất nhiều, bởi vì những kẻ tấn công có thể đã giới thiệu những thay đổi của họ với hệ thống trước khi công cụ kiểm tra trọn vẹn hoạt động lần đầu tiên.
Trong khi hầu như tất cả mọi công cụ đều yêu cầu một trạng thái baseline trước khi bị tấn công thì một vài công cụ lại dựa trên hiểu biết của chúng về các mối nguy hiểm. Một ví dụ đó là công cụ chkrootkit (http://www.chkrootkit.org). Nó tìm kiếm những dấu hiện xâm nhập phổ biến mà thường hiển hiện trên các hệ thống bị tổn thương.

Kiểm tra toàn vẹn cung cấp một giá trị lớn nhất nếu chúng có được một vài thông tin hướng dẫn. Trước hết, nó phải được phát triển trên một hệ thống hoàn toàn sạch sẽ sao cho nó không phải ghi nhận các trạng thái dở dang hoặc bị tổn thương như thông thường. Ví dụ, Tripwire nên được cài đặt trên một hệ thống khi nó còn nguyên bản từ nhà sản xuất với những ứng dụng cần thiết nhất, trước khi nó kết nối tới mạng.

Bởi vậy, ý kiến về việc lưu trữ dữ liệu về trạng thái tốt trên các bản ghi được đặt trên các thiết bị lưu trữ chỉ đọc như CDROMs là một ý kiến rất hay. Chúng ta sẽ luôn có 1 bản copy đầy đủ để so sánh khi cần phải giải quyết vấn đề. Tui nhiên, mặc dù có tất cả những biện pháp phòng ngừa đó thì hacker vẫn có thể vượt qua được tất cả hệ thống như thế.

19.1.2 Network IDSs

Network IDSs có thể được phân chia thành 2 loại: hệ thống dựa trên các dấu hiệu và hệ thống dựa trên những sự việc bất thường. Không giống như hệ thống dựa trên dấu hiệu, hệ thống sau là 1 sự pha lẫn giữa những công nghệ khác nhau và gần như nhau. Thêm vào đó, những NIDSs lai tạo đó đều nhắm tới việc làm cầu nối cho những thiếu sót bằng cách sử dụng những mánh lới được sử dụng trong mỗi loại NIDSs. Trong thực tế, tất cả những hệ thống NIDSs thương mại hiện đại đều sử dụng loại NIDS dựa trên những sự việc bất thường để phát triển NIDS dựa trên dấu hiệu. Ví dụ như ISS RealSecure, Cisco IDS, and Enterasys Dragon.

19.1.2.1 Signature matchers

Giống như những phần mềm quét virus truyền thống dựa trên chữ ký hex, phần lớn các IDS đều cố gắng phát hiện ra các cuộc tấn công dựa trên cơ sở dữ liệu về dấu hiệu của tấn công. Khi 1 hacker tìm cách khai thác lỗ hổng đã biết thì IDS cố gắng để đưa lỗi đó vào cơ sở dữ liệu của mình. Ví dụ như Snort (http://www.Snort.org), một IDS dựa trên dấu hiệu miễn phí được phát triển trên cả Unix và Windows.

Bởi vì nó là một phần mềm mã nguồn mở nên Snort có tiềm năng phát triển cơ sở dữ liệu chữ ký nhanh hơn bất kỳ một công cụ có sở hữu nào khác. Các dấu hiệu của Snort được sử dụng trong tất cả mọi thứ từ các firewall thương mại đến các phần mềm middleware như Hogwash. Snort bao gồm 1 bộ giải mã các gói tin, 1 thiết bị phát hiện, và 1 hệ thống nhỏ logging và cảnh bá. Snort là 1 IDS trạng thái , có nghĩa rằng nó có thể tập hợp lại và ghi nhận các tấn công dựa trên phân đoạn TCP.

Một vài bạn đọc có thể đã gặp nhiều khái niệm 1 firewall đa trạng thái hoặc firewall không trạng thái nhiều hơn là 1 hệ thống phát hiện xâm nhập. Tuy nhiên, cả 2 khái niệm đều như nhau. Firewalls không trạng thái (và NIDSs) làm việc với các gói tin riêng rẽ trong khi 1 firewall trạng thái lại cân nhắc đến các trạng thái kết nối. Ví dụ đơn giản nhất như sau: Nếu 1 kẻ tấn công chia nhỏ các gói tin, thì IDS không trạng thái sẽ bỏ lỡ nó (bởi vì 1 dấu hiệu không bao giờ xuất hiện trong 1 gói tin), tuy nhiên nó lại bị thiết bị IDS trạng thái phát hiện được bởi vì nó thu thập các phần đáng nghi không chỉ dựa trên 1 gói tin mà trên cả dòng dữ liệu trong quá trình kết nối.

Tuy nhiên, những NIDs trạng thái cũng không tránh khỏi việc bỏ lỡ những dấu hiệu xâm nhập. Trong chương này chúng tôi sẽ cung cấp 1 vài ví dụ.
Ví dụ cơ bản nhất cho dấu hiệu để phát hiện của IDS liên quan đến 1 cuộc tấn công web đó là dựa trên lỗi CGI scripts. Một công cụ phát hiện lỗi của hacker thường xuyên bao gồm việc quét lỗi CGI để phát hiện

những web server có lỗi CGI . Ví dụ như, một lỗi rất nổi tiếng phf cho phép 1 kẻ tấn công có thể quay lại bao nhiêu file thay thế cho các tài liệu html. Cuộc tấn công nỳ chỉ đơn giản sử dụng 1 script CGI nghèo nàn để truy cập đến các file và các thư mục được cho phép trên web server . Để phát hiện được tấn công dựa trên lỗi phf , công cụ quét NIDS phải tìm trên tất cả gói tin những phần của chuỗi sau:
GET /cgi-bin/phf?
Network IDSs sẽ tìm kiếm trong tất cả các dấu hiệu đã tồn tại để tìm các chuỗi tìm kiếm trong các gói tin mạng. Ví dụ, dấu hiệu Snort sau sẽ thích hợp với chuỗi trên:
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI phf

access";flow:to_server,established; uricontent:"/phf"; nocase; reference:bugtraq,629;

reference:arachnids,128; reference:cve,CVE-1999-0067; classtype:web-application-

activity; sid:886; rev:8;)
và cảnh báo sẽ được gửi
Chúng ta sẽ đề cập đầy đủ đến sự phát triển của Snort NIDS sau.

19.1.2.2 Phát hiện những dấu hiệu bất thường:

Phát hiện những dấu hiệu bất thường liên quan đến việc thiết lập 1 nền móng cơ bản của những hoạt động bình thường của hệ thống hoặc là các hành vi trên mạng, và sau đó cảnh báo chúng ta khi những sự trệch hướng xuất hiện. Lưu lượng trên mạng thay đổi một cách không đáng kể, chẳng hạn như thay đổi trong thiết kế để hướng IDS theo định dạng host – base nhiều hơn là NIDS, Tuy nhiên, một số mmạng lại có những cấu trúc thật khác thường đặc biệt là những mạng quân đội hoặc những mạng giao tiếp tình báo. Mặt khác, những hành động xảy ra trong một server rất lớn có thể không thể kiểm soát hết được, do đó mà mạng trở nên rất hỗn loạn. Nên lưu ý rằng, thỉnh thoảng chúng ta muốn tách rời những NIDS dựa trên những sự kiện bất thường thành những sự kiện chuyển động bất thường (bị trệch hướng từ 1 miêu tả chuyển động đã biết) và giao thức sự kiện bất thường (trệch hướng từ các chuẩn giao thức mạng) .

Như chúng ta sẽ thấy sau đây trong chương này, phát hiện những sự kiện bất thường cung cấp 1 độ nhạy cao nhưng lại ít đặc trưng. Sau đây, chúng ta sẽ đề cập đến những công cụ hữu ích nhất.

19.2 Bayesian Analysis

Những IDS nguyên bản rất không thuận tiện vì hacker luôn tìm thấy những lỗ hổng mới mà không thể tìm thấy trong cơ sở dữ liệu các dấu hiện, hơn nữa, giống như những chương trình quét virus, việc cập nhật những dấu hiệu mới vào cơ sở dữ liệu là một vấn đề đáng quan tâm. Hơn thế, NIDS lại luôn được kỳ vọng có thể đương đầu với những giải tần lớn. Bởi thế nên trạng thái tồn tại trong 1 mạng có tốc độ đường truyền cao trở thành 1 vấn đề đáng quan tâm về bộ nhớ và giá thành tiến trình

Nhiều hơn thế, việc giám sát những mạng lớn "switched networks" là một vấn đề tự động nảy sinh vởi các switch trên mạng bị rút ngắn các cảm biến IDS. Người ta cố gắng xử lý vấn đề này bằng cách tích hợp IDS trong switch hoặc kèm IDS vào các cổng giám sát switch. Tuy nhiên, giải pháp này có rất nhiều vấn đề không thể giải quyết được, chẳng hạn như tạo ra hàng loạt bộ những kết nối hàng gigabit đòi hỏi phát triển nhiều IDSs trong 1 cấu hình cân bằng load phức tạp bởi vì 1 IDS tự nó không có khả năng đối đầu với việc tải trọng.

Một giới hạn khác của IDS đó là nó có những lỗ hổng rất lớn có thể bị tấn công hoặc lảng tránh được. Ví dụ như, tấn công từ chối dịch vụ như SYN floods hoặc tấn công smurf có thể làm tê liệt IDS rất dễ dàng. Tương tự như thế thì việc tốc độ quét chậm các IP address có thể làm hỏng rất nhiều IDSs.

Phần này sẽ giới thiệu thuộc tính thống kê của các bước chẩn đoán kiểm tra và những những gợi ý của chúng cho việc biên dịch kết quả thử nghiệm. Chúng ta sử dụng một công thức thống kê được biết với cái tên định lý Bayes, định lý miêu tả mối quan hệ mà tồn tại trong một chuỗi những thuộc tính đơn giản và có điều kiện. Không gói gọn trong những phép tính toán học chi tiết mà có thể có được từ hàng trăm quyển sách thống kê khác, chúng tôi còn đề cập đến các thực thi thực tếcủa "Bayesian analysis" khi được áp dụng cho IDSs. Để hiểu được khái niệm và thực thi thực tế của nó sẽ cho phép bạn hiểu rõ hơn về làm thế nào để thiết lập những IDS khác nhau tại những điểm khác nhau trên mạng của bạn.

Sự tiếp cận tới việc sắp xếp các cảm ứng phát triển từ chẩn đoán của Bayesian đã dạy cho sinh viên y khoa bởi 1 trong những tác giả của ebook này

19.2.1 Những thuộc tính chống lại độ nhạy cảm

Cân nhắc một IDS thông dụng báo cáo giám sát được trình bày tại Figure 19-1. Một cột gọi là xâm nhập, đại diện cho những xâm nhập đang xuất hiện. Dấu (+) có nghĩa là nó thực sự là 1 cuộc xâm nhập, còn (-) có nghĩa là nó chưa phải là 1 cuộc xâm nhập. Cột khác, gọi là phản hồi từ IDS, miêu tả suy nghĩ của IDS khi nó phát hiện ra 1 cuộc xâm nhập, dấu (+) có nghĩa là IDS coi đó là 1 cuộc xâm nhập, còn dấu (-) có nghĩa là IDS không đánh giá nó là 1 cuộc xâm nhập. Giống như trong cuộc sống thật, thì nó cũng chỉ ra rằng IDS không phải lúc nào cũng đúng. Bạn có thể sử dụng những điểm rơi của mỗi một góc phần tư trong bảng 2 x 2 để giúp chúng ta hiểu về thuộc tính thống kê của 1 IDA.

Figure 19-1. IDS response matrix

TP = Xác nhận đúng (xâm nhập được phát hiện đúng)
FP = Xác nhận sai (cảnh báo nhầm)
FN = Phủ nhận sai (bỏ nhỡ xâm nhập)
TN = Phủ nhận đúng (phát hiện đúng toàn vẹn)

19.2.1.1 Độ nhạy

Độ nhạy được định nghĩa là 1 xác nhận đúng (phân bổ của xâm nhập được phát hiện bởi IDS). Về phương diện toán học, độ nhạy được biểu diễn như sau:
True positives / (true positives + false negatives)
Xác nhận đúng / (xác nhận đúng + Phủ nhận sai)

Tỉ lệ phủ nhận sai bằng 1 trừ đi độ nhạy. Độ nhạy của 1 IDS có được nhiều hơn bao nhiêu thì những xâm nhập không được phát hiện giảm đi bấy nhiêu.

IDSs nhạy rất có ích trong việc chỉ ra những cuộc tấn công trên các khu vực của mạng mà nó rất dễ để phát hiện ra hoặc không bao giờ bị bỏ sót. Kiểm tra tính nhạy hữu hiệu nhiều hơn cho việc kiểm tra khi bạn cần loại trừ những gì có thể là đại diện từ xa cho 1 cuộc xâm nhập. Trong số những IDS có độ nhạy cao thì kết quả phủ nhận có nhiều giá trị vốn có hơn là các kết quả khẳng định.

Ví dụ, bạn cần 1 IDS nhạy để giám sát thiết bị host trong 1 LAN được bảo vệ bởi firewall và router như hình Figure 19-2, Khu vực 2 đại diện cho loại thiết bị này. Tại thời điểm bộ đệm tải nặng, chúng ta không nên có bất kỳ 1 xâm nhập nào. Nó rất quan trọng để có độ nhạy cao để giám sát không bỏ sót thứ gì. Các đặc trưng ít quan trọng hơn bởi vì tại thời điểm đó trên mạng, tất cả những hoạt động bất thường đều có thể được khai thác. IDS không cần sự phân biệt bởi vì những xử lý của con người đều bắt buộc phải khai thác mỗi một cảnh báo.
Figure 19-2. Network segmentation for Bayesian optimization of IDS placement

19.2.1.2 Tính xác định:

Về mặt toán học, tính xác định được biểu diễn như sau:
True negatives / (true negatives + false positives)
Phủ nhận đúng / (phủ nhận đúng + xác nhận sai)

Phủ nhận đúng đại diện cho những trường hợp khi IDS báo cáo đúng không có xâm nhập. Xác nhận sai xuất hiện khi một IDS báo cáo sai về 1 xâm nhập mà trong thực tế là không xảy ra. Xác nhận sai được xác định bằng 1 trừ đi đặc tính.

1 IDS xác định có tiện ích tốt nhất cho người quản trị hệ thống. Đối với những chương trình đó, giá trị xác nhận là có ích hơn giá trị phủ nhận. Những kiểm tra tính xác định rất hiệu quả trong khi những kết quả xác nhận sai là rất khủng khiếp.

Chọn lựa một IDS với tỷ lệ đặc tính cao cho một khu vực của mạng mà tại đó tự động chẩn đoán là một sự chỉ trích. Ví dụ, khu vực 1 trong Figure 19-2 đại diện cho 1 firewall hợp tác đối mặt với các hiểm họa từ internet. Trong trường hợp này, bởi vì những cuộc tấn công có thể trở thành tai họa nếu không được phát hiện sớm. Tại thời điểm này trên mạng, chúng ta không quan tâm nhiều đến tổng thể độ nhạy bởi vì chúng ta chờ đợi một cuộc tấn công nhiều hơn là giám sát toàn bộ chuyển dịch trên mạng để tìm ra những hoạt động bất thường.

19.2.1.3 Độ chính xác:

Thông thường, sự cân bằng giữa độ nhạy và tính xác định dựa trên độ nhạy và tính xác định chúng thay đổi liên tục dựa trên 1 điểm thay đổi đột ngột. Những điểm thay đổi cho những dấu hiệu bất thường này có thể được lựa chọn 1 cách tùy tiện hoặc dè dặt . Tuy nhiên, có rất nhiều tình huống có thể xảy ra khi chúng ta muốn tiêu nhiều tiền hơn cho 1 thiết bị có cả độ nhạy cao lẫn tính xác định cao. Tính chính xác là một khái niệm mà bao quanh cả 2 đặc tính xác định và độ nhạy. Tính chính xác là 1 tỷ lệ cân xứng của tất cả các kết quả của IDS (cả xác nhận và phủ nhận) rằng chúng là chính xác.

Ví dụ như, chúng ta cần tính chính xác cao trong khu vực của mạng như khu vực 3 trong Figure 19-2. Trong trường hợp này, web server của chúng ta đặt dưới sự tấn công, và nó có thể gây nên sự lúng túng ngay lập tức cho chúng ta và thậm chí gây thiệt hại về mặt tài chính nếu chúng bị tổn thương. Chúng ta cần thực thi bất kỳ một hành động bất thường nào và thực hiện một cách tự động bởi vì lưu lượng chuyển dịch trên mạng là rất lớn. Trong thực tế, để đạt được độ nhạy cao nhất và tính xác định cao nhất, chúng ta cần phối hợp giữa các lớp trong IDS.

Đường các đặc điểm thực thi nhận được (ROC) là 1 phương pháp biểu diễn đồ họa mối quan hệ giữa độ nhạy và tính xác định. Một cung nhỏ ROC các xác nhận đúng (độ nhạy) tỷ lệ với tỉ lệ xác nhận sai (bằng 1 trừ đi tỉ lệ xác nhận đúng). Đồ thị này phục vụ giống như là 1 nomogram (Figure 19-3), đại diện đồ họa (từ trường của thống kê) mà giúp bạn có thể nhanh chóng so sánh chất lượng giữa 2 hệ thống.

Sau khi chọn 1 điểm giới hạn mong muốn, độ nhạy và tính chính xác của IDS có thể xác định được ngay trên đồ thị. Đường vòng cung tương quan với độ chính xác hoặc chất lượng của IDS. Đường thẳng chạy lên và sang phải 45 độ chỉ ra 1 IDS không hữu dụng. Mặt khác, một IDS mà đường ROC được tucked trong 1 góc trên trái thì có thông tin tốt nhất. Về mặt định lượng, khu vực dưới đường vòng liên kết trực tiếp với độ chính xác của IDS
Trong hình Figure 19-3, IDS B chính xác nhiều hơn IDS C và IDS A có độ chính xác cao nhất. Figure 19-3. Sample ROC curve

19.2.2 Giá trị xác nhận và khẳng định dự báo trước:

Về mặt lý thuyết, độ nhạy và tính xác định là những thuộc tính của IDS. Những thuộc tính này là độc lập đối với những mạng được giám sát. Bởi vậy, độ nhạy và tính xác định chỉ cho chúng ta cách mà IDS thao tác, nhưng nó không chỉ cho chúng ta IDS thao tác trong từng ngữ cảnh của những phần mạng nào.

Giá trị Predictive là những dự báo trong thực tế tổng hợp từ tất cả các dữ liệu có sẵn. Giá trị dự báo kết hợp giữa prior probability với kết quả của IDS để yield post-test probability, biểu thị như dự báo xác nhận và phủ nhận.

Sự kết hợp combination constitutes a practical application of Bayess theorem, which is a formula used in classic probability theory.
Thông tin dựa trên cuộc tấn công prevalence trong mạng của bạn được điều chỉnh bởi kết quả của IDS để sinh ra một prediction. Tất cả các nhà quản trị mạng đều đã thực thi những phân tích intuitively but imprecisely. Ví dụ, nếu bạn biết rằng slow ping sweeps have recently become prevalent against your network, bạn có thể sử dụng thông tin này để định giá trị dữ liệu cho IDS của bạn.

Bởi vì những predictors đều liên kết về mặt toán học, nên chúng phải được chyển đổi thành những số lẻ. Sau đó, chúng được đề cập đến như những likelihood ratios (LRs) hoặc những odds ratios (ORs) và có thể kết hợp được trong những phép toán đơn giản.

19.2.3 Likelihood Ratios

Các giá trị độ nhạy, tính xác định và giá trị dự báo predictive values are all stated in terms of probability: the estimated proportion of time that intrusions occur. Một khái niệm hữu hiệu khác đó là odds ((i.e., the ratio of two probabilities, ranging from zero [never] to infinity [always]). For example, the odds of 1 are equivalent to a 50% probability of an intrusion (i.e., just as likely to have occurred as not to have occurred). The mathematical relation between these concepts can be expressed as follows:
Odds = probability / (1 - probability)
Probability = odds / (1 + odds)
LRs and ORs are examples of odds. LRs yield a more sophisticated prediction because they employ all available data.
The LR for a positive IDS result is defined as the probability of a positive result in the presence of a true attack, divided by the probability of a positive result in a network not under attack (true-positive rate/false-positive rate). The LR for a negative IDS result is defined as the probability of a negative result in the absence of a true attack, divided by the probability of a negative result in a network that is under attack (true-negative rate/false-negative rate).
LRs enable more information to be extracted from a test than is allowed by simple sensitivity and specificity. When working with LRs and other odds, the post-test probability is obtained by multiplying together all the LRs. The final ratio can also be converted from odds to probability to yield a post-test probability. By applying these statistical methods, we can make informed choices about deploying IDSs throughout a network. Although currently fraught with inaccuracy, the field of intrusion detection is still nascent, and new and exciting developments are happening every day. As time goes on, use of the scientific method will improve this inexact and complex technology. By understanding the sensitivity and specificity of an IDS, we can learn its value and when to utilize it. In addition, increasing the use of likelihood ratios makes the data that we receive from our IDSs more meaningful.
(Ơ sao lại có đoạn chưa dịch này?? Chắc là ngủ quên Để đó, chiều về check lại).

19.3 Tấn công thông qua IDSs

Để giúp bạn xây dựng chiến lược an ninh, phần này sẽ chỉ cho bạn thấy những hacker thường khai thác lỗi trong IDS như thế nào.

19.3.1 Phân đoạn (Fragmentation)

Phân đoạn hoặc chia nhỏ các gói tin là 1 trong những cách tấn công chống lại hệ thống phát hiện xâm nhập của mạng, và nó thường (stump) tất cả những NIDSs thương mại được thiết kế cách đây vài năm. Bằng cách cắt các gói tin thành những mẩu nhỏ, các hacker có thể làm fool IDS. Một IDS trạng thái dịch ngược các gói tin để phân tích, nhưng khi lượng những gói tin tăng lên, tiến trình cũng tiêu thụ hết nhiều nguồn lực hơn và trở nên bắt đầu thiếu chính xác. Và dường như có 1 giá trị xác định con số những phân đoạn mà 1 IDS có thể xử lý, nếu vượt quá con số đó There is a seemingly infinite number of fragmentation tricks that one can employ, leading either to evasion or to overloading the NIDSs anti-evasion capabilities.

19.3.2 Giả mạo - snoofing

Ngoài phương pháp phân đoạn dữ liệu, còn có thể giả mạo TCP sequence number mà NIDS nhìn thấy. Ví dụ như, nếu gói tin tiền kết nối SYN cùng với 1 số thứ tự được chuyển, IDS sẽ trở thành 1 thiết bị desynchronized từ host bởi vì host drops những SYN không được đánh giá cao và không được trông đợi, trong khi đó IDS tự thiết lập lại để nhận số thứ tự mới. Bởi vậy, IDS bỏ qua dòng dữ liệu thực vì nó đang chờ 1 số thứ tự mới mà không tồn tại. Khi gửi 1 gói tin RST với địa chỉ forged mà chịu trách nhiệm cho forged SYN có thể làm kết thúc kết nối mới này tới IDS.

Nhìn chung, NIDS không biết bằng cách nào máy đích sẽ dịch những thông tin đầu vào. Bởi vậy, những giao tiếp mạng bất thường có thể được thiết lập để có thể nhìn thấy sự khác biệt từ chính IDS. Chỉ có địa chỉ đích thực sự mới có thể cho phép tất cả các vấn đề của NIDS được giải quyết.

19.3.3 Thay đổi giao thức - Protocol Mutation

Whisker by RFP (có tại http://www.wiretrip.net) là một công cụ phần mềm được thiết kế để hack webserver bằng cách giả tạo 1 yêu cầu HTTP để vượt qua IDS. Ví dụ, 1 yêu cầu CGI cổ điển sẽ theo chuẩn http sau:

GET /cgi-bin/script.cgi HTTP/1.0
Obfuscated HTTP requests can often fool IDSs that parse web traffic. Ví dụ, nếu 1 IDS quét để tìm kiếm những xâm nhập dựa trên phf: /cgi-bin/phf

Chúng ta có thể thường xuyên tạo ra fool bằng cách thêm các dữ liệu mở rộng vào các yêu cầu của chúng ta. Chúng ta có thể thay đổi yêu cầu như sau:
GET /cgi-bin/subdirectory/../script.cgi HTTP/1.0

Trong trường hợp này, chúng ta yêu cầu thư mục con và sử dụng /../ để chuyển tới thư mục mẹ và thực hiện script đích. Cách thức sneaking này trong các back door được đề cập tới như 1 thư mục thay đổi và nó là 1 trong những cách khai thác phổ biến nhất vào thời điểm hiện nay.

Whisker tự động tạo ra các cách tấn công chống lại IDS rất đa dạng. Kết quả là Whisker được biết đến như là 1 công cụ chống lại IDS (AIDS). Whisker được chia nhỏ thành 2 phần, whisker (scanner) và libwhisker (module Perl được sử dụng trong Whisker).

Những IDS hiện đại như Snort cố gắng để bình thường hóa mọi truyền thông trên mạng trước khi phân tích thông qua những sự sử dụng các various preprocessors. Kỹ thuật bình thường hóa đòi hỏi phải tạo được cho việc truyền thông giống như thêm nguyên tắc, (more uniform) ví dụ như, bằng cách gỡ bỏ ambiguities trong packet headers và payloads và bằng cách hiển thị dòng truyền thông đơn giản để đối chiếu với các mẫu xâm nhập. Tuy nhiên, con số những possible mutations chỉ là số ít một số bit được xác định. Do đó, cuộc đua vũ trang giữa bên tấn công và bên phòng thủ vẫn được tiếp tục.

19.3.4 Tấn công vào thiết bị kiểm tra tính toàn vẹn:

Như đã đề cập trước đây, những IDS là thiết bị kiểm tra tính toàn vẹn tính toán giá trị checksum và tập hợp thông tin về các file ở chế độ khởi tạo. Sau đó, chương trình sẽ kiểm tra những sự thay đổi, sử dụng chế độ "check mode". Thêm vào đó, người quản trị hệ thống có thể cập nhật nhũng dấu hiệu sau khi cấu hình lại hệ thống (chế độ "update mode". Phụ thuộc vào sự thực thi của host IDS mà mỗi một chế độ đều có thể bị tấn công.

Một kẻ tấn công có thể tự thay đổi phần mềm host IDS, sau đó gửi thông tin sai lệch đến bàn điều khiển host IDS trung tâm hoặc có thể làm cho hệ thống nhầm lẫn giữa những công việc kiểm tra tính toàn vẹn. Đồng thời, một số chương trình tấn công vào nhân cũng có thể bị IDS bỏ qua bởi vì chúng có thể tự làm đúng đối với hệ thống và lừa dối IDS thành công. Phân tích chi tiết những cuộc tấn công của host IDS được đề cập chi tiết trong "Ups and Downs of UNIX/Linux Host-Based Security Solutions" (Section 19.7).

19.4 Tương lai của IDSs

Những phát hiện xâm nhập còn là mới bắt đầu, trong khi các hacker ngày càng tiến triển, IDSs bắt buộc phải cố gắng để đối đầu với các cuộc tấn công đó. Table 19-1 chỉ ra những hiểm họa mà tương lai sẽ đe dọa IDS và những giải pháp tiềm năng.
Table 19-1. Những giải pháp tiềm năng cho khó khăn trong tương lai của IDS
Vấn đề Giải pháp
Encrypted traffic (IPSec) Nhúng IDS vào trong các stack của host
Tốc độ và độ phức tạp của cuộc tấn công tăng Ngăn cấm các phát hiện bất thường, những thiết bị nặng về NIDS, và đối chiếu thông minh
Switched networks Giám sát các host riêng rẽ, nhúng NIDS vào trong các switch
Gia tăng lượng thông tin cần biên dịch Hiển thị trực quan dữ liệu, tự động cảnh báo và liên kết
Những kỹ thuật evasion mới Kỹ thuật bình thường hóa truyền thông mới và bảo vệ host theo chiều sâu
Kỹ thuật tấn công dựa trên nhân mới Thiết bị an toàn cho nhân mới
Phần sau sẽ kiểm tra đến những sự phát triển của vấn đề vào các giải pháp dự định:

19.4.1 Embedded IDS

IPSec (viết tắt của IP Security) trở thành 1 chuẩn phổ biến cho an toàn dữ liệu trên mạng. Ipsec là một bộ những chuẩn về an toàn được thiết kế bởi IETF nhằm cung cấp sự bảo vệ end-to-end cho các dữ liệu cá

nhân. Việc thực thi các chuẩn này cho phép 1 thiết bị có thể chuyển dữ liệu trên 1 mạng không đáng tin cậy như Internet trong khi ngăn chặn các kẻ tấn công phá hỏng, ăn trộm hoặc spoofing giao tiếp riêng biệt này.

Bằng cách bảo vệ an toàn cho những gói tin tại lớp mạng, Ipsec cung cấp các dịch vụ mã hóa trong suốt đối với các ứng dụng cũng như bảo vệ truy cập cho an toàn mạng. Ví dụ, Ipsec có thể cung cấp sự an toàn đầu cuối cho các hệ thống cấu hình client-to-server, server-to-server, và client-to-client.

Thật không may mắn, Ipsec lại là 1 con dao 2 lưỡi cho IDS. Một mặt, Ipsec cho phép người sử dụng log an toàn vào mạng của họ từ nhà sử dụng mạng riêng ảo, mặt khác Ipsec mã hóa dữ liệu trên đường truyền, bởi vậy làm cho sniffing trong IDS làm việc kém hiệu quả. Nếu hacker tấn công vào thiết bị đăng nhập từ xa của người sử dụng, họ sẽ có 1 tunnel an toàn để hack toàn bộ mạng. Để sửa lỗi này của Ipsec, những IDS tương lai đều cần phải nhúng vào tại các tầng của TCP/IP stack tại host. Điều này sẽ cho phép IDS quản lý các dữ liệu như nó chưa bị unencapsulated và thực thi nó trong mỗi tầng của stack, phân tích những dữ liệu đã được mã hóa ở cấp độ cao hơn.

19.4.2 Ngăn cấm những dấu hiệu bất thường được phát hiện thấy

Bởi vì những cuộc tấn công vẫn tiếp diễn và ngày càng tốc độ và phức tạp, do đó IDSs càng ngày càng ít khả năng chống chọi. Trả lời cho tình huống này bằng cách ngăn cấm những dấu hiệu bất thường được phát hiện thấy: tất cả những dấu hiệu bất thường, bất kể chúng là chính hay phụ, đều được báo động bằng xác nhận đúng. Phương pháp này đòi hỏi các IDS phải được đưa về các host riêng rẽ hơn là để chúng trong toàn mạng. Một host riêng rẽ có thể có nhiều mẫu thông tin có thể dự báo được hơn là trên toàn mạng. Mỗi một host được nói đến đều có một IDS để phát hiện bất kỳ một dấu hiệu bất thường nào. Sau đó người quản trị có thể đưa ra những quy luật (ngoại lệ) cho các tùy biến có thể được chấp nhận. Theo cách này, IDS giám sát các hoạt động theo cách mà firewall giám sát truyền thông.

Vậy làm như thế nào chúng ta có thể thiết kế 1 IDS thực thi việc ngăn ngừa các hành động bất thường dựa trên các host? Chúng ta sẽ làm việc với các host riêng rẽ mà có phần nào đó được lai tạo giữa firewall và các router, và chúng ta có thể điều khiển IDS của chúng ta cho mỗi một host là độc nhất. Bởi vì chúng ta chỉ làm việc trên các host, do đó chúng ta gói tin nào nhận được là cho host xác định nào. Chúng ta có thể đặt độ nhạy của chúng lên cao để tìm kiếm bất cứ một dấu hiệu bất thường nào.

Ví dụ, tại mức độ gói tin, công cụ phát hiện dấu hiệu bất thường dựa trên host của chúng ta có thể quét các gói tin như khi chúng được thực thi trên stack. Chúng ta điều khiển IDS để giám sát mọi thứ :
- Các dấu hiệu không được mong chờ
- Xung đột TCP/IP
- Các gói tin có độ lớn bất thường
- Giá trị TTL thấp
- Giá trị checksum sai
- Hoặc những xung đột trong những giao thức khác

Tương tự như thế, tại tầng ứng dụng, chúng ta có thể buộc công cụ phát hiện dấu hiệu bất thường quét tất cả những sự thay đổi bất thường trong các đặc điểm sau đây của hệ thống:
- Sử dụng CPU
- Kích hoạt đĩa
- Đăng nhập người dùng
- Kích hoạt file
- Số những dịch vụ đang chạy
- Số những ứng dụng đang chạy
- Số những cổng đang mở
- Kích thước của file log

Khi một dấu hiệu bất thường được phát hiện, một cảnh báo sẽ được chuyển tới trung tâm điều kkhiển. Phương pháp này có độ nhạy cao, nhưng thật không may là nó tạo ra rất nhiều vấn đề cần phải giải quyết về mặt dữ liệu. Chúng ta sẽ giải quyết vấn đề này sau.

19.4.3 Host- Versus Network-Based IDSs

Sự gia tăng của những mạng switch làm cản trở 1 IDS trong việc giám sát mạng sử dụng chế độ pha tạp, phân tích giao thức thụ động. Nó trở nên ngày càng khó để giám sát nhiều host cùng lúc bởi vì sự gia tăng của đường truyền, những mạng ảo và những sự rắc rối khác. Thêm vào đó, việc ứng dụng gia tăng của các the growing use of encrypted traffic foils passive analysis off the wire. Bởi vậy, IDS đang trở nên các giám sát dựa trên host.

19.4.4 Visual Display of Data

Bởi vì đường truyền và hiệu quả các cuộc tấn công ngày càng tăng, nên việc tạo ra các cảnh báo chính xác ngày càng trở nên khó khăn. Lượng dữ liệu cảnh báo được tạo nên bởi IDS cps thể nhanh chóng overwhelm thao tác của con người. Thật không may, việc lọc dữ liệu cho con người thường sử dụng hạn chế những hiệu quả của nó.

Một giải pháp cho vấn đề này liên quan đến kỹ thuật phát triển visualization đồng thời được coi là hiển thị geometric dữ liệu. Con người hiểu được geometric shapes intuitively, bởi loại hiển thị này thường là cách dễ nhất để hiển thị một lượng dữ liệu (massive). Khi một theo tác cảm thấy 1 dấu hiệu bất thường trong màn hình đồ họa, nó có thể drill down muộn hơn để giải quyết vấn đề. Ví dụ, cho những ứng dụng bên trong, Airscanner Corporation mã hóa điều khiển linh hoạt ActiveX mà mimics a real-time human electrocardiogram (EKG). Tốc độ và giai điệu (màu sắc hoặc âm thanh) của dao động "heartbeat" trên màn hình để đáp trả lại sự thay đổi trên mạng. Giám sát giống như những người y tá trong bệnh viện đối với cardiac telemetry floor, Người quản trị mạng Airscanner có thể dễ dàng giám sát LAN bằng cách để ý đến màn hình.

19.5 Nghiên cứu Snort IDS

Phần này sẽ trình bày 1 ví dụ phát triển Snort IDS (http://www.Snort.org). Snort thường được gọi là "lightweight IDS," nhưng nó có tên gọi đó tại thời điểm đó chứ không liên quan gì đến lightweight.. Snort chỉ nên được gọi là lightweight nếu nó đề cập đến công cụ phát hiện hiệu quả và dung lượng nhớ các dấu hiệu nhỏ. Nó là 1 bộ dịch IDS đầy đủ mà có thể phát triển theo hướng tốc độ xử lý cao và cấu hình phân bổ mà có thể đạt được đến tốc độ hàng giga bit.

Thiết bị phát hiện xâm nhập được đề cập đến trong phần này xây dựng trên hệ điều hành Linux, cơ sở dữ liệu MySQL và một môi trường phân tích ACID. Tất cả mọi phiên bản Linux như Red Hat hoặc Debian đều có thể sử dụng. Bạn nên xây dựng 1 hệ thống Linux nhỏ nhất từ scratch (giống như những nhà bán phần mềm IDS thương mại bán IDS dựa trên Unix). Đối với việc phát triển những mạng nhỏ, bạn nên từ bỏ những biến Linux quá canned. Hệ thống phải được nhỏ gọn nhất và nhiều tính năng (tất cả những phần mềm không cần thiết đều nên gỡ bỏ).

Bạn nên có ít nhất 2 card mạng trên máy tính phát triển Snort. Bởi vì giao diện sniffing (để phát hiện những cuộc tấn công) và giao diện quản lý (sử dụng để quản lý dữ liệu các sự kiện nhạy cảm, cập nhật các quy định và những thay đổi cấu hình) phải được đặt riêng rẽ. Lý do chính đó là giao diện sniffing không có địa chỉ IP. Trong môi trường Linux, rất là dễ để kích hoạt một giao diện mạng mà không cần địa chỉ IP mà chỉ cần sử dụng lệnh như ifconfig eth1 up. Mặc dù không cung cấp 1 biện pháp an toàn tổng thể (bằng định nghĩa), nhưng biện pháp này tốt hơn là sử dụng một giao diện thông thường để phát hiện xâm nhập.

Snort và cơ sở dữ liệu có thể được cài đặt trên 1 máy, tuy nhiên trong trường hợp tốc độ truyền thông cao, bạn nên cài đặt cơ sở dữ liệu, Snort, và webserver trên những máy tính khác nhau. Tốt nhất là Snort trên 1 máy, còn cơ sở dữ liệu và webserver trên máy còn lại
Trong trường hợp cài đặt trên nhiều máy, các thành phần của IDS được kết nối với nhau qua mạng và do đó, các biện pháp an toàn phải được thực thi. Để bảo vệ đường truyền giữa thiết bị phân tích với cơ sở dữ liệu, chúng ta phải sử dụng kết nối SSL. Để hạn chế các truy cập bàp bàn điều khiển dựa trên ACID, chúng ta sẽ sử dụng những đặc chuẩn của Apache webserver, phương pháp xác thực HTTP cơ bản qua .htpasswd. Truyền thông giữa cảm biến snort với cơ sở dữ liệu có thể được tunneled qua SSL hoặc SSH.

19.5.1 Cài đặt hệ thống:

Đầu tiên bạn phải thiết lập 1 Linux hardened. Đối với Red Hat Linux, có thể chọn Custom Install từ những bộ cài đặt CD chính thức hoặc không chính thức, hoặc thu gọn các tùy chọn cài đặt của nó bằng cách gỡ bỏ các thành phần đồ họa . Phải chắc chắn rằng tất cả các gói tin MySQL server (có sẵn trên Red Hat CDs)

được cài đặt. Câu lệnh:
rpm -U /mnt/cdrom/RedHat/RPMS/mysql*rpm
sẽ quan tâm đến điều này, được cung cấp bởi Linux CD.

Trong trường hợp môi trường Linux sử dụng là Red Hat, rất nhiều gói phần mềm Snort RPM (Red Hat Package Manager) có thể download từ website Snort.org . Bạn cần gói Snort và Snort-mysql cho những cài đặt trên. Cài đặt chúng lên trên hệ thống của bạn. Nếu RPM đòi hỏi sự độc lập, hãy download gói cài đặt thích hợp cho nó (có thể sẽ cần thư viện libpcap) .
Cài đặt thêm phần mềm quan sát sự kiện ACID-IDS vào hệ thống. Trang chủ ACID có chứa tất cả các phần mềm và hướng dẫn cài đặt (http://acidlab.sourceforge.net). Các gói cài đặt ACID đòi hỏi phải được giải nén ở 1 thư mục có thể nhìn thấy được từ webserver (ví dụ trên Red Hat là /var/www/html). Bởi vậy ACID có thể được phát triển trên /var/www/html/acid. File cấu hình acid_conf.php là nơi chứa tất cả các sắp đặt cấu hình. Không có điều khiển truy cập nào được thiết lập bên trong, do đó bạn cần phải tạo.htpasswd trong /var/www/html/acid.

Nếu trong lựa chọn phát triển (chẳng hạn như cài đặt RedHat) không có web server thì 1 Apache web server cần được cài đặt trên môi trường đó thông qua CD.
rpm -U /mnt/cdrom/RedHat/RPMS/apache*rpm
Sau khi tất cả các thành phần đã được cài đặt, đến lượt chúng ta thiết lập cấu hình cho IDS. Đầu tiên, Snort phải được cấu hình để có thể log vào cơ sở dữ liệu. Sau đây là một số chỉ dẫn để làm điều đó:
1. Khởi động cơ sở dữ liệu MySQL :
/etc/init.d/mysql start
2. Tạo cơ sở dữ liệu Snort:
echo "CREATE DATABASE Snort_db;" | mysql -u root -p
3. Tạo người sử dụng để sử dụng cơ sở dữ liệu:
adduser Snort
4. Tạo các quyền cho người sử dụng này để thêm các dữ liệu cảnh báo vào trong cơ sở dữ liệu:
echo "grant INSERT,SELECT on Snort_db.* to Snort@localhost;" | mysql -u root -p
5. Sử dụng các script có sẵn trong nguồn của Snort (không đi kèm cùng với gói nhị phân RPM) để tạo cấu trúc dữ liệu:
cat ./contrib/create_mysql | mysql Snort_db
6. Thay đổi file cấu hình Snort để log vào cơ sở dữ liệu. Nói cách khác thay đổi /etc/Snort.conf như sau::
output database: log, mysql, user=Snort dbname=Snort_db host=localhost
7. Thay đổi script khởi tạo Snort (/etc/init.d/Snortd) để snort thực hiện lệnh sau:
/usr/sbin/Snort -D -l /var/log/Snort -i $INTERFACE -c /etc/Snort/Snort.conf
Định vị trí để các log của snort có thể được đánh giá ở đây.
Bây giờ, Snort có thể bắt đầu bằng lệnh:
/etc/rc.d/init.d/Snortd start

IDS đã được cấu hình và có thể log tới cơ sở dữ liệu. Hãy kiểm tra chúng như sau:
1. Kiểm tra rằng tiến trình đang chạy:
ps ax| grep Snort | grep -v grep
Nếu kết quả khả quan,bạn sẽ thấy dữ liệu trả về không trống.

Trên Linux, tồn tại 1 lệnh đơn giản tương tự:
ps u `pidof Snort`

2. Kiểm tra rằng Snort phát hiện thấy tấn công trên lynx http://www.someLOCALwebserver.com/cmd.exe và sau đó chạy lệnh:
tail /var/log/Snort/alert
Nếu có kết quả tốt, bạn sẽ nhìn thấy 1 thông điệp cảnh báo chỉ rằng có 1 tấn công IIS web. Đừng chạy bước kiểm tra này thông qua 1 kết nối URL từ xa mà hãy thử nó trên máy cục bộ của bạn. Phải chắc chắn rằng cảm biến có thể cảm nhận được cuộc tấn công (kết nối này được thiết lập thông qua mạng được giám sát bởi Snort).

Phương pháp quét cổng sử dụng nmap là 1 bước thử Snort hiệu quả, điều này đảm bảo cho việc phát hiện quét cổng được bật và được cấu hình chính xác. Trong thực tế, có tồn tại nhiều phương pháp để kiểm tra 1 IDS. Nhiều người thích sử dụng những gói tin ICMP lớn (có thể thực hiện bằng 1 lệnh ping đơn giản) hoặc là những phương pháp khác.

3. Kiểm tra logging cơ sở dữ liệu:
echo "SELECT count(*) FROM event" | mysql Snort_db -u root -p
Nếu tốt, bạn sẽ nhìn thấy một lượng khác rỗng dữ liệu được chứa trong cơ sở dữ liệu.

19.5.2 Cài đặt công cụ cảnh báo:

Bây giờ, hãy thiết lập công cụ cảnh báo qua ACID. ACID (Trung tâm phân tích dữ liệu xâm nhập) là 1 ứng dụng được xây dựng trên PHP mà cho phép phân tích dữ liệu Snort được chứa trong cơ sở dữ liệu.

ACID phải được phép truy cập vào cơ sở dữ liệu. Sử dụng lệnh sau để thực hiện điều này:
echo "grant CREATE,INSERT,SELECT,UPDATE,DELETE on Snort_db.* to acid@localhost;" |

mysql -u root -p
Để an toàn hơn, nên sử dụng SSL để quan sát cảnh báo. Hãy phát triển các gói SSL từ đĩa cài Red Hat:
rpm -U /mnt/cdrom/RedHat/RPMS/mod-ssl*rpm
và khởi động lại Apache thông qua /etc/init.d/httpd restart.
Để an toàn hơn nữa, chỉ những kết nối SSH mới được phép. Một host firewall script for the iptables Linux firewall có thể được sử dụng để chỉ cho phép TCP port 443 (HTTPS) và không cho phép TCP port 80 (HTTP).

Bây giờ, khởi động Apache web server và chỉ browser tới giao diện quản lý IP (hoặc địa chỉ 127.0.0.1 nếu chạy trên local browser). Địa chỉ đúng là: http://www.yourSnortServer.com/acid. Phàn mềm ACID sẽ hướng dẫn bạn lựa chọn khởi tạo cài đặt, cung cấp cho bạn theo những hướng dẫn trên. ACID có thể được sử dụng để quan sát các cảnh báo của Snort IDS trong những chế độ khác nhau, thực hiện việc tìm kiếm, và truy cập gói tin payload đầy đủ.

Nếu cài đặt cơ sở dữ liệu không đúng, bạn đơn giản chỉ cần chuyển tiếp những cảnh báo tới syslog và sau đó sử dụng công cụ phân tích syslog để giải quyết nó. Những công cụ như Snortsnarf tồn tại để tổng kết và quan sát các sự kiện Snort.

19.5.3 Điều chỉnh các quy tắc của IDS :

Thảo luận đầy đủ về việc điều chỉnh các quy tắc của IDS được đưa ra trong cốt lõi của chương này. Tuy nhiên, một khi chúng ta tiến tới việc khởi tạo các quy tắc, chúng ta sẽ mất rất nhiều thời gian cho những cảnh báo, phân tích chúng và theo đó giảm bớt các quy tắc. Công đoạn này thích hợp hơn đối với việc phát triển 1 NIDS nội bộ và mạng nhỏ. Một giải pháp khác đó là thu hẹp các bộ quy tắc để giám sát chỉ những dịch vụ đang bị nguy hiểm. Công việc này tốt hơn khi cài đặt DMZ với độ an toàn cao trong đó các thiết bị đều được thống kê một cách cẩn thận và vững chắc. Trong trường hợp này, cảnh báo CodeRed sẽ gia tăng một cách tuyệt đối bởi vì Unix web server sẽ không bị tổn thương bởi những đe dọa tầm thường đó.