



**SỞ THÔNG TIN VÀ TRUYỀN THÔNG
THÀNH PHỐ ĐÀ NẴNG
TRUNG TÂM CÔNG NGHỆ THÔNG TIN
VÀ TRUYỀN THÔNG**



TÀI LIỆU

**ĐÀO TẠO VỀ PHÂN TÍCH LOGFILE
VÀ CÁC CÔNG CỤ PHÒNG CHỐNG
AN TOÀN AN NINH THÔNG TIN**



Đà Nẵng, tháng 8 năm 2016



**SỞ THÔNG TIN VÀ TRUYỀN THÔNG
THÀNH PHỐ ĐÀ NẴNG
TRUNG TÂM CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

TÀI LIỆU

ĐÀO TẠO VỀ PHÂN TÍCH LOGFILE VÀ CÁC CÔNG CỤ PHÒNG CHỐNG AN TOÀN AN NINH THÔNG TIN

**Dành cho đối tượng là cán bộ chuyên trách CNTT tại các
sở, ban, ngành, UBND các quận, huyện trên địa bàn
thành phố Đà Nẵng, Trung tâm IID và DNICT**

Đà Nẵng, tháng 8 năm 2016

BAN BIÊN SOẠN

CHỦ BIÊN

THÀNH VIÊN BAN BIÊN SOẠN

Thạc sĩ Phạm Hồng Vĩnh

MỤC LỤC

PHẦN 1. LOGFILE VÀ PHÂN TÍCH LOGFILE	1
1.1. TỔNG QUAN VỀ LOG FILE	1
1.1.1. Một số khái niệm.....	1
1.1.2. Tác dụng của Logfile	2
1.1.3. Cơ chế ghi Log file.....	2
1.2. PHÂN TÍCH LOG FILE.....	3
1.2.1. Cơ bản của việc phân tích Log file	3
1.2.1.Một số kiểu log.....	4
1.2.3. Trạng thái logging	9
1.2.4. Khi nào cần phải quan sát các Log	10
1.2.5. Log Overflow và Aggregation	12
1.2.6. Những thử thách đối với việc phân tích log.....	13
1.2.7. Quản lý thông tin an toàn	14
1.2.8. Tích hợp Global Log	15
PHẦN 2. PHÂN TÍCH WINDOWSLOG FILE	17
2.1. WINDOWS EVENT LOG.....	17
2.1.1. Giới thiệuEvent log	17
2.1.2. Các kiểu Event	17
2.1.3. Cấu trúc Event log.....	17
2.1.4. Vị trí của Event log	18
2.1.5. Kịch bản phân tích Event log	18
2.2.WINDOWS FIREWALL LOG	23
2.2.1. Giới thiệuWindows Firewall Log	23
2.2.2. Cấu hình Windows Firewall Log	23
2.2.3. Vị trí Windows Firewall Log	24
2.2.4. Phân tích Firewall log	25
2.3.1. Giới thiệu IIS.....	25
2.3.2. IIS Log file	26

2.3.3. IIS Log File Format.....	27
2.3.4.W3C Extended Log Format	29
2.4. CÔNG CỤ PHÂN TÍCH.....	31
2.4.1. Windows Event Viewer	31
2.4.2. Event Log Explorer	37
2.4.3. Mandiant.....	47
2.4.4. Log Parser	48
PHẦN 3. PHÂN TÍCH LINUX LOG FILE	52
3.1. SYSLOG	52
3.1.1. Giới thiệu về Syslog.....	52
3.1.3. Cấu hình syslog	55
3.1.4. Sử dụng lệnh ghi log	56
3.1.5. Sự luân phiên log.....	57
3.1.6. Các vị trí log quan trọng.....	58
3.1.7. Thao tác với Log Files	58
3.1.8. Phân tích log file	60
3.2. APACHE LOG FILE	66
3.2.1. Security Warning	66
3.2.2. Error Log	66
3.2.3. Access log	67
3.2.4. Log Rotation.....	71
3.2.5. Piped Logs.....	72
3.3. CÔNG CỤ PHÂN TÍCH.....	73
3.3.1. Syslog-ng.....	73
3.3.2. ELK (Logstash, Elasticsearch, Kibana)	74
PHẦN 4. PHÂN TÍCH LOG FILE ỨNG DỤNG	79
4.1. EMAIL LOG FILE	79
4.1.1. Giới thiệu Email Log file	79
4.1.2. Phân tích Email Log file	79
4.1.3. Phát hiện Email giả mạo	80

4.2. FIREWALL LOG FILE	82
4.2.1. ISA Logging	82
4.2.2. Iptable log file	85
4.3. IDS/IPS LOG FILE	86
4.3.1. Giới thiệu IDS/IPS	86
4.3.2. Phân tích IDS/IPS log	86
4.4. DATABASE LOG FILE	87
4.4.1. SQL Server log file	87
4.4.2. MySQL log file	88
4.5. CÔNG CỤ PHÂN TÍCH	90
4.5.1. SolarWinds LEM	90
4.5.2. Splunk	91
PHẦN 5. NHẬN DIỆN TẤN CÔNG VÀO ỨNG DỤNG WEB TỪ LOGFILE	95
5.1. TẤN CÔNG VÀO ỨNG DỤNG WEB	95
5.1.1. Quy trình tấn công vào ứng dụng Web	95
5.1.2. Một số kiểu tấn công phổ biến vào ứng dụng Web	98
5.2. WEB LOG FILE	100
5.2.1. Web log file format	100
5.2.2. HTTP Status Code	102
5.3. NHẬN DIỆN TẤN CÔNG	108
5.3.1. Biểu thức chính quy	108
5.3.2. Nhận diện tấn công qua biểu thức chính quy	117
5.4. XÂY DỰNG CÔNG CỤ PHÂN TÍCH	119
5.4.1. Phân chia log file theo thời gian	119
5.4.2. Phân chia log file theo mã http status code	120
5.4.3. Phân chia log file theo địa chỉ IP	123
5.4.4. Phân chia log file theo kiểu tấn công	123
NGÂN HÀNG CÂU HỎI	Error! Bookmark not defined.

PHẦN 1. LOGFILE VÀ PHÂN TÍCH LOGFILE

1.1. TỔNG QUAN VỀ LOG FILE

1.1.1. Một số khái niệm

1.1.1.1. *Log file*

Trong máy tính, Log file là một tập tin ghi lại những sự kiện xảy ra trên một thiết bị phần cứng, một hệ điều hành, một phần mềm đang chạy hoặc thông điệp giữa người sử dụng khác nhau của một phần mềm truyền thông...

Ví dụ:

- Log file của các thiết bị firewall, router, modem, ...
- Log file của hệ điều hành Linux, Windows, ...
- Log file của ứng dụng Web, Mail, Database, ...
- Log file của các phần mềm truyền thông, ...

1.1.1.2. *Logging*

Logging là hành động lưu giữ một tập tin. Trong trường hợp đơn giản, thông điệp được ghi vào một tập tin duy nhất và trình tự theo thời gian. Những hành động làm sai lệch tập tin nhật ký phải được ngăn chặn ở mức tối đa.

1.1.1.3. *Transaction log*

Transaction log là một tập tin lưu trữ các thông tin liên lạc giữa một hệ thống với người sử dụng của hệ thống đó, hoặc một phương pháp thu thập dữ liệu sẽ tự động ghi lại kiểu, nội dung hoặc thời gian giao dịch thực hiện bởi một người từ một thiết bị đầu cuối với hệ thống đó.

Ví dụ trong cơ sở dữ liệu thì Transaction log là một dãy các record lưu trữ thông tin các thao tác cập nhật dữ liệu được thực hiện lên database.

1.1.1.4. *Logrotation*

Để ngăn cản những Log file ngày càng lớn và trở nên cồng kềnh, khó kiểm soát, một hệ thống quay vòng log file (Logrotation) nên được cài đặt. Hệ thống sẽ đưa ra các lệnh để thiết lập tên cho những log file mới, những file cũ

được đổi tên bằng cách thay những con số ở hậu tố. Sự luân phiên này được cấu hình cho một số lượng lớn các file và các log files cũ nhất có thể bị xoá khi sự luân phiên bắt đầu chạy.

Ví dụ:

Trong /var/log có các messages sau: messages, messages.1, messages-20071111, messages-20071118, ...

1.1.2. Tác dụng của Logfile

- Log file ghi lại liên tục các thông báo về hoạt động của cả hệ thống hoặc của các dịch vụ được triển khai trên hệ thống và file tương ứng.
- Phân tích nguyên nhân gốc rễ của một vấn đề.
- Giúp cho việc khắc phục sự cố nhanh hơn khi hệ thống gặp vấn đề.
- Giúp cho việc phát hiện, dự đoán một vấn đề có thể xảy ra đối với hệ thống.

1.1.3. Cơ chế ghi Log file

1.1.3.1. Cơ chế độc lập

- Các ứng dụng tự ghi nhật ký vào các thư mục riêng rẽ.
- Khó theo dõi các nhật ký.
- Nhật ký nhân hệ điều hành không phải là ứng dụng.
- Các ứng dụng khó sử dụng nhật ký của nhau.
- Khó phát hiện ứng dụng “có vấn đề”.

1.1.3.2. Cơ chế tập trung

- Các ứng dụng gửi thông báo chung cho một ứng dụng chịu trách nhiệm ghi nhật ký.
- Tùy theo mức độ ứng dụng nhật ký sẽ ghi các thông tin phù hợp vào nhật ký.
- Giúp quản trị viên có cái nhìn chi tiết về hệ thống, qua đó có định hướng tốt hơn về hướng giải quyết.

- Mọi hoạt động của hệ thống được ghi lại và lưu trữ ở một nơi an toàn (log server) nhằm đảm bảo tính toàn vẹn phục vụ cho quá trình phân tích điều tra các cuộc tấn công vào hệ thống.
- Log tập trung kết hợp với các ứng dụng thu thập và phân tích log khác nữa giúp cho việc phân tích log trở nên thuận lợi hơn, giảm thiểu nguồn nhân lực.

1.2. PHÂN TÍCH LOG FILE

1.2.1. Cơ bản của việc phân tích Log file

Phân tích các log file 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 kê những sự việc được sản sinh bởi máy tính. Phân tích log file không phải là ngành 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 nhầm lẫn và dường như chẳng liên quan gì đến nhau. Trong trường hợp một thiết bị một người sử dụng với rất ít các hoạt động, tất cả những bản ghi log mà chưa được nhìn trước là rất ít 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 một 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 một 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.

1.2.1. Một số kiểu log

1.2.1.1. Unix log

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 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 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 một ngày. Định dạng này bao gồm các trường sau:

Timestamp: Giờ hệ thống của thiết bị khi ghi nhận log (trường hợp log một 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 một bộ phận (ví dụ như PAM_unix) mà sản sinh ra thông báo log.

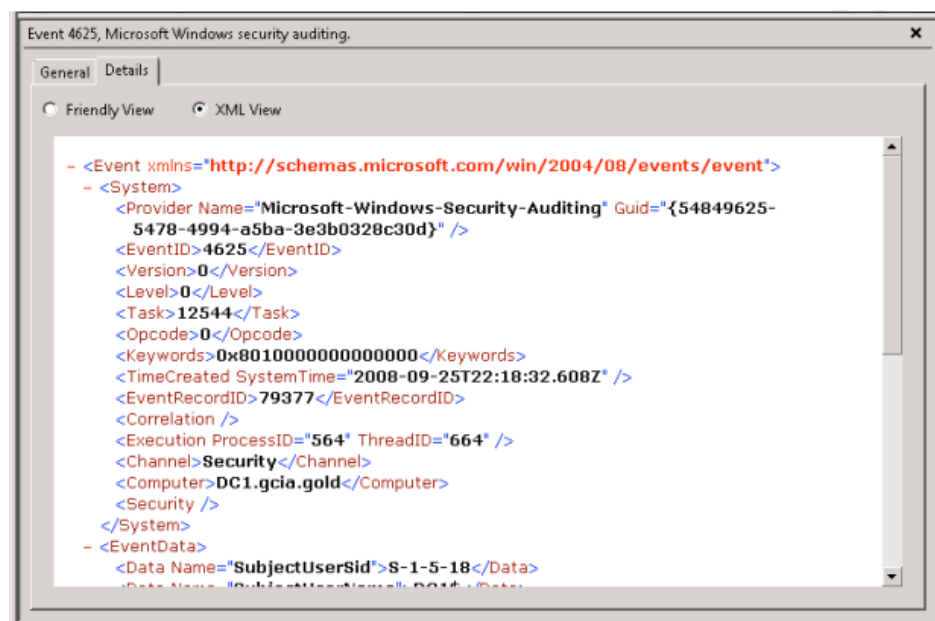
Log message: Thông báo log có thể có nhiều định dạng khác nhau, thông thường bao gồm tên ứng dụng, các biến tình trạng đa dạng, địa chỉ IP nguồn, giao thức ... Thành thạo định danh tiến trình của một tiến trình có thể tạo ra những bản ghi log và được ghi vào các chỗ trống.

Bốn thông báo log sau đây được chỉ ra đối với ví dụ trên, theo thứ tự:

- Có vấn đề xảy ra đối với DNS server thứ hai.
- Một người sử dụng đã đăng nhập vào thiết bị.
- Một truy cập DNS bị cấm xuất hiện.
- Một người sử dụng đã được cung cấp mật khẩu an toàn hệ thống đang đăng nhập từ xa từ địa chỉ IP 10.11.12.13.

1.2.1.2. Windows log

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).



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 một hoạt động

thông thường của Windows XP. 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 một dấu phẩy cho việc phân tích hoặc quan sát log qua các text editor.

1.2.1.3. 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 một kẻ tấn công có thể phát hiện ra.

Lựa chọn stealthy logging cơ bản nhất thực sự lại không phải là vùng trộm. Nó chỉ cung cấp một site backup cho việc lưu trữ log. Thêm vào việc chỉ định log server (có thể nhìn thấy đối với những kẻ tấn công), một 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 log tới một 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.sourceforge.net>) để lưu lại chỉ những chuyển dịch syslog từ xa giữa hai 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 hai gửi file log tới một host log mà không chạy syslog (hoặc là bất kỳ một dịch vụ mạng nào khác). 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 một file (sử dụng câu lệnh trên).

Nó có thể được thực thi để tránh viếng thăm 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 một 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 một 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:0C:0D: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 một 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 một server không tồn tại là hiệu quả hơn nếu một mức độ cao hơn của stealth là cần thiết. Phương pháp này có thể không áp dụng được cho

một mạng Lan truyền thống, nhưng nó có thể được ứng dụng trong rất nhiều trường hợp đặc biệt khác.

1.2.1.4. 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 một 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 hai 22:16
head S root stdin 0.00 secs Sat Nov hai 22:16
egrep root stdin 0.01 secs Sat Nov hai 22:16
grep S root stdin 0.01 secs Sat Nov hai 22:16
bash F root stdin 0.00 secs Sat Nov hai 22:16
bash SF root stdin 0.00 secs Sat Nov hai 22:16
dircolors root stdin 0.00 secs Sat Nov hai 22:16
stty root stdin 0.00 secs Sat Nov hai 22:16
bash SF root stdin 0.00 secs Sat Nov hai 22:16
tput root stdin 0.01 secs Sat Nov hai 22:16
bash SF root stdin 0.00 secs Sat Nov hai 22:16
tput root stdin 0.01 secs Sat Nov hai 22:16
su anton stdin 0.04 secs Sat Nov hai 22:16
head anton stdin 0.01 secs Sat Nov hai 22:16
```

Những bản ghi trên (được tạo ra bởi lệnh `lastcomm | 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 hai 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 có thể 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.

1.2.3. 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 một file log.

Một vài sự kiện mà máy tính có thể đặt vào log:

- Tắt, mở, restart hoặc bất cứ một hành động liên quan đến đầu cuối của hệ thống hoặc một phần mềm.
- Various thresholds được thực thi hoặc các cấp tìm kiếm nguy hiểm. giống như đầy dung lượng đĩa, exhausted bộ nhớ hoặc bộ xử lý hoạt động quá nhanh.
- Phản ứng thông báo rằng 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 một 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 một hệ thống và sẵn sàng cho việc phân tích. Công việc của 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

1.2.4. 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 kiếm 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, một 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 một 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 một 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 một 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 phân 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 lẻ hoặc chỉ một 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 một vụ xâm nhập ngoại trừ khi bạn thích thú đối với công việc đó hoặc là bạn đang chuẩn bị để lấy một 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 (attonish) việc chuyên nghiệp hạ các hành động bảo vệ an toàn hệ thống bởi những câu bình luận kiểu 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 một 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 một firewall và một 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. Tóm 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 nhỏ cho đến “Vâng, bạn phải làm như vậy” đối với những giao dịch lớn

1.2.5. Log Overflow và 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 gigabyte 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 một 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 độ 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 cứ log. Trong trường hợp này, những tài liệu minh chứng cần thận của một 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 một lượng rất lớn những thông tin log.

1.2.6. 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 nhận log 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 một số dấu hiệu của kẻ xâm nhập trong những file hệ thống, ví dụ như, một 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 một địa chỉ IP thì liệu bạn có thể chứng minh được điều gì? Rất dễ để thuyết phục một sự việc xảy ra đáp trả lại khi họ thực hiện việc chặn bắt đường truyền bằng một phiên của thiết bị ghi nhận một 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 đầu 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 một 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).

1.2.7. Quản lý thông tin an toàn

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 một 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 một 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 một giây và liên kết chúng lại trong thời gian thực cũng như cung cấp cho chúng khả năng phân tích và long term trending.

Một số công cụ cho phép phân tích thời gian thực và phức 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 một 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.

1.2.8. Tích hợp Global Log

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>), và 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 một 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 một 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 tự 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 nhắc một trong những dịch vụ này để có thể quen thuộc hơn với dữ liệu log của bạn và để xây dựng một mạng internet an toàn hơn.

PHẦN 2. PHÂN TÍCH WINDOWS LOG FILE

2.1. WINDOWS EVENT LOG

2.1.1. Giới thiệu Event log

Event log ghi lại các sự kiện diễn ra trong việc thực hiện của một hệ thống nhằm cung cấp cơ sở để kiểm tra lại hệ thống hoặc có thể được sử dụng để tìm hiểu hoạt động và chẩn đoán các vấn đề nếu có đối với hệ thống.

Chúng rất cần thiết cho việc tìm hiểu các hoạt động của các hệ thống phức tạp, đặc biệt là trong trường hợp các ứng dụng với tương tác người dùng (chẳng hạn như các ứng dụng máy chủ).

Nó cũng rất hữu ích trong việc kết hợp các log file từ nhiều nguồn khác nhau. Sử dụng các biện pháp kết hợp với phân tích thống kê các log file này, có thể xác định các mối tương quan giữa các sự kiện dường như không liên quan trên các máy chủ khác nhau.

2.1.2. Các kiểu Event

Hệ điều hành Windows phân loại các sự kiện thành 5 loại:

- Information event: Mô tả sự thành công của một công việc, chẳng hạn như cài đặt xong một ứng dụng.
- Warning event: Thông báo cho quản trị viên một vấn đề tiềm ẩn, chẳng hạn không gian đĩa thấp.
- Error message: Mô tả một vấn đề quan trọng mà có thể dẫn đến tính năng nào đó bị vô hiệu hóa.
- Success audit event: Mô tả một hoạt động thành công, chẳng hạn như một người dùng cuối đăng nhập thành công vào hệ thống.
- Failure audit event: Mô tả một hoạt động không thành công, chẳng hạn như một người dùng cuối nhận được thông báo khi nhập mật khẩu không chính xác.

2.1.3. Cấu trúc Event log

Mỗi sự kiện trong một bản ghi chứa các thông tin sau:

- Date: Ngày mà sự kiện xảy ra.
- Time: Thời gian diễn ra sự kiện.
- User: Tên người sử dụng của người dùng đã đăng nhập vào khi sự kiện xảy ra.
- Computer: Tên của máy tính.
- Event ID: Số định danh tương ứng với sự kiện.
- Source: Chương trình hoặc một ứng dụng thực hiện sự kiện..
- Type: Kiểu của sự kiện (information event, warning event, error message, security success audit event hoặc failure audit event).

2.1.4. Vị trí của Event log

- NT/Win2000/XP/Server 2003:
 - + Kiểu tập tin là **.evt*
 - + *%System root%\System32\config*
 - + Tên tập tin: *SecEvent.evt, AppEvent.evt, SysEvent.evt*
- Vista/Win7/Server 2008/Server 2012:
 - + Kiểu tập tin là **.evtx*
 - + *%System root%\System32\winevt\logs*
 - + Remote log server
 - + Tên tập tin: *Security.evtx, Application.evtx, System.evtx,*
 - + Vị trí mặc định có thể được thay đổi trong Registry.

2.1.5. Kịch bản phân tích Event log

Bởi vì nhiều tiến trình Windows chạy trước user hoặc ngay cả khi user logged on (boot time) hoặc không cần biết ngữ cảnh user (tiến trình bảo trì hệ thống). Windows thiết đặt một số tài khoản dịch vụ (service accounts) mà có thể sử dụng để cho phép các tiến trình chạy trong các phạm vi ngữ cảnh khác nhau.

SYSTEM: local account có quyền cao nhất; không hạn chế truy cập hệ thống.

LOCAL SERVICE: Bị giới hạn quyền (tương tự như một chuẩn USER account bình thường) được sử dụng cho các services mà không yêu cầu truy cập mạng. Chỉ có thể truy cập tài nguyên mạng thông qua NULL session.

NETWORK SERVICE: tương tự như LOCAL SERVICE nhưng lại dùng cho các dịch vụ trên network (cho phép đóng giả một chuẩn computer accounts và được chứng thực trên mạng). Có thể truy cập tài nguyên network tương tự như user account đã được chứng thực.

ANONYMOUS LOGON: Được thiết kế với các giao tiếp không cần có thông tin tài khoản tường minh (NULL session được sử dụng để xác minh truy cập tài nguyên). Tài khoản này vẫn được sử dụng trên Windows network để thoải mái thực hiện các thao tác như file and print sharing, browser list.

2.1.5.1. Theo dõi việc sử dụng Account

Kịch bản: Xác định tài khoản nào đang thử thực hiện đăng nhập và theo dõi tài khoản bị thỏa hiệp.

Các Event ID liên quan:

- 528 – Successful Logon
- 529 – Failed Logon
- 538 – Successful Logoff
- 540 – Successful Network Logon

Chú ý khi điều tra: Event ID 528-552 thường liệt kê các hoạt động logon vượt ngưỡng; danh sách các codes thông thường nhất xuất hiện.

Lưu ý: Logon event không thể ghi nhận khi backdoor, exploited services hoặc các hành động độc hại tương tự gán quyền truy cập đến hệ thống. Bởi vì, những hành động này sử dụng backchannel và sử dụng các APIs để đạt được việc truy cập.

Logon Type Code:

Type	Logon Title	Description
2	Interactive	Một người sử dụng đăng nhập vào giao diện điều khiển từ máy tính này.
3	Network	Một người sử dụng máy tính hoặc đăng nhập

		vào máy tính này từ mạng.
4	Batch	Kiểu batch logon này được sử dụng bởi các máy chủ, nơi mà quá trình có thể được thực hiện thay mặt cho một người sử dụng mà không cần can thiệp trực tiếp của họ.
5	Service	Một dịch vụ đã được bắt đầu bởi Service Control Manager.
7	Unlock	Máy trạm này đã được mở khóa.
8	NetworkCleartext	Một người sử dụng đăng nhập vào máy tính này từ mạng. Mật khẩu của người dùng đã được thông qua để gói xác thực trong hình thức unhashed của nó. Việc xây dựng trong các gói chứng thực tất cả thông tin băm trước khi gửi chúng qua mạng. Các thông tin không đi qua mạng không được mã hóa, còn gọi là dạng cleartext.
9	NewCredentials	Một người nhận bản token và các thông tin chỉ mới hiện nay cho các kết nối ra bên ngoài. Các phiên đăng nhập mới có thuộc tính như nhau, nhưng sử dụng thông tin khác nhau cho các kết nối mạng khác.
10	RemoteInteractive	Một người sử dụng đăng nhập vào máy tính này từ xa sử dụng Terminal Services hoặc Remote Desktop.
11	CachedInteractive	Một người sử dụng đăng nhập vào máy tính này với các thông tin mạng được lưu trữ cục bộ trên máy tính. Các bộ điều khiển miền không được liên lạc để xác minh các thông tin.

2.1.5.2. Phân tích truy cập tập tin và thư mục

Kịch bản: Nhận diện users thử truy cập các tập tin, thư mục, registry key, ... được bảo vệ.

Các Event ID liên quan:

- 560 – Object accessed
- 564 – Object deleted
- 567 – Kiểm tra các quyền trên đối tượng (read, write, delete, ...)

Chú ý khi điều tra:

- Event bao gồm timestamp, file hoặc folder và user account thử thực hiện truy cập.
- Lọc các Event ID 560 Failure Events để nhận diện users thử truy cập.

2.1.5.3. Tìm kiếm tài khoản lạ

Kịch bản: Giả sử có một tài khoản local account có tên là root mà có hành động map một chia sẻ network.

Các Event ID liên quan:

- Event ID 680 sẽ chỉ ra rằng root account thành công trong việc đăng nhập đến máy.
- Event ID 540 sẽ chỉ ra hành động network logon thành công ngay sau đó

2.1.5.4. Sự cài đặt của ứng dụng

Kịch bản: Quan sát logs để nhận ra các phần mềm không được phép hoặc độc hại được cài đặt, theo dõi quá trình phần mềm độc hại được gỡ bỏ trước đó và nhận diện việc cố thử cài đặt phần mềm.

Các Event ID liên quan:

- 1033 – Installation completed (trạng thái success/failure)
- 1034 – Application removal completed (trạng thái success/failure)
- 11707 – Installation completed successfully
- 11708 – Installation operation failed
- 11724 – Application removal completed successfully

Chú ý khi điều tra:

- Tất cả Event ID đều nằm ở Application Log
- Events chỉ logged khi ứng dụng sử dụng Windows Installer API

2.1.5.5. Các dịch vụ độc hại

Kịch bản: Phân tích logs cho việc phát hiện các dịch vụ độc hại chạy lúc khởi động và quan sát các dịch vụ started hoặc stopped xung quanh thời điểm bị thỏa hiệp.

Các Event ID liên quan:

- 7034 – Service crashed unexpectedly
- 7035 – Service sent a Start/Stop control
- 7036 – Service started or stopped
- 7040 - Start type changed (Boot|On Request|Disabled)

Chú ý khi điều tra:

- Tất cả Event ID đều nằm trong System Log
- Đa số malware sử dụng Service
- Service “lạ” được started tại lúc khởi động (malware)
- Services có thể bị crash có thể do tấn công như process injection

2.1.5.6. Xóa bỏ Event Log

Kịch bản: Xác định liệu rằng Event Logs có bị thay đổi

Event ID liên quan: 517 - Audit log cleared

Ghi chú khi điều tra:

- Chỉ có quyền Administrator mới được xóa log.
- Sau khi log bị xóa bỏ thì event 517 sẽ được đặt trong log.
- Không xây dựng cơ chế cho việc lựa chọn xóa bỏ các events.

2.1.5.7. Các thiết bị phần cứng không được xác thực

Kịch bản: Xác định liệu rằng có thiết bị phần cứng được cài đặt trên hệ thống.

Các Event ID liên quan: 20001 - Plug and Play driver install attempted (từ Vista và sau này).

Ghi chú khi điều tra:

- Nhận diện các thiết bị và serial number.
- Chỉ hiển thị lần đầu tiên thiết bị được gắn vào.

2.2.WINDOWS FIREWALL LOG

2.2.1. Giới thiệu Windows Firewall Log

Việc ghi log chỉ sẵn sàng đối với các thiết bị đã bật Windows Firewall. Tuy nhiên không phải toàn bộ các kết nối đều được ghi log lại, mà chỉ những kết nối được cấu hình bảo vệ bởi Windows Firewall.

Tất cả các lưu lượng đi ra gửi đến đích thành công đều không ghi log lại. Ngoài ra, lưu lượng gửi đi mà không bị chặn cũng không được ghi log.

Kích thước mặc định của file log trong Windows Firewall là 4.096 kilobyte (KB); kích thước tối đa là 32,767 KB.

Khi file log đạt đến kích thước giới hạn mà bạn thiết lập, các file log sẽ được đổi tên và một file log mới được tạo ra.

2.2.2. Cấu hình Windows Firewall Log

Phiên bản áp dụng thực hiện đối với Windows 7 và Windows 2008 R2.

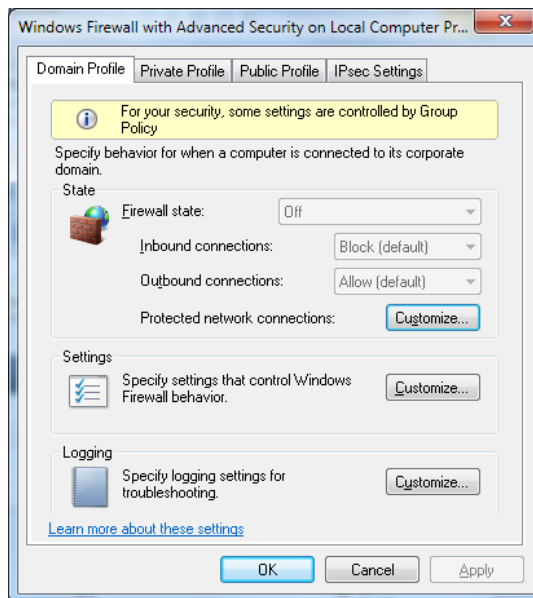
Windows Firewall là một trong những thành phần được cải tiến rất nhiều từ Windows XP đến Windows 7. Trong Windows 7 nó hoạt động như tường lửa hai chiều (Inbound/Outbound) với nhiều tính năng bảo mật tiên tiến. Điều này cũng được gọi là "Windows Firewall với Advanced security" trong Window 7.

Nhiều tính năng hơn, có nhiều thông tin chi tiết bạn cần nó cho vấn đề xử lý sự cố. Kể từ phiên bản này của hành vi tường lửa hoạt động hai chiều, bạn cần tốt số lượng ghi log để khắc phục vấn đề ở cấp kết nối mạng trên Windows 7 và Windows 2008 R2.

Phần này nói về việc làm thế nào để bật tính năng ghi log, vị trí của các file log, và ít lựa chọn hơn có sẵn trong khai thác gỡ.

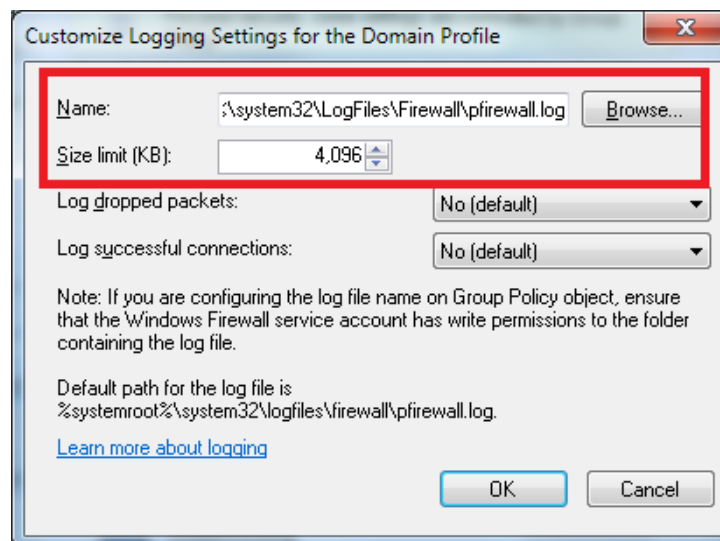
Để kích hoạt trên một máy tính duy nhất, chỉ cần vào Start -> Run -> và gõ wf.msc, điều này mở ra "Windows Firewall with Advanced Security".

Bây giờ hãy nhấp chuột phải vào "Windows Firewall with Advanced Security on Local Computer" bảng thuộc tính đó sẽ mở ra bên dưới.



Chọn tùy chọn “Customize” trong phần ghi log.

Thiết lập vị trí các tập tin log mà bạn muốn đặt các file log và cũng chọn giới hạn kích thước các file log.



Việc ghi log của bạn sẽ không bắt đầu cho đến khi bạn chọn "YES" với một trong các tùy chọn sau:

- Log dropped packets (ghi log đối với các gói tin bị chặn)
- Log Successful connections (ghi log tất cả các kết nối thành công)

Nhấn OK hai lần để hoàn thành cấu hình của bạn.

2.2.3. Vị trí Windows Firewall Log

- XP/Server 2003:

\systemroot\pfirewall.log

- Vista/Server 2008/Win7

\systemroot\System32\LogFiles\Firewall\pfirewall.log

2.2.4. Phân tích Firewall log

Các thông tin cần quan tâm khi phân tích Windows Firewall Log như:

- Sự kiện đã diễn ra: Hành động
- Date/Time: Date, Time
- Computers được đề cập: Source IP, Destination IP
- Ports được đề cập: Source port, Destination port
- Kiểu packet: Protocol, Tcpip Flags

Event 5152, Microsoft Windows security auditing.

test-pfirewall.log - Notepad

The Windows Filtering Platform has blocked a packet

Application Information:

- Process ID: 0
- Application Name: -

Network Information:

- Direction: Inbound
- Source Address: 189.185
- Source Port: 48451
- Destination Address: .211.25
- Destination Port: 443
- Protocol: 6

Filter Information:

- Filter Run-Time ID: 125691
- Layer Name: Transport
- Layer Run-Time ID: 13

Date/Time	Action	Protocol	Source IP	Source Port	Destination IP	Destination Port	Action/Flags
2012-02-10 12:29:29	ALLOW	TCP	.189.185	48451	.211.25	443	0 - 0 0 0 - - - SEND
2012-02-10 12:29:29	ALLOW	TCP	.189.185	48450	.211.25	443	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:29	ALLOW	TCP	.189.185	48451	.211.25	443	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:29	ALLOW	TCP	.189.185	54580	.211.25	8080	0 - 0 0 0 - - - SEND
2012-02-10 12:29:30	ALLOW	TCP	.189.185	48452	.211.25	443	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:30	ALLOW	TCP	.189.185	48453	.211.25	443	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:30	ALLOW	TCP	.189.185	48454	.211.25	443	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:30	ALLOW	TCP	.189.185	54596	.211.25	80	0 - 0 0 0 - - - SEND
2012-02-10 12:29:30	ALLOW	TCP	.189.185	54597	.211.25	80	0 - 0 0 0 - - - SEND
2012-02-10 12:29:30	ALLOW	TCP	.189.185	54598	.211.25	80	0 - 0 0 0 - - - SEND
2012-02-10 12:29:32	ALLOW	UDP	.189.185	53019	.211.25	902	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:32	ALLOW	UDP	.189.185	58689	.211.25	902	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:39	ALLOW	UDP	.189.185	58185	.211.25	902	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:42	ALLOW	UDP	.189.185	54931	.211.25	902	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:42	ALLOW	UDP	.189.185	31	.211.25	53	0 - 0 0 0 - - - SEND
2012-02-10 12:29:42	ALLOW	TCP	.189.185	54621	.211.25	443	0 - 0 0 0 - - - SEND
2012-02-10 12:29:44	ALLOW	UDP	.189.185	57998	.211.25	902	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:44	ALLOW	TCP	.189.185	54622	.211.25	5989	0 - 0 0 0 - - - SEND
2012-02-10 12:29:45	ALLOW	TCP	.189.185	54623	.211.25	5989	0 - 0 0 0 - - - SEND
2012-02-10 12:29:49	ALLOW	UDP	.189.185	56010	.211.25	902	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:52	ALLOW	UDP	.189.185	50920	.211.25	902	0 - 0 0 0 - - - RECEIVE
2012-02-10 12:29:54	ALLOW	UDP	.189.185	58103	.211.25	902	0 - 0 0 0 - - - RECEIVE

2.3. IIS LOG FILE

2.3.1. Giới thiệu IIS

IIS là viết tắt của từ Internet Information Services(các dịch vụ cung cấp thông tin Internet), IIS được đính kèm với các phiên bản của Windows. IIS là các dịch vụ dành cho máy chủ chạy trên nền Hệ điều hành Window nhằm cung cấp và phân tán các thông tin lên mạng, nó bao gồm nhiều dịch vụ khác nhau như Web Server, FTP Server,...

Nó có thể được sử dụng để xuất bản nội dung của các trang Web lên Internet/Intranet bằng việc sử dụng “Phương thức chuyển giao siêu văn bản” – Hypertext Transport Protocol (HTTP).

Như vậy, sau khi bạn thiết kế xong các trang Web của mình, nếu bạn muốn đưa chúng lên mạng để mọi người có thể truy cập và xem chúng thì bạn phải nhờ đến một Web Server, ở đây là IIS.

Nhiệm vụ của IIS là tiếp nhận yêu cầu của máy trạm và đáp ứng lại yêu cầu đó bằng cách gửi về máy trạm những thông tin mà máy trạm yêu cầu.

Bạn có thể sử dụng IIS để:

- Xuất bản một Website của bạn trên Internet
- Tạo các giao dịch thương mại điện tử trên Internet (hiện các catalog và nhận được các đơn đặt hàng từ người tiêu dùng)
- Chia sẻ file dữ liệu thông qua giao thức FTP.
- Cho phép người ở xa có thể truy xuất database của bạn (gọi là Database remote access).

IIS sử dụng các giao thức mạng phổ biến là HTTP và FTP (File Transfer Protocol) và một số giao thức khác như SMTP, POP3,... để tiếp nhận yêu cầu và truyền tải thông tin trên mạng với các định dạng khác nhau.

Một trong những dịch vụ phổ biến nhất của IIS mà chúng ta quan tâm trong giáo trình này là dịch vụ WWW (World Wide Web), nói tắt là dịch vụ Web.

Dịch vụ Web sử dụng giao thức HTTP để tiếp nhận yêu cầu (Requests) của trình duyệt Web (Web browser) dưới dạng một địa chỉ URL (Uniform Resource Locator) của một trang Web và IIS phản hồi lại các yêu cầu bằng cách gửi về cho Web browser nội dung của trang Web tương ứng.

2.3.2. IIS Log file

2.3.2.1. Kích hoạt IIS Logging

Bạn phải kích hoạt việc ghi log cho mỗi Website, FTP và các máy chủ SMTP riêng để thu thập và chuyển đổi dữ liệu sử dụng. Hãy chắc chắn rằng bạn đã chọn Enable Logging trên trang Windows Internet Information Services.

Các bước thực hiện như sau:

- Trong cửa sổ Windows Internet Information Services, kích chuột phải vào trang web hoặc máy chủ, và sau đó nhấp vào Properties.
- Hộp thoại properties được hiển thị.
- Trên tab Web Site, chọn Enable Logging và nhấp W3C Extended Format Log File từ Active danh sách định dạng log.
- Bấm Apply, và sau đó nhấp vào Properties.
- Hộp thoại Logging thuộc tính mở rộng được hiển thị.
- Trên tab General Properties, thiết lập các thuộc tính chung như lịch trình log (hàng ngày, hàng tuần, hàng tháng, vv) và log vị trí file.
- Trên tab Properties Extended, chọn các thuộc tính mà bạn muốn ghi log. Chọn tất cả các thuộc tính được khuyến khích. Bạn cũng có thể chọn các thuộc tính trình Accounting; Tuy nhiên, thông tin này không phải là hữu ích cho các phản hồi và không được ghi vào file CSR.
- Nhấn OK để lưu các thiết lập và đóng hộp thoại, và sau đó nhấn OK lần nữa để đóng hộp thoại thuộc tính.

2.3.2.2. Vị trí IIS log file

IIS 5 – IIS 6: %windir%\System32\LogFiles\exYYMMDD.log

IIS 7 – IIS 7.5: %SystemDrive%\inetpub\logs\LogFiles

u_exYYMMDD.log (utf-8)

exYYMMDD.log (ANSI)

2.3.3. IIS Log File Format

Việc sử dụng và quản lý Accounting Data Collector cho Microsoft IIS thu thập những dữ liệu được chứa trong một file log được tạo ra bởi IIS. Các lĩnh vực được chứa trong các file log được xác định bởi các tính chất đó đã được lựa chọn khi ghi nhật ký đã được kích hoạt cho hệ điều hành IIS Server. File log này cung cấp số liệu hữu ích như: Số byte được gửi từ một địa chỉ IP client đến một địa chỉ IP server, số byte gửi từ một địa chỉ IP của máy chủ đến một địa chỉ IP của khách hàng, tên người dùng và địa chỉ IP, tên trang web IIS, tên máy chủ và Địa chỉ IP.

Field Name	Description/Values
Date	The date that the action occurred.
Time	The time that the action occurred.
c-ip (client IP address)	The IP address of the client that accessed the server.
cs-username (user name)	The name of the authenticated user who accessed the server. This does not include anonymous users, which are represented by a hyphen (-).
s-sitename (service name)	The Internet service and instance number that was accessed by the client.
s-computername (server name)	The name of the server on which the log entry was generated.
s-ip (server IP address)	The IP address of the server on which the log entry was generated.
s-port (server port)	The port number the client was connected to.
cs-method (method)	The action the client was trying to perform (for example, a GET method).
cs-uri-stem (URI stem)	The resource accessed (for example, Default.htm).
cs-uri-query (URI query)	The query, if any, the client was trying to perform.
sc-status (protocol status)	The status of the action, in HTTP or FTP terms.
sc-win32-status (protocol status)	The status of the action, in terms used by Windows.
sc-bytes (bytes sent)	The number of bytes sent by the server.
cs-bytes (bytes received)	The number of bytes received by the server.
time-taken	The length of time the action took.
cs-version (protocol version)	The protocol (HTTP, FTP) version used by the client. For HTTP, this will be either HTTP 1.0 or HTTP 1.1.
cs-host (host)	Displays the content of the host header.
cs(User-Agent) (user agent)	The browser used on the client.
cs(Cookie) (cookie)	The content of the cookie sent or received, if

Field Name	Description/Values
	any.
cs(Referrer)	The previous site visited by the user. This site provided a link to the current site.

2.3.4.W3C Extended Log Format

Khi W3C Extended Log được kích hoạt trên máy chủ phiên nó có chức năng như hình thức tập trung khai thác gổ cho tất cả các nhóm URL dưới phiên máy chủ. Một File log duy nhất là duy trì cho tất cả các nhóm URL trong phiên máy chủ.

Field	Appears As	Description
Date	Date	The date on which the activity occurred.
Time	Time	The time, in coordinated universal time (UTC), at which the activity occurred.
Service Name and Instance Number	s-sitename	The Internet service name and instance number that was running on the client.
Server Name	s-computername	The name of the server on which the log file entry was generated.
Server IP Address	s-ip	The IP address of the server on which the log file entry was generated.
Method	cs-method	The requested verb, for example, a GET method.
URI Stem	cs-uri-stem	The target of the verb, for example, Default.htm.
URI Query	cs-uri-query	The query, if any, that the client was trying to perform. A Universal Resource Identifier (URI) query is necessary only for dynamic pages.
Server Port	s-port	The server port number that is configured for the service.

User Name	cs-username	The name of the authenticated user that accessed the server. Anonymous users are indicated by a hyphen.
Client IP Address	c-ip	The IP address of the client that made the request.
Protocol Version	cs-version	The HTTP protocol version that the client used.
User Agent	cs(User-Agent)	The browser type that the client used.
Cookie	cs(Cookie)	The content of the cookie sent or received, if any.
Referrer	cs(Referrer)	The site that the user last visited. This site provided a link to the current site.
Host	cs-host	The host header name, if any.
HTTP Status	sc-status	The HTTP status code.
Protocol Substatus	sc-substatus	The substatus error code.
Win32 Status	sc-win32-status	The Windows status code.
Bytes Sent	sc-bytes	The number of bytes sent by the server.
Bytes Received	cs-bytes	The number of bytes received and processed by the server.
Time Taken	time-taken	The length of time that the action took, in milliseconds.

Ví dụ về W3C Extended Log Format

```
#Software: Microsoft HTTP Server API 2.0
#Version: 1.0      // the log file version as it's described by
"http://www.w3.org/TR/WD-logfile".
#Date: 2002-05-02 17:42:15  // when the first log file entry was
recorded, which is when the entire log file was created.
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-stem
cs-uri-query sc-status cs(User-Agent)
```

```

2002-05-02 17:42:15 172.22.255.255 - 172.30.255.255 80 GET
/images/picture.jpg - 200
Mozilla/4.0+(compatible;MSIE+5.5;+Windows+2000+Server)

```

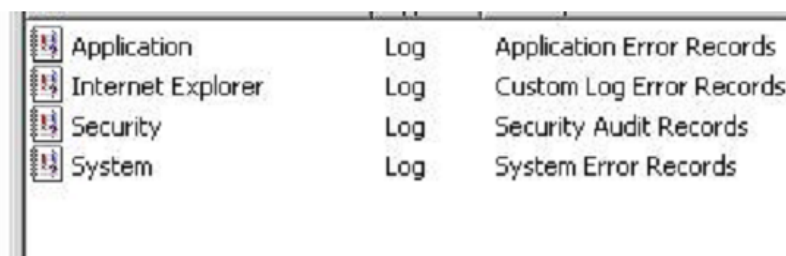
2.4. CÔNG CỤ PHÂN TÍCH

2.4.1. Windows Event Viewer

Quản lý hệ thống mạng không thể thiếu phần giám sát hệ thống, ngoài việc bạn ghi lại các tiến trình trong hệ thống bạn còn phải biết điều chỉnh thiết lập chỉ ghi lại những yếu tố cần thiết. Chẳng hạn một máy chủ File Server bạn chỉ cần giám quá trình truy cập tài nguyên, máy chủ Active Directory giám sát quá trình log on vào hệ thống.

Trong phần hai của bài viết tôi sẽ giới thiệu với các bạn công một công cụ ghi lại các event của hệ thống đó là Event Viewer. Event viewer là một công cụ tích hợp trong Windows cho phép bạn xem lại các sự kiện đã xảy ra trong hệ thống một cách chi tiết với nhiều tham số cụ thể như: user, time, computer, services... Các sự kiện rời rạc được lọc lại thành những sự kiện giống nhau giúp chúng ta lấy được những thông tin cần thiết một cách nhanh nhất. Trong Event viewer đã phân vùng các sự kiện riêng biệt cho từng ứng dụng, một máy chủ cài đặt mặc định sẽ có ba phân vùng trong event viewer:

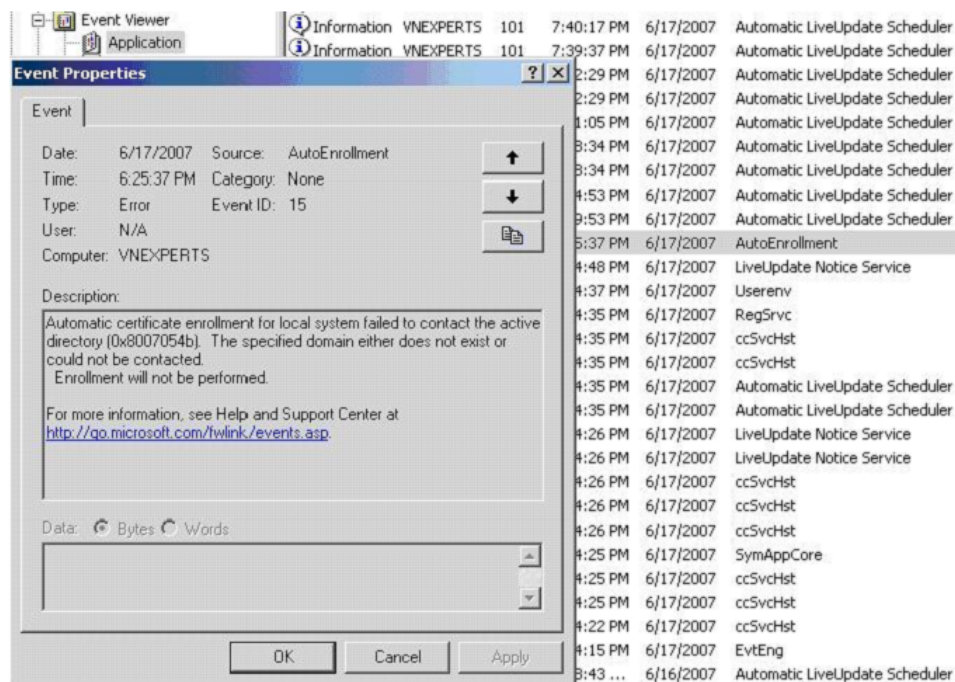
- Application
- Security
- System



2.4.1.1. Application log

Application log ghi lại sự kiện của các ứng dụng khác từ các nhà sản xuất khác như symantec hay các ứng dụng mail... Thường thiết lập trong application

là mặc định của các ứng dụng nên chúng ta chỉ có thể đọc nó mà không thiết lập được.



Trong ví dụ trên application log chỉ được phần mềm symantec sử dụng.

2.4.1.2. Security log

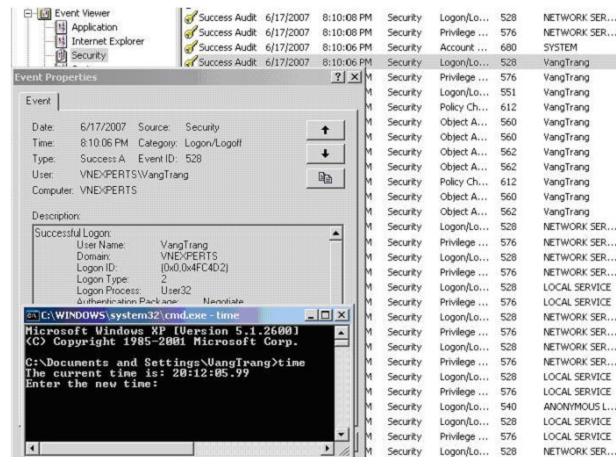
Đây là một trong những log quan trọng nhất trong hệ thống, nó ghi lại toàn bộ các thiết lập audit trong group policy. Nhưng trong các thiết lập group policy quan trọng nhất là thiết lập giám sát quá trình login vào hệ thống, truy cập dữ liệu.



Trong thiết lập này tôi chỉ thiết lập giám sát quá trình truy cập login log-off hệ thống. Nếu với thiết lập như trên toàn bộ người dùng logon hay logoff vào hệ thống đều được ghi lại sau khi thiết lập trong group policy các bạn nên logoff

hoặc restart lại máy bởi các thông tin chỉnh trong group policy bản chất là chỉnh các thông số trong registry.

Giờ tôi logoff ra và login vào sẽ thấy ghi lại trong security log.



Sau khi login vào máy tính mở event viewer ra xem và tôi phát hiện ra hệ thống đã lưu lại username: vangtrang computer: vnexperts, event: success audit, time: 8:10:06PM

Vậy ý nghĩa của việc xem lại log này là gì: bạn hãy tưởng tượng một dữ liệu trong máy của bạn đã bị mất và trong log ghi lại là đã được xóa lúc 12h đêm vậy bạn cần quy trách nhiệm đó cho ai, bạn cần biết trong thời điểm đó những ai đang online và login, logoff trong thời gian đó.

Thiết lập giám sát một folder dữ liệu quan trọng, với yêu cầu đặt ra là giám sát toàn bộ các quá trình truy cập các action cụ thể với folder này. Trong ổ E có thư mục quan trọng VNEDATA việc cần thiết của bạn là đưa ra các thiết lập giám sát toàn bộ truy cập vào folder này.

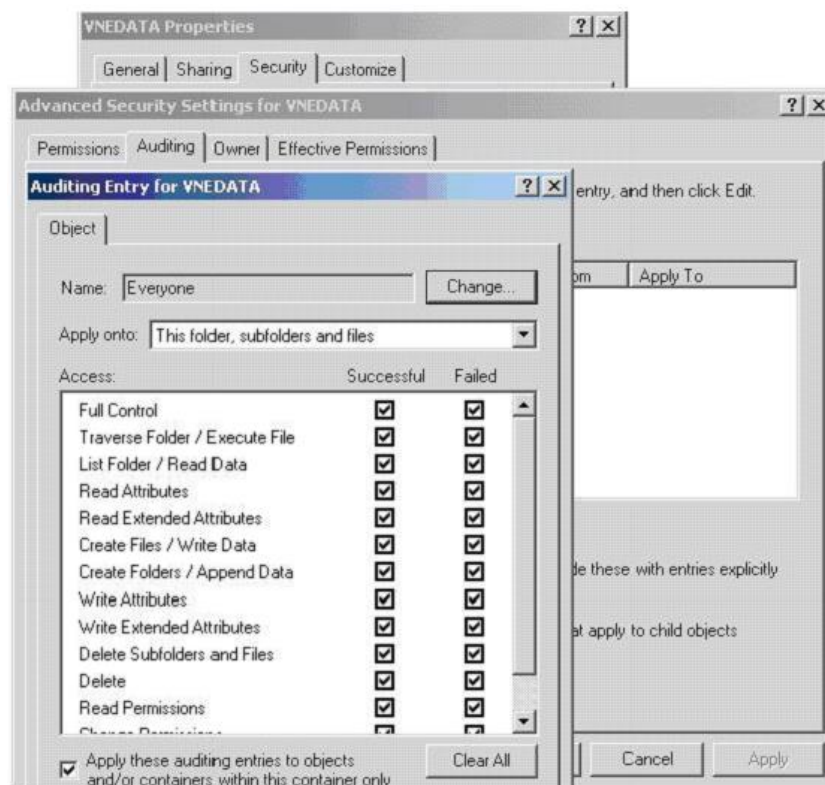
Bước 1: thiết lập audit object access trong group policy ở cả chế độ success và fails



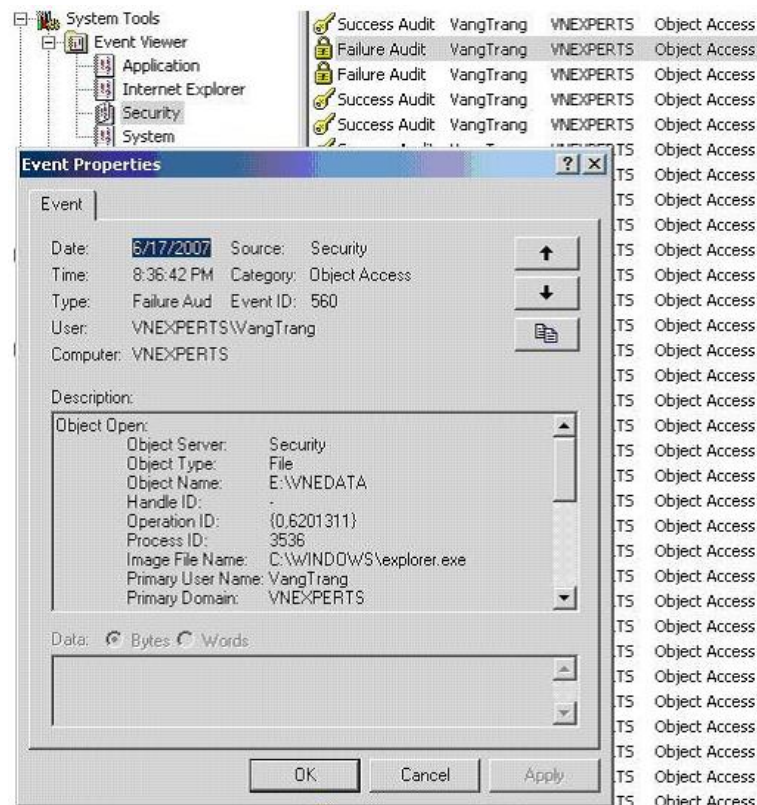
Với thiết lập trong group policy có nghĩa bạn chỉ enable tính năng cho phép hệ thống ghi lại mà thôi, mặc định hệ thống sau khi thiết lập này sẽ ghi lại event với các đối tượng hệ thống như registry... Còn muốn một quá trình truy cập vào folder mà được lưu lại thì phải thiết lập trên folder đó

Bước 2: thiết lập audit trên folder

Chuột phải vào folder chọn properties sang tab security chọn advanced, chuyển sang tab audit chọn trong cửa sổ add chúng ta add group với tên là everyone (everyone là một system group trong Windows).



Sau khi thiết lập bạn restart lại máy và thử truy cập vào folder này xem trong event viewer có ghi các sự kiện với folder này không.

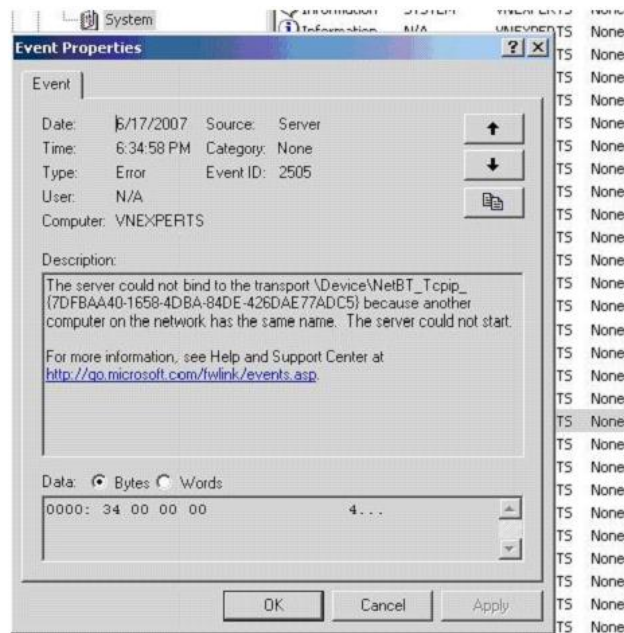


Nhìn vào event ta nhận thấy vào lúc 8:36:42 PM user với tên là vangtrang từ máy tính có tên VNEXPERS đã bị failure trong quá trình truy cập vào folder E:\VNEDATA. Ứng dụng của tính năng audit và xem lại các event cho ta phát hiện những kẻ truy cập bất hợp pháp và quy trách nhiệm cụ thể cho những kẻ phá hoại.

2.4.1.3. System Log

System log được thiết lập mặc định của hệ thống giúp chúng ta xem lại các sự kiện: Bật, tắt, pause, disable, enable các services của hệ thống.

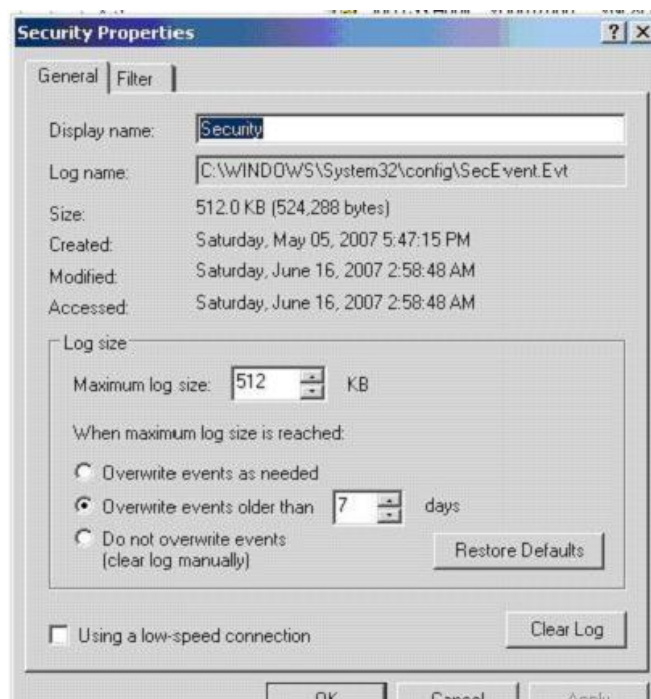
Ví như một service bật bị lỗi trong thời điểm nào nó sẽ ghi lại trong system log của event viewer.



Với thông event service với tên là Server đã bị lỗi do trong mạng LAN có máy tính trùng tên hoặc trùng địa chỉ IP.

2.4.1.4. Log Properties

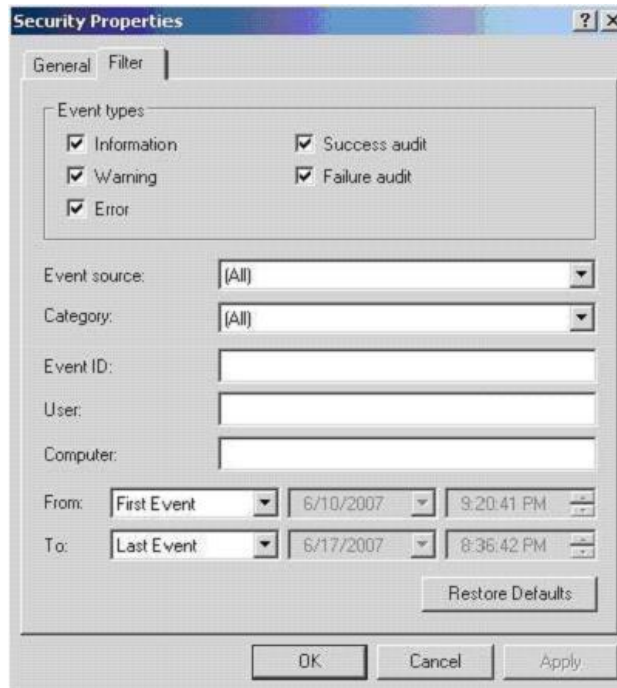
Log properties giúp chúng ta cấu hình dung lượng file log, cách xóa các event cũ đi như thế nào, và những tính năng lọc các sự kiện.



Đây là thiết lập cho security properties: Với file log tên là gì và ở đâu:
C:\Windows\System32\Config\SecEvent.Evt

Dung lượng tối đa cho file log này là 512 KB bạn có thể cấu hình lại to hơn hoặc nhỏ hơn, nếu dung lượng file long lớn hơn 512 KB hệ thống sẽ tự xóa các sự kiện cũ theo thuật toán First in First out – (vào trước vào thì ra trước).

Nếu dung lượng chưa được 512KB nhưng với thiết lập mặc định các event sẽ bị xóa sau 7 ngày.



Trong tab này với khi chưa cấu hình lọc mặc định sẽ hiển thị toàn bộ các sự kiện bạn có thể lọc chỉ hiển thị theo "event types: như information, warning, error, success audit, hay failure audit".

Hoặc có thể thiết lập lọc các sự kiện theo thời gian và ID của các sự kiện Event viewer là một tool quan trọng trong việc giám sát hệ thống dựa vào công cụ này người quản trị sẽ phát hiện ra những kẻ truy cập bất hợp pháp vào những thời điểm cụ thể, với tính năng lọc giúp bạn giới hạn những sự kiện cần thiết giám sát.

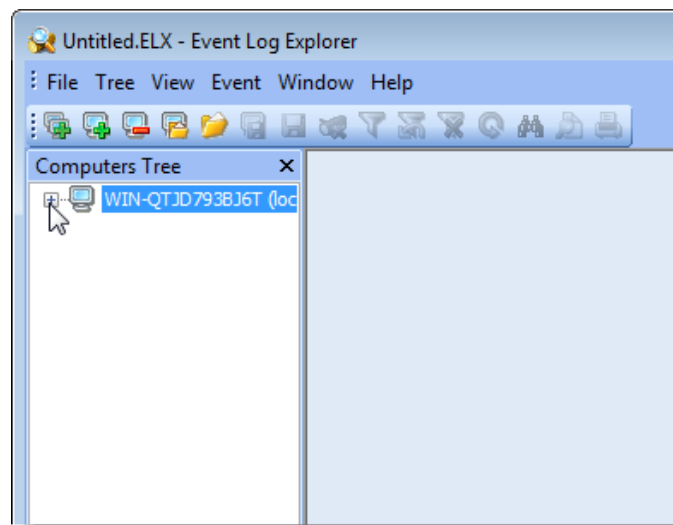
2.4.2. Event Log Explorer

Công cụ miễn phí với đăng ký sử dụng cá nhân, Event Log Explorer, có thể thay thế cho Windows Event Viewer. Event Log Explorer hiển thị số lượng thông tin như Event Viewer, nhưng ngoài ra nó còn cho phép tra cứu nhanh các Event ID trên Internet. Chỉ cần kích phải vào một sự kiện, bạn sẽ có thể tra cứu

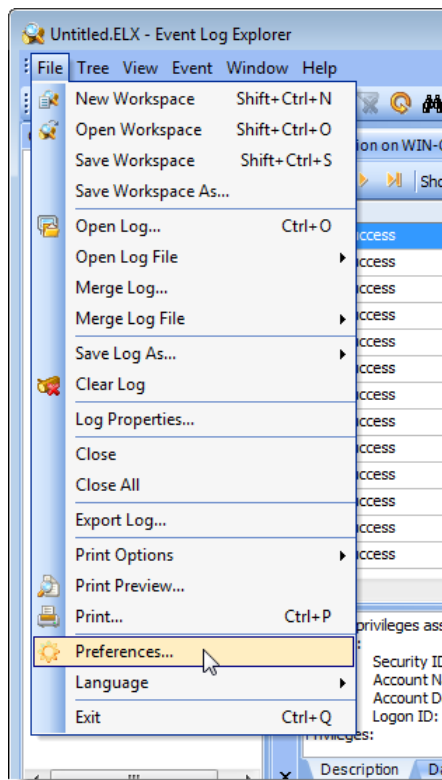
Event ID trong cơ sở dữ liệu EventID.Net hoặc trong Microsoft Knowledge Base.

Để cài đặt Event Log Explorer, giải nén file *.zip*, sau đó kích đúp file *.exe*. Thực hiện theo các chỉ dẫn trong wizard cài đặt. Nếu trước đó bạn không chọn khởi chạy Event Log Explorer ở cuối quá trình cài đặt, hãy khởi chạy chương trình từ desktop hoặc menu Start.

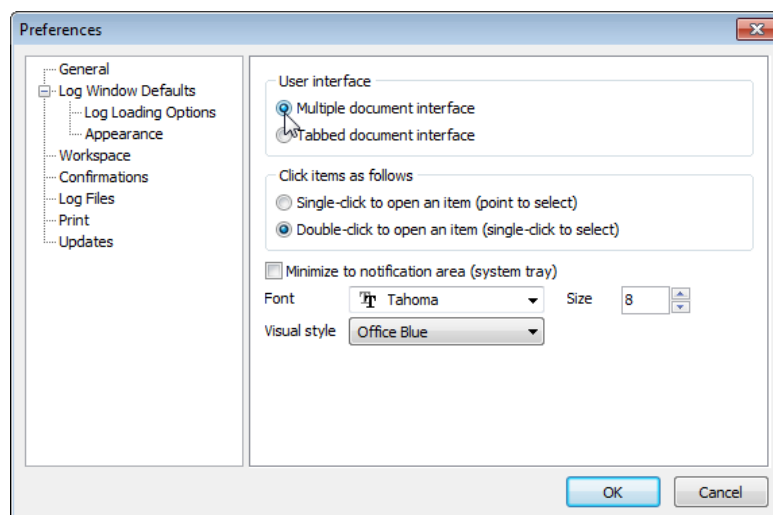
Khi cửa sổ Event Log Explorer xuất hiện, kích dấu cộng trong mục Computer Tree để mở danh sách.



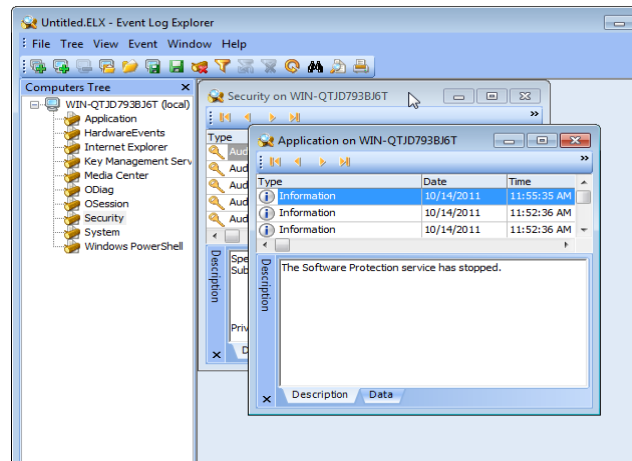
Có hai phương pháp xem bản ghi sự kiện, các tab và Multiple document interface (MDI). Để thay đổi khung nhìn, hãy chọn Preferences từ menu File.



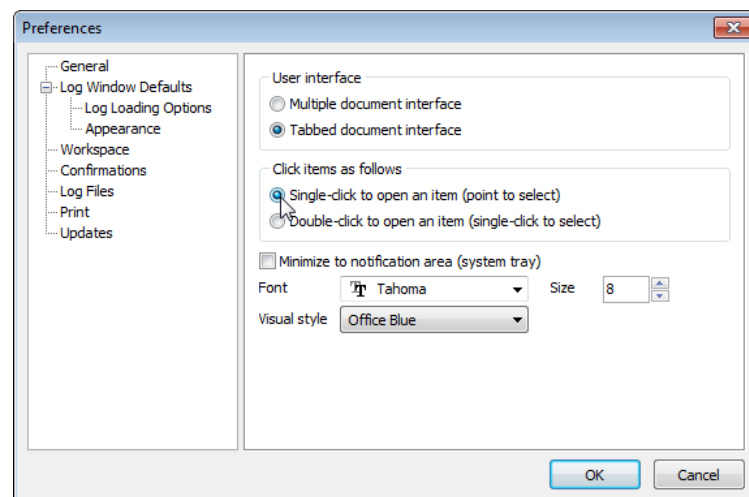
Trong hộp thoại Preferences, hãy chọn General trong menu cây bên trái. Chọn Multiple document interface hoặc Tabbed document interface trong hộp thoại User interface. Kích OK để lưu các thay đổi.



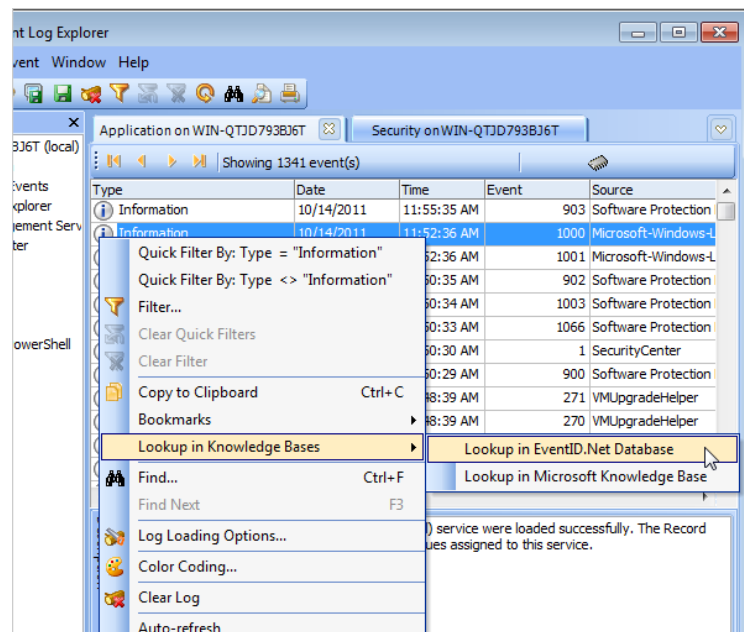
Multiple document interface sẽ hiển thị như hình bên dưới. Mỗi tài liệu được hiển thị trong một cửa sổ riêng biệt trong ứng dụng.



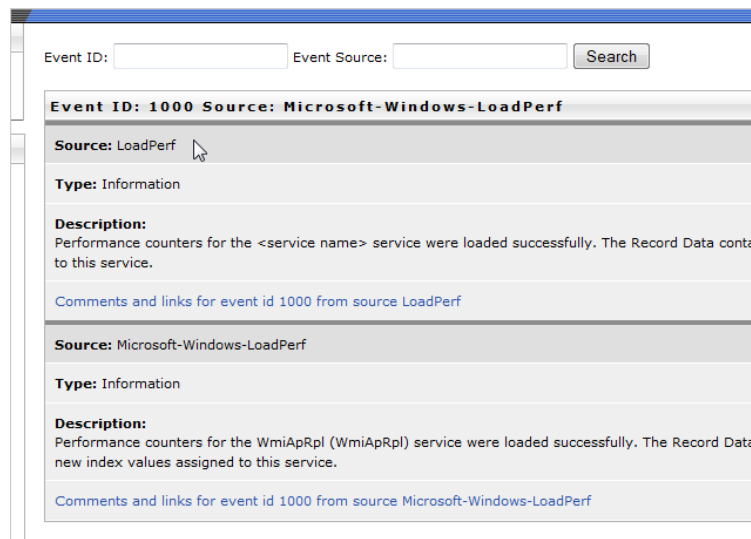
Bạn cũng có thể thiết lập để cho phép mở một bản ghi khi kích đơn hoặc kích đúp vào nó bằng cách chọn tùy chọn trong cửa sổ General của hộp thoại Preferences.



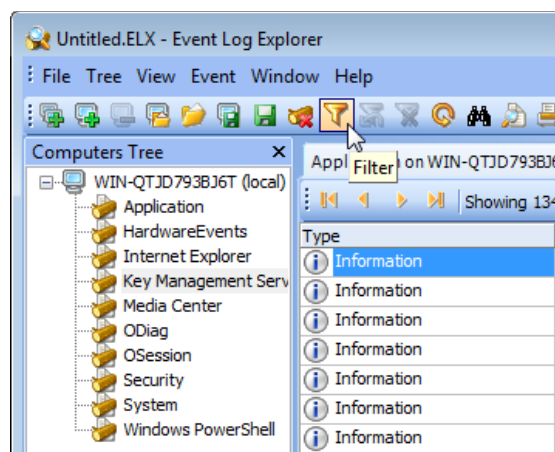
Một trong những tính năng hữu dụng nhất của Event Log Explorer (hữu dụng hơn Windows Event Log Viewer) là khả năng tra cứu dễ dàng các Event ID trong hai cơ sở dữ liệu trực tuyến khác nhau. Để thực hiện điều này, kích phải vào một sự kiện trong panel bên phải và chọn Lookup in Knowledge Bases từ menu xuất hiện. Có hai tùy chọn hiển thị trên menu con. Chọn tùy chọn nào là phụ thuộc vào việc bạn muốn tra cứu event ID trong cơ sở dữ liệu EventID.Net hay trong Microsoft Knowledge Base.



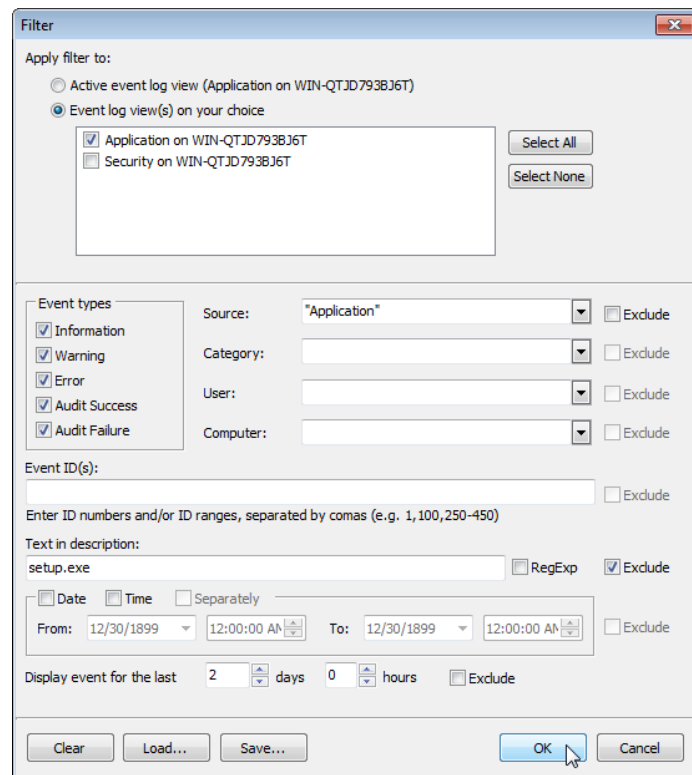
Cho ví dụ, hình dưới đây hiển thị Event ID 1000 trong EventID.Net.



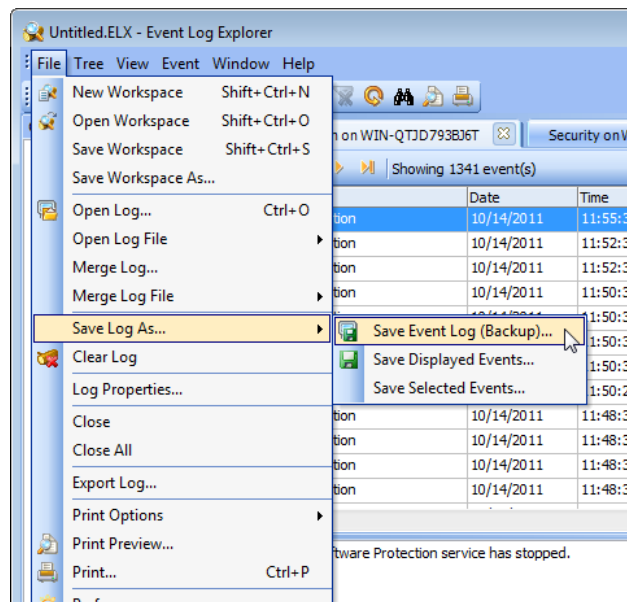
Có thể lọc các bản ghi. Để thực hiện điều này, kích Filter trên thanh công cụ. Lưu ý: Có thể chọn Filter từ menu View hoặc nhấn Ctrl + L.



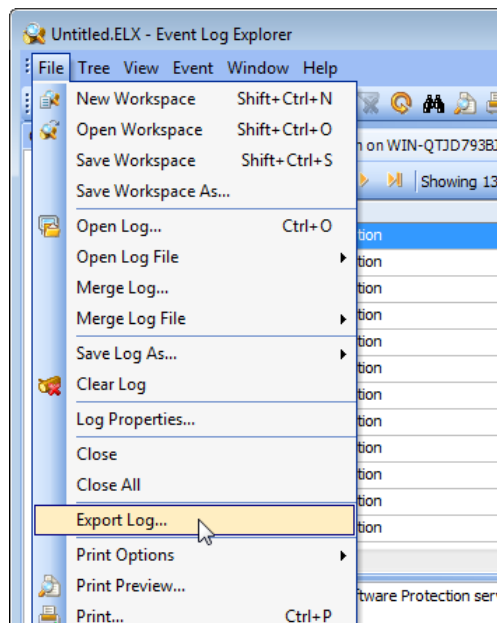
Sử dụng hộp thoại Filter để chỉ định bản ghi nào sẽ được lọc và nhập vào các tiêu chuẩn lọc. Kích OK để chấp nhận thay đổi và xem danh sách lọc trên cửa sổ chính của Event Log Explorer.



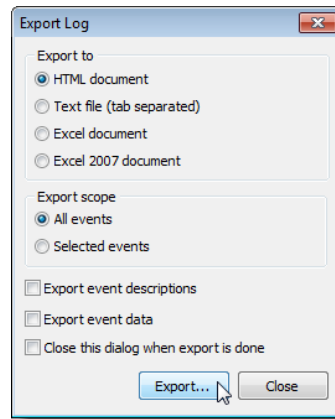
Bạn cũng có thể backup các bản ghi sự kiện. Để thực hiện, hãy chọn Save Log As | Save Event Log từ menuFile. Nhập vào tên của file cần backup và chọn kiểu file là .evt hay .evtx. Sử dụng kiểu file .evt cho các file backup bản ghi sự kiện khi muốn mở chúng trong Windows XP hoặc các phiên bản cũ hơn. Kiểu file .evtx áp dụng cho các file backup bản ghi sự kiện được mở trong Windows 7 hoặc Vista.



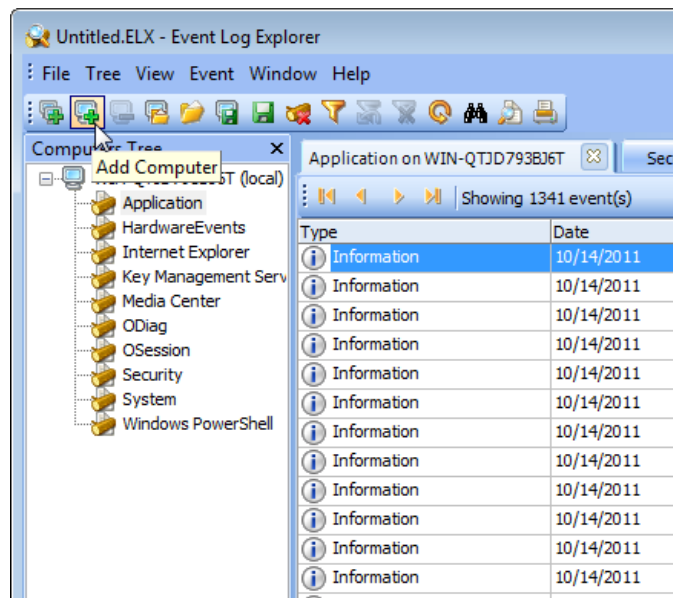
Nếu muốn xem thông tin bản ghi sự kiện bên ngoài Event Log Explorer, bạn có thể export các bản ghi thành các định dạng khác. Để export bản ghi hiện đang mở, hãy chọn Export Log từ menu File.



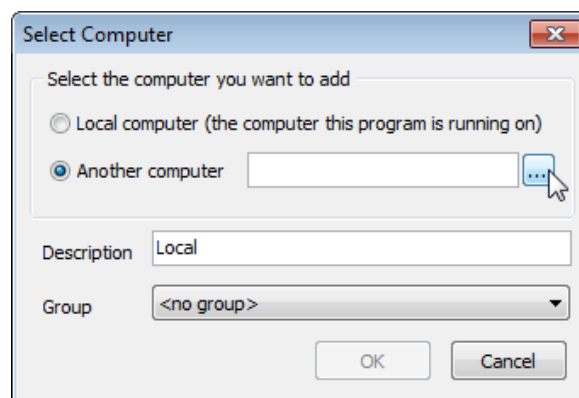
Hộp thoại Export Log sẽ xuất hiện. Chọn định dạng cho file bản ghi được export từ hộp thoại Export to và chọn xem bạn muốn export tất cả hay chỉ các sự kiện được chọn từ hộp thoại Export scope. Có thể chỉ định để export thông tin mô tả và dữ liệu sự kiện nếu muốn. Để tự động đóng hộp thoại Export Log khi quá trình export kết thúc, chọn Close this dialog when export is done. Kích Export để bắt đầu quá trình export.



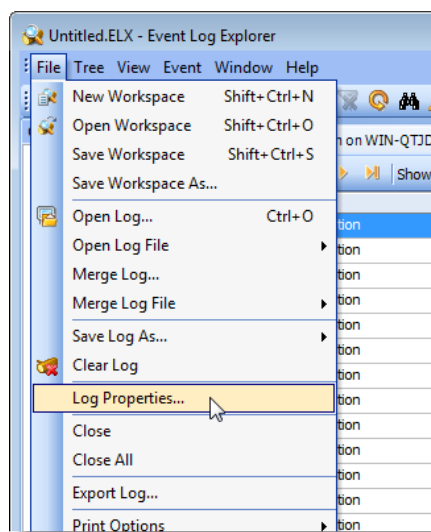
Nếu muốn xem các bản ghi sự kiện trên các máy tính khác, kích Export trên thanh công cụ. Lưu ý: Bạn cũng có thể chọn Add Computer từ menu.



Chọn tùy chọn Add Computer và sử dụng nút ... để chọn máy tính trong mạng. Nhập vào thông tin mô tả, chọn nhóm và kích OK để kết nối máy tính.



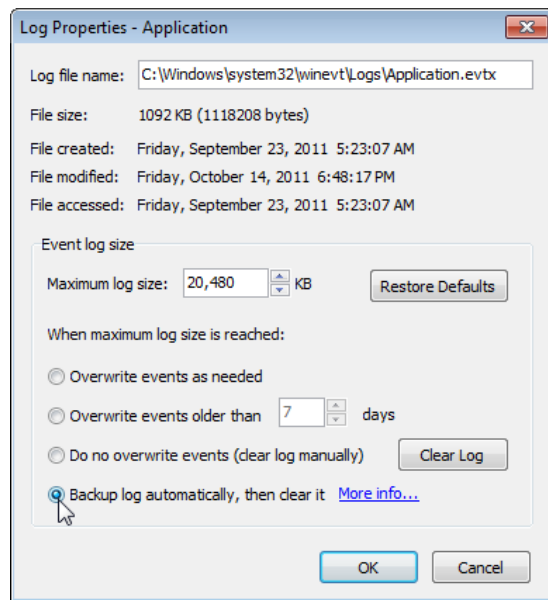
Để thay đổi thuộc tính của một bản ghi sự kiện hiện đang được chọn, hãy chọn Properties từ menu File. Lưu ý: Bạn cũng có thể kích phải vào bản ghi sự kiện trong menu bên trái và chọn Properties từ menu xuất hiện.



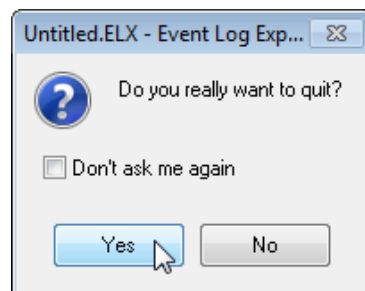
Hộp thoại Log Properties sẽ hiển thị. Bản ghi sự kiện có các thuộc tính này sẽ hiển thị trên thanh tiêu đề của hộp thoại.

Trong Event Log Explorer chúng ta hoàn toàn có thể thay đổi kích thước cực đại của bản ghi sự kiện. Để thực hiện điều này, nhập vào kích thước của file bản ghi trong hộp Maximum log size hoặc sử dụng mũi tên để chọn kích thước. Ngoài các tùy chọn phổ biến, bạn còn có một tùy chọn mở rộng là backup các bản ghi tự động khi đạt đến kích thước tối đa. Để có thêm thông tin về backup tự động các file bản ghi, kích liên kết More info để mở những chủ đề trợ giúp tương ứng. File trợ giúp sẽ mô tả nơi các file được lưu và cách đặt tên file được sử dụng như thế nào.

Lưu ý: Cần bảo đảm bạn không cho phép backup quá nhiều file bản ghi vì theo thời gian công việc này sẽ tốn rất nhiều không gian trên ổ cứng máy tính. Kiểm tra định kỳ các file và chuyển chúng sang ổ cứng khác hoặc xóa đi nếu cảm thấy không cần thiết.

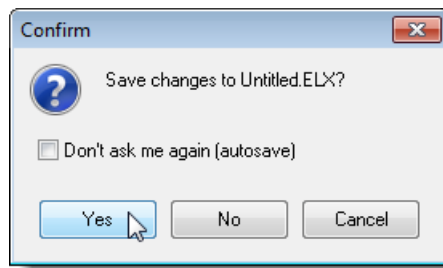


Để đóng Event Log Explorer, chọn Exit từ menu File để đóng Event Log Explorer. Hộp thoại dưới đây sẽ hỏi bạn có chắc chắn muốn thoát chương trình hay không. Nếu không muốn thấy hộp thoại này mỗi khi đóng Event Log Explorer, hãy tích hộp kiểm Don't ask me again. Kích Yes để tiếp tục đóng chương trình.

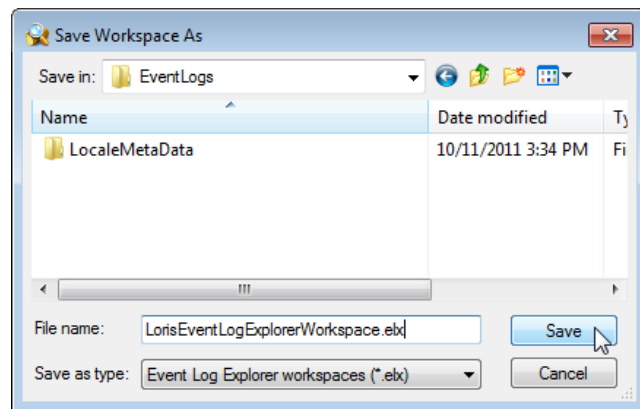


Event Log Explorer sẽ lưu workspace của bạn vào một file để khi mở lại file này, bạn sẽ thấy các tab, các thiết lập vẫn được duy trì như cũ. Nếu thực hiện một số thay đổi đối với workspace hiện hành trong Event Log Explorer, bạn sẽ thấy xuất hiện hộp thoại dưới đây. Nếu không lưu workspace, tên file sẽ được liệt kê là Untitled.ELX. Nếu muốn lưu các thay đổi của workspace, hãy kích Yes.

Bạn sẽ thấy có tùy chọn Don't ask me again. Nếu muốn chọn tùy chọn đó để lưu các thay đổi đối với workspace, khi đó bất cứ thay đổi nào bạn thực hiện trong Event Log Explorer ở về sau này sẽ tự động được lưu.



Nếu đã chọn lưu các thay đổi của workspace và đây là thời điểm đầu tiên lưu workspace, khi đó hộp thoại Save Workspace As sẽ xuất hiện. Điều hướng đến vị trí bạn muốn lưu các thiết lập của workspace, nhập vào tên cho workspace trong mục File name sau đó kích Save. Lưu ý là bạn có thể lưu nhiều workspace trong Event Log Explorer.

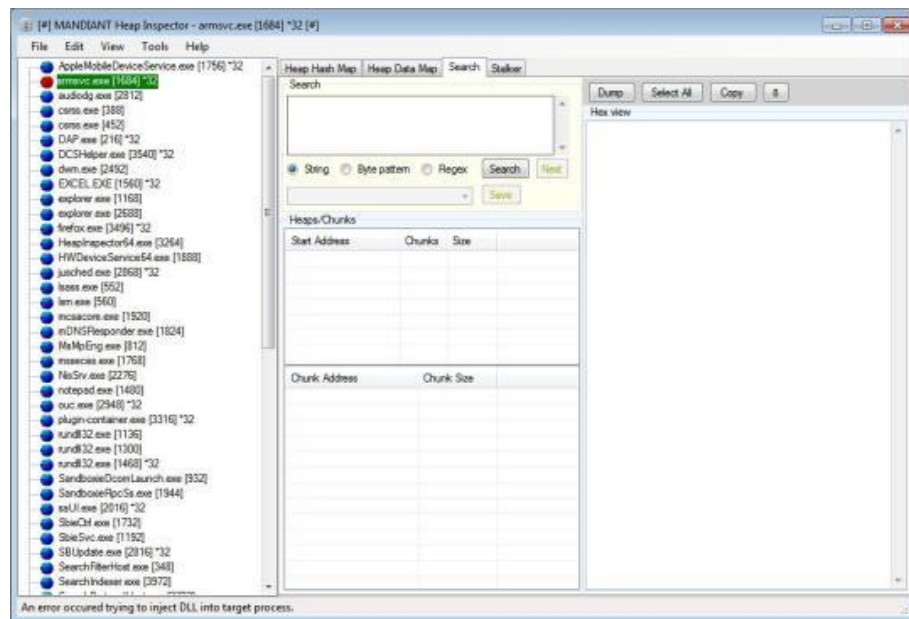


Có thể nói Event Log Explorer là một công cụ hữu dụng đáng để bổ sung vào hộp công cụ phần mềm của mỗi người. Chỉ có một hạn chế của phiên bản miễn phí là không cho phép người dùng kết nối quá ba máy tính.

2.4.3. Mandiant

Mandiant Highlighter là một tiện ích miễn phí được thiết kế chủ yếu cho các nhà phân tích an ninh và quản trị hệ thống. Highlighter cung cấp người dùng với ba quan điểm của bản ghi hoặc tập tin văn bản được phân tích: một quan điểm văn bản cho phép người sử dụng để làm nổi bật các từ khóa thú vị và loại bỏ các nếp với "tiếng tốt" nội dung, một cái nhìn đầy đủ nội dung đồ họa cho thấy tất cả nội dung và cấu trúc đầy đủ của tập tin kết xuất như một hình ảnh mà là tự động có thể chỉnh sửa thông qua giao diện người dùng, và xem một biểu đồ thể hiện mô hình trong tập tin theo thời gian.

Mô hình sử dụng trở nên trực quan rõ ràng và cung cấp các giám khảo với siêu dữ liệu hữu ích mà không có sẵn trong người xem văn bản khác hoặc biên tập viên.



2.4.4. Log Parser

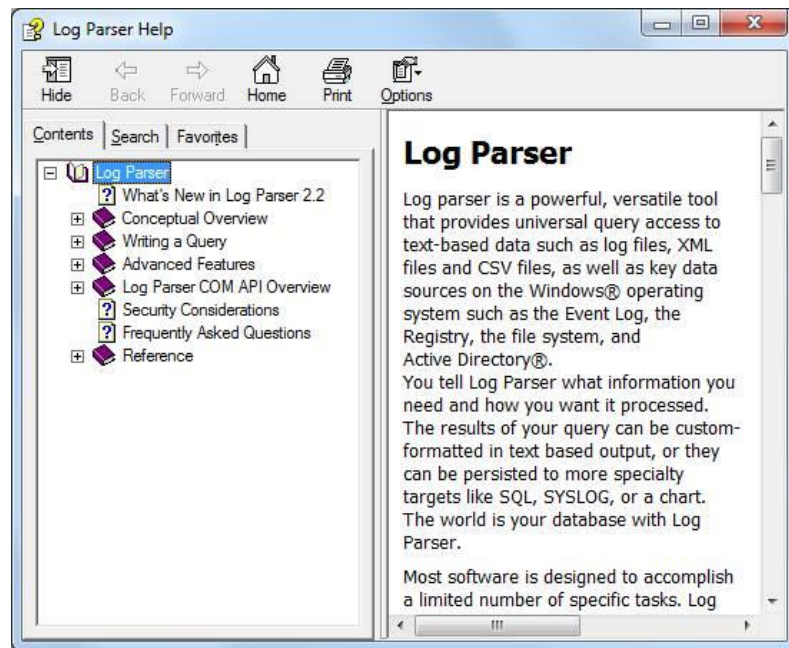
2.4.4.1. Giới thiệu

Log Parser 2.2 là một tiện ích dòng lệnh (logparser.exe) được hỗ trợ của Microsoft hoàn toàn miễn phí. Log Parser được biết đến như một công cụ khá mạnh, cho phép bạn truy vấn dữ liệu văn bản chẳng hạn như các file bản ghi, các file XML, các file CSV cũng như nguồn dữ liệu chính trên hệ điều hành Windows, Event Log, Registry, IIS, hệ thống file hoặc thậm chí cả Active Directory.

Ngoài khả năng cung cấp các thông tin phân tích cú pháp, Log Parser còn cho kết quả các truy vấn dưới định dạng tùy chỉnh ở đầu ra (Output), chẳng hạn như lưới dữ liệu datagrid, hoặc có thể chuyển đổi thành các biểu đồ trực quan.

2.4.4.2. Cách sử dụng

Đầu tiên, chúng ta hãy tham khảo file trợ giúp hướng dẫn Log Parser 2.2\LogParser.chm mà tiện ích này cung cấp rất chi tiết và cụ thể. Cũng như một vài ví dụ cụ thể tại Log Parser 2.2\Samples để ví dụ thực tiễn.



Bên cạnh đó, chúng ta cũng cần phải biết hai tham số cơ bản được sử dụng bởi công cụ là Input Format và Output Format. Nếu bạn có bất cứ câu hỏi nào liên quan đến việc sử dụng công cụ, hãy sử dụng LogParser -h, khi đó sẽ có một bảng tóm tắt tất cả các tùy chọn được hiển thị.

```
Administrator: Log Parser 2.2
Microsoft (R) Log Parser Version 2.2.10
Copyright (C) 2004 Microsoft Corporation. All rights reserved.

Usage:  LogParser [-i:<input_format>] [-o:<output_format>] [<SQL query>] !
        file:<query_filename>[?param1=value1+...]
        [<input_format_options>] [<output_format_options>]
        [-q[:ON|OFF]] [-e:<max_errors>] [-iw[:ON|OFF]]
        [-stats[:ON|OFF]] [-saveDefaults] [-queryInfo]

        LogParser -e -i:<input_format> -o:<output_format> <from_entity>
        <into_entity> [<where_clause>] [<input_format_options>]
        [<output_format_options>] [-multiSite[:ON|OFF]]
        [-q[:ON|OFF]] [-e:<max_errors>] [-iw[:ON|OFF]]
        [-stats[:ON|OFF]] [-queryInfo]

-i:<input_format>      : one of IISW3C, NCSA, IIS, IISODBC, BIN, IISMSID,
                        HTTPERR, URLSCAN, CSU, TSU, W3C, XML, EVT, ETW,
                        NETMON, REG, ADS, TEXTLINE, TEXTWORD, FS, COM (if
                        omitted, will guess from the FROM clause)
-o:<output_format>    : one of CSU, TSU, XML, DATAGRID, CHART, SYSLOG,
                        NEUROVIEW, NAT, W3C, IIS, SQL, TPL, NULL (if omitted,
                        will guess from the INTO clause)
-q[:ON|OFF]          : quiet mode; default is OFF
-e:<max_errors>       : max # of parse errors before aborting; default is -1
                        (ignore all)
-iw[:ON|OFF]         : ignore warnings; default is OFF
-stats[:ON|OFF]      : display statistics after executing query; default is
                        ON
-c                   : use built-in conversion query
-multiSite[:ON|OFF]  : send BIN conversion output to multiple files
                        depending on the SiteID value; default is OFF
-saveDefaults        : save specified options as default values
-restoreDefaults     : restore factory defaults
-queryInfo           : display query processing information (does not
                        execute the query)

Examples:
LogParser "SELECT date, REVERSE(DNS(c-ip)) AS Client, COUNT(*) FROM file.log
WHERE sc-status<>200 GROUP BY date, Client" -e:10
LogParser file:myQuery.sql?myInput=C:\temp\ex*.log*myOutput=results.csv
LogParser -c -i:BIN -o:W3C file1.log file2.log "ComputerName IS NOT NULL"

Help:
-h GRAMMAR           : SQL Language Grammar
-h FUNCTIONS [ <function> ] : Functions Syntax
-h EXAMPLES          : Example queries and commands
-h -i:<input_format>   : Help on <input_format>
-h -o:<output_format>  : Help on <output_format>
-h -c                : Conversion help

C:\Program Files\Log Parser 2.2>
```

Hai chuyển đổi lệnh cơ bản mà bạn phải lưu tâm:

-i:<input_format>—Tùy chỉnh định dạng đầu vào, trong trường hợp này là IISW3C

-o:<output_format>—Định dạng dữ liệu sẽ được hiển thị, trong trường hợp này, chúng ta sẽ xuất file CSV

Đề xuất từ các **log file** theo định dạng chuẩn **log W3C** (bạn có thể download các file log tổng hợp này từ công cụ quản trị hosting VD: Plesk Panel...) thành file CSV để tiện cho việc theo dõi thông kê hoạt động của website, chúng ta dùng lệnh sau:

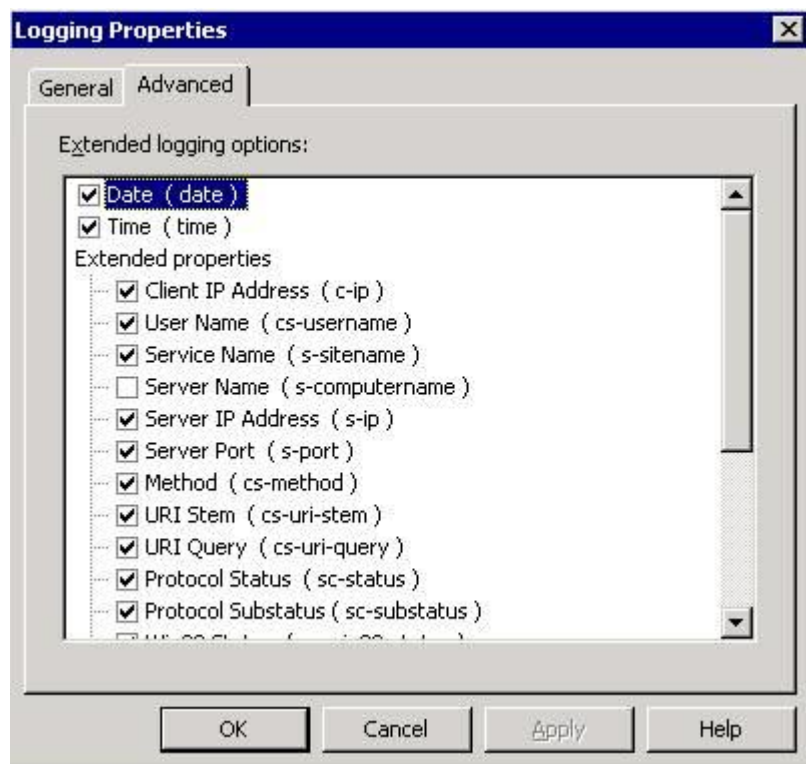
```
C:\Program Files\Log Parser2.2>logparser-i:IISW3C-o:CSV"SELECT *  
FROM *.log">*.csv
```

2.4.4.3. Ví dụ

Ở đây chúng tôi sử dụng file *test.log* đặt tại thư mục *E:\DATA* và xuất thành file *test.csv*.

```
C:\Program Files\Log Parser2.2>logparser-i:IISW3C-o:CSV"SELECT *  
FROM E:\DATA\test.log">E:\DATA\test.csv
```

Các bạn có thể tham chiếu cho các tên cột mà bạn đã sử dụng với tiện ích Log Parser trong file excel bằng các field trong tab *Advance Logging Properties* của IIS.



Từ đây để thống kê lưu lượng thật về hoạt động của website , đơn giản bạn chỉ việc cộng (SUM) giá trị của hai cột: *sc-bytes* (Bytes Sent) và *cs-bytes* (Bytes Recevied) lại với nhau.

PHẦN 3. PHÂN TÍCH LINUX LOG FILE

3.1. SYSLOG

3.1.1. Giới thiệu về Syslog

Các hệ thống Unix có hệ thống ghi log rất mạnh và linh động, mà cho bạn khả năng để ghi lại hầu hết mọi thứ bạn có thể tưởng tượng và sau đó thao tác sự ghi log này để truy xét thông tin bạn yêu cầu.

Rất nhiều phiên bản của Unix cung cấp một phương tiện dễ dàng ghi log với mục đích chung gọi là syslog. Mỗi chương trình cần thông tin ghi log được gửi tới syslog.

Syslog trong Unix là một host có thể định hình, là phương tiện ghi log hệ thống đồng dạng. Hệ thống sử dụng một tiến trình ghi log hệ thống trung tâm mà chạy chương trình/etc/syslogd hoặc /etc/syslog.

Hoạt động của hệ thống ghi log là không phức tạp. Các chương trình gửi cửa vào ghi log tới syslogd, mà tham vấn từ file định cấu hình /etc/syslogd.conf hoặc /etc/syslog và khi một kết nối được tìm thấy, nó ghi thông tin ghi log tới file ghi log đã yêu cầu.

Bảng dưới liệt kê 4 mục syslog cơ bản mà bạn nên hiểu:

Mục	Miêu tả
Facility (phương tiện)	Dấu hiệu nhận diện được sử dụng để miêu tả ứng dụng hoặc tiến trình mà đệ trình tới thông báo log. Các ví dụ là mail, kernel, và ftp.
Priority (quyền ưu tiên)	Một chỉ dẫn quan trọng của thông báo. Các mức được xác định trong syslog như một guideline, từ việc chỉnh lỗi thông tin tới các sự kiện quan trọng.
Selector (bộ chọn)	Một sự kết nối của một hoặc nhiều phương tiện và mức độ. Khi một sự kiện mới đến kết nối với một bộ chọn, một hành động được thực hiện.
Action (hành động)	Điều gì xảy ra khi một thông tin mới đến kết nối với một bộ chọn. Các hành động có thể ghi thông tin tới file ghi log, phản xạ thông tin tới một bàn điều khiển hoặc thiết bị khác, ghi thông báo tới hệ thống ghi log của người sử dụng hoặc gửi thông báo cùng với máy chủ syslog khác.

3.1.1.1. Facility syslog

Ví dụ: Bạn là người quản lý một tòa nhà. Một ngày bạn nhận được rất nhiều các video ghi lại từ các camera an ninh. Một câu hỏi được đưa ra là: có quá nhiều video đến, và làm thế nào để “quy hoạch” lại chúng ?

“Facility” sẽ giúp bạn làm việc này, trước khi các video được chuyển đến cho bạn nó đã được “dán nhãn” trước (Floor1_video1; Floor1_video2; Floor3_video1...) và bạn chỉ việc đưa chúng vào các Thư mục tương ứng để tiện việc sử dụng.

Dưới đây là các phương tiện có sẵn cho bộ chọn:

Facility	Miêu tả
Auth	Các hoạt động liên quan đến yêu cầu tên và mật khẩu (getty, su, login)
Authpriv	Tương tự như auth nhưng ghi log tới một file mà chỉ có thể được đọc bởi những người dùng được chọn.
Console	Sử dụng để bắt các thông báo mà thường trực tiếp gửi tới bàn điều khiển hệ thống.
Cron	Các thông báo từ người lập hệ thống cron.
Daemon	Hệ thống daemon nhận tất cả.
ftp	Các thông báo liên quan đến hệ thống ftp daemon.
Kern	Các thông báo kernel.
local0.local7	Các phương tiện nội bộ được xác định cho mỗi site.
Lpr	Các thông báo từ dòng hệ thống in.
Mail	Các thông báo liên quan tới hệ thống mail.
Mark	Các sự kiện giả được sử dụng để tạo timestamp trong các file hệ thống.
News	Các thông báo liên quan tới mạng lưới giao thức tin tức (network news protocol)
Ntp	Các thông báo liên quan đến giao thức thời gian mạng.
User	Các tiến trình người dùng thông thường.
Uucp	Hệ thống phụ UUCP.

3.1.1.2. *Priority syslog*

Các quyền ưu tiên syslog được tổng hợp ở dưới bảng sau:

Priority	Miêu tả
Emerg	Tình trạng khẩn cấp, như một sự ngưng hoạt động hệ thống sắp xảy ra, thường được thông báo tới tất cả người dùng.
Alert	Tình trạng mà nên được chỉnh lại cho đúng ngay lập tức, như một dữ liệu hệ thống bị hư hỏng.
Crit	Tình trạng nghiêm trọng, như lỗi phần cứng.
Err	Các lỗi thông thường.
Warning	Cảnh báo.
Notice	Tình trạng mà không là lỗi, nhưng có lẽ nên được thực hiện theo một cách đặc biệt.
Info	Thông báo mang tính thông tin.
Debug	Các thông báo mà được sử dụng khi chỉnh lỗi các chương trình.
None	Các mức giả tạo được sử dụng để xác định không log các thông báo.

Sự kết nối của các phương tiện và các mức cho bạn khả năng để thấy rõ về những gì được ghi log và nơi mà các thông tin bắt nguồn.

Khi mỗi chương trình gửi các thông báo của nó một cách nghiêm túc tới hệ thống ghi log, trình ghi log tạo các quyết định về những gì để theo dõi nó và những gì để loại bỏ nó ở các mức được xác định trong bộ chọn.

Khi bạn xác định một mức, hệ thống sẽ theo dõi mọi thứ tại mức đó và cao hơn.

3.1.1.3. *Action syslog*

Trường hành động xác định một trong 5 hành động sau:

- Thông tin ghi log tới một file hoặc một thiết bị. Ví dụ, /var/log/lpr.log hoặc /dev/console.
- Gửi một thông báo tới một người sử dụng. Bạn có thể xác định nhiều tên sử dụng bằng việc ngăn cách chúng bởi dấu phẩy (ví dụ root, amrood).

- Gửi một thông báo tới tất cả người dùng. Trong trường hợp này, trường hành động bao gồm một dấu *.
- Gửi một thông báo thông qua pipe tới một chương trình. Trong trường hợp này, chương trình được xác định sau ký hiệu pipe (|).
- Gửi thông báo tới syslog trên một host khác. Trong trường hợp này, trường hành động bao gồm một tên host, được đặt trước bởi một dấu hiệu (ví dụ: @tutorialspoint.com)

3.1.3. Cấu hình syslog

Tập này điều khiển nơi các thông báo được log. Một tệp **syslog.conf** đặc trưng có thể trông giống như thế này:

```
*.err;kern.debug;auth.notice /dev/console
daemon,auth.notice          /var/log/messages
lpr.info                     /var/log/lpr.log
mail.*/var/log/mail.log
ftp.*/var/log/ftp.log
auth.*@prep.ai.mit.edu
auth.*                       root,amrood
netinfo.err                  /var/log/netinfo.log
install.*/var/log/install.log
*.emerg                      *
*.alert                      |program_name
mark.*/dev/console
```

Mỗi dòng của file chứa hai phần:

- Một bộ chọn thông báo mà xác định loại thông báo để log. Ví dụ, tất cả các thông báo lỗi hoặc tất cả các thông báo chỉnh lỗi từ kernel.
- Một trường hành động mà nói những gì nên được làm với thông báo đó. Ví dụ, đặt nó trong một file hoặc gửi thông báo tới terminal của một người dùng.

Dưới đây là các điểm đáng chú ý cho sự định cấu hình trên:

- Các bộ chọn thông báo gồm hai phần: một phương thức và một quyền ưu tiên. Ví dụ, *kern.debug* chọn tất cả các thông báo debug (có Priority) được tạo bởi kernel (có Facility).

- Bộ chọn thông báo kern.debug chọn tất cả các quyền ưu tiên mà ưu tiên hơn chỉnh lỗi.
- Một dấu sao * trong vị trí hoặc của phương thức hoặc quyền ưu tiên ám chỉ rằng “tất cả”. Ví dụ, *.debug nghĩa là tất cả các thông báo chỉnh lỗi, trong khi kern.* nghĩa là tất cả các thông báo được tạo ra bởi kernel.
- Bạn cũng có thể sử dụng các dấu phẩy để xác định nhiều phương thức. Hai hoặc nhiều bộ chọn có thể được nhóm lại với nhau bằng cách sử dụng một dấu chấm phẩy (;)

3.1.4. Sử dụng lệnh ghi log

Unix cung cấp lệnh **logger**, mà là một lệnh thực sự hữu ích để giải quyết hệ thống ghi log. Lệnh logger gửi các thông báo ghi log tới **syslogd daemon**, và do đó kích thích hệ thống ghi log.

Điều này có nghĩa là chúng ta có thể kiểm tra từ dòng lệnh tại bất cứ thời gian nào. Lệnh **logger** cung cấp một phương thức để thêm cổng vào một dòng tới hệ thống ghi log file từ dòng lệnh.

Định dạng của lệnh là:

```
logger [-i][-f file][-p priority][-t tag][message]...
```

Dưới đây là chi tiết về các tham số.

Chức năng	Miêu tả
-f filename	Sử dụng nội dung của tên file như thông báo để log.
-i	log ID tiến trình của tiến trình logger với mỗi dòng.
-p priority	Nhập thông báo với quyền ưu tiên được xác định (lỗi vào bộ chọn được xác định); quyền ưu tiên thông báo có thể được xác định ở dạng số hoặc như là cặp phương thức.quyền ưu tiên. Quyền ưu tiên mặc định là user.notice.
-t tag	Đánh dấu mỗi dòng được thêm tới hệ thống log với thẻ đã xác định.
Message	Các tham số chuỗi mà nội dung được kết nối cùng nhau theo thứ tự xác định, riêng rẽ bởi khoảng trống.

Bạn có thể sử dụng trang trợ giúp (Manpage Help) để kiểm tra cú pháp của các lệnh này.

3.1.5. Sự luân phiên log

Các log file có thiên hướng tăng lên rất nhanh và chiếm một khoảng lớn của không gian đĩa. Để cho phép khả năng luân phiên log, hầu hết phiên bản sử dụng các công cụ như `newsyslog` hoặc `logrotate`.

Những công cụ này nên được gọi trên một khoảng không gian thường xuyên bằng cách sử dụng `cron daemon`. Bạn truy cập vào trang trợ giúp (Manpage Help) để biết thêm chi tiết về `newsyslog` hoặc `logrotate`.

`Logrotate` là một công cụ hữu hiệu hỗ trợ cho việc quản trị các file log. Với `Logrotate` các file log có thể được định kỳ sử dụng lại theo ngày tuần tháng hay theo kích thước file log, nén, xóa bỏ file log...

Bạn có thể tham khảo thêm tại link: <http://linux.die.net/man/8/logrotate>

Các tham số thường dùng:

- `compress`: nén những file log đã sử dụng
- `nocompress`: ngược lại với tùy chọn `compress`
- `create mode owner group`: khi sử dụng một file log mới, file log mới được tạo sẽ có các thuộc tính (mode, owner group).
- `nocreate`: không tạo file log mới.
- `mail address`: khi hết chu kỳ sử dụng file log sẽ được gửi tới địa chỉ (address).
- `nomail`: ngược lại với tùy chọn trên.
- `daily`: chu kỳ sử dụng file log theo ngày
- `weekly`: chu kỳ sử dụng file log theo tuần.
- `monthly`: chu kỳ sử dụng file log theo tháng.
- `rotate count`: xác định số lần luân phiên sử dụng file log.
- `size size`: chu kỳ sử dụng file log được xác định theo kích thước.
- `include /etc/logrotate.d`: đọc thêm các thông tin cấu hình tại các file trong thư mục `/etc/logrotate`. Các tham số khai báo ở các file này có độ ưu tiên cao hơn các tham số khai báo trong file `/etc/logrotate.conf`.

3.1.6. Các vị trí log quan trọng

Tất cả ứng dụng hệ thống tạo các tệp log trong **/var/log** và các thư mục phụ của nó. Dưới đây là một số ứng dụng quan trọng và các thư mục log tương ứng của chúng.

Ứng dụng	Thư mục
Httpd	/var/log/httpd
Samba	/var/log/samba
Cron	/var/log/
Mail	/var/log/
Mysql	/var/log/

3.1.7. Thao tác với Log Files

Bạn có thể học được nhiều về cách hoạt động của hệ thống bằng cách xem lại các log files được tạo. Một lúc nào đó sẽ cần thiết để gỡ rối vấn đề gặp phải từ những thông tin đã được logged đó. Phần lớn các log files có dạng plain text, nó rất dễ dàng để xem lại với một vài lệnh như: tail, less, và grep.

Câu lệnh	Cú pháp	Ý nghĩa	Ghi chú thêm
More	more [file]	Dùng xem toàn bộ nội dung của thư mục	Đối với câu lệnh này nội dung được xem theo từng trang. Bạn dùng dấu "cách" để chuyển trang
Tail	tail [file]	In ra 10 dòng cuối cùng nội dung của file	thêm tùy chọn -n [số dòng] sẽ in ra số dòng theo yêu cầu
Head	head [file]	In ra 10 dòng đầu tiên của nội dung file	
tail -f	tail -f [file]	Dùng để xem ngay lập tức khi có log đến	Đây là câu lệnh dùng phổ biến nhất nó giúp ta có thể xem ngay lập tức log mới đến, và nó sẽ in ra 10 dòng cuối cùng trong nội dung file đó

Một dòng thông báo log được tạo với các thông tin sau, ngăn cách bởi một khoảng trắng (space):

- Date/time
- Origin hostname
- Message sender
- Message text

Một đoạn messages điển hình, trông giống như sau:

```
openvpn.net/howto.html#mitm for more info.
Nov 21 14:42:13 localhost openvpn[1964]: Re-using SSL/TLS context
Nov 21 14:42:13 localhost openvpn[1964]: LZO compression initialized
Nov 21 14:42:13 localhost openvpn[1964]: Control Channel MTU parms [
L:1542 D:138 EF:38 EB:0 ET:0 EL:0 ]
Nov 21 14:42:13 localhost openvpn[1964]: Data Channel MTU parms [
L:1542 D:1450 EF:42 EB:135 ET:0 EL:0 AF:3/1 ]
Nov 21 14:42:13 localhost openvpn[1964]: Local Options hash (VER=V4):
'41690919'
Nov 21 14:42:13 localhost openvpn[1964]: Expected Remote Options hash
(VER=V4): '530fdded'
Nov 21 14:42:13 localhost openvpn[1964]: Socket Buffers: R=[110592-
>131072] S=[110592->131072]
Nov 21 14:42:13 localhost openvpn[1964]: UDPv4 link local: [undef]
Nov 21 14:42:13 localhost openvpn[1964]: UDPv4 link remote:
192.168.1.3:1194
Nov 21 14:42:13 localhost openvpn[1964]: read UDPv4 [ECONNREFUSED]:
Connection refused (code=111)
```

Trong trường hợp này, hostname là localhost và messages đến từ dịch vụ openvpn. Bất cứ lúc nào bạn cũng có thể xem lại nội dung log files của bạn bằng cách dùng less

```
$ less /var/log/messages
```

hoặc tail

```
$ tail -f /var/log/messages
```

Để tìm kiếm các thông báo xác định về mouse, bạn có thể dùng grep

```
# grep '[Mm]ouse' /var/log/messages
Dec 8 00:15:28 smp kernel: Detected PS/2 Mouse Port.
Dec 8 10:55:02 smp gpm: Shutting down gpm mouse services:
```

Thông thường, nếu bạn sử dụng grep để tìm kiếm một item đặc biệt trong /var/log/messages, bạn sẽ cần phải tìm kiếm tất cả các rotated files với một wildcard (tạm dịch: ký tự đại diện). Chẳng hạn, để tìm kiếm tất cả các messages từ sendmail bạn cần:

```
# grep 'sendmail:' /var/log/messages*
```

Khi bạn xác định vấn đề từ log files, hãy nhìn vào hostname và sender đầu tiên, sau đó là messages text. Trong nhiều trường hợp bạn có thể xác định được cái gì là sai từ thông báo đó. Thi thoảng, những thông báo lỗi đó chỉ là đầu mối, và việc xem lại toàn bộ các logs của bạn là cần thiết. Trong trường hợp này, sẽ hữu ích nếu bạn tạm thời thay đổi level trong /etc/syslog.conf thành debug để log lại nhiều thông tin hơn giúp bạn giải quyết vấn đề.

3.1.8. Phân tích log file

Log một hệ thống Unix được quản lý bởi 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 một 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 một máy trạm của syslogd. Trong mẫu chuẩn riêng, klogd chuyển các thông báo kernel ra một file, còn trong mẫu kết hợp, nó đẩy thông báo tới một 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 một 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 đây, nó cho phép bạn có 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 trên) 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 tắc nghẽn mạng và những vấn đề khác (bởi vì nó nhận 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 một nền tảng syslog. Một vài vấn đề về syslog sẽ xuất hiện một cách hiển nhiên trong khi làm việc.

Đây là một danh sách ngắn:

- 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.

- Việc lọc các thông điệp 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.

- Các chuyển dịch trên mạng dựa trên UDP là không thể tin tưởng được, nếu những cái nhận được kết thúc của một 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.

- 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 là một vấn đề trầm trọng bởi syslog được sử dụng trong các mạng nội bộ có thể tin tưởng được hoặc thậm chí là một mạng LAN được chỉ định quản lý.

- 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 một 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 hai này.

- 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 một lệnh grep hoàn chỉnh trên một 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, một cơ sở dữ liệu quan hệ là thực sự cần thiết.

- Các log được lưu trữ là điểm yếu để sửa chữa hoặc xóa đi, đặc biệt là khi lưu 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 hai sự thay thế khá nổi tiếng đó là thay thế syslog-ng bởi Balabit (<http://www.balabit.hu/en/downloads/syslogng>); 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 buffering, 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 chuyên 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 tìm ra những thay đổi trong các log đã được lưu giữ.

Trên các máy trạm mà tạo ra hoặc chuyển tiếp các file log, 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 kết 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 hình 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:

```
/var/log/critical file is ok
```

Nếu logfile đã bị thay đổi, bạn sẽ thấy:

```
/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 một 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 một bảng 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 "rant 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 thị 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.

3.2. APACHE LOG FILE

Trong việc quản một máy chủ web hiệu quả, log file là rất cần thiết để thu được thông tin phản hồi về hoạt động và hiệu suất của máy chủ cũng như bất kỳ vấn đề có thể được xảy ra. Apache HTTP Server cung cấp khả năng khai thác log rất toàn diện và linh hoạt.

3.2.1. Security Warning

Người mà có thể ghi vào thư mục, nơi Apache đang ghi log file thì gần như chắc chắn có thể được truy cập vào uid của máy chủ, thường là tài khoản root. Không nên cung cấp cho người dùng quyền ghi vào các thư mục được lưu trữ log file, vì như vậy sẽ gây hậu quả nghiêm trọng.

Ngoài ra, file log có thể chứa các thông tin được cung cấp trực tiếp từ client. Do đó, nó có thể bị chen các ký tự điều khiển trong các file log, vì vậy cần phải thận trọng trong việc nhận diện các log file nguyên bản.

3.2.2. Error Log

Các error log máy chủ, tên và vị trí được thiết lập bởi ErrorLog, là log file rất quan trọng. Đây là nơi mà Apache httpd sẽ gửi thông tin chẩn đoán và ghi lại bất kỳ lỗi nào mà nó gặp trong các yêu cầu xử lý. Đây là nơi đầu tiên để xem xét khi một vấn đề xảy ra với khởi chạy máy chủ hoặc với những hoạt động của máy chủ, vì nó sẽ thường chứa các thông tin chi tiết về những gì đã xảy ra và làm thế nào để sửa chữa nó.

Error log thường được ghi vào một tập tin (thường là error_log trên các hệ thống Unix và error.log trên Windows). Trên các hệ thống unix nó cũng có thể có các máy chủ gửi lỗi tới syslog.

Định dạng của các error log là tương đối dạng tự do. Nhưng có một số thông tin được chứa trong hầu hết các mục error log. Ví dụ, đây là một thông điệp điển hình.


```
[Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client denied  
by server configuration: /export/home/live/ap/htdocs/test
```

Mục đầu tiên trong mục ghi là ngày và thời gian của tin nhắn. Các mục thứ hai liệt kê mức độ nghiêm trọng của lỗi được báo cáo. Các LogLevel chỉ được sử dụng để kiểm soát các loại lỗi được gửi đến các error log bằng cách hạn chế mức độ nghiêm trọng. Các mục thứ ba cung cấp cho các địa chỉ IP của client đã tạo ra lỗi. Ngoài ra là thông điệp chính nó, mà trong trường hợp này chỉ ra rằng các máy chủ đã được cấu hình để từ chối truy cập từ client. Các máy chủ báo cáo các đường dẫn tập tin hệ thống (như trái ngược với con đường web) của tài liệu được yêu cầu.

Một loạt rất nhiều thông điệp khác nhau có thể xuất hiện trong error log. Hầu hết nhìn tương tự như ví dụ trên. Các error log cũng sẽ chứa kết quả xử lý từ các kịch bản CGI. Bất kỳ thông tin bằng văn bản cho stderr bởi một CGI script sẽ được sao chép trực tiếp vào error log.

Nó không phải là có thể tùy chỉnh các error log bằng cách thêm hoặc loại bỏ thông tin. Tuy nhiên, các mục error log đối phó với các yêu cầu đặc biệt có các mục tương ứng trong access log.

Trong thời gian thử nghiệm, nó thường hữu ích để liên tục theo dõi các error log cho bất kỳ vấn đề. Trên các hệ thống Unix, bạn có thể thực hiện điều này bằng cách sử dụng:

```
tail -f error_log
```

3.2.3. Access log

Các access log máy chủ ghi lại tất cả các yêu cầu xử lý bởi máy chủ. Các vị trí và nội dung của access log được kiểm soát bởi CustomLog. Các LogFormat chỉ có thể được sử dụng để đơn giản hóa việc lựa chọn các nội dung của các bản ghi. Phần này mô tả làm thế nào để cấu hình các máy chủ để ghi lại thông tin trong access log.

Tất nhiên, lưu trữ các thông tin trong access log chỉ là sự bắt đầu của quản lý log file. Bước tiếp theo là phân tích các thông tin này để đưa ra các thống kê hữu ích.

Các phiên bản khác nhau của Apache httpd đã sử dụng mô-đun và các chỉ thị khác để kiểm soát truy cập việc ghi log, kể cả `mod_log_referer`, `mod_log_agent`, và `TransferLog`

Định dạng của access log là cấu hình cao. Các định dạng được chỉ định bằng một chuỗi định dạng trông giống như một `printf` C-style định dạng chuỗi. Một số ví dụ được trình bày trong các phần tiếp theo.

3.2.3.1. Common Log Format

Một cấu hình điển hình cho các access log có thể như sau.

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common
```

Điều này xác định biệt danh chung và liên kết nó với một chuỗi định dạng log cụ thể. Chuỗi định dạng bao gồm các ký tự phân trăm, mỗi trong số đó với máy chủ đăng nhập một đoạn cụ thể của thông tin. Ký tự chữ cũng có thể được đặt trong chuỗi định dạng và sẽ được sao chép trực tiếp vào đầu ra đăng nhập. Các nhân vật báo (") phải được thoát ra bằng cách đặt một back-slash trước khi nó để ngăn chặn nó được hiểu như là sự kết thúc của chuỗi định dạng. Các chuỗi định dạng cũng có thể chứa các ký tự kiểm soát đặc biệt "\n" cho thêm hàng mới và "\t" cho tab.

Các CustomLog thành lập một file log mới sử dụng biệt danh được xác định. Các tên tập tin cho access log là tương đối so với ServerRoot trừ khi nó bắt đầu với một dấu gạch chéo.

Cấu hình trên sẽ viết entry bản ghi trong một định dạng được gọi là Common Log Format (CLF). Định dạng chuẩn này có thể được sản xuất bởi nhiều máy chủ web khác nhau và đọc bởi nhiều chương trình phân tích log. Các mục file bản ghi sản xuất tại CLF sẽ giống như thế này:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif
HTTP/1.0" 200 2326
```

Mỗi một phần của dòng log này được mô tả dưới đây.

```
127.0.0.1 ( %h )
```

Đây là địa chỉ IP của máy khách (máy chủ từ xa) mà thực hiện các yêu cầu đến máy chủ.

```
- ( %l )
```

Những "gạch nối" ở đầu ra cho thấy rằng các mảnh yêu cầu các thông tin không có sẵn.

```
frank ( %u )
```

Đây là userid của người yêu cầu các tài liệu được xác định bằng cách xác thực HTTP.

```
[10/Oct/2000:13:55:36 -0700] ( %t )
```

Thời gian mà máy chủ đã hoàn thành xử lý yêu cầu. Định dạng là:

```
[day/month/year:hour:minute:second zone]
day = 2*digit
month = 3*letter
year = 4*digit
hour = 2*digit
minute = 2*digit
second = 2*digit
zone = ('+' | '-') 4*digit
```

Nó có thể có thời gian hiển thị trong định dạng khác bằng cách xác định `%{format}t` trong chuỗi định dạng nhật ký.

```
"GET /apache_pb.gif HTTP/1.0" ( \"%r\" )
```

Dòng yêu cầu từ khách hàng được đưa ra trong dấu ngoặc kép. Dòng yêu cầu có chứa rất nhiều thông tin hữu ích. Đầu tiên, các phương pháp được sử dụng bởi các khách hàng là GET. Thứ hai, khách hàng yêu cầu tài nguyên /apache_pb.gif, và thứ ba, khách hàng sử dụng các giao thức HTTP/1.0. Nó cũng có thể ghi log một hoặc nhiều bộ phận của dòng yêu cầu độc lập. Ví dụ, chuỗi định dạng "%m %U%q %H" sẽ ghi log phương pháp, đường dẫn, truy vấn-string, và giao thức, kết quả chính xác đầu ra tương tự như "%r".

```
200 ( %>s )
```

Đây là mã trạng thái mà máy chủ gửi lại cho khách hàng (HTTP status code). Thông tin này là rất có giá trị, bởi vì nó cho thấy liệu các yêu cầu dẫn đến một phản ứng thành công, chuyển hướng, lỗi gây ra bởi client, một lỗi trong các máy chủ.

Các mục cuối cùng chỉ ra kích thước của đối tượng trả lại cho khách hàng, không bao gồm các tiêu đề ứng. Nếu không có nội dung được trả lại cho khách hàng, giá trị này sẽ được " - ". Để đăng nhập " 0 " cho không có nội dung, sử dụng %B để thay thế.

3.2.3.2. Combined Log Format

Một chuỗi định dạng thường được sử dụng được gọi là Combined Log Format. Nó có thể được sử dụng như sau.

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
CustomLog log/acces_log combined
```

Định dạng này là chính xác giống như các Format Common Log, với việc bổ sung thêm hai trường khác. Mỗi một trường bổ sung sử dụng ký tự phần trăm % {header}i, header có thể là bất kỳ yêu cầu HTTP header. Các access log dưới dạng này sẽ giống như:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif
HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en]
(Win98; I ;Nav)"
```

Các trường bổ sung là:

```
"http://www.example.com/start.html" ("%{Referer}i")
"Mozilla/4.08 [en] (Win98; I ;Nav)" ("%{User-agent}i")
```

3.2.3.3. Multiple Access Logs

Multiple Access Logs có thể được tạo ra chỉ đơn giản bằng cách xác định nhiều CustomLog trong file cấu hình. Ví dụ, các chỉ dẫn sau sẽ tạo ra ba access log. Việc đầu tiên chứa thông tin CLF cơ bản, trong khi thứ hai và thứ ba chứa referer và thông tin trình duyệt. Hai dòng CustomLog cuối cùng cho thấy làm thế nào để bắt chước những tác động của các ReferLog và Agent Log

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common
CustomLog logs/referer_log "%{Referer}i -> %U"
CustomLog logs/agent_log "%{User-agent}i"
```

Ví dụ này cũng cho thấy rằng nó không phải là cần thiết để xác định một biệt danh với chỉ thị LogFormat. Thay vào đó, các định dạng đăng nhập có thể được chỉ định trực tiếp trong các chỉ thị CustomLog.

Conditional Logging

Nó là thuận tiện để loại trừ các mục nhất định từ các access log dựa trên đặc điểm của các yêu cầu của khách hàng. Điều này rất dễ thực hiện với sự giúp đỡ của các biến môi trường. Đầu tiên, một biến môi trường phải được thiết lập để chỉ ra rằng các yêu cầu đáp ứng các điều kiện nhất định. Điều này thường được thực hiện với SetEnvIf. Sau đó, các env= của CustomLog được sử dụng để loại trừ các yêu cầu trong đó các biến môi trường được thiết lập. Vài ví dụ:

```
# Mark requests from the loop-back interface
SetEnvIf Remote_Addr "127\.0\.0\.1" dontlog
# Mark requests for the robots.txt file
SetEnvIf Request_URI "^/robots\.txt$" dontlog
# Log what remains
CustomLog logs/access_log common env=!dontlog
```

3.2.4. Log Rotation

Ngay cả trên một máy chủ khá bận rộn, số lượng thông tin được lưu trữ trong các tập tin đăng nhập là rất lớn. Các tập tin access log thường có dung lượng 1 MB cho mỗi 10.000 yêu cầu. Do đó cần thiết phải định kỳ luân phiên các file bản ghi bằng cách di chuyển hoặc xóa các bản ghi hiện có. Điều này không thể được thực hiện trong khi máy chủ đang chạy, vì Apache sẽ tiếp tục viết vào file log cũ miễn là nó nắm giữ các tập tin mở. Thay vào đó, các máy chủ phải được khởi động lại sau khi các tập tin nhật ký được di chuyển hoặc xóa để nó sẽ mở file log mới.

Bằng cách sử dụng một khởi động lại, các máy chủ có thể được hướng dẫn để mở file log mới mà không mất bất kỳ kết nối hiện có hoặc cấp phát từ client. Tuy nhiên, để thực hiện điều này, máy chủ phải tiếp tục ghi vào các tập tin log cũ trong khi nó kết thúc phục vụ yêu cầu cũ. Do đó, cần thiết phải chờ đợi một thời gian sau khi khởi động lại trước khi làm bất cứ xử lý trên các tập

tin log. Một kịch bản điển hình mà chỉ đơn giản xoay các bản ghi và nén các bản ghi cũ để tiết kiệm không gian là:

```
mv access_log access_log.old
mv error_log error_log.old
apachectl graceful
sleep 600
gzip access_log.old error_log.old
```

Một cách khác để thực hiện luân phiên đăng nhập được sử dụng bản ghi piped logs như trình bày trong phần tiếp theo.

3.2.5. Piped Logs

Apache httpd có khả năng viết các tập tin lỗi và access log thông qua một pipe đến một quá trình, chứ không phải là trực tiếp vào một tập tin. Khả năng này làm tăng đáng kể sự linh hoạt của ghi log, mà không cần thêm mã vào các máy chủ chính. Để ghi log vào pipe, chỉ cần thay tên tập tin với các ký tự gạch "|", theo sau là tên của file thực thi mà nên chấp nhận các log trên đầu vào tiêu chuẩn của nó. Apache sẽ bắt đầu quá trình piped-log khi máy chủ bắt đầu, và sẽ khởi động lại nó nếu nó bị treo trong khi máy chủ đang chạy

Tiến trình Piped log được sinh ra bởi tiến trình cha là httpd Apache, và kế thừa userid của tiến trình đó. Điều này có nghĩa rằng các chương trình đăng nhập piped thường chạy như là người chủ. Do đó, nó là rất quan trọng để giữ cho các chương trình đơn giản và an toàn.

Một ứng dụng quan trọng các piped log là cho phép luân phiên log mà không cần phải khởi động lại máy chủ. Apache HTTP Server bao gồm một chương trình đơn giản được gọi là rotatelog cho mục đích này. Ví dụ, để luân phiên các bản ghi mỗi 24 giờ, bạn có thể sử dụng:

```
CustomLog "|/usr/local/apache/bin/rotatelog" /var/log/access_log
86400" common
```

Một chương trình luân phiên log tương tự, nhưng linh hoạt hơn gọi là cronolog có sẵn tại một trang bên ngoài.

Như với ghi log có điều kiện, các bản ghi piped là một công cụ rất mạnh mẽ, nhưng họ không nên được sử dụng khi một giải pháp đơn giản như off-line sau xử lý có sẵn.

3.3. CÔNG CỤ PHÂN TÍCH

3.3.1. Syslog-ng

Syslog-ng là một công cụ thu thập Log rất hiệu quả và linh hoạt là sự lựa chọn của rất nhiều nhà quản trị mạng trong việc xây dựng một hệ thống log tập trung. Syslog-ng được xây dựng dựa trên chuẩn syslog trên nền tảng Unix và các hệ điều hành tương tự. Gồm xây dựng với hai thành phần Syslog-ng client và Syslog-ng Server. Các Client thực hiện việc thu thập log quan trọng gửi tới máy chủ tập trung và lưu trữ.

Syslog-ng là một phần mềm mã nguồn mở được phát triển trên nền tảng của Syslogd. Hiện nay nó có hai phiên bản và được phát triển bởi Balabit IT Security Ltd.

Phiên bản miễn phí: Syslog-ng Open Source Edition (OSE).

Phiên bản trả phí độc quyền: Premium Edition (PE).

3.3.1.1. Tính năng

Thu thập dữ liệu: Syslog-client thực hiện việc tập trung log từ các host và gửi về Syslog server. Syslog-ng thực hiện việc thu thập log từ các server khác nhau dựa trên giao thức TCP, đảm bảo không bị mất mát thông tin trên đường truyền. Syslog-ng cung cấp một cơ chế truy xuất log an toàn dựa trên SSL/TLS.

Định dạng log: Theo mặc định Syslog-ng chỉ hỗ trợ chuẩn, Syslog trong Unix. Theo mặc định Windows không hỗ trợ Syslog. Tuy nhiên chúng ta có thể sử dụng một số biện pháp để chuyển các loại log về dạng Syslog. Syslog-ng cũng hoạt động rất tốt trên những môi trường (hệ điều hành, phần cứng) khác nhau: Linux, BSD, Sun Solaris, HP-UX, AIX và Unix khác.

Lưu trữ: Với Syslog-ng, ta có thể lưu trữ dữ liệu vào cơ sở dữ liệu cho phép tìm kiếm và truy vấn dễ dàng. Syslog-ng hỗ trợ các hệ CSDL: MSSQL, MYSQL, Oracle và PostgreSQL.

Lọc và phân loại: Syslog-ng cung cấp cơ chế lọc nhằm phân loại các Log message và cũng hạn chế lượng dữ liệu đổ về server log từ các client. Cơ chế lọc của Syslog-ng dựa trên các thông số khác nhau như source host, ứng dụng, sự ưu tiên trong Log message.

Cơ chế thu thập Log: Syslog-ng client được đặt trên các client sẽ thực hiện việc thu thập các loại Log trên client đó. Sau đó dữ liệu sẽ được đi qua bộ phận lọc của syslog-ng (gồm những luật đã được cấu hình trước). Sau đó mới được gửi đến các Server log hoặc chuyển đến một Relay server rồi mới chuyển tới Log Server.

3.3.1.2. Nhược điểm

Syslog-ng không phải là một phần mềm phân tích cho nên syslog-ng chỉ có thể lọc những log message phù hợp với một số tiêu chí định trước. Syslog-ng không thể làm tốt nhiệm vụ phân tích và cảnh báo các nguy cơ đến người quản trị.

3.3.1.3. Triển khai

Để triển khai một hệ thống syslog-ng ta cần có hai thành phần là một server được cài đặt syslog-ng server và các client được cài đặt trên các client để thu thập log. Một điểm đáng chú ý là Syslog không hỗ trợ windows.

3.3.2. ELK (Logstash, Elasticsearch, Kibana)

3.3.2.1. Giới thiệu

Với những hệ thống lớn việc quản lý log và phân loại log bằng việc xem file log của server để xác định thông tin của log, phân loại log là khá khó khăn. Cần thiết phải có một công cụ quản lý log một cách tốt hơn, sớm phát hiện những lỗi phát sinh của server hoặc kiểm tra các thông tin về log. Hiện nay cũng có khá nhiều công cụ để quản lý log khác nhau. Qua tìm hiểu thì bộ công cụ Logstash, Elasticsearch, Kibana có nhiều ưu điểm như phần mềm mã nguồn mở hoàn toàn miễn phí, cung cấp dịch vụ quản lý log rất tốt và dễ sử dụng. Dưới đây tôi sẽ giới thiệu về bộ công cụ này.



Logstash: Đây là một công cụ sử dụng để thu thập, xử lý log được viết bằng java. Nhiệm vụ chính của logstash là thu thập log sau đó chuyển vào Elasticsearch. Mỗi dòng log của logstash được lưu trữ dưới dạng json.

Elasticsearch: sử dụng cơ sở dữ liệu NoSQL dựa trên nền tảng của Apache Lucene engine. Dùng để lưu trữ dữ liệu và cung cấp interface cho phép truy vấn đến cơ sở dữ liệu.

Kibana: Đây là giao diện sử dụng dành cho người dùng trên môi trường web. Kibana sẽ sử dụng Elasticsearch để tìm kiếm các dữ liệu phù hợp với yêu cầu của người dùng.

3.3.2.2. Cài đặt ELK

- Cài đặt java:

Logstash được viết bằng java nên cần có một Java Virtual Machine để hoạt động. Vì vậy trước tiên bạn cần cài đặt Java, nên cài phiên bản mới nhất của java hiện nay Java8

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
```

- Cài đặt Logstash:

```
sudo wget http://download.elastic.co/logstash/logstash/packages/debian/logstash_1.5.3-1_all.deb
sudo dpkg -i logstash_1.5.3-1_all.deb
```

```
sudo update-rc.d logstash defaults 95 10
sudo /etc/init.d/logstash restart
```

- Cài đặt Elasticsearch:

```
sudo wget https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.7.1.deb
sudo dpkg -i elasticsearch-1.7.1.deb
sudo update-rc.d elasticsearch defaults 95 10
sudo /etc/init.d/elasticsearch restart
```

- Cài Kibana:

```
cd /opt
sudo wget https://download.elasticsearch.org/kibana/kibana/kibana-4.1.1-linux-x64.tar.gz
sudo tar xvfz kibana-4.1.1-linux-x64.tar.gz
sudo ln -s kibana-4.1.1-linux-x64 kibana
cd /etc/init.d
sudo wget https://raw.githubusercontent.com/akabdog/scripts/master/kibana4_init
sudo chmod 755 kibana4_init
sudo update-rc.d kibana4_init defaults 95 10
sudo /etc/init.d/kibana4_init restart
```

Như vậy bạn đã xong việc cài đặt bộ công cụ này.

Chú ý: Kibana mặc định hoạt động ở cổng 5601 <http://localhost:5601>

- Kiểm tra hoạt động của logstash, Elasticsearch, Kibana:

Bạn chạy lệnh sau để khởi động logstash. Theo config này logstash sẽ nhận input là stdin (dữ liệu nhập từ màn hình) và dữ liệu được xuất ra Elasticsearch ở địa chỉ localhost

```
sudo -u logstash /opt/logstash/bin/logstash -e 'input { stdin { } }
output { elasticsearch { host => localhost } }'
```

3.3.2.3. Sử dụng Logstash, Elasticsearch, Kibana

Để sử dụng Logstash, Elasticsearch, Kibana hiệu quả thì log của Rails cũng cần được lưu trữ dưới dạng Json. Một số gem hỗ trợ việc lưu trữ log dưới dạng Json cho Rails, trong đó gem logstash-logger khá tiện lợi, dễ dùng và được đánh giá cao. Việc cài đặt và sử dụng logstash-logger bạn có thể tham khảo ở <https://github.com/dwbutler/logstash-logger>

Dưới đây là ví dụ một số log được lưu trữ bởi logstash-logger

```
{ "test":1, "@timestamp":"2015-09-19T09:54:09.309+07:00", "@version":"1", "severity":"INFO", "host":"somehost" }
{ "test":2, "@timestamp":"2015-09-19T09:54:09.309+07:00", "@version":"1", "severity":"ERROR", "host":"somehost" }
{ "test":3, "@timestamp":"2015-09-19T09:54:09.309+07:00", "@version":"1", "severity":"DEBUG", "host":"somehost" }
```

Sau đó bạn chạy logstash và config cho logstash đọc thông tin log thì file log của server như sau, example.config:

```
input {
  file {
    type => "rails logs"
    path => "#path_your_log_file"
    codec => json {
      charset => "UTF-8"
    }
  }
}

output {
  stdout {
    codec => rubydebug
  }
  elasticsearch {
    embedded => true
  }
}
```

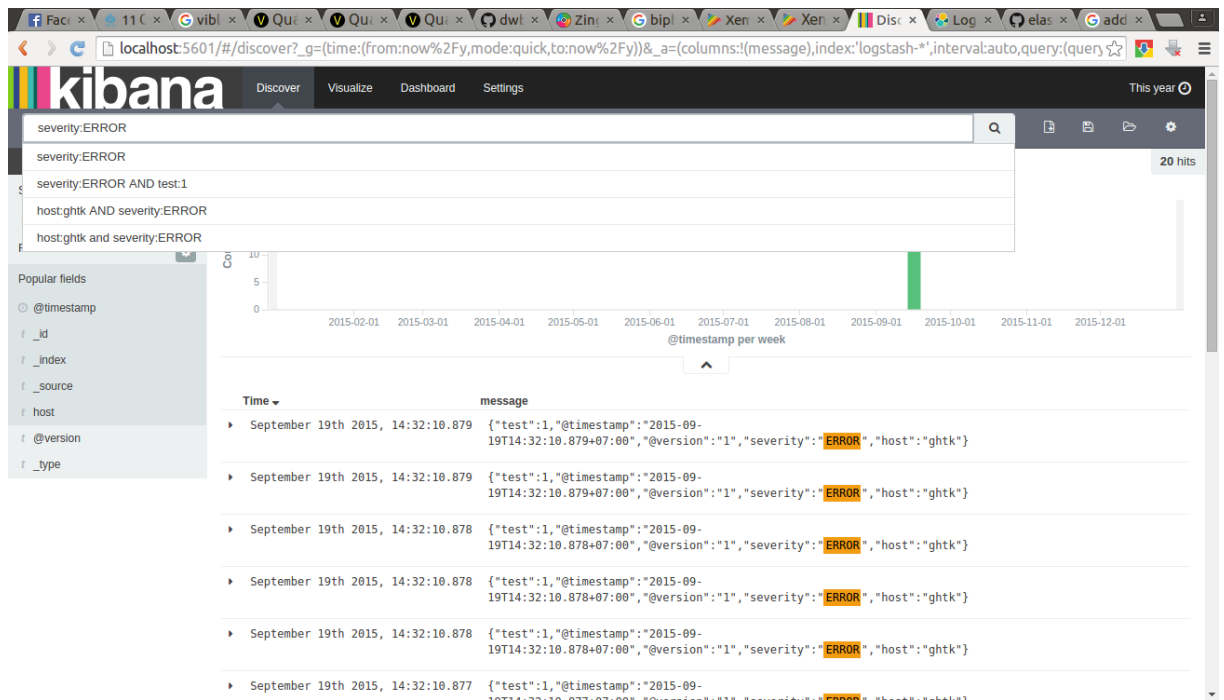
Sau đó bạn khởi động logstash và sử dụng file config này để cấu hình cho logstash

```
sudo -u logstash /opt/logstash/bin/logstash -f example.config
```

Khởi động lại elasticsearch

```
sudo /etc/init.d/elasticsearch restart
```

Sau đó bạn vào địa chỉ localhost:5601 và có thể thấy các log đã được hiển thị lên. Bạn có thể tìm kiếm các log, hoặc tạo biểu đồ cho log. Như vậy việc cài đặt và sử dụng bộ công cụ Logstash, Elasticsearch, Kibana đã hoàn tất.



PHẦN 4. PHÂN TÍCH LOG FILE ỨNG DỤNG

4.1. EMAIL LOG FILE

4.1.1. Giới thiệu Email Log file

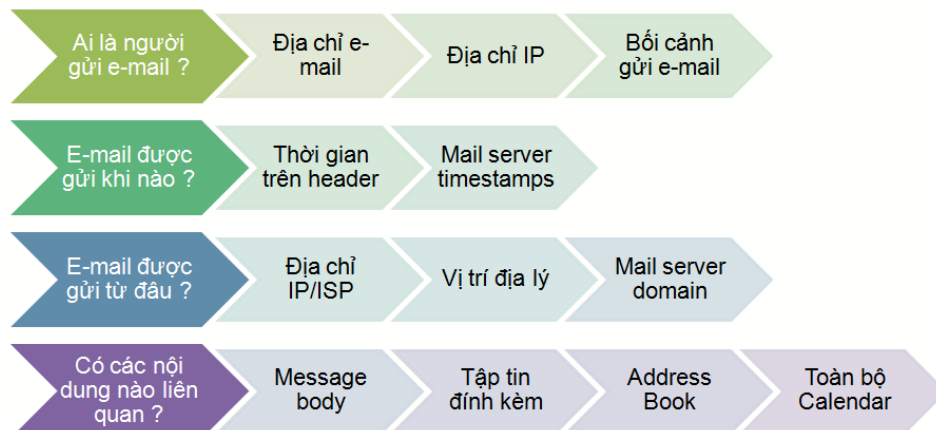
Email log là các tập tin có chứa thông tin về tất cả các email được gửi thông qua Mail Server trong khoảng thời gian yêu cầu. Các thông tin bao gồm:

- Các địa chỉ email của từng người gửi và người nhận email
- Ngày và thời gian mỗi email được gửi đi
- Các trạng thái phân phối của từng email
- Bất kỳ mã lỗi liên quan với mỗi email

Bạn có thể sử dụng log email để khắc phục các lỗi gặp phải.

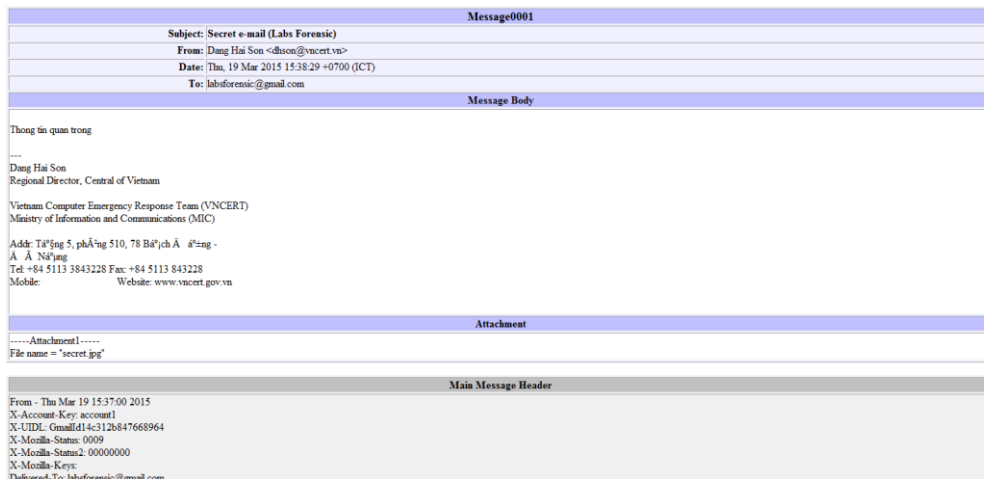
4.1.2. Phân tích Email Log file

Phân tích Email cho biết các thông tin:



Các thành phần có thể phân tích:

- Mail Header
- Message Body
- Attachments



4.1.3. Phát hiện Email giả mạo

4.1.3.1. Phương thức tạo thư giả mạo của tin tặc

Thông thường khi soạn và gửi thư điện tử, người gửi thư chỉ biên soạn nội dung, tiêu đề thư (title), địa chỉ nơi nhận, lựa chọn các tệp tin đính kèm, các thông tin còn lại khác sẽ do máy chủ gửi thư tự động cập nhật như: địa chỉ hòm thư nhận phản hồi khi thư bị trả lại (Return-Path); địa chỉ hòm thư tiếp nhận thư trả lại (Reply-To) và địa chỉ hòm thư người gửi (from).

Để đánh lừa người nhận tin, bước đầu tin tặc sẽ tìm cách tự biên soạn thư điện tử

với các thông tin giả mạo về: địa chỉ hòm thư nhận phản hồi khi thư bị trả lại (Return-Path); địa chỉ hòm thư tiếp nhận thư trả lời (Reply-To) và địa chỉ hòm thư người gửi (from). Sau đó tin tặc sẽ tìm một máy chủ thư điện tử hoặc tự cài đặt một phần mềm gửi thư (MTA) không yêu cầu xác thực hòm thư người gửi để phát tán thư điện tử giả mạo tới người cần lừa đảo.

4.1.3.2. Tìm hiểu nguồn gốc thật phát tán của thư điện tử

Trong nội dung thư điện tử gửi đến người nhận bao gồm các đầy đủ thông tin về: địa chỉ IP của máy gửi thư; địa chỉ hòm thư nhận; địa chỉ hòm thư nhận phản hồi khi thư bị trả lại (Return-Path); địa chỉ hòm thư tiếp nhận thư trả lời (Reply-To) và địa chỉ hòm thư người gửi (from); nội dung thư; Tiêu đề thư; Các tệp tin đính kèm. Nhưng trong chế độ hiển thị thông thường (mặc định) để đơn giản hóa giao diện, hầu hết các chương trình duyệt thư điện tử chỉ hiện các thông

tin: địa chỉ hòm thư tiếp nhận thư trả lời (Reply); Địa chỉ hòm thư người nhận; nội dung thư; tiêu đề thư; các tệp tin đính kèm và các thời gian liên quan. Các thông tin chi tiết về nguồn gốc của thư như: địa chỉ IP của máy gửi thư; địa chỉ hòm thư nhận phản hồi khi thư bị trả lại (Return-Path); địa chỉ hòm thư tiếp nhận thư trả lời (Reply-To) và địa chỉ hòm thư người gửi (from) được lưu trong phần đầu (header) của thư sẽ chỉ hiện thị chi tiết khi người nhận thư sử dụng các chức năng cho xem nguồn gốc (original) của thư hoặc xem nội dung phần đầu (header) của thư (Chú ý: đối với mỗi trình duyệt và hệ quản trị thư điện tử khác nhau sẽ có những cách khác nhau để xem nguồn gốc của thư điện tử, tuy nhiên tất cả các phần mềm trên đều hỗ trợ chức năng show original).

4.1.3.3. Phát hiện thư giả mạo

Qua phân tích các thư điện tử giả mạo đã gửi đến các cơ quan nhà nước trong thời gian vừa qua, có hai dấu hiệu chính để có thể phát hiện ra các thư giả mạo theo phương thức này là:

- Khi mở xem nguồn gốc chi tiết của thư điện tử, địa chỉ hòm thư “Return-Path” không trùng với địa chỉ hòm thư người gửi đến (From). Hầu hết các thư điện tử được gửi từ các hệ thống thư điện tử của cơ quan nhà nước (có đuôi .gov.vn) đều có hai địa chỉ này trùng nhau.

- Địa chỉ IP của máy chủ gửi thư không trùng với địa chỉ IP của hệ thống thư điện tử thật nơi bị giả mạo là gửi thư điện tử. Hiện nay, các địa chỉ IP giả mạo này thường có nguồn gốc từ nước ngoài trong khi địa chỉ IP các hệ thống cơ quan nhà nước thường có địa chỉ IP trong nước.

Ví dụ minh họa :

Dưới đây là ví dụ minh họa một thư điện tử giả mạo ông Vũ Xuân Hoàng có địa chỉ thư điện tử là hoangvx@abc.gov.vn được tin tặc gửi tới hòm thư của chị Nguyễn Thanh Huyền có địa chỉ huyennt@xyz.gov.vn. Tin tặc tạo ra thư giả mạo ông Vũ Xuân Hoàng có tiêu đề là “Thông báo lớp đào tạo” với các địa chỉ hòm thư gửi, và hòm thư nhận là hoangvx@abc.gov.vn, hòm thư trả lại là root@nbr.com. Sau đó tin tặc sử dụng máy gửi thư có địa chỉ IP

xxx.xxx.xxx.xxx để gửi thư giả mạo đã soạn tới hòm thư của chị Nguyễn Thanh Huyền có địa chỉ huyennt@xyz.gov.vn.

Return-Path: root@nbr.com

Địa chỉ hòm thư trả lại - Return

Received: from xyz.gov.vn (LHLO xyz.gov.vn) (yyy.yyy.yyy.yyy) by xyz.gov.vn with

LMTP; Sat, 18 May 2013 15:24:12 +0700 (ICT)

Received: from localhost (localhost [127.0.0.1])

by xyz.gov.vn (Postfix) with

for <huyennt@xyz.gov.vn>; Sat, 18 May 2013 15:24:12 +0700 (ICT)

Địa chỉ IP của máy chủ gửi thư

Received: from nbr.com (unknown [xxx.xxx.xxx.xxx])

by xyz.gov.vn (Postfix) with ESMTPT id C4493288FE8

for <huyennt@xyz.gov.vn> Sat, 18 May 2013 15:23:43 +0700 (ICT)

Date: Sat, 18 May 2013 12:25

Message-Id: <201305181625.r4

Địa chỉ hòm thư nhận cần lừa đảo

To: huyennt@xyz.gov.vn

Subject: Thông báo lớp đào tạo

From: Vu Xuan Hoang <hoangvx@abc.gov.vn>

Địa chỉ hòm thư gửi (bị giả)

Reply-To: hoangvx@abc.gov.vn

Địa chỉ hòm thư nhận trả lời (bị giả mạo)

Một ví dụ khác:

Real E-mail

```
Return-Path: labsforensic@gmail.com
Received: from localhost (LHLO mail.vncert.vn) (127.0.0.1) by mail.vncert.vn
with LMTP; Tue, 17 Mar 2015 16:51:10 +0700 (ICT)
Received: from localhost (localhost [127.0.0.1])
by mail.vncert.vn (Postfix) with ESMTPT id DC5A26CE6F
for <dhsn@vncert.vn>; Tue, 17 Mar 2015 16:51:10 +0700 (ICT)
X-Virus-Scanned: amavisd-new at vncert.vn
X-Spam-Flag: NO
X-Spam-Score: 1.61
X-Spam-Level: *
X-Spam-Status: No, score=1.61 tagged_above=-10 required=6
tests=[BAYES_40=-0.001, DKIM_SIGNED=0.1, DKIM_VALID=0.1,
DKIM_VALID_AU=0.1, DNS_FROM_AHBL_RHSBL=2.699, FREEMAIL_FROM=0.001,
HTML_MESSAGE=0.001, SPF_PASS=0.001, TVD_SPACE_RATIO=0.001,
T_TO_NO_BRKTS_FREEMAIL=0.01, DSPAM:Innocent=-1.000] autolearn=no
Authentication-Results: mail.vncert.vn (amavisd-new); dkim=pass
header.i=@gmail.com
Received: from mail.vncert.vn ([127.0.0.1])
by localhost (mail.vncert.vn [127.0.0.1]) (amavisd-new, port 10024)
with ESMTPT id 50V1aXN14000; Tue, 17 Mar 2015 16:51:04 +0700 (ICT)
Received: from mail-010-f65.google.com (mail-010-f65.google.com [209.85.218.65])
by mail.vncert.vn (Postfix) with ESMTPT id 0941E282C00
for <dhsn@vncert.vn>; Tue, 17 Mar 2015 16:51:03 +0700 (ICT)
Received: by oiba3 with SMTP id a3e01789101b.0
for <dhsn@vncert.vn>; Tue, 17 Mar 2015 02:47:03 -0700 (PDT)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
d=gmail.com; s=20120113;
h=mime-version:date:message-id:subject:from:to:content-type;
bh=CwEjN0D0CfPqP/L/3C0M0aJ9e0017R0e0x2G0a;
b=E0eX0LjYkdu0Ck0X0T0a0pW0C020g0P0a0e04IAC0R0z0m0H0K0e0h0V0B0v
F0j0a0k0p0x0Kj08010p0x07J0d0Kj0p020a0b070u050X0E0a060v0z0v0v0F0e0Kj
I0i0f0n0a0l0u0P0W0t0x0F0W0b0j0b0q0H0q0T0E0X0d0z0Q0i0d0b0U0M0d0p0v0W0e0c0R0Q0W0
h0C0J020a0w0i0r0T0u0b0d0e020F0J0V0C0M0e0a0F0i0I0H0a0E0C0L0a0r0C0F0U0C0T0C0P0b0J0k0
a0c0a0G0M0I0Q0Z0f0v0K0f0p0n0K03020T0Z0Lj0X0X0f0b0c0T0H0c0E0T0X0H0Q0W0U0Z0a0C0Z0X0J0L0
W0r0w0=
MIME-Version: 1.0
X-Received: by 10.182.128.199 with SMTP id nq7mr29159310bb.47.1426585423131;
Tue, 17 Mar 2015 02:47:03 -0700 (PDT)
Received: by 10.202.172.2 with HTTP; Tue, 17 Mar 2015 02:47:03 -0700 (PDT)
Date: Tue, 17 Mar 2015 16:47:03 +0700
Message-Id: <CApFudm:1Pw==T0C-4Tl0gXKl58U0gDE9ngIc0bm3=idWfkt-Yv@mail.gmail.com>
Subject: Test forward
From: Forensic Labs <labsforensic@gmail.com>
To: dhsn@vncert.vn
Content-Type: multipart/alternative; boundary=089e01537e764ef9e5051178d99e
--089e01537e764ef9e5051178d99e
Content-Type: text/plain; charset=UTF-8
Test
```

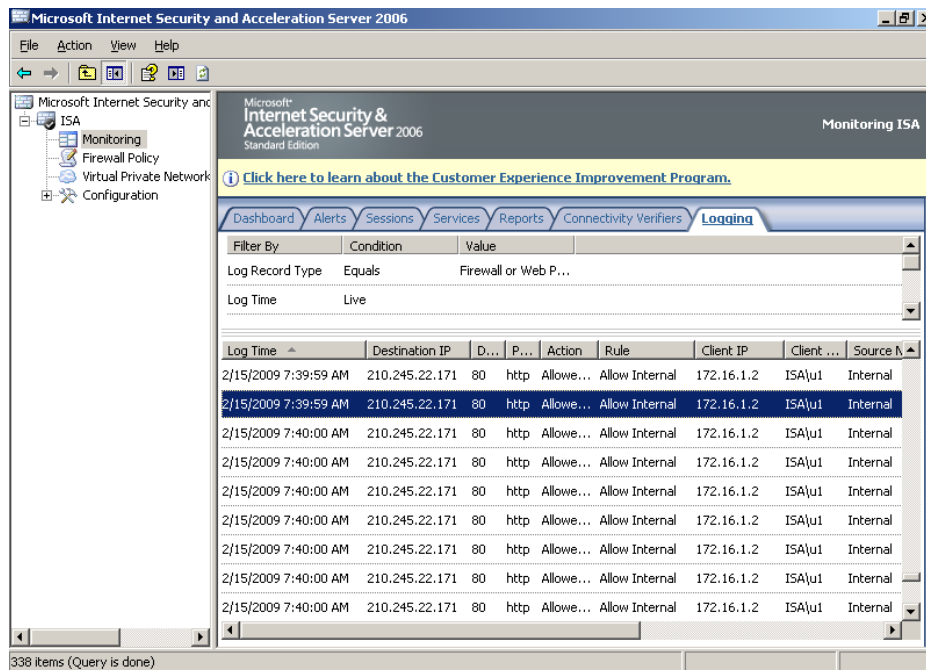
Fake E-mail

```
Return-Path: labsforensic@gmail.com
Received: from localhost (LHLO mail.vncert.vn) (127.0.0.1) by mail.vncert.vn
with LMTP; Tue, 17 Mar 2015 20:04:49 +0700 (ICT)
Received: from localhost (localhost [127.0.0.1])
by mail.vncert.vn (Postfix) with ESMTPT id 6D93328CF42
for <dhsn@vncert.vn>; Tue, 17 Mar 2015 20:04:49 +0700 (ICT)
X-Virus-Scanned: amavisd-new at vncert.vn
X-Spam-Flag: NO
X-Spam-Score: 4.776
X-Spam-Level: ****
X-Spam-Status: No, score=4.776 tagged_above=-10 required=6
tests=[BAYES_60=1.5, DKIM_ADSP_CUSTOM_MED=0.001,
DNS_FROM_AHBL_RHSBL=2.699, FREEMAIL_FROM=0.001,
NML_ADSP_CUSTOM_MED=0.9, SPF_SOFTFAIL=0.665,
T_TO_NO_BRKTS_FREEMAIL=0.01, DSPAM:Innocent=-1.000] autolearn=no
Received: from mail.vncert.vn ([127.0.0.1])
by localhost (mail.vncert.vn [127.0.0.1]) (amavisd-new, port 10024)
with ESMTPT id wqkDRGU7J3y5; Tue, 17 Mar 2015 20:04:42 +0700 (ICT)
Received: from emkei.cz (emkei.cz [46.167.245.72])
by mail.vncert.vn (Postfix) with ESMTPT id 0D4F528CF29
for <dhsn@vncert.vn>; Tue, 17 Mar 2015 20:04:42 +0700 (ICT)
Received: by emkei.cz (Postfix, from userid 33)
id E9476D5759; Tue, 17 Mar 2015 14:00:39 +0100 (CET)
To: dhsn@vncert.vn
Subject: Fake Test Forensic
From: "Labs Forensic" <labsforensic@gmail.com>
X-Priority: 3 (Normal)
Importance: Normal
Errors-To: labsforensic@gmail.com
Reply-To: labsforensic@gmail.com
Content-Type: text/plain; charset=utf-8
Message-Id: <20150317130039.E9476D5759@emkei.cz>
Date: Tue, 17 Mar 2015 14:00:39 +0100 (CET)
Test
```

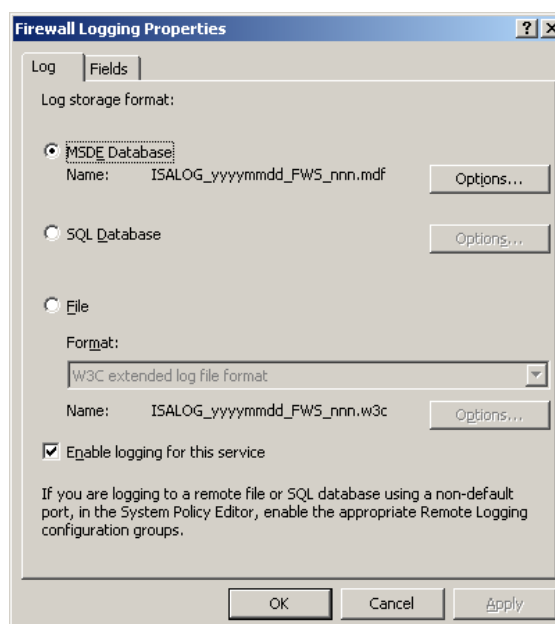
4.2. FIREWALL LOG FILE

4.2.1. ISA Logging

Như chúng ta đã biết trong ISA Server 2006 có chức năng Logging, bạn có thể theo dõi việc truy cập Internet của user theo thời gian thực (real time).

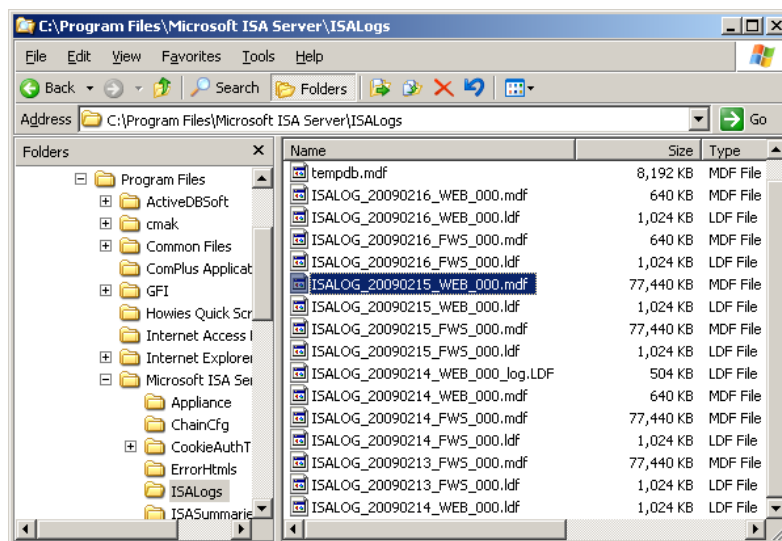


Đồng thời bạn cũng có thể lập báo cáo (reporting) mỗi ngày (daily) hoặc lập báo cáo theo mốc thời gian. Ví dụ bạn sử dụng ISA Server từ ngày 01/01/2008, nhưng bạn vẫn có thể lập báo cáo theo mốc thời gian từ ngày 01/01/2008 đến hết ngày 14/02/2009. Tại sao ISA có thể làm như vậy? Vì ISA lưu các sự kiện đó theo dạng *.mdf và *.ldf trong thư mục C:\Program Files\Microsoft ISA Server\ISALogs. Hình bên dưới chỉ định log file được lưu mặc định theo định dạng MSDE database với tên *.mdf



Các file *.mdf và *.ldf được tạo ra theo định dạng ISALOG_yyyymmdd_FWS_nnn.mdf, ISALOG_yyyymmdd_WEB_nnn.mdf và ISALOG_yyyymmdd_EML_nnn.mdf . Trong đó:

- yyyy: năm tạo file log
- mm: tháng tạo file log
- dd: ngày tạo file log
- FWS log của firewall
- WEB log của proxy
- EML log của Email (SMTP)
- nnn: số thứ tự file log trong trường hợp nhiều file log trong 1 ngày



Quay lại ví dụ trên, hôm nay là 15/02/2009, bạn muốn xem các sự kiện trong ngày 15/2, do bạn không xem logging theo thời gian thực nên bạn lập báo cáo trong ngày 15/2 thì ISA Server không cho phép. ISA chỉ cho xem các sự kiện đến cuối ngày 14/02 mà thôi. Mở thư mục C:\Program Files\Microsoft ISA Server\ISALogs thì có nhiều file *.mdf được tạo ra trong ngày 15/2 . Để mở được file này đòi hỏi phải có SQL Server (phải mua license), attach file *.mdf vào SQL Server và sử dụng các công cụ mà SQL Server có sẵn thì mới đọc được nội dung file *.mdf. Và bạn cũng đã từng hỏi tại sao cài xong ISAServer thì tại system tray lại có 1 biểu tượng giống SQL Server, nhưng đó không phải là SQL Server mà là MSDE. Có thể nói MSDE là một dạng SQL thu nhỏ.

4.2.2. Iptable log file

Kích hoạt ghi log trên iptables là hữu ích cho việc giám sát traffic đến máy chủ của chúng ta. Chúng ta cũng có thể tìm thấy số lượng truy cập thực hiện từ bất kỳ ip. Phần này sẽ giúp cho phép ghi log trong iptables cho tất cả các gói dữ liệu được lọc bởi iptables.

4.2.2.1. Kích hoạt Iptables log

Sử dụng lệnh sau để kích hoạt việc ghi log trên iptables.

```
$ iptables -A INPUT -j LOG
```

Xác định ip nguồn hoặc phạm vi mà log sẽ được tạo ra.

```
$ iptables -A INPUT -s 192.168.10.0/24 -j LOG
```

Để xác định cấp độ LOG tạo ra bởi iptables phía sau tham số `--log-level`

```
$ iptables -A INPUT -s 192.168.10.0/24 -j LOG --log-level 4
```

Chúng tôi cũng có thể thêm một số tiền tố trong Log được tạo ra, Vì vậy, nó sẽ được dễ dàng để tìm kiếm các nhật ký trong một tập tin rất lớn.

```
$ iptables -A INPUT -s 192.168.10.0/24 -j LOG --log-prefix '**  
SUSPECT **'
```

4.2.2.2. Xem Iptables Log

Sau khi kích hoạt ghi log trên iptables, kiểm tra lại file log để xem các bản ghi được tạo ra bởi iptables theo hệ điều hành.

Trên Ubuntu và Debian: Iptables log được tạo ra bởi các kernel. Để kiểm tra log file kernel, dùng lệnh sau:

```
$ tailf /var/log/kern.log
```

Trên CentOS/RHEL và Fedora

```
# cat /var/log/messages
```

4.2.2.3. Thay đổi tên Log File Iptables

Để thay đổi tên tập tin log trong iptables, chỉnh sửa file cấu hình `/etc/rsyslog.conf`

```
# vi /etc/syslog.conf
```

Chèn thêm dòng:

```
kern.warning /var/log/iptables.log
```

Bây giờ thì khởi động lại hệ thống bằng lệnh:

```
$ service rsyslog restart
```

4.3. IDS/IPS LOG FILE

4.3.1. Giới thiệu IDS/IPS

IDS (Intrusion Detection System) hay còn gọi là hệ thống phát hiện xâm nhập là một hệ thống phòng chống, nhằm phát hiện các hành động tấn công vào một mạng. Mục đích của nó là phát hiện và ngăn ngừa các hành động phá hoại đối với vấn đề bảo mật hệ thống, hoặc những hành động trong tiến trình tấn công như sưu tập, quét các cổng. Một tính năng chính của hệ thống này là cung cấp thông tin nhận biết về những hành động không bình thường và đưa ra các báo cảnh thông báo cho quản trị viên mạng khóa các kết nối đang tấn công này. Thêm vào đó công cụ IDS cũng có thể phân biệt giữa những tấn công bên trong từ bên trong tổ chức (từ chính nhân viên hoặc khách hàng) và tấn công bên ngoài (tấn công từ hacker).

IPS (Intrusion Prevention System) hay còn gọi là hệ thống phòng chống xâm nhập được định nghĩa là một phần mềm hoặc một thiết bị chuyên dụng có khả năng phát hiện xâm nhập và có thể ngăn chặn các nguy cơ gây mất an ninh.

IDS và IPS có rất nhiều điểm chung, do đó hệ thống IDS và IPS có thể được gọi chung là IDP (Intrusion Detection và Prevention).

4.3.2. Phân tích IDS/IPS log

Cung cấp các thông tin về:

- Cảnh báo về loại gói tin đáng ngờ
- Giúp trong việc xác định các thiết bị thăm dò
- Giúp trong việc tạo dấu hiệu phát hiện tấn công mới
- Thống kê Attack (Host/Network based)

4.4. DATABASE LOG FILE

4.4.1. SQL Server log file

Khi bạn nghĩ về SQL Server log file, suy nghĩ đầu tiên của bạn chính là transaction log, trong đó ghi lại các giao dịch cơ sở dữ liệu gần đây và được sử dụng để đảm bảo tính toàn vẹn cơ sở dữ liệu trong trường hợp của một hệ thống khôi phục lại. Tuy nhiên, nhiều tập tin nhật ký khác cũng giúp chẩn đoán và khắc phục sự cố. Dưới đây là năm các file log có vai trò quan trọng trong SQL Server.

4.4.1.1. SQL Server Setup Log

Bạn có thể đã quen thuộc với SQL Server Setup log, vị trí của nó nằm ở `\ProgramFiles\Microsoft SQL Server\90\Setup Bootstrap\LOG\Summary.txt`. Nếu tập tin log summary.txt sẽ hiển thị một thành phần thất bại, bạn có thể điều tra các nguyên nhân gốc rễ bằng cách nhìn vào log của các thành phần, trong đó bạn sẽ tìm thấy trong thư mục `\Program-Files\Microsoft SQL Server\90\Setup Bootstrap\LOG\Files`.

4.4.1.2. SQL Server Profiler Log

SQL Server Profiler, công cụ truy tìm ứng dụng chính trong SQL Server, nắm bắt hoạt động cơ sở dữ liệu hiện tại của hệ thống và viết nó vào một tập tin để phân tích sau. Bạn có thể tìm thấy các Profiler log trong file log .trc trong thư mục `ProgramFiles\Microsoft SQL Server\MSSQL.1\MSSQL\LOG`.

4.4.1.3. SQL Server Agent Log

SQL Server Agent, là hệ thống phụ lập lịch trình công việc SQL Server, duy trì một bộ các tập tin nhật ký với thông điệp cảnh báo và báo lỗi về việc nó đã chạy, được viết vào thư mục `ProgramFiles\Microsoft SQL Server\MSSQL.1\MSSQL\LOG`. SQL Server sẽ duy trì tối đa chín SQL Server Agent lỗi trong file log. Các tập tin log hiện được đặt tên là `SQLAgent .OUT`, trong khi tập tin lưu trữ được đánh số tuần tự. Bạn có thể xem SQL Server Agent Log bằng cách sử dụng SQL Server Management Studio (SSMS).

4.4.1.4. Windows Event Log

Một nguồn thông tin quan trọng cho việc xử lý sự cố lỗi của SQL Server, Windows Event Log chứa các bản ghi rất hữu ích. Các application log ghi lại các sự kiện trong SQL Server và SQL Server Agent và có thể được sử dụng bởi và SQL Server Integration Services (SSIS). Các security log ghi lại thông tin xác thực, và system log ghi lại việc khởi động và tắt máy. Để xem các Windows Event log, vào Administrative Tools, Event Viewer.

4.4.1.5. SQL Server Error Log

Error Log, là log file quan trọng nhất trong SQL Server, được sử dụng để khắc phục sự cố hệ thống. SQL Server giữ lại bản sao lưu của sáu bản ghi trước đó, đặt tên mỗi file bản ghi lưu trữ theo tuần tự. Các file error log hiện tại được đặt tên ERRORLOG. Để xem các error log, nằm trong thư mục Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\ERRORLOG.

4.4.2. MySQL log file

Liên quan đến các máy chủ cơ sở dữ liệu MySQL nổi tiếng bạn cần phải tham khảo các Log file sau đây:

The Error Log: Chứa thông tin về các lỗi xảy ra trong khi máy chủ đang hoạt động (start và stop máy chủ)

The General Query Log: Các log chung liên quan đến MySQL (connect, disconnect, queries)

The Slow Query Log: Log ghi lại các query thực thi lâu đến MySQL (as indicated by its name).

4.4.2.1. Kích hoạt ghi log từ cấu hình MySQL

Chỉnh sửa tập tin cấu hình MySQL:

```
nano /etc/mysql/my.cnf
```

Đây là thiết lập mặc định cho Logging và Replication (Debian). Trong bản phân phối khác cấu trúc có thể khác nhau.

```
# * Logging and Replication
# Both location gets rotated by the cronjob.
```

```

# Be aware that this log type is a performance killer.
# As of 5.1 you can enable the log at runtime!
#general_log_file      = /var/log/mysql/mysql.log
#general_log           = 1
#      Error           logging      goes      to      syslog      due      to
/etc/mysql/conf.d/mysqld_safe_syslog.cnf.
# Here you can see queries with especially long duration
#log_slow_queries      = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
# The following can be used as easy to replay backup logs or for
replication.
# note: if you are setting up a replication slave, see README.Debian
about
#      other settings you may need to change.
#server-id             = 1
#log_bin               = /var/log/mysql/mysql-bin.log
expire_logs_days      = 10
max_binlog_size        = 100M
#binlog_do_db          = include_database_name
#binlog_ignore_db      = include_database_name

```

Tất cả các tập tin nhật ký không được kích hoạt theo mặc định thiết lập MySQL (ngoại trừ các bản ghi lỗi trên Windows). Mặc định Debian thiết lập gửi error log đến syslog. Các log file khác không được kích hoạt.

4.4.2.2. Error Log

Gửi error log đi đến syslog trong `/etc/mysql/conf.d/mysqld_safe_syslog.cnf`, trong đó có những điều sau đây:

```

[mysqld_safe]
syslog

```

Đây là phương pháp được khuyến nghị. Nếu vì một số lý do, nếu không muốn gửi Error log đến syslog, thêm ký tự `#` trước dòng trên trong `/etc/mysql/conf.d/mysqld_safe_syslog.cnf` hoặc xóa hoàn toàn tập tin này. Sau đó, thêm vào trong `/etc/mysql/my.cnf` các dòng sau:

```

[mysqld_safe]
log_error=/var/log/mysql/mysql_error.log
[mysqld]
log_error=/var/log/mysql/mysql_error.log

```

Khởi động lại MySQL server sau khi thay đổi:

```
service mysql restart
```

4.4.2.3. General Query Log

Để kích hoạt General Query Log, thêm dòng sau:

```
general_log_file = /var/log/mysql/mysql.log  
general_log      = 1
```

Khởi động lại MySQL server sau khi thay đổi:

```
service mysql restart
```

4.4.2.4. Slow Query Log

Để kích hoạt Slow Query Log, thêm dòng sau:

```
log_slow_queries      = /var/log/mysql/mysql-slow.log  
long_query_time = 2  
log-queries-not-using-indexes
```

Khởi động lại MySQL server sau khi thay đổi:

```
service mysql restart
```

4.5. CÔNG CỤ PHÂN TÍCH

4.5.1. SolarWinds LEM

Để giúp bảo vệ dữ liệu nhạy cảm và làm giảm nguy cơ mất dữ liệu, người dùng được khuyên nên sử dụng công nghệ Security Information và Event Management (SIEM), cụ thể như SolarWinds Log & Event Manager (LEM).

SolarWinds LEM là một sản phẩm SIEM toàn diện, được đóng gói trong một thiết bị ảo rất dễ sử dụng, một ứng dụng có tất cả trong một. Với chức năng out-of-the-box, bất cứ ai cũng có thể sử dụng và chạy nó mà không cần phải là một chuyên gia bảo mật! Quan trọng nhất, nó bao gồm tính năng phát triển chuyên môn để giúp phòng chống mất mát dữ liệu.

Tính năng của SolarWinds LEM:

- Log File Analysis
- Operating System Log Files
- Network Infrastructure Log Files (Syslog)
- IDS/IPS Log Files

- Endpoint Security: Antivirus (AV) & Malware Log Files
- Endpoint Security: Identity Authentication and Endpoint Protection Log Files
- Web Server and Application Log Files
- FTP and Content Management Log Files

Link download và tham khảo: <http://www.solarwinds.com/log-event-manager/log-file-analysis.aspx>

4.5.2. Splunk

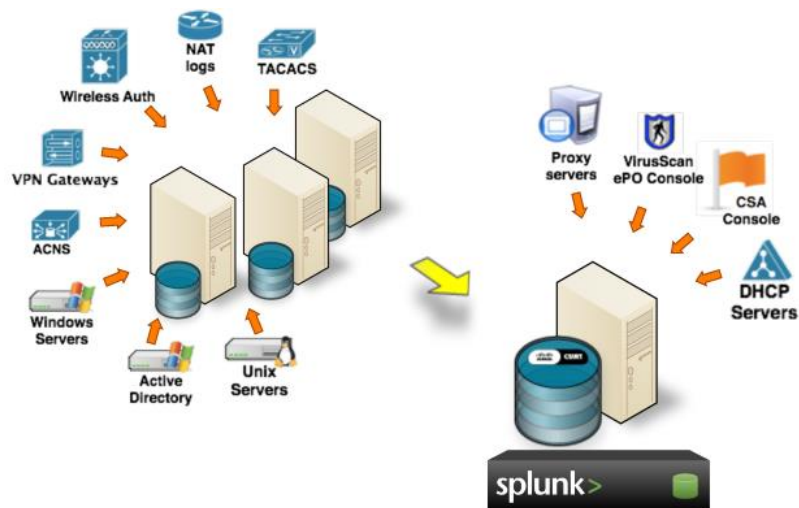
4.5.2.1. Giới thiệu

Splunk là phần mềm cho phép CNTT có thể tìm kiếm và duyệt logs và các dữ liệu IT trong thời gian thực. Người dùng có thể ngay lập tức phát hiện ra sự cố ở bất cứ ứng dụng nào, hoặc ở các máy chủ và thiết bị; cảnh báo các nguy cơ tiềm ẩn và báo cáo các hoạt động của các dịch vụ và thành phần khác nhau trong mạng. Và đây cũng là giải pháp troubleshoot cho hệ thống.

Splunk là một công cụ dữ liệu rất linh hoạt và khả năng mở rộng cho các dữ liệu máy tính được tạo ra bởi cơ sở hạ tầng CNTT của CNTT. Nó thu thập, lập chỉ mục và khai thác những dữ liệu được tạo ra từ bất cứ nguồn nào, định dạng hoặc vị trí bao gồm cả đóng gói và các ứng dụng tùy chỉnh, máy chủ ứng dụng, máy chủ web, cơ sở dữ liệu, mạng, máy ảo, hypervisors, hệ điều hành và nhiều hơn nữa mà không cần phải phân tích cú pháp tùy chỉnh, bộ điều hợp hoặc một cơ sở dữ liệu trên các phụ trợ.

Splunk được sử dụng để cung cấp một cái nhìn rõ ràng, chi tiết về toàn bộ hệ thống công nghệ thông tin. Nó liên kết các dữ liệu riêng biệt của các thiết bị, ứng dụng riêng biệt lại với nhau một cách tự động, giúp quá trình tìm kiếm điều tra trở nên nhanh chóng, đơn giản đi rất nhiều.

Chủ động giám sát các dữ liệu để phát hiện các bất thường theo thời gian thực. Cho phép người dùng ngay lập tức đi sâu chi tiết vào vấn đề gặp phải để có hướng giải quyết một cách chính xác và nhanh chóng nhất.



4.5.2.2. Lợi thế của Splunk

- Linh hoạt mềm dẻo khi sử dụng

Linh hoạt, khả năng mở rộng và đủ linh hoạt để tìm kiếm trên terabyte dữ liệu từ bất kỳ nguồn dữ liệu như các nguồn truyền thống an ninh, các ứng dụng tùy chỉnh, và cơ sở dữ liệu. Splunk tự động cung cấp một cái nhìn chi tiết theo dòng thời gian của tất cả các dữ liệu thu thập được.

Thời gian này có thể được sử dụng để tập trung vào thời điểm chính xác trong thời gian một sự kiện an ninh xảy ra. Bất kỳ kết quả tìm kiếm có thể được biến thành một báo cáo để phân phối. Điều này đặc biệt hữu ích cho các truy vấn ad-hoc hỗ trợ các sáng kiến tuân thủ như PCI, SOX hoặc HIPAA.

- Điều tra theo thời gian thực

Một khi một cuộc điều tra pháp y là hoàn chỉnh, tìm kiếm Splunk có thể được lưu lại và theo dõi trong thời gian thực. Cảnh báo thời gian thực có thể được chuyển đến các thành viên trong nhóm bảo mật thích hợp để theo dõi. Tương quan qua hệ thống dữ liệu của nhà cung cấp hoặc kiểu dữ liệu được hỗ trợ trong dễ sử dụng tìm kiếm Splunk của ngôn ngữ tìm kiếm của Splunk hỗ trợ tương quan có thể tạo ra các cảnh báo dựa trên sự kết hợp các điều kiện cụ thể, mô hình trong hệ thống dữ liệu hoặc khi một ngưỡng cụ thể đạt được.

Splunk cho phép bạn xem thông tin thời gian thực từ an ninh và thiết bị mạng, hệ điều hành, cơ sở dữ liệu và các ứng dụng, trên một thời gian cho phép

các đội an ninh để nhanh chóng phát hiện và hiểu được ý nghĩa end-to -end của một sự kiện an ninh.

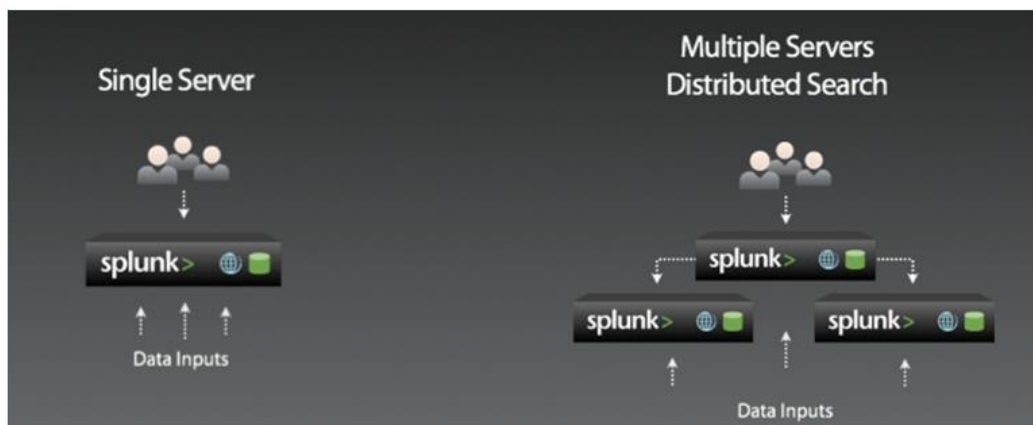
Splunk sẽ giải quyết được khó khăn của các hệ thống bảo mật hiện tại khi tìm kiếm và phát hiện các hành vi nguy hiểm đang hoạt động trong hệ thống. Với khả năng phát hiện từng hành vi bất hợp pháp phát nhỏ nhất, Splunk sẽ giúp phát hiện những cuộc tấn công tinh vi nhằm vào hệ thống một cách nhanh chóng và hiệu quả nhất.

- Liên kết thông tin theo thời gian thực và cảnh báo

Tương quan của thông tin từ bộ dữ liệu khác nhau có thể làm giảm tính giả tích cực và cung cấp cái nhìn sâu sắc thêm và bối cảnh. Splunk có thể liên kết với tất cả các thông tin dữ liệu từ mọi nguồn trên hệ thống một cách nhanh chóng và chính xác theo thời gian thực.

4.5.2.3. Mô hình triển khai Splunk

Do Splunk là Software multi-platform nên việc triển khai rất đơn giản, ta có hai mô hình triển khai cơ bản như sau:



Single Server: Splunk được cài đặt trên một server đóng vai trò là một host tập trung và lên mục lục các dữ liệu log. Mô hình triển khai dễ dàng, và sau này bạn cũng có thể mở rộng thêm.

Multiple Server: Splunks được cài đặt thành nhiều cấp và điều này có thể làm tăng hiệu quả hoạt động và gia tăng thêm khả năng mở rộng hệ thống.

Ngoài ra đối với các thành phần cộng thêm thì tùy vào từng mô hình cụ thể sẽ triển khai. Do Splunk instance có thể đóng nhiều vai trò như Indexer,

Search Head, Forwarder,... nên việc triển khai các thành phần cộng thêm sẽ rất dễ dàng và không cần thay đổi mô hình hiện tại.

4.5.2.4. Các thành phần của Splunk

Đối với Forwarder thì ta có các thành phần nhỏ như sau:

- Universal Forwarder: cài đặt tại thành phần cần lấy log, hoặc động như một agent của Splunk, chiếm rất ít performance của hệ thống. Chức năng chính là forward log.
- Heavy Forwarder: cũng có chức năng forward log, nhưng Heavy Forwarder lại mang gần như đầy đủ các tính năng của Indexer (ngoại trừ khả năng thực hiện distributed search). Log trước khi được forward đi sẽ được phân tích, cũng như có khả năng route log theo điều kiện

Tham khảo: https://www.splunk.com/en_us/download/splunk-enterprise.html

PHẦN 5. NHẬN DIỆN TẤN CÔNG VÀO ỨNG DỤNG WEB TỪ LOGFILE

5.1. TẤN CÔNG VÀO ỨNG DỤNG WEB

5.1.1. Quy trình tấn công vào ứng dụng Web

Quy trình thực hiện tấn công vào hệ thống thông tin bao gồm 5 bước được mô tả theo sơ đồ sau:



5.1.1.1. Thu thập thông tin

Thu thập thông tin là hoạt động tìm kiếm, tập hợp thông tin về hệ thống đích một cách nhiều nhất có thể. Các thông tin cụ thể cần được thu thập như là hệ điều hành, nền tảng, công nghệ web sử dụng hoặc tìm lỗ hổng bảo mật và khai thác liên quan đến hệ thống đích.

Đối tượng thu thập:

- Thông tin dãy mạng: Tên miền, tên miền con, dãy địa chỉ mạng, địa chỉ IP, các dịch vụ TCP/UDP đang tồn tại, ...
- Thông tin hệ thống: Tài khoản, nhóm người dùng, banners hệ thống, bảng định tuyến, thông tin SNMP, ...
- Thông tin tổ chức: Thông tin nhân viên, website của tổ chức/đơn vị, cấu trúc tổ chức, ...

Phương pháp thu thập:

- Tiết lộ của nhân viên tổ chức mục tiêu: Trong giai đoạn đầu của cuộc kiểm thử, đại diện của tổ chức mục tiêu có thể cung cấp một danh sách các mục tiêu ban đầu.
- Được phát hiện bởi tìm kiếm Google: Google là một công cụ tìm kiếm thông tin rất phong phú và hữu ích.
- Được phát hiện bởi chuyển vùng DNS: DNS cung cấp rất nhiều thông tin, nếu việc chuyển vùng được cho phép.
- Được phát hiện bởi tra cứu ngược DNS: Chúng ta có thể tìm thấy các máy chủ bằng cách thực hiện tra cứu ngược DNS.
- Được phát hiện trong quá trình quét mạng: Có rất nhiều phương pháp để quét mạng để phát hiện máy chủ.
- Được phát hiện trong quá trình đánh giá vật lý: Nếu việc kiểm thử bao gồm cả kiểm thử mạng không dây, chúng ta có thể tìm thấy một số máy chủ thông qua phương pháp này.
- Được phát hiện bởi sự thỏa hiệp vào một máy chủ: Đây là một trong những phương pháp thú vị nhất để phát hiện máy chủ - từ một mục tiêu và tìm kiếm các mục tiêu khác xung quanh

Cách thức thu thập:

- Tìm kiếm Whois: Để tìm kiếm những thông tin chi tiết hơn về một tên miền, chúng ta có thể sử dụng những cơ sở dữ liệu Whois.
- Tìm kiếm thông tin từ trang web: Tìm kiếm thêm thông tin về mục tiêu thông qua các nguồn thông tin được công bố công khai. Những trang web trên toàn thế giới là một kho tàng thông tin, rất có ích. Tìm kiếm các thông tin như: Thông tin về hoạt động, thông tin về tuyển dụng, thông tin liên quan đến con người
- Phân tích siêu dữ liệu: Một nguồn thông tin rất hữu ích trong quá trình khảo sát là các siêu dữ liệu được lưu trữ bên trong các tài liệu mà người kiểm thử xâm nhập có thể thu thập từ trang web và nhân viên mục tiêu.

- Tìm kiếm DNS: Tìm kiếm danh sách các máy chủ DNS liên quan đến mục tiêu bằng cách tra cứu Whois. Xác định những hệ thống đang trực tiếp và gián tiếp liên quan đến mục tiêu. Các máy chủ DNS được liệt kê theo thứ tự máy chủ tên miền chính (primary), thứ cấp (secondary) và cấp ba (tertiary – nếu có). Các máy chủ tên miền tập trung phân giải tên miền thành địa chỉ IP, nhưng đó không phải là chức năng duy nhất.
- Công cụ tìm kiếm: Sử dụng công cụ tìm kiếm có thể truy cập công khai để tìm những dấu hiệu của các lỗ hổng trên hệ thống. Google, Yahoo và Bing của Microsoft đều có chứa một lượng lớn thông tin có thể chỉ ra sự hiện diện của các lỗ hổng trong hệ thống liên quan đến môi trường mục tiêu. Bằng cách gửi các truy vấn phù hợp với các công cụ tìm kiếm, chúng ta có thể xác định lỗ hổng hệ thống mà không thực sự gửi bất kỳ gói dữ liệu trực tiếp nào cho các hệ thống

5.1.1.2. Quét và rà soát mạng

Quét là một bước tiếp theo trong tiến trình tấn công hệ thống. Giai đoạn này giúp chúng ta xác định được nhiều thông tin của mục tiêu cần tấn công. Tức là sau khi chúng ta tìm được vài thông tin có liên quan đến máy tính cần tấn công, công đoạn tiếp theo là thu thập thông tin về máy tính đó. Những thông tin cần thu thập như tên máy, địa chỉ ip, cấu hình máy tính, hệ điều hành, dịch vụ đang chạy, port đang mở... Những thông tin này sẽ giúp cho hacker có kế hoạch tấn công, cũng như việc chọn kỹ thuật tấn công nào. Quét còn giúp định vị hệ thống còn hoạt động trên mạng hay không.

Quét được sử dụng để xác định một hệ thống có trên mạng hay không và có đang sẵn sàng hoạt động. Công đoạn quét sẽ thu thập thông tin về một hệ thống như địa chỉ IP, hệ điều hành và các dịch vụ chạy trên các máy tính mục tiêu. Có ba loại quét chủ yếu:

- Port scanning.
- Network scanning.

- Vulnerability scanning.

Đối tượng mà chúng ta đang nhắm tới chính là hệ thống máy tính với những thành phần của nó. Khi tiến hành quét hệ thống, chúng ta chú ý đến các mục đích sau: Live System, Port, Operating System, Service, IP Address

Phương pháp quét sẽ là kiểm tra xem hệ thống có tồn tại, có đang hoạt động hay không, kiểm tra các port nào đang được mở mà chúng ta có thể tương tác được, nhận biết các dịch vụ tương ứng với những port đang mở, phát họa sơ đồ mạng, đặc biệt chú ý đến những host dễ bị tổn thương, ghi dấu hệ điều hành và những thông tin có liên quan đến hệ điều hành.

5.1.1.3. Thực hiện thâm nhập

- Thực hiện kết nối và truy cập trực tiếp đến hệ thống mục tiêu.
- Thâm nhập ở mức hệ điều hành, ứng dụng, môi trường mạng.
- Thực hiện leo thang đặc quyền.

Ví dụ: bẻ khóa mật khẩu, tràn bộ đệm, từ chối dịch vụ, chèn phiên...

5.1.1.4. Duy trì kết nối

- Thực hiện sau khi chiếm quyền hệ thống.
- Sử dụng backdoor, Rookits, Trojans.
- Có thể upload, download, thực thi dữ liệu, ứng dụng, thay đổi cấu hình.
- Lợi dụng tấn công các hệ thống khác.

5.1.1.5. Xóa dấu vết

- Ẩn hoạt động thâm nhập: Rootkits, backdoors, Steganography
- Thực hiện các mục đích khác trong tương lai: Spam, từ chối dịch vụ
- Làm sai lệch thông tin nhật ký: Xóa nhật ký (máy chủ, web, FTP...)

5.1.2. Một số kiểu tấn công phổ biến vào ứng dụng Web

5.1.2.1. Injection

Sai sót trong nhập liệu, chẳng hạn như SQL injection, OS injection hay LDAP injection... Điều này xảy ra khi các thông tin sai lệch được đưa vào cùng với các biến dữ liệu đầu vào như một phần của lệnh hay câu truy vấn. Kẻ tấn

công có thể lợi dụng sơ hở này để thực hiện các lệnh không mong muốn hay truy cập các dữ liệu bất hợp pháp.

5.1.2.2. Broken Authentication and Session Management

Xác thực hay quản lý phiên thiếu chính xác. Sơ hở này cho phép kẻ tấn công có thể lợi dụng để đạt được mật khẩu, khóa hay phiên làm việc, từ đó mạo danh phiên làm việc và danh tính của người dùng thông thường.

5.1.2.3. Cross-Site Scripting (XSS)

Sai sót trong kiểm duyệt nội dung đầu vào cũng dẫn đến rủi ro này. Các dữ liệu bất hợp pháp được gửi đến trình duyệt web mà ko cần sự xác nhận thông thường. Nó cho phép kẻ tấn công thực thi các kịch bản trên trình duyệt web của nạn nhân làm thay đổi nội dung trang web, chuyển hướng nạn nhân hay đánh cắp phiên làm việc được lưu trên trình duyệt.

5.1.2.4. Insecure Direct Object References

Điều này xảy ra thì nhà phát triển cho thấy có các tham chiếu trực tiếp đến một đối tượng nội bộ hay của người dùng khác, ví dụ như một tập tin, thư mục, hay cơ sở dữ liệu quan trọng, mà ko có sự kiểm tra hay bảo vệ an toàn cần thiết. Điều này cho phép kẻ tấn công có thể truy cập các tài liệu này một cách trái phép.

5.1.2.5. Security Misconfiguration

Một hệ thống bảo mật tốt là hệ thống triển khai cho khung ứng dụng, máy chủ ứng dụng, máy chủ cơ sở dữ liệu, nền tảng... các phương pháp bảo mật cần thiết, thống nhất và liên kết với nhau. Điều này nhằm tránh những nguy cơ bị khai thác vào ứng dụng, ví dụ để lộ ra những thông tin quan trọng khi trao đổi các gói tin.

5.1.2.6. Sensitive Data Exposure

Các dữ liệu nhạy cảm không được lưu trữ và bảo vệ cẩn thận, dẫn đến khi bị kẻ tấn công khai thác gây ra những ảnh hưởng to lớn cho hệ thống máy chủ,

doanh nghiệp, client. Ví dụ như việc lưu trữ thẻ tín dụng mà không thông qua các khâu mã hóa, hay các gói tin TLS bị bẻ khóa và nghe lén thông qua lỗ hổng CRIME.

5.1.2.7. Missing Function Level Access Control

Thiếu các điều khoản trong việc phân quyền quản trị các mức, dẫn đến việc kẻ tấn công có thể lợi dụng và truy ra các điểm yếu trên hệ thống, hay lợi dụng để leo thang đặc quyền.

5.1.2.8. Cross-Site Request Forgery (CSRF)

Lợi dụng sơ hở của nạn nhân, kẻ tấn công có thể lừa nạn nhân thực hiện các hành động nguy hiểm mà nạn nhân không hề hay biết, ví dụ như chuyển tiền từ tài khoản nạn nhân sang tài khoản kẻ tấn công, thông qua các lỗ hổng XSS.

5.1.2.9. Using Known Vulnerable Components

Sử dụng các thư viện, plugin, module... có chứa các lỗ hổng đã được công khai, dễ dàng dẫn đến việc bị kẻ tấn công lợi dụng để tấn công vào hệ thống một cách nhanh chóng.

5.1.2.10. Unvalidated Redirects and Forwards

Chuyển hướng không an toàn người dùng đến một đường dẫn bên ngoài có thể bị kẻ tấn công lợi dụng để chuyển hướng nạn nhân đến một trang đích được chuẩn bị sẵn của kẻ tấn công.

5.2. WEB LOG FILE

5.2.1. Web log file format

Các web server như Apache và IIS thường tạo ra các thông điệp đăng nhập (logging message) mặc định trong đặc tả Common Log Format (CLF). Các tập tin CLF log chứa một dòng riêng biệt cho mỗi yêu cầu HTTP. Một dòng gồm nhiều thẻ ngăn cách bởi khoảng trắng, nó mang những thông tin sau:

- Host: Chứa tên miền đầy đủ của client, hoặc địa chỉ IP của nó.

- Ident: Nếu chỉ thị IdentityCheck được kích hoạt và các máy client chạy identd, thì đây là thông tin nhận dạng báo cáo của các client.
- Authuser: Nếu các URL được yêu cầu cần xác thực thành công Basic HTTP, sau đó tên người sử dụng là giá trị của ký hiệu này.
- Date: Ngày và thời gian yêu cầu.
- Request: Dòng yêu cầu từ client, được đặt trong dấu ngoặc kép (“”).
- Status: Các mã trạng thái HTTP gồm 3 chữ số được trả lại cho client.
- Byte: Số byte trong đối tượng trả lại cho client, bao gồm tất cả các HTTP header.

Ví dụ:

```
127.0.0.1 - frank [10/Oct/2007:13:55:36 -0700] "GET /index.html
HTTP/1.0" 200 2326 "http://www.example.com/links.html" "Mozilla/4.0
(compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322)"
```

Trong đó:

- 127.0.0.1: địa chỉ IP của client.
- frank: đây là userid của người yêu cầu.
- [10 / Oct / 2007: 13: 55: 36 0700]: Thời gian mà các máy chủ xử lý xong yêu cầu.
- "GET /index.html HTTP / 1.0": Dòng yêu cầu từ client được đặt trong dấu ngoặc kép.
- 200: Đây là mã trạng thái mà các máy chủ sẽ gửi lại cho client.
- 2326: kích thước của đối tượng trả lại cho client, không bao gồm các response header.
- "http://www.example.com/links.html": "Referer" (sic) HTTP request header.
- "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322)": User-Agent HTTP request header.

5.2.2. HTTP Status Code

Trước tiên chúng ta sẽ tìm hiểu một chút về HTTP Response. Khi nhận và phiên dịch một HTTP Request, Server sẽ gửi tín hiệu phản hồi là một HTTP Response bao gồm các thành phần sau:

- Một dòng trạng thái (Status-Line)
- Không hoặc nhiều hơn các trường Header (General|Response|Entity) được theo sau CRLF
- Một dòng trống chỉ dòng kết thúc của các trường Header
- Một phần thân thông báo tùy ý

Dưới đây là một ví dụ về một HTTP Response:

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
Content-Length: 88
Content-Type: text/html
Connection: Closed

<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Bây giờ chúng ta sẽ tập trung chính vào dòng Status-Line: Một dòng Status Line bao gồm phiên bản giao thức (HTTP-Version) sau đó là mã hóa trạng thái số (Status-Code) và cụm từ thuần văn bản được liên kết của nó. Các thành phần được phân biệt bởi dấu cách

Status-Line = HTTP-Version <Space> Status Code <Space> Reason-Phrase
CRLF

Ví dụ:

```
HTTP/1.1 404 Not Found
HTTP-Version: HTTP/1.1
Status-Code: 404
Reason-Phrase: Not Found
```

Status code (Mã hóa trạng thái thường được gọi là mã trạng thái) là một số nguyên 3 ký tự, trong đó ký tự đầu tiên của Status-Code định nghĩa loại Response và hai ký tự cuối không có bất cứ vai trò phân loại nào. Có 5 giá trị của ký tự đầu tiên:

- 1xx: Information (Thông tin): Khi nhận được những mã như vậy tức là request đã được server tiếp nhận và quá trình xử lý request đang được tiếp tục.
- 2xx: Success (Thành công): Khi nhận được những mã như vậy tức là request đã được server tiếp nhận, hiểu và xử lý thành công
- 3xx: Redirection (Chuyển hướng): Mã trạng thái này cho biết client cần có thêm action để hoàn thành request
- 4xx: Client Error (Lỗi Client): Nó nghĩa là request chứa cú pháp không chính xác hoặc không được thực hiện.
- 5xx: Server Error (Lỗi Server): Nó nghĩa là Server thất bại với việc thực hiện một request nhìn như có vẻ khả thi.

Status-Code HTTP là có thể co giãn và ứng dụng HTTP không được yêu cầu để hiểu ý nghĩa của tất cả các mã trạng thái được đăng ký.

5.2.2.1. 1xx Information (Thông tin)

- 100 Continue: Chỉ một phần của Request được nhận bởi Server (có thể là header và Client cần gửi tiếp body), nhưng miễn là nó không bị loại bỏ, Client nên tiếp tục với Request.
- 101 Switching Protocols: Requester đã hỏi Server về việc thanh đổi Protocol và Server đã chấp nhận điều đó

5.2.2.2. 2xx Success (Thành công)

- 200 OK: Request đã được tiếp nhận và xử lý thành công. Các Response thực tế trả về sẽ phụ thuộc vào phương thức HTTP của Request. Trong một GET Request, Response sẽ chứa một thực thể tương ứng với các tài nguyên yêu cầu, trong một POST Request, Response sẽ chứa một thực thể mô tả hoặc chứa các kết quả của các action.

- 201 Created: Request được chấp nhận cho xử lý, nhưng việc xử lý chưa hoàn thành.
- 203 Non-authoritative Information (Xuất hiện từ HTTP/1.1): Server là nơi chuyển đổi proxy (ví dụ một Web accelerator) đã nhận được 200 OK nhưng nó trả về một phiên bản thay đổi (có thể là header) của Response nguyên gốc.
- 204 No Content: Server đã xử lý thành công request nhưng không trả về bất cứ content nào.
- 205 Reset Content: Server đã xử lý thành công request nhưng không trả về bất cứ content nào. Không giống với 204 No Content Response này yêu cầu phía Client phải thiết lập lại document view.
- 206 Partial Content: Server chỉ trả về một phần của resource(dạng byte) do một range header được gửi bởi phía Client. Các Range Header được sử dụng bởi Client để cho phép nối lại các phần của file download bị gián đoạn hoặc chia thành nhiều luồng download.

5.2.2.2. 3xx Redirection (Sự chuyển hướng lại)

- 300 Multiple Choices: Một danh sách các link. Người sử dụng có thể chọn một link và tới vị trí đó. Tối đa 5 địa chỉ. Ví dụ: List các file video với format khác nhau
- 301 Moved Permanently: Request hiện tại và các request sau được yêu cầu di chuyển tới một URI mới.
- 302 Found: Đây là một ví dụ cho thấy sự mâu thuẫn giữa thực tiễn và quy chuẩn. Ở phiên bản HTTP/1.0 nó có nghĩa là yêu cầu Client chuyển hướng đến một URL tạm thời (tương tự như là 301 Moved Permanently) nhưng phần lớn các browser lại thực hiện nó với ý nghĩa của 303 See Other(sẽ nói sau đây). Do đó từ phiên bản HTTP/1.1 có thêm hai mã 303 và 307 để phân biệt rõ hành vi, nhưng một số ứng dụng web và framework vẫn sử dụng 302 như thể là 303.

- 303 See Other (Xuất hiện từ HTTP/1.1): Response trả về của Request có thể tìm thấy ở một URL khác bằng cách sử dụng phương thức GET.
- 304 Not Modified: Đây là Status-Code tới một If-Modified-Since hoặc If-None-Match header, nơi mà URL không được chỉnh sửa từ ngày cụ thể.
- 305 Use Proxy: Tài nguyên yêu cầu chỉ có sẵn thông qua một proxy, địa chỉ mà được cung cấp trong các Response. Nhiều HTTP Client (như Mozilla và Internet Explorer) không xử lý một cách chính xác phản ứng với mã trạng thái này, chủ yếu là vì các lý do an ninh.
- 306 Switch Proxy: Mã này hiện không còn được sử dụng, ý nghĩa ban đầu của nó là "Các Request tiếp theo nên sử dụng các proxy được chỉ định".
- 307 Temporary Redirect (xuất hiện từ HTTP/1.1): Trong trường hợp này, Request hiện tại cần được lập lại một URI khác nhưng các Request trong tương lai vẫn sử dụng URI gốc.

5.2.2.4. 4xx: Client Error (Lỗi Client)

- 400 Bad Request: Server không thể xử lý hoặc sẽ không xử lý các Request lỗi của phía client (ví dụ Request có cú pháp sai hoặc Request lừa đảo định tuyến ...)
- 401 Unauthorized: Tương tự như 403 Forbidden nhưng được sử dụng khi yêu cầu xác thực là bắt buộc và đã không thành công. Các Response bắt buộc phải có thành phần WWW-Authenticate chứa các thách thức với tài nguyên được yêu cầu. Một số trang web vẫn đề HTTP 401 khi một địa chỉ IP bị cấm từ các trang web (thường là các tên miền trang web) và địa chỉ cụ thể là từ chối quyền truy cập một trang web.
- 402 Payment Required: Hiện tại mã này chưa được sử dụng và nó được dự trữ cho tương lai. Mục đích ban đầu là mã này có thể được sử dụng như là một phần của đề án tiền mặt hoặc micropayment kỹ thuật số, nhưng điều đó đã không xảy ra, và mã này thường không được sử dụng.

Google API sử dụng Status-Code này nếu một nhà phát triển đặc biệt đã vượt quá giới hạn số lần yêu cầu.

- 403 Forbidden: Request là hợp lệ nhưng server từ chối đáp ứng nó. Nó có nghĩa là trái phép, người dùng không có quyền cần thiết để tiếp cận với các tài nguyên.
- 404 Not Found: Các tài nguyên hiện tại không được tìm thấy nhưng có thể có trong tương lai. Các request tiếp theo của Client được chấp nhận.
- 405 Method Not Allowed: Request method không được hỗ trợ cho các tài nguyên được yêu cầu. Ví dụ Một GET request đến một POST resource, PUT Request gọi đến một tài nguyên chỉ đọc.
- 406 Not Acceptable: Server chỉ có thể tạo một Response mà không được chấp nhận bởi Client.
- 407 Proxy Authentication Required: Bạn phải xác nhận với một Server ủy quyền trước khi Request này được phục vụ.
- 408 Request Timeout: Request tốn thời gian dài hơn thời gian Server được chuẩn bị để đợi.
- 409 Conflict: Request không thể được hoàn thành bởi vì sự xung đột, ví dụ như là xung đột giữa nhiều chỉnh sửa đồng thời.
- 410 Gone: Các resource được yêu cầu không còn nữa và sẽ không có sẵn một lần nữa, khi gặp mã lỗi này Client không nên có gắng tìm kiếm các tài nguyên này ở những lần sau.
- 411 Length Required: Content-Length không được xác định rõ. Server sẽ không chấp nhận Request nào không có nó.
- 412 Precondition Failed: Server sẽ không đáp ứng một trong những điều kiện tiên quyết của Client trong Request.
- 413 Payload Too Large: Server sẽ không chấp nhận yêu cầu, bởi vì đối tượng yêu cầu là quá rộng. Trước đây nó gọi là "Request Entity Too Large".

- 414 URI Too Long: URI được cung cấp là quá dài để Server xử lý, thường là kết quả của quá nhiều dữ liệu được mã hóa như là một truy vấn chuỗi của một GET Request, trong trường hợp đó nó phải được chuyển đổi sang một POST Request. Trước đây được gọi là "Request-URI Too Long"
- 415 Unsupported Media Type: Server sẽ không chấp nhận Request, bởi vì kiểu phương tiện không được hỗ trợ. Ví dụ khi Client upload một ảnh có định dạng image/svg+xml, nhưng server yêu cầu một định dạng khác.
- 416 Range Not Satisfiable: Client yêu cầu một phần của tập tin nhưng server không thể cung cấp nó. Trước đây được gọi là "Requested Range Not Satisfiable"
- 417 Expectation Failed: Máy chủ không thể đáp ứng các yêu cầu của trường Expect trong header.

5.2.2.5. 5xx: Server Error (Lỗi Server)

- 500 Internal Server Error: Một thông báo chung chung, được đưa ra khi Server gặp phải một trường hợp bất ngờ, Message cụ thể là không phù hợp.
- 501 Not Implemented: Server không công nhận các Request method hoặc không có khả năng xử lý nó.
- 502 Bad Gateway: Server đã hoạt động như một gateway hoặc proxy và nhận được một Response không hợp lệ từ máy chủ nguồn.
- 503 Service Unavailable: Server hiện tại không có sẵn (Quá tải hoặc được down để bảo trì). Nói chung đây chỉ là trạng thái tạm thời.
- 504 Gateway Timeout: Server đã hoạt động như một gateway hoặc proxy và không nhận được một Response từ máy chủ nguồn.
- 505 HTTP Version Not Supported: Server không hỗ trợ phiên bản “giao thức HTTP”.

Trên đây là danh sách các Status Code phổ biến, ngoài ra còn nhiều các Status-Code tiêu chuẩn đã được nhưng nghĩa như: 418, 421, 506, 507, 508, 511...

5.3. NHẬN DIỆN TẤN CÔNG

5.3.1. Biểu thức chính quy

5.3.1.1. Định nghĩa

Biểu thức chính quy (regular expression, viết tắt là regexp, regex hay regxp) là một chuỗi miêu tả một bộ các chuỗi khác, theo những quy tắc cú pháp nhất định. Biểu thức chính quy thường được dùng trong các trình biên tập văn bản và các tiện ích tìm kiếm và xử lý văn bản dựa trên các mẫu được quy định. Nhiều ngôn ngữ lập trình cũng hỗ trợ biểu thức chính quy trong việc xử lý chuỗi, chẳng hạn như C#, Perl có bộ máy mạnh mẽ để xử lý biểu thức chính quy được xây dựng trực tiếp trong cú pháp của chúng. Bộ các trình tiện ích (gồm trình biên tập sed và trình lọc grep) đi kèm các bản phân phối Unix có vai trò đầu tiên trong việc phổ biến khái niệm biểu thức chính quy.

Thuật ngữ regular expression xuất phát từ lý thuyết toán học và khoa học máy tính, nó phản ánh một đặc điểm của các biểu thức toán học được gọi là chính quy (regularity). Một biểu thức có thể được thực hiện trong một phần mềm bằng cách sử dụng một bộ xác định giới hạn tự động (Deterministic Finite Automation – DFA). DFA là một trạng thái xác định và không sử dụng cơ chế quay lui (backtracking).

Nếu bạn sử dụng tốt những kỹ năng về regular expression. Chúng sẽ đơn giản hơn nhiều trong lập trình và quá trình xử lý văn bản, và có những vấn đề sẽ không thể giải quyết được nếu không sử dụng regular expression. Bạn sẽ cần đến hàng trăm thủ tục để trích xuất tất cả các địa chỉ email từ một số tài liệu, đây có thể nói là một việc làm tẻ nhạt và vất vả. Nhưng với regular expression bạn chỉ cần một số dòng lệnh hoặc thậm chí một dòng lệnh để làm việc này.

Regular expression là một công cụ mạnh mẽ trong việc thao tác và trích xuất văn bản trên máy tính. Do đó nắm vững các biểu thức chính quy sẽ giúp bạn tiết kiệm nhiều thời gian và công sức.

5.3.1.2. Cách viết một mẫu biểu thức chính quy

Một mẫu biểu thức chính quy là một tập các kí tự thường, như `/abc/`, hay một tập kết hợp cả kí tự thường và kí tự đặc biệt như `/ab*c/` hoặc `/Chapter` hoặc `(\d+)\.\d*/`

Các mẫu đơn giản là các mẫu có thể được xây dựng từ các kí tự có thể tìm kiếm một cách trực tiếp. Ví dụ, mẫu `/abc/` sẽ tìm các đoạn 'abc' theo đúng thứ tự đó trong các chuỗi. Mẫu này sẽ khớp được với "Hi, do you know your abc's?" và "The latest airplane designs evolved from slabcraft.", vì cả hai chuỗi này đều chứa đoạn 'abc'. Còn với chuỗi 'Grab crab', nó sẽ không khớp vì chuỗi này không chứa 'abc' theo đúng thứ tự, mà chỉ chứa 'ab c'.

Các mẫu có thể chứa các kí tự đặc biệt cho các mục đích tìm kiếm nâng cao mà tìm kiếm trực tiếp sẽ khó khăn như tìm một đoạn chứa một hoặc nhiều hơn một kí tự b, hay tìm một hoặc nhiều kí tự dấu cách (while space). Ví dụ, mẫu `/ab*c/` có thể tìm các đoạn có chứa: một kí tự 'a', theo sau là không có hoặc có một hoặc có nhiều kí tự 'b', cuối cùng là một kí tự 'c' như chuỗi "cbbabbbbbcdebc," sẽ được khớp với xâu con 'abbbbc'.

Kí tự	Ý nghĩa
\	<p>Tìm với luật dưới đây:</p> <p>Một dấu gạch chéo ngược sẽ biến một kí tự thường liền kề phía sau thành một kí tự đặc biệt, tức là nó không được sử dụng để tìm kiếm thông thường nữa. Ví dụ, trường hợp kí tự 'b' không có dấu gạch chéo ngược này sẽ được khớp với các kí tự 'b' in thường, nhưng khi nó có thêm dấu gạch chéo ngược, '\b' thì nó sẽ không khớp với bất kì kí tự nào nữa, lúc này nó trở thành kí tự đặc biệt.</p> <p>Tuy nhiên nếu đứng trước một kí tự đặc biệt thì nó sẽ biến kí tự này thành một kí tự thường, tức là bạn có thể tìm kiếm kí tự đặc biệt này trong xâu chuỗi của bạn như các kí tự thường khác. Ví dụ, mẫu <code>/a*/</code> có '*' là kí tự đặc biệt và mẫu này sẽ bị phụ thuộc vào kí tự</p>

Kí tự	Ý nghĩa
	<p>này, nên được hiểu là sẽ tìm khớp với 0 hoặc nhiều kí tự a. Nhưng, với mẫu /a*/ thì kí tự '*' lúc này được hiểu là kí tự thường nên mẫu này sẽ tìm kiếm sâu con là 'a*'. Đừng quên \ cũng là một kí tự đặc biệt, khi cần so khớp chính nó ta cũng phải đánh dấu nó là kí tự đặc biệt bằng cách đặt \ ở trước.</p>
^	<p>Khớp các kí tự đứng đầu một chuỗi. Nếu có nhiều cò này thì nó còn khớp được cả các kí tự đứng đầu của mỗi dòng (sau kí tự xuống dòng).</p> <p>Ví dụ, /^A/ sẽ không khớp được với 'A' trong "an A" vì 'A' lúc này không đứng đầu chuỗi, nhưng nó sẽ khớp "An E" vì lúc này 'A' đã đứng đầu chuỗi.</p> <p>Ý nghĩa của '^' sẽ thay đổi khi nó xuất hiện như một kí tự đầu tiên trong một lớp kí tự.</p>
\$	<p>So khớp ở cuối chuỗi. Nếu gắn cò multiline (đa dòng), nó sẽ khớp ngay lập tức trước kí tự xuống dòng.</p> <p>Ví dụ, /t\$/ không khớp với 't' trong chuỗi "eater" nhưng lại khớp trong chuỗi "eat".</p>
*	<p>Cho phép kí tự trước nó lặp lại 0 lần hoặc vô số lần. Tương đương với cách viết {0,}.</p> <p>Ví dụ, /bo*/ khớp với 'boooo' trong chuỗi "A ghost boooed" nhưng không khớp trong chuỗi "A birth warbled".</p>
+	<p>Cho phép kí tự trước nó lặp lại một lần hoặc vô số lần. Tương đương với cách viết {1,}.</p> <p>Ví dụ, /a+/ khớp với 'a' trong chuỗi "candy" và khớp với tất cả kí tự a liền nhau trong chuỗi "caaaaaaandy".</p>
?	<p>Cho phép kí tự trước nó lặp lại 0 lần hoặc một lần duy nhất. Tương đương với cách viết {0,1}.</p> <p>Ví dụ, /e?le?/ khớp với 'el' trong chuỗi "angel" và 'le' trong chuỗi "angle" hay 'l' trong "oslo".</p>

Kí tự	Ý nghĩa
	Nếu sử dụng kí tự này ngay sau bất kì kí tự định lượng nào trong số *,+,? hay {}, đều làm bộ định lượng "chán ăn" (dừng so khớp sau ngay khi tìm được kí tự phù hợp), trái ngược với đức tính "tham lam" vốn sẵn của chúng (khớp tất cả kí tự chúng tìm thấy). Ví dụ, áp dụng biểu mẫu <code>\d+</code> cho <code>"123abc"</code> ta được <code>"123"</code> . Nhưng áp <code>\d+?</code> cho chính chuỗi trên ta chỉ nhận được kết quả là <code>"1"</code> .
.	Dấu . khớp với bất kì kí tự đơn nào ngoại trừ kí tự xuống dòng. Ví dụ, <code>/n/</code> khớp với <code>'an'</code> và <code>'on'</code> trong chuỗi <code>"no, an apple is on the tree"</code> , nhưng không khớp với <code>'no'</code> .
(x)	Khớp <code>'x'</code> và nhớ kết quả so khớp này, như ví dụ ở dưới. Các dấu ngoặc tròn được gọi là các dấu ngoặc có nhớ. Biểu mẫu <code>/(foo) (bar) \1 \2/</code> khớp với <code>'foo'</code> và <code>'bar'</code> trong chuỗi <code>"foo bar foo bar"</code> . <code>\1</code> và <code>\2</code> trong mẫu khớp với hai từ cuối. Chú ý rằng <code>\1</code> , <code>\2</code> , <code>\n</code> được sử dụng để so khớp các phần trong regex, nó đại diện cho nhóm so khớp đứng trước. Ví dụ: <code>/(foo) (bar) \1 \2/</code> tương đương với biểu thức <code>/(foo) (bar) foo bar/</code> . Cú pháp <code>\$1</code> , <code>\$2</code> , <code>\$n</code> còn được sử dụng trong việc thay thế các phần của một regex. Ví dụ: <code>'bar foo'.replace(/(...) (...)/, '\$2 \$1')</code> sẽ đảo vị trí hai từ <code>'bar'</code> và <code>'foo'</code> cho nhau.
(?:x)	Khớp <code>'x'</code> nhưng không nhớ kết quả so khớp. Những dấu ngoặc tròn được gọi là những dấu ngoặc không nhớ, nó cho phép bạn định nghĩa những biểu thức con cho những toán tử so khớp. Xem xét biểu thức đơn giản <code>/(?:foo){1,2}/</code> . Nếu biểu thức này được viết là <code>/foo{1,2}/</code> , <code>{1,2}</code> sẽ chỉ áp dụng cho kí tự <code>'o'</code> ở cuối chuỗi <code>'foo'</code> . Với những dấu ngoặc không nhớ, <code>{1,2}</code> sẽ áp dụng cho cả cụm <code>'foo'</code> .
x(?:=y)	Chỉ khớp <code>'x'</code> nếu <code>'x'</code> theo sau bởi <code>'y'</code> . Ví dụ, <code>/Jack(?:=Sprat)/</code> chỉ khớp với <code>'Jack'</code> nếu đằng sau nó là <code>'Sprat'</code> . <code>/Jack(?:=Sprat Frost)/</code> chỉ khớp <code>'Jack'</code> nếu theo sau nó là <code>'Sprat'</code> hoặc <code>'Frost'</code> . Tuy nhiên, cả <code>'Sprat'</code> và <code>'Frost'</code> đều không phải là một phần của kết quả so khớp trả về.
x(?:!y)	Chỉ khớp <code>'x'</code> nếu <code>'x'</code> không được theo sau bởi <code>'y'</code> . Ví dụ: <code>\d+(?!\.)</code> chỉ khớp với số không có dấu . đằng sau. Biểu thức <code>\d+(?!\.)</code> .exec(<code>"3.141"</code>) cho kết quả là <code>'141'</code> mà không phải <code>'3.141'</code> .

Kí tự	Ý nghĩa
x y	Khớp 'x' hoặc 'y' Ví dụ, /green red/ khớp với 'green' trong chuỗi "green apple" và 'red' trong chuỗi "red apple".
{n}	Kí tự đứng trước phải xuất hiện n lần. n phải là một số nguyên dương. Ví dụ, /a{2}/ không khớp với 'a' trong "candy", nhưng nó khớp với tất cả kí tự 'a' trong "caandy", và khớp với hai kí tự 'a' đầu tiên trong "caandy".
{n,m}	Kí tự đứng trước phải xuất hiện từ n đến m lần. n và m là số nguyên dương và $n \leq m$. Nếu m bị bỏ qua, nó tương đương như ∞ . Ví dụ, /a{1,3}/ không khớp bất kì kí tự nào trong "cndy", kí tự 'a' trong "candy", hai kí tự 'a' đầu tiên trong "caandy", và 3 kí tự 'a' đầu tiên trong "caaaaaandy". Lưu ý là "caaaaaandy" chỉ khớp với 3 kí tự 'a' đầu tiên mặc dù chuỗi đó chứa 7 kí tự 'a'.
[xyz]	Lớp kí tự. Loại mẫu này dùng để so khớp với một kí tự bất kì trong dấu ngoặc vuông, bao gồm cả <u>escape sequences</u> . Trong lớp kí tự, dấu chấm (.) và dấu hoa thị (*) không còn là kí tự đặc biệt nên ta không cần kí tự thoát đứng trước nó. Bạn có thể chỉ định một khoảng kí tự bằng cách sử dụng một kí tự gạch nối (-) như trong ví dụ dưới đây: Mẫu [a-d] so khớp tương tự như mẫu [abcd], khớp với 'b' trong "brisket" và 'c' trong "city". Mẫu /[a-z.]+/ và /[w.]+/ khớp với toàn chuỗi "test.i.ng".
[^xyz]	Lớp kí tự phủ định. Khi kí tự ^ đứng đầu tiên trong dấu ngoặc vuông, nó phủ định mẫu này. Ví dụ, [^abc] tương tự như [^a-c], khớp với 'r' trong "brisket" và 'h' trong "chop" là kí tự đầu tiên không thuộc khoảng a đến c.
[\\b]	Khớp với kí tự dịch lùi - backspace (U+0008). Bạn phải đặt trong dấu ngoặc vuông nếu muốn so khớp một kí tự dịch lùi. (Đừng nhầm lẫn với mẫu \\b).
\\b	Khớp với kí tự biên. Kí tự biên là một kí tự giả, nó khớp với vị trí mà một kí tự không được theo sau hoặc đứng trước bởi một kí tự khác. Tương đương với mẫu (^\\w \\w\$ \\W\\w \\w\\W). Lưu ý rằng một kí tự biên được khớp sẽ không bao gồm trong kết quả so khớp. Nói

Kí tự	Ý nghĩa
	<p>cách khác, độ dài của một kí tự biên là 0. (Đừng nhầm lẫn với mẫu <code>[\b]</code>)</p> <p>Ví dụ:</p> <p><code>^bm/</code> khớp với 'm' trong chuỗi "moon";</p> <p><code>/oo\b/</code> không khớp 'oo' trong chuỗi "moon", bởi vì 'oo' được theo sau bởi kí tự 'n';</p> <p><code>/oon\b/</code> khớp với 'oon' trong chuỗi "moon", bởi vì 'oon' ở cuối chuỗi nên nó không được theo sau bởi một kí tự;</p> <p><code>^w\b/w/</code> sẽ không khớp với bất kì thứ gì, bởi vì một kí tự không thể theo sau một kí tự biên và một kí tự thường.</p>
<code>\B</code>	<p>Khớp với kí tự không phải kí tự biên. Mẫu này khớp tại vị trí mà kí tự trước và kí tự sau nó cùng kiểu: hoặc cả hai là kí tự hoặc cả hai không phải là kí tự. Bắt đầu và kết thúc chuỗi không được xem là những kí tự.</p> <p>Ví dụ, <code>^B./</code> khớp với 'oo' trong "noonday", và <code>/y\B./</code> khớp với 'ye' trong "possibly yesterday."</p>
<code>\cX</code>	<p>X là một kí tự trong khoảng A tới Z. Mẫu này khớp với một kí tự điều khiển trong một chuỗi.</p> <p>Ví dụ: <code>^cM/</code> khớp với control-M (U+000D) trong chuỗi.</p>
<code>\d</code>	<p>Khớp với một kí tự số. Tương đương với mẫu <code>[0-9]</code>.</p> <p>Ví dụ: <code>^d/</code> hoặc <code>/[0-9]/</code> khớp với '2' trong chuỗi "B2 is the suite number."</p>
<code>\D</code>	<p>Khớp với một kí tự không phải là kí tự số. Tương đương với mẫu <code>[^0-9]</code>.</p> <p>Ví dụ; <code>^D/</code> hoặc <code>/[^0-9]/</code> khớp với 'B' trong "B2 is the suite number."</p>
<code>\f</code>	Matches a form feed (U+000C).
<code>\n</code>	Matches a line feed (U+000A).
<code>\r</code>	Matches a carriage return (U+000D).
<code>\s</code>	<p>Khớp với một kí tự khoảng trắng, bao gồm space, tab, form feed, line feed.</p> <p>Ví dụ: <code>^s\w*/</code> khớp với ' bar' trong "foo bar."</p>
<code>\S</code>	Khớp với một kí tự không phải khoảng trắng.

Kí tự	Ý nghĩa
	Ví dụ: <code>\S\w*</code> / khớp với 'foo' trong chuỗi "foo bar."
<code>\t</code>	Matches a tab (U+0009).
<code>\v</code>	Matches a vertical tab (U+000B).
<code>\w</code>	Khớp với tất cả kí tự là chữ, số và gạch dưới. Tương đương với mẫu <code>[A-Za-z0-9_]</code> . ví dụ, <code>\w/</code> khớp với 'a' trong "apple," '5' trong "\$5.28," và '3' trong "3D."
<code>\W</code>	Khớp với tất cả kí tự không phải là chữ. Tương đương với mẫu <code>[^A-Za-z0-9_]</code> . ví dụ, <code>\W/</code> hoặc <code>/[^A-Za-z0-9_]/</code> khớp với '%' trong "50%."
<code>\n</code>	Trong đó, n là một số nguyên dương, một tham chiếu ngược tới chuỗi khớp thứ n trong biểu thức (đếm từ trái sang, bắt đầu bằng 1). Ví dụ: <code>/apple(,)\sorange\1/</code> hay <code>/apple(,)\sorange,/</code> khớp với 'apple, orange,' trong chuỗi "apple, orange, cherry, peach."

Ngoặc tròn bao quanh bất kỳ phần nào của biểu thức chính quy sẽ khiến phần kết quả so khớp được nhớ. Mỗi lần nhớ, chuỗi con có thể được gọi lại để sử dụng.

Ví dụ, mẫu `/Chapter (\d+)\.\d*/` khớp đúng với 'Chapter ' theo sau bởi một hoặc nhiều kí tự số, sau nữa là một dấu chấm thập phân, cuối cùng có thể là 0 hoặc nhiều kí tự số. Bên cạnh đó, dấu ngoặc tròn được sử dụng để nhớ một hoặc nhiều kí tự số đầu tiên được khớp.

Mẫu này được tìm thấy trong chuỗi "Open Chapter 4.3, paragraph 6", nhớ '4' nhưng không được tìm thấy trong chuỗi "Chapter 3 and 4", bởi vì chuỗi đó không có dấu chấm sau kí tự số '3'.

Để so khớp một chuỗi con không nhớ, đặt `?:` ở vị trí đầu tiên trong ngoặc. Ví dụ, `(?:\d+)` khớp với một hoặc nhiều kí tự số nhưng không nhớ kết quả so khớp.

5.3.1.3. Làm việc với biểu thức chính quy

Biểu thức chính quy được sử dụng với phương thức `test` và `exec` của lớp `RegExp` hoặc phương thức `match`, `replace`, `search` và `split` của chuỗi.

Phương thức	Mô tả
exec	Một phương thức của RegExp dùng để tìm kiếm chuỗi phù hợp với mẫu so khớp. Nó trả về một mảng chứa kết quả tìm kiếm.
test	Một phương thức của RegExp dùng để kiểm tra mẫu có khớp với chuỗi hay không. Nó trả về giá trị true hoặc false.
match	Một phương thức của chuỗi dùng để tìm kiếm chuỗi phù hợp với mẫu so khớp. Nó trả về một mảng chứa kết quả tìm kiếm hoặc null nếu không tìm thấy.
search	Một phương thức của chuỗi dùng để tìm kiếm chuỗi phù hợp với mẫu so khớp và trả về vị trí của chuỗi đó hoặc -1 nếu không tìm thấy.
replace	Một phương thức của chuỗi dùng để tìm kiếm một chuỗi theo mẫu so khớp và thay thế chuỗi con được khớp với một chuỗi thay thế.
split	Một phương thức của chuỗi dùng một biểu mẫu chính quy hoặc một chuỗi bất biến để ngắt chuỗi đó thành một mảng các chuỗi con.

5.3.1.4. Ví dụ về biểu thức chính quy

Kiểm tra email có hợp lệ: `[a-zA-Z0-9_\.]+\@[a-zA-Z]+\.[a-zA-Z]+(\.[a-zA-Z]+)*`

- Đầu tiên ta kiểm tra phần tên email, tên email bao gồm kí tự thường, in hoa, chữ số, dấu _ và .. Vì vậy nên ta phải gom tất cả vào một nhóm `[a-zA-Z_\.]` (bởi vì giữa kí tự thường và in hoa có các kí tự đặc biệt nên ta không thể dùng `A-z` mà phải tách ra `a-zA-Z`). Tuy nhiên ta cần phải thêm dấu `+` vì nhóm đó chỉ đại diện cho một kí tự duy nhất.
- Kí tự `@` là luôn luôn có
- Phần tên miền, ở đây mình quy định tên miền chỉ bao gồm các kí tự thường và hoa, có dấu . phân cách. Vì vậy nên ta ghi vào `[a-zA-Z]+\.[a-zA-Z]+`. Tuy nhiên nó chỉ đúng với vài tên miền dạng như `gmail.com`, `yahoo.com`, `live.com`,... nhưng sẽ không đúng với những tên miền dạng như `yahoo.com.vn`, `dauan.com.xyz.vn`,... chẳng hạn. Vì vậy ta cần phải

thêm vào một group bao gồm dấu . và các kí tự chữ (\.[a-zA-Z]+). Group này có thể có cũng được, không có cũng được nên ta đặt dấu * ngay đuôi (khớp với 0 lần trở lên).

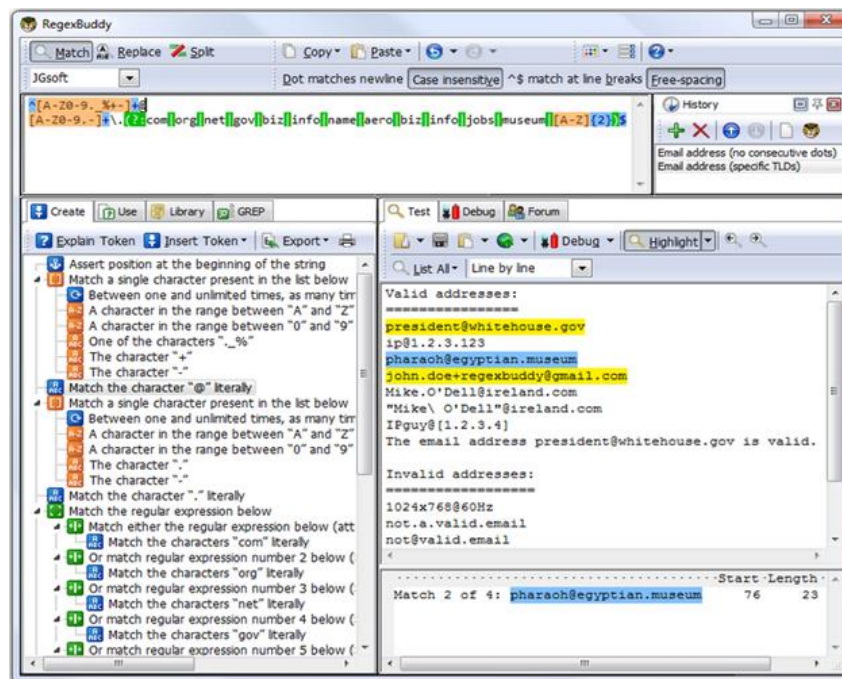
Kiểm tra số điện thoại hợp lệ: (\\+84|0)\\d{9,10}

- Đầu tiên, do số điện thoại có phần mở đầu có thể là +84 (ở Việt Nam) hoặc là 0, nên ta cần đặt vào trong group và thêm dấu |
- Kế tiếp, một số điện thoại bao gồm 10 hoặc 11 chữ số, nhưng ta không tính phần đầu của số điện thoại nên chỉ còn khoảng 9 – 10 chữ số

5.3.1.5. Một số công cụ làm việc với biểu thức chính quy

- RegexBuddy:

RegexBuddy là một công cụ đầy đủ các tính năng nhất hiện nay để tạo lập, kiểm tra và thực thi các biểu thức chính quy. Nó có bộ biểu thức chính quy cho những ngôn ngữ lập trình khác nhau như: .NET, Java,... Và cho phép chuyển đổi biểu thức chính quy giữa các ngôn ngữ lập trình.

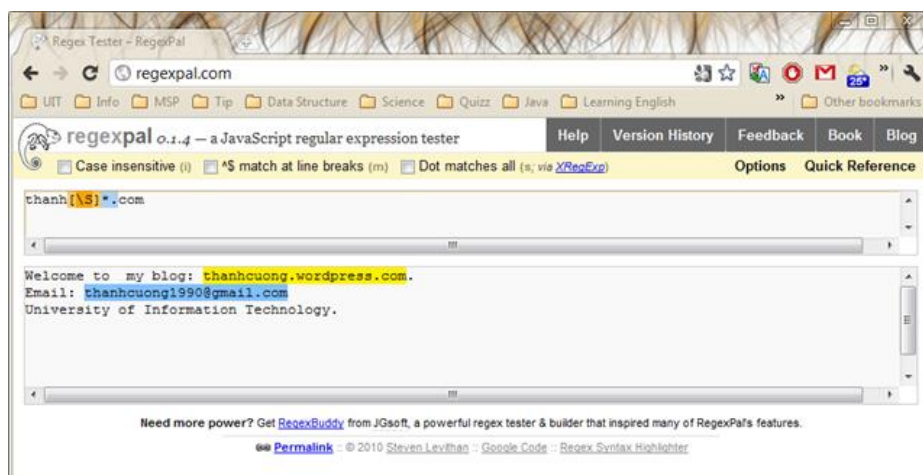


RegexBuddy được thiết kế bởi Jan Goyvaerts (là một chuyên gia về Regular Expression).

RegexBuddy sẽ tự động tô sáng những phần phù hợp với biểu thức chính quy của bạn, và thông báo những lỗi xuất hiện trong biểu thức chính quy (nếu có).

- RegexPal:

RegexPal là một trình kiểm tra biểu thức chính quy online được tạo bởi tác giả Steven Levithan. Bạn chỉ cần một trình duyệt web hiện đại và một đường truyền internet là có thể sử dụng nó. RegexPal được viết hoàn toàn bằng JavaScript, do đó nó chỉ hỗ trợ ngôn ngữ JavaScript.



RegexPal sẽ tự động tô màu các chuỗi phù hợp với chuỗi biểu thức chính quy được nhập ở ô phía trên. Nếu bạn nhập một cú pháp sai thì RegexPal sẽ tô sáng những phần sai đó. RegexPal là công cụ tôi thường sử dụng khi viết các chương trình trên .Net để kiểm tra biểu thức chính quy của mình. Bạn có thể sử dụng và tìm hiểu thêm về RegexPal tại: <http://regexpal.com/>

5.3.2. Nhận diện tấn công qua biểu thức chính quy

5.3.2.1. Cross Site Scripting (XSS)

Lỗi XSS xảy ra khi ứng dụng web nhận các dữ liệu độc hại và chuyển nó đến trình duyệt cho người dùng mà không xác nhận lại dữ liệu đó có hợp lệ hay không. Kiểu tấn công này cho phép kẻ tấn công thực thi các đoạn mã độc trong trình duyệt của nạn nhân và có thể cướp phiên người dùng hoặc chuyển hướng người dùng đến các trang độc hại khác.

Để phát hiện các tấn công XSS, có thể sử dụng các biểu thức chính quy:

```
(?:,\s*(?:alert|showmodal|dialog|eval)\s*,)|(?:::\s*eval\s*[\^\\s])|([\^:\  
s\\w,\\.\/?+-]\s*)?(?<![a-  
z\\/_@]) (\s*return\s*)?(?: (?:document\s*\.)?(?:.+\/)?(?:alert|eval|msgbox|sh  
owmod(?:al|eless)|dialog|showhelp|prompt|confirm|dialog|open)) \s*(?:([\^.a-  
z\\s\\-]|(?:\s*[\^\\s\\w,\\.@\\/+  
]))| (?:java[\s\\/]*\.[\s\\/]*lang) | (?:\w\s*=\s*new\s+\w+) | (?:&\s*\w+\s*\)[\^,]  
) | (?:\+[\\W\\d]*new\s+\w+[\\W\\d]*\+)| (?:document\\.\\w) ]
```

Hoặc:

```
(?:=\s*(?:top|this>window|content|self|frames|_content))| (?:\\/\\s*[gimx  
]*\s*{ } ) ) | (?:[\^\\s]\s*=\s*script) | (?:\\.\\s*constructor) | (?:default\s+xml\s+n  
amespace\s*=) | (?:\\/\\s*\+[\^+]\s*\+\\s*\/) ]
```

5.3.2.2. Nhận diện tấn công SQL Injection

SQL Injection là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (chèn vào) và thi hành các câu lệnh SQL bất hợp pháp. SQL Injection có thể cho phép những kẻ tấn công thực hiện các thao tác, delete, insert, update,... trên cơ sở dữ liệu của ứng dụng, thậm chí là server mà ứng dụng đó đang chạy, lỗi này thường xảy ra trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase..

Để phát hiện các tấn công SQL Injection, có thể sử dụng các biểu thức chính quy:

```
(?: "\s*or\s*"?d) | (?: \\x(?:23|27|3d) ) | (?: ^.? "$) | (?: (?: ^["\\"] * (?: [\\d"]  
+ | [^"] +) ) + \s* (?: n?and|x?or|not|\\|\\| |&&) \s* [\\w" [+&!@() , . -  
] ) | (?: [^\\w\\s] \\w+ \s* [|-] \s* " \s* \s* \w | (?: @\\w+ \s+ (and|or) \s* ["\\d] +) | (?: @ [\\w-  
] + \s (and|or) \s* [^\\w\\s] ) | (?: [^\\w\\s:] \s* \\d\\w+ [^\\w\\s] \s* ". ) | (?: \\Winformation_s  
chema|table_name\\w) ]
```

Hoặc:

```
(?: "\s*.*+(?:or|id)\\W*"d) | (?: \\^") | (?: ^ [\\w\\s"-  
] + (?: <=and\\s) (?: <=or\\s) (?: <=xor\\s) (?: <=nand\\s) (?: <=not\\s) (?: <=\\|\\| ) (?: <=\\&\\&) \\w+ ( ( )  
 ) | (?: " [\\s\\d] * [^\\w\\s] + \\W* \\d\\W* . * ["\\d] ) | (?: "\s* [^\\w\\s?] + \s* [^\\w\\s] + \s*" ) | (?: "  
\\s* [^\\w\\s] + \s* [\\W\\d] . * (?: #|--) ) | (?: ". * \\s* \\d ) | (?: "\s*or\\s [^\\d] + [\\w-  
] + . * \\d ) | (?: [ ( ) * <>%+- ] [\\w-] + [^\\w\\s] + " [^, ] ) ]
```

5.3.2.3. Nhận diện tấn công directory traversal

Directory Traversal là một dạng tấn công cho phép kẻ tấn công truy cập vào thư mục và file cấm trên web server. Nếu như truy cập này thành công thì kẻ tấn công có thể xem được các file, thư mục cấm và thực thi các câu lệnh trên web server. Đa số các web server mắc lỗi này đều không kiểm soát đầu vào dữ liệu được gửi từ client. Dạng tấn công này còn có tên gọi khác như Dot-Dot-Slash, Directory Climbing, Path Traversal và Backtracking.

Để phát hiện các tấn công Directory Traversal, có thể sử dụng các biểu thức chính quy:

```
(?: (?:\|/|\|) ?\.(?:\|/|\|) (?:\|/|\|) ?) | (?:\w\.exe\|s) | (?:\s*\w+\s*\|/|\w*-|+|/)| (?:\d\|dx\|) | (?:%(?:c0\|af\|5c\|) | (?:\|/ (?:%2e) {2}) ]
```

Hoặc:

```
(?:%c0%ae\|/)| (?: (?:\|/|\|) (home|conf|usr|etc|proc|opt|s?bin|local|dev|tmp|kern| [br]oot|sys|system|windows|winnt|program| %[a-z_-]{3,}% ) (?:\|/|\|) ) | (?: (?:\|/|\|) inetpub|localstart\|asp|boot\|ini) ]
```

5.4. XÂY DỰNG CÔNG CỤ PHÂN TÍCH

Dựa vào một số yêu cầu cụ thể, có thể tự xây dựng công cụ phân tích log file riêng nhằm phục vụ công tác chuyên môn. Sử dụng ngôn ngữ lập trình hệ thống bất kỳ để có thể thực hiện, ví dụ dưới đây sử dụng ngôn ngữ lập trình Python cung một số rule được trình bày theo dạng xml.

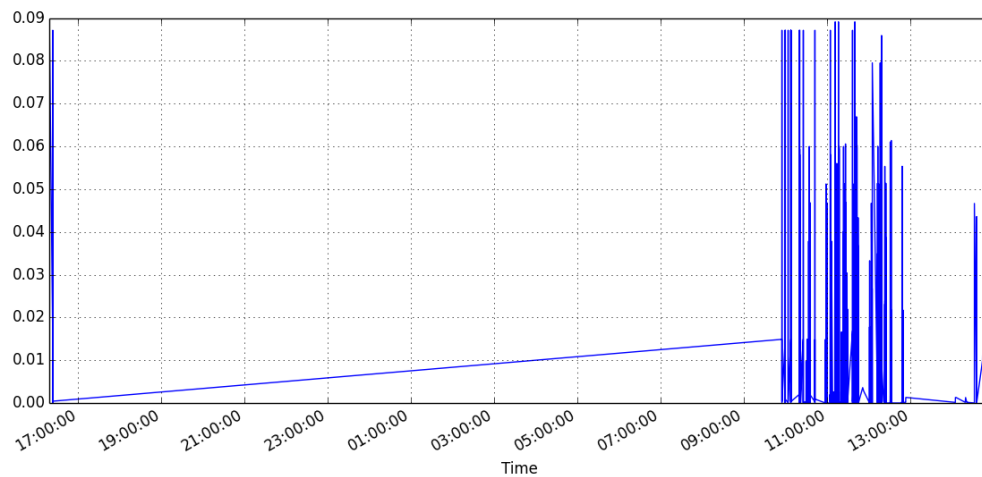
Trình tự các việc xây dựng công cụ phân tích, trước tiên cần đọc log file từ đầu đến cuối theo trình tự từng dòng, với mỗi dòng phân tích, cần trích lọc các thông tin về thời gian, địa chỉ ip của client, http status code nhằm sử dụng cho việc phân chi log thành nhiều file nhỏ theo yêu cầu.

5.4.1. Phân chia log file theo thời gian

Sử dụng biểu thức chính quy để nhận diện thời gian xảy ra sự cố nhằm phân chia file log ban đầu thành các file log nhỏ, một file log lúc này sẽ chứa các dòng log trong một ngày hoặc một khoảng thời gian xảy ra sự cố.

```
(.*) (\d{2}) / (\w{3}) / (\d{4}) .*
```

Mục đích của việc này là dựa trên phán đoán khoảng thời gian xảy ra sự cố và co hẹp lại phạm vi các log file để tiến hành phân tích cho hợp lý nhằm tiết kiệm thời gian phân tích.



Đoạn mã sau mô tả việc đọc file log và ghi vào các file log nhỏ, mỗi log file sẽ chứa các dòng log trong một ngày hoặc một khoảng thời gian xảy ra sự cố:

```
match = re.search(r"(.*) (\d{2})/(\w{3})/(\d{4}).*", line)
if match:
    date = match.group(2)
    month = match.group(3)
    year = match.group(4)
    file_name=time_log_output+"/"+year+"_"+month+"_"+date+".log"
    log_file = open(file_name,'a')
    log_file.write(line)
    log_file.close()
```

5.4.2. Phân chia log file theo mã http status code

Sử dụng biểu thức chính quy để nhận diện mã http status code nhằm phân chia file log ban đầu thành các file log nhỏ, một file log lúc này sẽ chứa tất cả các dòng log tương ứng với từng loại mã http status code, chẳng hạn như: Information, Success, Redirection, Client Error (Lỗi Client), Server Error (Lỗi Server).

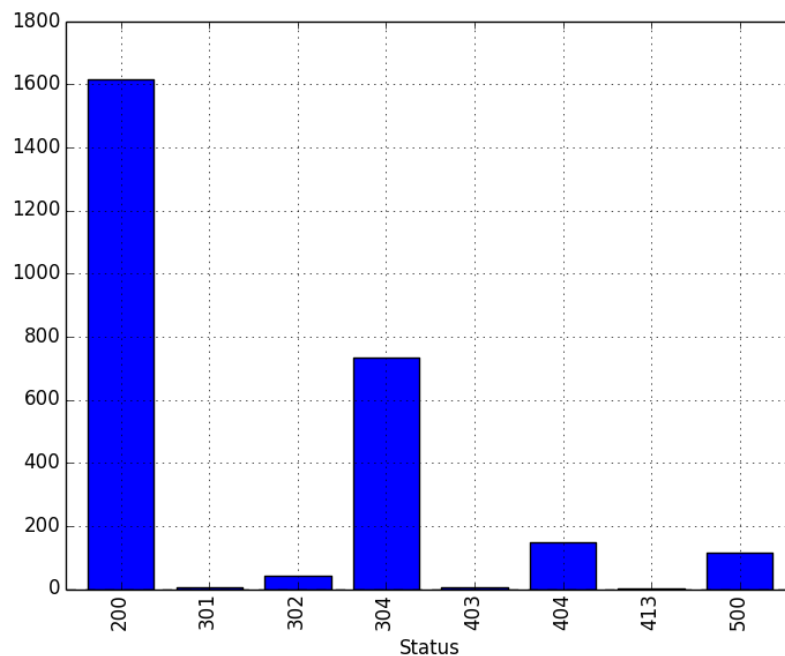
```
# Informational (100-102) -> 10\d
HTTP/1\.\d" 10[0-2]
# Success (200-208, 226) -> 2\d\d
```

```

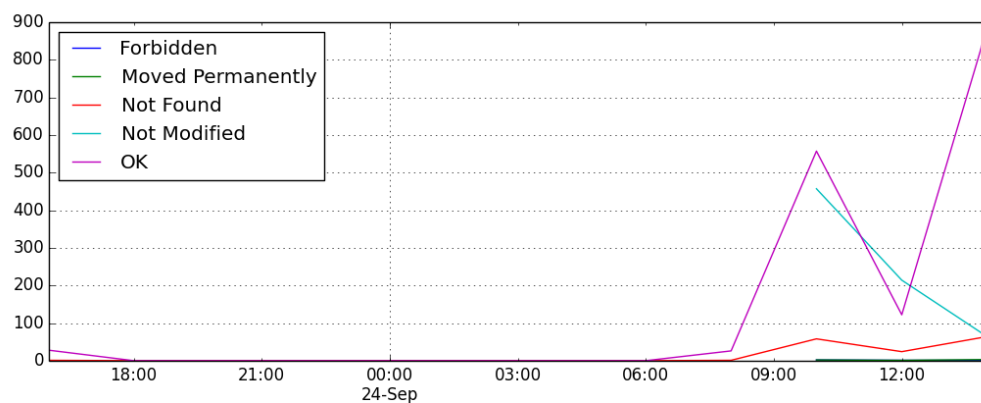
HTTP/1\.\d" (20[0-8]|226)
# Redirection (300-308) -> 30\d
HTTP/1\.\d" 30[0-8]
# Client Error (400-409, 410-419, 420-429, 431, 440, 444, 449, 450-
451, 494-497, 499) -> 4\d\d
HTTP/1\.\d" 4(0[0-9]|1[0-9]|2[0-9]|31|4(0|4|9)|5[0-1]|9([4-7]|9))
# Server Error (500-511, 520, 522, 523, 524, 598, 599) -> 5\d\d
HTTP/1\.\d" 5(0[0-9]|1[0-1]|2(0|[2-4])|9[8-9])

```

Mục đích của việc này là cho phép chúng ta xem xét riêng log file hoặc dữ liệu mà client đệ trình lên máy chủ có thể bị chuyển hướng, có thể bị lỗi máy chủ, lỗi client hoặc là đệ trình thành công.



Hoặc là xem xét các loại mã lỗi theo trình tự thời gian nhằm phân tích các nhận định về tấn công tương ứng.



Đoạn mã sau mô tả việc đọc file log và ghi vào các file log nhỏ, mỗi log file sẽ chứa tất cả các dòng log tương ứng với từng loại mã http status code:

```
# Informational (100-102) -> 10\d
regex_informational = re.search(r'HTTP/1\.\d" 10[0-2]', line)
if regex_informational:
    file_name=err_code_output+"/Informational.log"
    log_file = open(file_name, "a")
    log_file.write(line)
    log_file.close()

# Success (200-208, 226) -> 2\d\d
regex_success = re.search(r'HTTP/1\.\d" (20[0-8]|226)', line)
if regex_success:
    file_name=err_code_output+"/Success.log"
    log_file = open(file_name, "a")
    log_file.write(line)
    log_file.close()

# Redirection (300-308) -> 30\d
regex_redirection = re.search(r'HTTP/1\.\d" 30[0-8]', line)
if regex_redirection:
    file_name=err_code_output+"/Redirection.log"
    log_file = open(file_name, "a")
    log_file.write(line)
    log_file.close()

# Client Error (400-409, 410-419, 420-429, 431, 440, 444, 449, 450-
451, 494-497, 499) -> 4\d\d
regex_client_error = re.search(r'HTTP/1\.\d" 4(0[0-9]|1[0-9]|2[0-
9]|31|4(0|4|9)|5[0-1]|9([4-7]|9))', line)
if regex_client_error:
    file_name=err_code_output+"/Client_Error.log"
    log_file = open(file_name, "a")
    log_file.write(line)
    log_file.close()

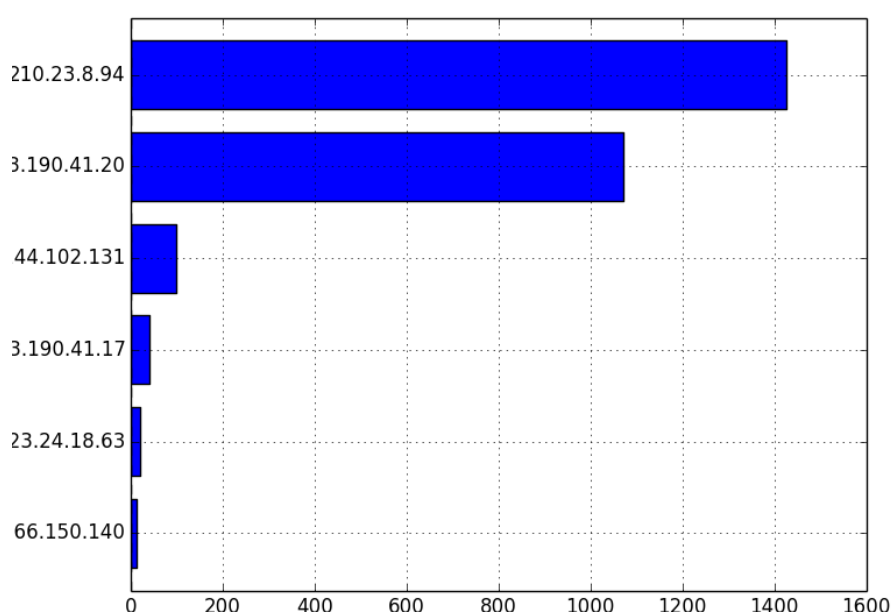
# Server Error (500-511, 520, 522, 523, 524, 598, 599) -> 5\d\d
regex_server_error = re.search(r'HTTP/1\.\d" 5(0[0-9]|1[0-1]|2(0|[2-
4])|9[8-9])', line)
if regex_server_error:
    file_name=err_code_output+"/Server_Error.log"
    log_file = open(file_name, "a")
    log_file.write(line)
    log_file.close()
```


5.4.3. Phân chia log file theo địa chỉ IP

Sử dụng biểu thức chính quy để nhận diện địa chỉ client IP nhằm phân chia file log ban đầu thành các file log nhỏ, một file log lúc này sẽ là một địa chỉ IP.

```
(?: (?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.){3}(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])
```

Mục đích của việc này là dựa trên dung lượng của từng file có thể nhận diện IP nào truy cập nhiều nhất hoặc nghi ngờ những tấn công từ chối dịch vụ.



Đoạn mã sau mô tả việc đọc file log và ghi vào các file log nhỏ, mỗi log file lưu trữ thông tin của một địa chỉ IP:

```
ip_address=re.findall(r'(?: (?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.){3}(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])', line)
if (ip_address!=[]):
    name_file_ip=ip_log_output+"/"+ip_address[0]+".log"
    log_file = open(name_file_ip, "a")
    log_file.write(line)
    log_file.close()
```

5.4.4. Phân chia log file theo kiểu tấn công

Khi một kẻ tấn công thực hiện các tấn công vào ứng dụng web thì chúng sẽ để trình các dữ liệu tương ứng với các tấn công. Và đương nhiên, log file của

chúng ta sẽ lưu lại điều đó. Để tìm kiếm các tấn công trên trong log file, chúng ta sẽ dựa vào các biểu thức chính quy để có thể nhận diện một cách nhanh chóng.

Một số rule nhận diện tấn công bao gồm:

```
<filter>
<id>1</id>
<rule><![CDATA[(?:\"[^\"]*\"[^\"]*>)|(?:\"[^\"]*\"[^\"]*>)|(?:\"[^\"]*\"[^\"]*>)]></rule>
<description>finds html breaking injections including whitespace attacks</description>
</filter>
<filter>
<id>2</id>
<rule><![CDATA[(?:\".+\"[<=\\s\"\"^\"]+\")|(?:\"\\s*\\w+\\s*=\")|(?:>\\w=\\/)|(?:#.+.\\)\"[\\s]*>)|(?:\"\\s*(?:src|style|on\\w+)\\s*=\\s*\")|(?:\"[^\"]?\"[,;\\s]+\\w*\\[\\[\\(\\)\\]]></rule>
<description>finds attribute breaking injections including whitespace attacks</description>
</filter>
<filter>
<id>3</id>
<rule><![CDATA[(?:^>[\\w\\s]*<\\/?\\w{2,}>)]></rule>
<description>finds unquoted attribute breaking injections</description>
</filter>
<filter>
<id>4</id>
<rule><![CDATA[(?:[+\\/\\s*name[\\W\\d]*\\[\\])+)|(?:;\\W*url\\s*=)|(?:\"[^\"]*\"[^\"]*>)]></rule>
<description>Detects url-, name-, JSON, and referrer-contained payload attacks</description>
</filter>
<filter>
<id>5</id>
<rule><![CDATA[(?:\\W\\s*hash\\s*\"[^\"]*\"[^\"]*>)]></rule>
<description>Detects hash-contained xss payload attacks, setter usage and property overloading</description>
</filter>
<filter>
<id>6</id>
```

```

<rule><![CDATA[(?:with\s*\(\s*.\s*\)\s*\w+\s*\()|(?:do|while|for)\s*\([^\)]*\)\s*\{|(?:\[w\s]*\[W*w])></rule>
<description>Detects self contained xss via with(), common loops and regex to string conversion</description>
</filter>
<filter>
<id>7</id>

<rule><![CDATA[(?:[=(.|\.|\?|\:|;)|(?:with\([^\)]*\)\)|(?:\.\s*source\W)]]></rule>
<description>Detects JavaScript with(), ternary operators and XML predicate attacks</description>
</filter>
<filter>
<id>8</id>

<rule><![CDATA[(?:\/\w*\s*)\s*\()|(?:\[w\s]+\([w\s]+\)[w\s]+\)|(?:<!(?:mozilla\/\d\.\d\s)\([^\]]+[[^\]]+\)[^\]]*\)|(?:^[s!]{([([^[{]]+{[([^[^\\]])+[]\)}\])\s+",\d]*[]\)}\))|(?:"\)?\W*[]|(?:=\s*^[s:;]+\s*{([([^[^\\]])+[]\)}\)}\))></rule>
<description>Detects self-executing JavaScript functions</description>
</filter>
<filter>
<id>9</id>

<rule><![CDATA[(?:\\u00[a-f0-9]{2})|(?:\\x0*[a-f0-9]{2})|(?:\\d{2,3})]]></rule>
<description>Detects the IE octal, hex and unicode entities</description>
</filter>
<filter>
<id>10</id>

<rule><![CDATA[(?:(?:\/|\\)\?\.+(\|\/|\\)(?:\.+)?|(?:\w\.exe\??\s)|(?:;\s*\w+\s*\[\/\w*-\]|+\/)|(?:\d\.\dx\|)|(?:%(?:c0\.|af\.|5c\.)|(?:\/(?:%2e){2})]]></rule>
<description>Detects basic directory traversal</description>
</filter>...

```

Các rule này được lưu thành tập tin rules.xml, đoạn code sau cho phép đọc các rule này và tìm kiếm tương ứng trong log file các dòng tương ứng có sự trùng khớp dữ liệu. Nếu phát hiện có sự trùng khớp, sẽ tiến hành lưu thành tập tin riêng với id tương ứng với mã lỗi.

```

dom_xml=dx.parse('rules.xml')
for rule in dom_xml.getElementsByTagName('filter'):
    parten=rule.getElementsByTagName('rule')[0].childNodes[0].data
    if re.search(parten, line):














```

```

id_rule = rule.getElementsByTagName('id')[0].childNodes[0].data
log_file=open(att_log_output+"/id_"+id_rule+".log", "a")
log_file.write(line)
log_file.close()
break

```

Kết quả của quá trình sẽ đưa ra như sau, tuy nhiên các kết quả cần phải kiểm tra lại vì việc khớp các rule chỉ mang tính tương đối.

Name	Date modified	Type
 id_1	12/29/2015 9:58 AM	Text Document
 id_7	12/29/2015 9:58 AM	Text Document
 id_10	12/29/2015 9:53 AM	Text Document
 id_11	12/29/2015 9:44 AM	Text Document
 id_14	12/29/2015 9:52 AM	Text Document
 id_16	12/29/2015 9:33 AM	Text Document
 id_21	12/29/2015 9:53 AM	Text Document
 id_22	12/29/2015 9:58 AM	Text Document
 id_23	12/29/2015 9:58 AM	Text Document
 id_30	12/29/2015 9:57 AM	Text Document
 id_35	12/29/2015 9:58 AM	Text Document
 id_42	12/29/2015 9:58 AM	Text Document
 id_44	12/29/2015 9:58 AM	Text Document