BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ

ĐỀ TÀI THỰC TẬP CƠ SỞ

TÌM HIỂU HỆ THỐNG PHÁT HIỆN XÂM NHẬP MẠNG SURICATA

Khoa: Công nghệ thông tin
Chuyên ngành: An toàn thông tin

Sinh viên thực hiện:
Trần Quang Huy
Lớp: AT12G

Người hướng dẫn:
Lý Bá Cường
Khoa Công nghệ thông tin – Học viện Kỹ thuật mật mã

Hà Nội, Tháng 1/2020

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

Điểm chuyên cần:……
Điểm báo cáo:……..

Hà nội, ngày…..tháng…..năm…..
Xác nhận của giảng viên hướng dẫn

MỤC LỤC

LỜI CẢM ƠN

*Em xin cảm ơn sâu sắc đến Thầy **LÝ BÁ CƯỜNG** – giảng viên Học viện kỹ thuật Mật mã đã tạo đầy đủ điều kiện để em có thể hoàn thành đề tài của mình. Thầy luôn nhắc nhở và giúp đỡ em trong việc hoàn thành bài báo cáo.*

Nhờ vào sự nhiệt tình của Thầy, và các thầy cô trong khoa em đã có thể hoàn thành đề tài đúng tiến độ theo yêu cầu của nhà trường.

Xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Cũng như bất kỳ một thành tựu khoa học nào của nhân loại, khi mà các thành tựu càng được ứng dụng rộng rãi trong đời sống xã hội thì càng dễ bị lợi dụng, sử dụng hoặc là mục tiêu của tội phạm. Các thành tựu do công nghệ thông tin đem lại cũng không nằm ngoài quy luật đó. Vì vậy, trong thế giới mà công nghệ thông tin đã tạo ra cho con người đã hình thành một khái niệm về tội phạm - tội phạm trong lĩnh vực công nghệ thông tin hay còn được biết đến với các tên khác nhau như: tội phạm mạng (Cyber Crimes), tội phạm máy tính hay tội phạm liên quan đến máy tính (Computer Crimes). Đây là vấn đề đang gây rất nhiều khó khăn không chỉ đối với Việt Nam mà còn đối với nhiều nước trên thế giới. Do vậy, việc tìm ra các giải pháp hữu hiệu trong việc phòng chống lại các tội phạm công nghệ thông tin đang là vấn đề được quan tâm hàng đầu và cũng là những thách thức trong ngành công nghệ thông tin của mỗi quốc gia.

Với sự phát triển ngày càng nhiều cùng những thử đoạn ngày càng tinh vi và nguy hiểm của các attacker hiện nay thì đối với một hệ thống mạng của doanh nghiệp vấn đề an toàn an ninh phải được đặt lên hàng đầu. Sử dụng một tường lửa với những công nghệ hiện đại và các chức năng có thể ngăn chặn các cuộc tấn công của các hacker là một giải pháp hữu hiệu cho các doanh nghiệp trong việc phát hiện, ngăn chặn các mối đe doạ nguy hiểm để bảo vệ hệ thống mạng.

Báo cáo đề tài **"TÌM HIỂU HỆ THỐNG PHÁT HIỆN XÂM NHẬP MẠNG SURICATA"** tập trung phân tích các chức năng của tường lửa thế hệ mới, cấu trúc, chức năng và tập luật của tường lửa thế hệ mới Suricata. Phân tích một số cuộc tấn công xâm nhập từ đó hình thành các tập luật phát hiện và ngăn chặn tương ứng

SINH VIÊN THỰC HIỆN

Trần Quang Huy

CHƯƠNG 1. TỔNG QUAN VỀ SURICATA

1.1. Giới thiệu về Suricata

Suricata là một hệ thống phát hiện xâm nhập dựa trên mã nguồn mở. Nó được phát triển bởi Open Information Security Foundation (OISF).

Công cụ này được phát triển không nhằm cạnh tranh hay thay thế các công tụ hiện có, nhưng nó sẽ mang lại những ý tưởng và công nghệ mới trong lĩnh vực an ninh mạng.

Suricata là công cụ IDS/IPS (Intrusion Detection System / Intrusion Prevention System) phát hiện và ngăn chặn xâm nhậpdựa trên luật để theo dõi lưu lượng mạng và cung cấp cảnh báo đến người quản trị hệ thống khi có sự kiện đáng ngờ xảy ra. Nó được thiết kế để tương thích với các thành phần an ninh mạng hiện có. Bản phát hành đầu tiên chạy trên nền tảng linux 2.6 có hỗ trợ nội tuyến (inline) và cấu hình giám sát lưu lượng thụ động có khả năng xử lý lưu lượng lên đến gigabit. Suricata là công cụ IDS/IPS miễn phí trong khi nó vẫn cung cấp những lựa chọn khả năng mở rộng cho các kiến trúc an ninh mạng phức tạp nhất.

Là một công cụ đa luồng, Suricata cung cấp tăng tốc độ và hiệu quả trong việc phân tích lưu lượng mạng. Ngoài việc tăng hiệu quả phần cứng (với phần cứng và card mạng giới hạn), công cụ này được xây dựng để tận dụng khả năng xử lý cao được cung cấp bởi chip CPU đa lõi mới nhất.

1.2. Nhu cầu ứng dụng

Với sự phát triển ngày càng nhiều các hình thức tấn công của tội phạm mạng (Cyber Crimes) hiện nay thì đối với một hệ thống mạng của doanh nghiệp, vấn đề an ninh phải được đặt lên hàng đầu.

Sử dụng một tường lửa với những công nghệ hiện đại và các chức năng có thể ngăn chặn các cuộc tấn công của các hacker là một giải pháp hữu hiệu cho các doanh nghiệp trong việc phát hiện, ngăn chặn các mối đe doạ nguy hiểm để bảo vệ hệ thống mạng.

1.3. Lịch sử phát triển và các RFC liên quan

1.3.1. Lịch sử phát triển

Vào tháng 12 năm 2009, một phiên bản beta đã được phát hành.Bản chuẩn đầu tiên phát hành tiếp theo vào tháng 7 năm 2010.

1.3.2. RFC liên quan

Jonkman, Matt (2009-12-31). "Suricata IDS Available for Download!". Seclists.org. Retrieved 2011-11-08.

"Suricata Features". Retrieved 2012-10-06.

"Suricata All Features". Retrieved 2012-10-06.

1.4. Công nghệ áp dụng

1.4.1. Hệ thống phát hiện xâm nhập mạng IDS

IDS (Intrusion Detection System) là hệ thống giám sát lưu thông mạng (có thể là phần cứng hoặc phần mềm), có khả năng nhận biết những hoạt động khả nghi hay những hành động xâm nhập trái phép trên hệ thống mạng trong tiến trình tấn công, cung cấp thông tin nhận biết và đưa ra cảnh báo cho hệ thống, nhà quản trị. IDS có thể phân biệt được các cuộc tấn công từ nội bộ hay tấn công từ bên ngoài.

IDS phát hiện dựa trên các dấu hiệu đặc biệt về nguy cơ đã biết hay dựa trên so sánh lưu thông mạng hiện tại với baseline (thông số chuẩn của hệ thống có thể chấp nhận được) để tìm ra các dấu hiệu bất thường.

1.4.2. Hệ thống ngăn chặn xâm nhập mạng IPS

IPS (Intrusion Prevention System) là một hệ thống có thể phát hiện và ngăn chặn sự xâm nhập từ bên ngoài vào các hệ thống máy tính.

IPS là một phương pháp tiếp cận an ninh mạng bằng cách ưu tiên sử dụng các công nghệ tiên tiến để phát hiện và ngăn chặn các nỗ lực xâm nhập vào hệ thống máy tính. IPS kiểm tra các luồng lưu lượng ra/ vào một hệ thống máy tính hoặc mạng máy tính nhằm mục đích vi phạm an ninh. Nếu phát hiện mối đe dọa thì nó sẽ có những hành động bảo vệ như ngăn chặn gói tin hoặc ngắt toàn bộ kết nối. IPS kiểm tra, ghi chép lại một cách chi tiết các hành động đang đăng nhập vào hệ thống và gửi cảnh báo cho hệ thống hoặc quản trị mạng. Các IPS khác nhau có phương thức khác nhau trong việc kiểm tra các luồng dữ liệu để phát hiện các mối đe dọa, xâm nhập.

1.4.3. Công nghệ giám sát an ninh mạng NSM

Giám sát an ninh mạng (Network Security Mornitoring) là việc thu thập các thông tin trên các thành phần của hệ thống, phân tích các thông tin, dấu hiệu nhằm đánh giá và đưa ra các cảnh báo cho người quản trị hệ thống.

Đối tượng của giám an ninh mạng là tất cả các thành phần, thiết bị trong hệ thống mạng:

- Các máy trạm.
- Cơ sở dữ liệu.

- Các ứng dụng.
- Các server.
- Các thiết bị mạng.

1.4.4. Sử dụng PCAP log lại thông tin của lưu lượng dữ liệu mạng

PCAP (packet capture) bao gồm những giao diện lập trình ứng dụng (API) dùng để chặn bắt và phân tích các lưu lượng dữ liệu trên mạng. PCAP thực hiện các chức năng lọc gói dữ liệu theo những luật của người dùng khi chúng được truyền tới ứng dụng, truyền những gói dữ liệu thô tới mạng, thu thập thông tin thống kê lưu lượng mạng. Đối với các hệ thống thuộc họ Unix ta có thư viện libpcap, còn đối với Window ta có thư viện được port từ libpcap là winpcap.

1.5. Luật trong Suricata

1.5.1. Rules format

Chữ ký đóng một vai trò rất quan trọng trong Suricata. Trong hầu hết các dịp mọi người đang sử dụng các quy tắc hiện có.

Tài liệu Rules Suricata này giải thích tất cả về chữ ký, làm thế nào để đọc, điều chỉnh và tạo ra chúng.

Một quy tắc/chữ ký bao gồm:

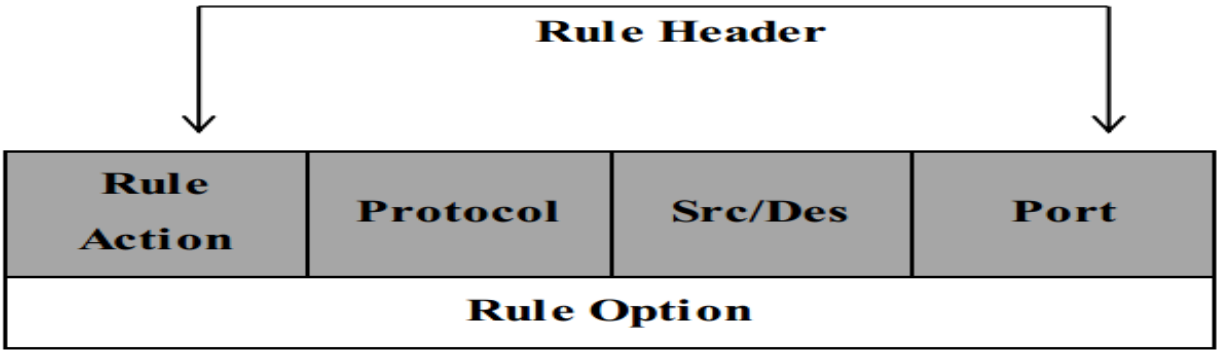
- Các Action , xác định những gì sẽ xảy ra khi các chữ ký.
- Các Header , xác định giao thức, địa chỉ IP, cổng và chỉ đạo của các quy tắc.
- Các rules option, xác định các chi tiết cụ thể của quy tắc.

Một ví dụ về quy tắc như sau:

```
drop tcp $HOME_NET any -> $EXTERNAL_NET
any (msg:"ET TROJAN Likely Bot Nick in IRC
(USA +..)"; flow:established,to_server;
flowbits:isset,is_proto_irc; content:"NICK ";
pcre:"/NICK .*USA.*[0-9]{3,}/i";
reference:url,doc.emergingthreats.net/2008124
; classtype:trojan-activity; sid:2008124;
rev:2;)
```

Trong ví dụ này, Drop là Action, tcp là Header và phần trong ngoặc là rules option.

1.5.2. Rule Header



Hình Tổng quan về suricata.1: Cấu trúc luật trong Suricata

a) Rule Action

Mục đầu tiên trong một luật đó chính là phần rule action, rule action sẽ nói cho Suricata biết phải làm gì khi thấy các gói tin phù hợp với các luật đã được quy định sẵn. Có 4 hành động mặc định trong Suricata đó là: pass (cho qua), drop (chặn gói tin), reject, alert (cảnh báo):

- ***Pass:*** nếu signature được so sánh trùng khớp và chỉ ra là pass thì Suricata sẽ thực hiện dừng quét gói tin và bỏ qua tất cả các luật phía sau đối với gói tin này.
- ***Drop:*** nếu chương trình tìm thấy một signature hợp lệ và nó chỉ ra là drop thì gói tin đó sẽ bị hủy bỏ và dừng truyền ngay lập tức, khi đó gói tin không thể đến được nơi nhận.
- ***Reject:*** là hành động bỏ qua gói tin, bỏ qua ở cả bên nhận và bên gửi. Suricata sẽ tạo ra một cảnh báo với gói tin này.
- ***Alert:*** nếu signature được so sánh là hợp lệ và có chứa một alert thì gói tin đó sẽ được xử lý giống như với một gói tin không hợp lệ. Suricata sẽ tạo ra một cảnh báo.

b) Protocol

Trường tiếp theo trong luật đó là protocol. Các giao thức mà Suricata hiện đang phân tích các hành vi bất thường đó là TLS, SSH, SMTP (tải thư điện tử qua mạng internet), IMAP (đặt sự kiểm soát email trên mail server), MSN, SMB (chia sẻ file), TCP, UDP, ICMP và IP, DNS, HTTP, HTTPS.

c) IP Address

Mục tiếp theo của phần header đó là địa chỉ IP. Các địa chỉ này dùng để kiểm tra nơi đi và nơi đến của một gói tin. Địa chỉ ip đó có thể là địa chỉ của một máy

đơn hoặc cũng có thể là địa chỉ của một lớp mạng. Từ khóa "any" được sử dụng để định nghĩa một địa chỉ bất kỳ.

Một địa chỉ ip sẽ được viết dưới dạng *ip_address/netmask*. Điều này có nghĩa là nếu netmask là /24 thì lớp mạng đó là lớp mạng C, /16 là lớp mạng B hoặc /32 là chỉmột máy đơn. Ví dụ: địa chỉ 192.168.1.0/24 có nghĩa là một dải máy có địa chỉ IP từ 192.168.1.1-192.168.1.255.

Trong hai địa chỉ IP trong một luật Suricata thì sẽ có một địa chỉ IP nguồn và một địa chỉ IP đích. Việc xác định đâu là địa chỉ nguồn, đâu là địa chỉ đích phụ thuộc vào "→".

Ngoài ra toán tử phủ định có thể được áp dụng cho việc định địa chỉ IP. Có nghĩa là khi sử dụng toán tử này thì Suricata sẽ bỏ qua việc kiểm tra địa chỉ của gói tin đó. Toán tử đó là "!". Ngoài ra ta có thể định nghĩa một danh sách các địa chỉ IP bằng cách viết liên tiếp chúng cách nhau bởi một dấu ",".

Ví dụ:

Alert TCP any any → ![192.168.1.0/24, 172.16.0.0/16] 80 (msg: "Access")

d) Port

Port có thể được định nghĩa bằng nhiều cách. Với từ khóa "any" giống như địa chỉIP để chỉ có thể sử dụng bất kỳ port nào. Gán một port cố định, ví dụ như gán kiểm tra ở port 80 http hoặc port 22 ssh. Ngoài ra ta cũng có thể sử dụng toán tử phủ định để bỏ qua một port nào đó hoặc liệt kê một dải các port.

Ví dụ:

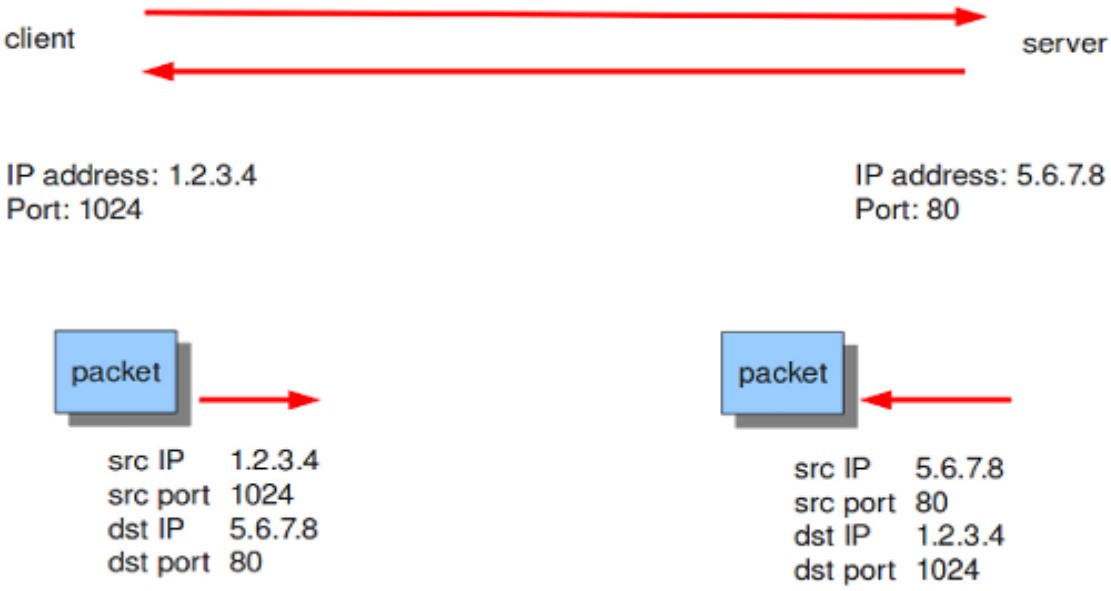
Alert UDP any any → 192.168.1.0/24 1:1024 - port bất kỳ tới dãy port từ 1 - 1024.

Alert UDP any any → 192.168.1.0/24 :6000 - port bất kỳ tới dãy port nhỏ hơn 6000.

Alert UDP any any → 192.168.1.0/24 !6000:6010 - port bất kỳ tới bất kỳ port nào, bỏ qua dãy port từ 6000 – 6010.

e) Điều hướng

Toán tử hướng "→" chỉ ra đâu là hướng nguồn, đâu là hướng đích. Phần địa chỉ IP và port ở phía bên trái của toán tử được coi như là địa chỉ nguồn và port nguồn, phần bên phải được coi như địa chỉ đích và port đích. Ngoài ra còn có toán tử "<>" Suricata sẽ xem cặp địa chỉ/port nguồn và đích là như nhau. Nghĩa là nó sẽ ghi/phân tích ở cả hai phía của cuộc hội thoại.



Hình Tổng quan về suricata.2: Có một máy khách có địa chỉ IP 1.2.3.4 và cổng 1024 và máy chủ có địa chỉ IP 5.6.7.8, nghe trên cổng 80 (thường là HTTP)

Ví dụ:

Alert tcp 1.2.3.4 1024 -> 5.6.7.8 80

1.5.3. Rule Option

Rule Options chính là trung tâm của việc phát hiện xâm nhập. Nội dung chứa các dấu hiệu để xác định một cuộc xâm nhập. Nó nằm ngay sau phần Rule Header và được bọc bởi dấu ngoặc đơn "()". Tất cả các rule options sẽ được phân cách nhau bởi dấu chấm phẩy ";", phần đối số sẽ được tách ra bởi dấy hai chấm ":".

Có 4 loại rule options chính bao gồm:

- ***General:*** Tùy chọn này cung cấp thông tin về luật đó nhưng không có bất cứảnh hưởng nào trong quá trình phát hiện.
- ***Payload:***Tùy chọn liên quan đến phần tải trong một gói tin.
- ***Non-payload:***Bao gồm các tùy chọn không liên quan đến phần tải của gói tin (header).
- ***Post-detection:*** Các tùy chọn này sẽ gây ra những quy tắc cụ thể sau khi một luật đã được kích hoạt.

Các thành phần khác trong Rule:

a) General

❖ ***Msg***

Msg (Message): được dùng để cho biết thêm thông tin về từng signature và các cảnh báo. Phần đầu tiên sẽ cho biết tên tập tin của signature và phần này quy ước là phải viết bằng chữ in hoa. Định dạng của msg như sau:

```
msg: ".....";
```

❖ ***Sid***

Sid (signature id): cho ta biết định danh riêng của mỗi signature. Định danh này được bắt đầu với số. Định dạng của sid như sau:

```
sid:123;
```

❖ ***Rev***

Rev (revision): mỗi sid thường đi kèm với một rev. Rev đại diện cho các phiên bản của signature. Mỗi khi signature được sửa đổi thì số rev sẽ được tăng lên bởi người tạo ra. Định dạng của rev như sau:

```
rev:123;
```

❖ ***Reference***

Reference: cung cấp cho ta địa chỉ đến được những nơi chứa các thông tin đầy đủ về signature. Các tham chiếu có thể xuất hiện nhiều lần trong một signature. Ví dụ về một tham chiếu như sau:

```
reference: url, www.info.nl
```

system	URL Prefix
bugtraq	http://www.securityfocus.com/bid
cve	http://cve.mitre.org/cgi-bin/cvename.cgi?name=
nessus	http://cgi.nessus.org/plugins/dump.php3?id=
arachnids	http://www.whitehats.com/info/IDS
mcafee	http://vil.nai.com/vil/dispVirus.asp?virus_k=
url	http://

Hình Tổng quan về suricata.3: ***Bảng các tuỳ chọn của Reference***

❖ ***Classtype***

Classtype: cung cấp thông tin về việc phân loại các lớp quy tắc và cảnh báo. Mỗi lớp bao gồm một tên ngắn gọn, một tên đầy đủ và mức độ ưu tiên.

Ví dụ:

```
Config classification: web-application-attack,Web Application Attack,1
config classification: not-suspicious,Not Suspicious Traffic,3
```

Signature	Classification.config	Alert
web-attack	web-attack, Web Application Attack, priority:1	Web Application Attack
not-suspicious	not-suspicious, Not Suspicious Traffic, priority:3	Not Suspicious Traffic

Hình Tổng quan về suricata.4: ***Thông tin phân loại lớp quy tắc***

❖ ***Priority***

Priority: chỉ ra mức độ ưu tiên của mỗi signature. Các giá trị ưu tiên dao động từ 1 đến 255, nhưng thường sử dụng các giá trị từ 1 -> 4. Mức ưu tiên cao nhất là 1. Những signature có mức ưu tiên cao hơn sẽ được kiểm tra trước. Định dạng như sau:

```
priority:1;
```

❖ ***Metadata***

Metadata: Suricata sẽ bỏ qua những gì viết sau metadata. Định dạng như sau:

```
metadata:.....;
```

b) Payload

❖ ***Content***

Content: thể hiện nội dung chúng ta cần viết trong signature, nội dung này được đặt giữa 2 dấu nháy kép. Nội dung là các byte dữ liệu, có 256 giá trị khác nhau (0-255). Chúng có thể là các ký tự thường, ký tự hoa, các ký tự đặc biệt, hay là các

mã hexa tương ứng với các ký tự và các mã hexa này phải được đặt giữa 2 dấu gạch dọc. Định dạng của một nội dung như sau:

```
content: ”.....”;
```

❖ ***Nocase***

Nocase: được dùng để chỉnh sửa nội dung thành các chữ thường, không tạo rasự khác biệt giữa chữ hoa và chữ thường. Nocase cần được đặt sau nội dung cần chỉnh sửa.

Ví dụ:

```
content: “abC”; nocase;
```

❖ ***Depth***

Depth: sau từ khóa depth là một số, chỉ ra bao nhiêu byte từ đầu một payload cần được kiểm tra. Depth cần được đặt sau một nội dung.

Ví dụ: Ta có một payload : abCdefghij. Ta thực hiện kiểm tra 3 byte đầu của payload.

```
content: “abC”; depth:3;
```

❖ ***Offset***

Offset: chỉ độ lệch byte trong tải trọng sẽ được kiểm tra.

Ví dụ: độ lệch là 3 thì sẽ kiểm tra từ byte thứ 4 trong tải trọng.

```
content: “def”; offset:3;
```

Ví dụ:

Alert TCP 192.168.1.0/24 any -> any any (content: \"HTTP"; offset: 4; depth: 40; msg: "HTTP matched";)

❖ ***Distance***

Distance: xác định khoảng cách giữa các nội dung cần kiểm tra trong payload.Khoảng cách này có thể là một số âm.

Ví dụ:

```
content: “abC”; content: “efg”; distance:1;
```

❖ ***Within***

Within: được dùng cùng với distance, để chỉ độ rộng của các byte cần kiểm tra sau một nội dung với khoảng cách cho trước đó.

Ví dụ:

content:"GET"; depth:3 content:"download"; distance:10 \within:9;

Luật có nghĩa là tìm “GET” trong 3 byte đầu tiên của trường dữ liệu, di chuyển thêm 10 byte bắt đ ầu từ “GET” và tìm khớp “download”. Tuy nhiên, “download” phải xuất hiện trong 9 byte tiếp theo.

❖ ***Dsize***

Dsize: được dùng để tìm một payload có độ dài bất kỳ.

```
dsize:min<>max;
```

❖ ***Rpc***

Rpc (Remote Procedure Call): là một ứng dụng cho phép một chương trình máy tính thực hiện một thủ tục nào đó trên một máy tính khác, thường được sử dụng cho quá trình liên lạc. Định dạng của rpc như sau:

```
rpc:<application number>, [<version number>|*], [<procedure number>|*]>;
```

❖ ***Replace***

Replace được dùng để thay đổi nội dung của payload, điều chỉnh lưu lượng mạng. Việc sửa đổi nội dung của payload chỉ có thể được thực hiện đối với gói dữ liệu cá nhân. Sau khi thực hiện thay đổi nội dung xong thì Suricata sẽ thực hiện tính toán lại trường checksum.

c) Non-Payload

1. IP

❖ ***ttl***

Được sử dụng để kiểm tra về thời gian sống, tồn tại tên mạng của một địa chỉ IP cụ thể trong phần đầu của mỗi gói tin. Giá trị time-to-live (thời gian sống), xác định thời gian tối đa mà mỗi gói tin có thể được lưu thông trên hệ thống mạng. Nếu giá trị này về 0 thì gói tin sẽ bị hủy bỏ. Thời gian sống được xác định dựa trên số

hop, khi đi qua mỗi hop/router thì thời gian sống sẽ bị trừ đi 1. Cơ chế này nhằm hạn chế việc gói tin lưu thông trên mạng vô thời hạn. Định dạng của một ttl như sau:

```
ttl:<number>;
```

❖ ***ipopts***

Chúng ta có thể xem và tùy chỉnh các tùy chọn cho việc thiết lập các địa chỉ IP. Việc thiết lập các tùy chọn cần được thực hiện khi bắt đầu một quy tắc. Một số tùy chọn có thể sử dụng:

IP-option	Description
rr	Record Route
eol	End of List
nop	No Op
ts	Time Stamp
sec	IP Security
esec	IP Extended Security
lsrr	Loose Source Routing
ssrr	Strict Source Routing
satid	Stream Identifier
any	any IP options are set

Hình Tổng quan về suricata.5: ***Một số tuỳ chọn của Ipopts***

Định dạng của một ipopts như sau:

```
ipopts: <name>;
```

❖ ***sameip***

Mỗi gói tin sẽ có một địa chỉ IP nguồn và đích. Chúng ta có thể sử dụng sameip để kiểm tra xem địa chỉ IP nguồn và đích có trùng nhau hay không. Định dạng của sameip như sau:

```
sameip;
```

❖ ***Ip_proto***

Được dùng để giúp ta lựa chọn giao thức. Ta có thể chọn theo tên hoặc số tương ứng với từng giao thức. Có một số giao thức phổ biến sau:

```
1    ICMP       Internet Control Message
6    TCP        Transmission Control Protocol
17    UDP        User Datagram
47    GRE        General Routing Encapsulation
50    ESP        Encap Security Payload for IPv6
51    AH         Authentication Header for Ipv6
58    IPv6-ICMP   ICMP for Ipv6
```

Định dạng của ip_proto như sau:

```
ip_proto:<number/name>;
```

❖ ***Id***

Được sử dụng để định danh cho các phân mảnh của gói tin được truyền đi. Khi gói tin truyền đi sẽ được phân mảnh, và các mảnh của một gói tin sẽ có ID giống nhau. Việc này giúp ích cho việc ghép lại gói tin một cách dễ dàng. Định dạng như sau:

```
id:<number>;
```

❖ ***Geoip***

Cho phép xác định địa chỉ nguồn, đích để gói tin lưu thông trên mạng.

❖ ***Fragbits***

Được dùng để kiểm tra các phân mảnh của gói tin. Nó bao gồm các cơ chế sau:

```
M - More Fragments
D - Do not Fragment
R - Reserved Bit
+        match on the specified bits, plus any others
*        match if any of the specified bits are set
!        match if the specified bits are not set
```

Định dạng của một Fragbits như sau:

```
fragbits:[*+!]<[MDR]>;
```

❖ ***Fragoffset***

Kiểm tra sự phù hợp trên các giá trị thập phân của từng mảnh gói tin trên trường offset. Nếu muốn kiểm tra phân mảnh đầu tiên của gói tin, chúng ta cần kết hợp fragoffset 0 với các tùy chọn fragment khác. Các tùy chọn fragment như sau:

```
<       match if the value is smaller than the specified value
>       match if the value is greater than the specified value
!       match if the specified value is not present
```

Định dạng của fragoffset:

```
fragoffset:[!|<|>]<number>;
```

2. TCP

❖ ***Sed***

Là một số ngẫu nhiên được tạo ra ở cả bên nhận và bên gửi gói tin để kiểm tra số thứ tự của các gói tin đến và đi. Máy khách và máy chủ sẽ tự tạo ra một số seq riêng của mình. Khi một gói tin được truyền thì số seq này sẽ tăng lên 1. Seq giúp chúng ta theo dõi được những gì diễn ra khi một dòng dữ liệu được truyền đi.

❖ ***Ack***

Được sử dụng để kiểm tra xem gói tin đã được nhận bởi nơi nhận hay chưa trong giao thức kết nối TCP. Số thứ tự của ACK sẽ tăng lên tương ứng với số byte dữ liệu đã được nhận thành công.

❖ ***Window***

Được sử dụng để kiểm tra kích thước của cửa sổ TCP. Kích thước cửa sổ TCP là một cơ chế dùng để kiểm soát các dòng dữ liệu. Cửa sổ được thiết lập bởi người nhận, nó chỉ ra số lượng byte có thể nhận để tránh tình trạng bên nhận bị tràn dữ liệu. Giá trị kích thước của cửa sổ có thể chạy từ 2 đến 65.535 byte.

3. ICMP

❖ ***Itype***

Cung cấp cho việc xác định các loại ICMP. Các thông điệp khác nhau sẽ được phân biệt bởi các tên khác nhau hay các giá trị khác nhau.

Định dạng của itype như sau:

```
itype:min<>max;
itype:[<|>]<number>;
```

Type	Name	Reference
0	Echo Reply	[RFC792]
3	Destination Unreachable	[RFC792]
4	Source Quench	[RFC792]
5	Redirect	[RFC792]
6	Alternate Host Address	[JBP]
7	Unassigned	[JBP]
8	Echo	[RFC792]
9	Router Advertisement	[RFC1256]
10	Router Selection	[RFC1256]
11	Time Exceeded	[RFC792]
12	Parameter Problem	[RFC792]
13	Timestamp	[RFC792]
14	Timestamp Reply	[RFC792]
15	Information Request	[RFC792]
16	Information Reply	[RFC792]
17	Address Mask Request	[RFC950]
18	Address Mask Reply	[RFC950]
19	Reserved (for Security)	[Solo]
20-29	Reserved (for Robustness Experiment)	[ZSu]
30	Traceroute	[RFC1393]
31	Datagram Conversion Error	[RFC1475]
32	Mobile Host Redirect	[David Johnson]
37	Domain Name Request	[RFC1788]
38	Domain Name Reply	[RFC1788]
39	SKIP	[Markson]
40	Photuris	[RFC2521]

Hình Tổng quan về suricata.6: ***Bảng Type của ICMP Header***

❖ ***Icode***

Cho phép xác định mã của từng ICMP để làm rõ hơn cho từng gói tin ICMP. Định dạng của icode như sau:

```
icode:min<>max;
icode:[<|>]<number>;
```

❖ ***Icmp_id***

Mỗi gói tin ICMP có một giá trị ID khi chúng được gửi. Tại thời điểm đó, người nhận sẽ trả lại tin nhắn với cùng một giá trị ID để người gửi sẽ nhận ra và kết nối nó đúng với yêu cầu ICMP đã gửi trước đó. Định dạng của một icmp_id như sau:

```
icmp_id:<number>;
```

❖ ***Icmp_seq***

Được sử dụng để kiểm tra số thứ tự của ICMP. Định dạng của icmp_seq như sau:

```
icmp_seq:<number>;
```

4. HTTP

Có các sửa đổi nội dung bổ sung có thể cung cấp các khả năng dành riêng cho giao thức ở lớp ứng dụng. Thông tin thêm có thể được tìm thấy tại Payload Keywords Các từ khóa này đảm bảo chữ ký chỉ kiểm tra các phần cụ thể của lưu lượng mạng. Chẳng hạn, để kiểm tra cụ thể về URI yêu cầu, cookie hoặc cơ quan phản hồi hoặc yêu cầu HTTP, v.v.

Tất cả các từ khóa HTTP là sửa đổi. Lưu ý sự khác biệt giữa sửa đổi nội dung và bộ đệm dính. Xem từ khóa sửa đổi để biết thêm thông tin.

- **content modifiers** nhìn lại quy tắc, ví dụ:

  alert http any any -> any any (content:"index.php"; http_uri; sid:1;)

- **sticky buffers** được đặt đầu tiên và tất cả các từ khóa theo sau nó áp dụng cho bộ đệm đó, ví dụ:

  alert http any any -> any any (http_response_line; content:"403 Forbidden"; sid:1;)

Các từ khóa yêu cầu:

Keyword	Sticky or Modifier	Direction
http_uri	Modifier	Request
http_raw_uri	Modifier	Request
http_method	Modifier	Request
http_request_line	Sticky Buffer	Request
http_client_body	Modifier	Request
http_header	Modifier	Both
http_raw_header	Modifier	Both
http_cookie	Modifier	Both
http_user_agent	Modifier	Request
http_host	Modifier	Request
http_raw_host	Modifier	Request
http_accept	Sticky Buffer	Request
http_accept_lang	Sticky Buffer	Request
http_accept_enc	Sticky Buffer	Request
http_referer	Sticky Buffer	Request
http_connection	Sticky Buffer	Request
http_content_type	Sticky Buffer	Both
http_content_len	Sticky Buffer	Both
http_start	Sticky Buffer	Both
http_protocol	Sticky Buffer	Both
http_header_names	Sticky Buffer	Both

Các từ khóa phản hồi:

Keyword	Sticky or Modifier	Direction
http_stat_msg	Modifier	Response
http_stat_code	Modifier	Response
http_response_line	Sticky Buffer	Response
http_header	Modifier	Both
http_raw_header	Modifier	Both
http_cookie	Modifier	Both
http_server_body	Modifier	Response
http.server	Modifier	Response
http.location	Modifier	Response
file_data	Sticky Buffer	Response
http_content_type	Sticky Buffer	Both
http_content_len	Sticky Buffer	Both
http_start	Sticky Buffer	Both
http_protocol	Sticky Buffer	Both
http_header_names	Sticky Buffer	Both

- http_method: chỉ ra các phương thức được áp dụng với các request http. Các phương thức http: GET, POST, PUT, HEAD, DELETE, TRACE, OPTIONS, CONNECT và PATCH.
- http_uri và http_raw_uri: chỉ ra đường dẫn tới nơi chứa nội dung yêu cầu.
- http_header: chỉ ra phương thức sử dụng, địa chỉ cần truy cập tới và tình trạng kết nối.
- http_cookie.
- http_user_agent: là một phần của http_header, chỉ ra thông tin về trình duyệt của người dùng.
- http_client_body: chỉ ra các yêu cầu của máy trạm.
- http_stat_code: chỉ ra mã trạng thái của server mà máy trạm yêu cầu kết nối tới.
- http_stat_msg: các dòng tin thông báo về tình trạng máy chủ, hay tình trạng về việc đáp ứng các yêu cầu kết nối của máy trạm.

- http_server_body: chỉ ra nội dung đáp trả các yêu cầu từ máy trạm của máy chủ.
- File_data: chỉ ra nội dung, đường dẫn tới file chứa dữ liệu được yêu cầu.

5. FLOW

❖ *Flowbits*

Gồm 2 phần, phần đầu mô tả các hành động được thực hiện, phần thứ 2 là tên của flowbit. Các hành động của flowbit:

flowbits: set, name	*Được dùng để thiết lập các điều kiện/tên cho các flow.*
flowbits: isset, name	*Có thể được sử dụng trong các luật để đảm bảo rằng sẽ tạo ra một cảnh báo khi các luật là phù hợp và các điều kiện sẽ được thiết lập trong flow.*
flowbits: toggle, name	*Dùng để đảo ngược các thiết lập hiện tại.*
flowbits: unset, name	*Được sử dụng để bỏ các thiết lập về điều kiện trong luật.*
flowbits: isnotset, name	*Được sử dụng để đảm bảo rằng sẽ tạo ra một cảnh báo khi các luật là phù hợp và các điều kiện sẽ không được thiết lập trong flow.*

❖ *Flow*

Có thể được sử dụng để kết nối các thư mục chứa các flow lại với nhau. Các flow có thể được đi từ hoặc đến từ Client/Server và các flow này có thể ở trạng thái được thiết lập hoặc không. Việc kết nối các flow có thể xảy ra các trường hợp sau:

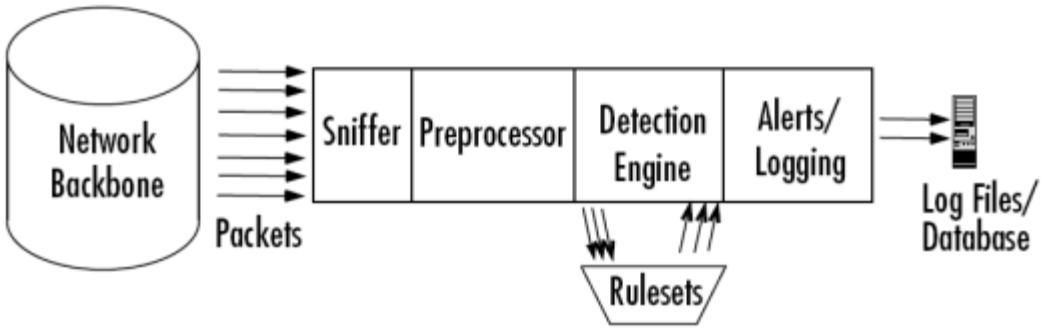
```
to_client              established/ stateless
from_client            established/ stateless
to_server              established/ stateless
from_serverestablished/ stateless
```

1.6. Kiến trúc Của Suricata

Suricata được phát triển dựa trên snort nên nó vẫn giữ nguyên kiến trúc bên trong của snort. Kiến trúc của nó có nhiều thành phần, với mỗi thành phần có một chức năng riêng.

Các thành phần chính là:

- modul giải mã gói tin
- modul tiền xử lý (preprocessors)
- modul phát hiện
- modul bản ghi và cảnh báo (logging and alerting system)
- modul kết xuất thông tin



Hình Tổng quan về suricata.7: Kiến trúc của suricata

Khi suricata hoạt động nó sẽ thực hiện lắng nghe và thu bắt tất cả các gói tin nào di chuyển qua nó. Các gói tin sau khi bị bắt được đưa và modul giải mã gói tin. Tiếp theo gói tin sẽ được đưa vào modul tiền xử lý, rồi đưa vào modul phát hiện. Tại đây, tùy theo việc có phát hiện được xâm nhập hay không mà gói tin có thể được lưu thông tiếp hay được đưa vào modul bản ghi và cảnh báo để xử lý. Khi các cảnh báo được xác định modul kết xuất thông tin sẽ thực hiện việc đưa cảnh báo ra theo đúng định dạng mong muốn. Sau đây ta sẽ đi sâu vào nghiên cứu chi tiết hơn.

1.6.1. Module giải mã gói dữ liệu

Suricata sử dụng thư viện PCap để bắt mọi gói tin trên mạng lưu thông qua hệ thống. Mỗi gói tin sau khi được giải mã sẽ được đưa tiếp vào modul tiền xử lý.

1.6.2. Module tiền xử lý

Modul tiền xử lý là một modul rất quan trọng đối với bất kỳ hệ thống IDS nào để có thể chuẩn bị gói dữ liệu đưa vào cho modul phát hiện phân tích. Ba nhiệm vụ chính của modul này là:

- Kết hợp lại các gói tin: khi một lượng dữ liệu lớn được gửi đi, thông tin sẽ bị chia nhỏ thành nhiều gói tin. Khi suricata nhận được các gói tin này thì nó phải thực hiện ghép lại thành hình dạng ban đầu, từ đó mới thực hiện các công việc xử lý tiếp. Như ta đã biết khi một phiên làm việc diễn ra, sẽ có rất nhiều gói tin được trao đổi trong phiên đó. Một gói tin riêng rẽ sẽ không có trạng thái và nếu công việc phát hiện xâm nhập chỉ dựa vào gói tin đó sẽ không đem lại hiệu quả cao. Modul tiền xử lý giúp suricata hiểu được các phiên làm việc khác nhau từ đó giúp đạt được hiệu quả cao hơn trong việc phát hiện xâm nhập.
- Giải mã và chuẩn hóa giao thức (decode/normalize): công việc phát hiện xâm nhập dựa trên dấu hiệu nhận dạng nhiều khi bị thất bại khi kiểm tra các giao thức có dữ liệu có thể được thực hiện dưới nhiều hình thức khác nhau. Ví dụ: một web server có thể chấp nhận nhiều dạng URL như URL viết dưới dạng mã hexa/unicode, URL chấp nhận cả dấu / hay \ hoặc nhiều ký tự này liên tiếp cùng lúc. Chẳng hạn ta có dấu hiệu nhậ dạng “scripts/iiaadmin”, kẻ tấn công có thể vượt qua bằng cách tùy biến các yêu cầu gửi đến web server như sau:

```
“scripts//iisadmin”
“scripts/examples/../iisadmin”
“scripts\iisadmin”
“scripts/.\iisadmin”
```

- Hoặc thực hiện mã hóa các chuỗi này dưới dạng khác. Nếu suricata chi thực hiện đơn thuần việc so sánh dữ liệu nhận dạng sẽ xảy ra tình trạng bỏ sót các hành vi xâm nhập. Do vậy, một số modul tiền xử lý

phải có nhiệm vụ giải mã và chỉnh sửa, sắp xếp lại các thông tin đàu vào này để thông tin khi đưa đến modul phát hiện cơ thể phát hiện được mà không bỏ sót.

- Phát hiện các xâm nhập bất thường (nonrule/anormal):thường dùng để đối phó với các xâm nhập không thể hoặc rất khó phát hiện được bằng luật thông thường hoặc các dấu hiệu bất thường trong giao thức. Các modul tiền xử lý dạng này có thể phát hiện xâm nhập theo bất cứ cách nào mà ta nghĩ ra từ đó tăng thêm tính năng cho suricata. Ví dụ: một plugin tiền xử lý có nhiêm vụ thống kê thông lượng mạng tại thời điểm bình thường để rồi khi có thông lượng bất thường xảy ra nó có thể tính toán, phát hiện và đưa ra cảnh báo.

1.6.3. Module phát hiện

Đây là modul quan trong nhất. Nó chịu trách nhiệm phát hiện các dấu hiệu xâm nhập. Modul phát hiện sử dụng các luật được định nghĩa sẵn để so sánh với dữ liệu thu thập được từ đó xác định có xâm nhập xảy ra hay không. Rồi tiếp theo mới có thể thực hiện công việc ghi log, tạo báo cáo, kết xuất thông tin.

Một vấn đề quan trọng trong modul phát hiện là vấn đề thời gian xử lý các gói tin: IDS thường nhận được rất nhiều gói tin và bản thân nó cũng có rất nhiều luật xử lý. Vì vậy có thể mất những khoảng thời gian khác nhau cho việc xử lý các gói tin khác nhau. Và khi thông lượng qua mạng quá lớn có thể xảy ra việc bỏ sót hoặc không phản hồi đúng lúc. Khả năng xử lý của modul phát hiện dựa trên yếu tố như: số lượng các luật, tốc độ của hệ thống mạng.

Một modul phát hiện cũng có khả năng tách các phần của gói tin ra và áp dụng các luật trên từng phần của gói tin. Các phần đó có thể là:

- IP header
- Header của tầng vận chuyển: TCP, UDP
- Header của tầng ứng dụng: DNS header, HTTP header, …
- Phần tải của gói tin (bạn cũng có thể áp dụng các luật lên các phần dữ liệu được truyền đi trong gói tin).

Một vấn đề trong modul phát hiện là việc xử lý thế nào khi một gói tin bị phát hiện đã được đánh thứ tự ưu tiên nên một gói tin khi bị phát hiện bởi nhiều luật khác nhau, cảnh báo được đưa ra ứng với luật có mức ưu tiên cao nhất.

1.6.4. Module bản ghi và cảnh báo

Tùy thuộc vào modul phát hiện có nhận dạng được xâm nhập hay không mà gói tin có thể bị ghi bản ghi hoặc đưa ra cảnh báo. Các file log là các file text, dữ liệu trong đó có thể được ghi dưới nhiều định dạng khác nhau chẳng hạn tcpdump.

1.6.5. Module kết xuất thông tin

Modul này có thể thức hiện các thao tác khác nhau tùy thuộc theo việc bạn muốn lưu kết quả xuất ra như thế nào. Tùy theo việc cấu hình hệ thống mà có thể thực hiện các công việc như là:

- Ghi log file
- Ghi syslog: syslog là một chuẩn lưu trữ các file log được sử dụng rất nhiều trên các hệ thống unix, linux.
- Ghi cảnh báo vào cơ sở dữ liệu.
- Tạo file log dạng xml: việc này rất thuận tiện cho việc trao đổi dữ liệu và chia sẻ dữ liệu.
- Cấu hình lại router, firewall.
- Gửi các cảnh báo được gói trong gói tin sử dụng giao thức SNMP. Các gói tin dạng này sẽ được gửi tới một SNMP server từ đó giúp cho việc quản lý các cảnh báo và hệ thống IDS một cách tập trung và thuật tiện.
- Gửi thông điệp SMB (server message block) tới các máy tính windows.

Ta cũng có thể tự viết modul kết xuất thông tin riêng tùy thoe mục đích sử dụng.

1.7. Suricata IPS

Suricata là một *IPS* (Hệ thống ngăn chặn xâm nhập), một hệ thống để phân tích xâm nhập mạng. Phần mềm phân tích tất cả lưu lượng truy cập trên tường lửa để tìm kiếm các cuộc tấn công và sự bất thường đã biết.

Khi phát hiện một cuộc tấn công hoặc sự bất thường, hệ thống có thể quyết định có chặn lưu lượng truy cập hay chỉ đơn giản là lưu sự kiện trên nhật ký (`/var/log/suricata/fast.log`).

Suricata có thể được cấu hình bằng cách sử dụng các bộ quy tắc được tổ chức trong các danh mục thống nhất. Mỗi thể loại có thể được đặt thành:

- **Enable**: quy tắc khớp lưu lượng truy cập từ danh mục này sẽ được báo cáo.
- **Block**: quy tắc khớp lưu lượng truy cập từ danh mục này sẽ bị loại bỏ.
- **Disable:** quy tắc từ danh mục này được bỏ qua.

Việc sử dụng IPS tác động đến tất cả lưu lượng truy cập trên tường lửa. Hãy chắc chắn rằng bạn hiểu đầy đủ tất cả các hàm ý trước khi kích hoạt nó. Cụ thể, chú ý đến việc chặn các quy tắc có thể dừng cập nhật cho chính hệ thống.

Loại quy tắc:

- **Activex:** Tấn công và lỗ hổng (CVE, v.v.) liên quan đến Activex.
- **Phản ứng tấn công:**

Phản hồi chỉ ra sự xâm nhập khi tải xuống tập tin LMhost, một số biểu ngữ nhất định, lệnh tiêu diệt Metasploit Meterpreter được phát hiện, v.v ... Chúng được thiết kế để bắt các kết quả của một cuộc tấn công thành công. Những thứ như Id id = root, hoặc thông báo lỗi cho thấy sự thỏa hiệp có thể đã xảy ra.

- **Botcc (Bot Command and Control):**

Chúng được tự động phát sinh từ một số nguồn Botnet đang hoạt động đã biết và đã được xác nhận và các máy chủ chỉ huy và điều khiển khác. Cập nhật hàng ngày, nguồn dữ liệu chính là Shadowserver.org. Các quy tắc khối lệnh và điều khiển của Bot được tạo từ Shadowerver.org, cũng như spyeyetracker, palevotracker và zeustracker. Các quy tắc được nhóm theo cổng cung cấp độ trung thực cao hơn với cổng đích được sửa đổi theo quy tắc.

- **Botcc Portgrouped:**

Tương tự như trên, nhưng được nhóm theo cổng đích.

- **Chat:**

Xác định lưu lượng truy cập liên quan đến nhiều khách hàng trò chuyện, irc và hoạt động đăng ký có thể.

- **CIArmy:**

Trí tuệ tập thể đã tạo ra các quy tắc IP để chặn dựa trên www.cinsscore.com.

- **Compromised:**

Đây là danh sách các máy chủ bị xâm nhập đã biết, được xác nhận và cập nhật hàng ngày. Bộ này thay đổi từ một trăm đến một vài quy tắc hunderd tùy thuộc vào các nguồn dữ liệu. Đây là một bản tổng hợp của một số nguồn dữ liệu riêng tư nhưng có độ tin cậy cao. Hâm nóng: Snort không xử lý IP phù hợp với tải thông minh. Nếu cảm biến của bạn đã được đẩy đến giới hạn, bộ này sẽ thêm tải đáng kể. Chúng tôi khuyên bạn nên ở lại với các quy tắc botcc trong trường hợp tải cao.

- **Current Events:**

Danh mục cho các chiến dịch hoạt động và sống ngắn. Danh mục này bao gồm các bộ dụng cụ khai thác và phần mềm độc hại sẽ bị lão hóa và loại bỏ nhanh chóng do tính chất ngắn ngủi của mối đe dọa. Chẳng hạn, các mục có cấu hình cao mà chúng tôi không mong đợi sẽ có các chiến dịch lừa đảo lâu dài liên quan đến thảm họa. Đây là những quy tắc mà chúng tôi không có ý định giữ trong quy tắc lâu dài hoặc cần phải được kiểm tra trước khi chúng được xem xét để đưa vào. Thông thường, đây sẽ là các trang web đơn giản cho URL nhị phân Storm trong ngày, các trang web để bắt các ứng dụng dễ bị tổn thương mới được tìm thấy của CLSID, nơi chúng tôi không có bất kỳ chi tiết nào về khai thác, v.v.

Decoder-events:

Các quy tắc này ghi nhật ký các sự kiện chuẩn hóa liên quan đến giải mã.

DNS:

Quy tắc cho các cuộc tấn công và lỗ hổng liên quan đến DNS. Ngoài ra loại lạm dụng dịch vụ cho những thứ như đường hầm.

DOS:

Nỗ lực phát hiện tấn công từ chối dịch vụ. Dự định ngăn chặn tấn công DOS gửi đến và chỉ dẫn.

Drop:

Các quy tắc để chặn các thư rác được liệt kê trên mạng. Dựa trên IP. Đây là danh sách được cập nhật hàng ngày của danh sách Spamhaus DROP (Không lộ trình hoặc ngang hàng). Chủ yếu là những kẻ gửi thư rác chuyên nghiệp.

Dshield:

Quy tắc dựa trên IP cho những kẻ tấn công được xác định Dshield. Danh sách cập nhật hàng ngày của danh sách những kẻ tấn công hàng đầu DShield. Cũng rất đáng tin cậy.

Khai thác:

Khai thác không được bao gồm trong thể loại dịch vụ cụ thể. Quy tắc phát hiện khai thác trực tiếp. Nói chung nếu bạn đang tìm kiếm một cửa sổ khai thác, Veritas, v.v., chúng sẽ ở đây. Những thứ như SQL injection và những thứ tương tự, chúng được khai thác, có thể loại riêng.

- **Các tập tin:**

Ví dụ quy tắc sử dụng chức năng xử lý và trích xuất tệp trong Suricata.

- **FTP:**

Quy tắc cho các cuộc tấn công, khai thác và lỗ hổng liên quan đến FTP. Cũng bao gồm cơ bản không có hoạt động FTP độc hại nào cho mục đích ghi nhật ký, chẳng hạn như đăng nhập, v.v.

- **Games:**

Quy tắc xác định lưu lượng truy cập trò chơi và các cuộc tấn công chống lại các trò chơi đó. World of Warcraft, Starcraft và các trò chơi trực tuyến phổ biến khác có hoạt động ở đây. Chúng tôi không có ý định gắn nhãn những thứ xấu xa này, chỉ là chúng không phù hợp với mọi môi trường.

- **HTTP-Events:**

Quy tắc để ghi nhật ký các sự kiện cụ thể của giao thức HTTP, thường hoạt động bình thường.

- **Inappropriate:**

Quy tắc để xác định các hoạt động liên quan đến các trang web đen. Bao gồm trang web Khiêu dâm, Khiêu dâm Kiddy, các trang web bạn không nên truy cập tại nơi làm việc, v.v. Chúng thường khá nặng Regex và do đó tải cao và thường xuyên bị lỗi. Chỉ chạy những thứ này nếu bạn thực sự quan tâm.

- **Phần mềm độc hại:**

Phần mềm độc hại và phần mềm gián điệp liên quan, không có ý định tội phạm rõ ràng. Ngưỡng để đưa vào bộ này thường là một số hình thức theo dõi dừng hoạt động tội phạm rõ ràng. Bộ này ban đầu được dự định chỉ là phần mềm gián điệp. Điều đó đủ cho một số loại quy tắc thực sự. Ranh giới giữa phần mềm gián điệp và những thứ độc hại hoàn toàn đã bị xóa nhòa kể từ khi chúng tôi bắt đầu thiết lập này. Có nhiều thứ hơn là phần mềm gián điệp ở đây, nhưng hãy yên tâm rằng không có gì ở đây là thứ bạn muốn chạy trên mạng hoặc PC của bạn. Có các móc nối URL cho các bản cập nhật đã biết được mô tả, các chuỗi Tác nhân người dùng của phần mềm độc hại đã biết và tải các phần mềm khác.

- **Phần mềm độc hại di động:**

Cụ thể đối với các nền tảng di động: Phần mềm độc hại và phần mềm gián điệp liên quan, không có mục đích phạm tội rõ ràng.

- **Netbios:**

Các quy tắc để nhận dạng, cũng như các cuộc tấn công, khai thác và lỗ hổng liên quan đến Netbios. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký hoạt động của người dùng.

- **P2P:**

Quy tắc xác định lưu lượng truy cập ngang hàng và các cuộc tấn công chống lại. Bao gồm torrents, edonkey, Bittorrent, Gnutella, Limewire, v.v.

- **Chính sách:**

Danh mục nhận dạng ứng dụng. Bao gồm chữ ký cho các ứng dụng như DropBox và Google Apps, v.v. Cũng bao gồm các giao thức cổng, DLP cơ bản như số thẻ tín dụng và số an sinh xã hội. Bao gồm trong bộ này là các quy tắc cho những thứ thường không được chính sách của công ty hoặc tổ chức không cho phép. Myspace, Ebay, v.v.

- **Shellcode:**

Phát hiện Shellcode từ xa. Shellcode từ xa được sử dụng khi kẻ tấn công muốn nhắm mục tiêu vào một quá trình dễ bị tấn công đang chạy trên một máy khác trên mạng cục bộ hoặc mạng nội bộ. Nếu được thực hiện thành công, shellcode có thể cung cấp cho kẻ tấn công quyền truy cập vào máy đích trên toàn mạng. Mã hóa từ xa thường sử dụng các kết nối ổ cắm TCP / IP tiêu chuẩn để cho phép kẻ tấn công truy cập vào vỏ trên máy mục tiêu. Shellcode như vậy có thể được phân loại dựa trên cách thiết lập kết nối này: nếu shellcode có thể thiết lập kết nối này, nó được gọi là shell shell ngược hoặc một shellcode kết nối lại vì shellcode kết nối lại với máy của kẻ tấn công.

- **SMTP:**

Các quy tắc cho các cuộc tấn công, khai thác và các lỗ hổng liên quan đến SMTP. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký.

- **TELNET:**

Quy tắc cho các cuộc tấn công và lỗ hổng liên quan đến dịch vụ TELNET. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký.

- **VOIP:**

Quy tắc cho các cuộc tấn công và lỗ hổng liên quan đến môi trường VOIP. SIP, h.323, RTP, v.v.

- **SQL:**

  Các quy tắc cho các cuộc tấn công, khai thác và các lỗ hổng liên quan đến SQL. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký.

- **SNMP:**

Quy tắc cho các cuộc tấn công, khai thác và lỗ hổng liên quan đến SNMP. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký.

- **Trojan:**

Phần mềm độc hại có ý định tội phạm rõ ràng. Các quy tắc ở đây phát hiện phần mềm độc hại đang truyền, đang hoạt động, lây nhiễm, tấn công, cập nhật và bất cứ điều gì khác có thể phát hiện trên dây. Đây cũng là một quy tắc rất quan trọng để chạy nếu phải chọn.

- **EveBox:**

EveBox là một trang web dựa trên công cụ quản lý sự kiện và cảnh báo cho các sự kiện do Suricata tạo ra.

CHƯƠNG 2. CÀI ĐẶT VÀ CẤU HÌNH SURICATA

2.1. Chuẩn bị

Một máy ảo Ubuntu Desktop 16.04 LTS với địa chỉ IP: 192.168.150.136.

Một máy ảo windows 7 với địa chỉ IP: 192.168.150.145.

Yêu cầu: Hai máy ảo nằm trong một mạng cục bộ.

Kịch bản: Cài đặt và cấu hình hệ thống phát hiện xâm nhập suricata trên máy ảo Ubuntu 16.04. Sau đó từ máy ảo windows 7, thực hiện các giao thức hay các cuộc tấn công như Telnet, ICMP, DDOS, v.v lên máy ảo Ubuntu 16.04. Suricata sẽ ngăn chặn và cảnh báo về cuộc tấn công đó.

2.2. Cài đặt và cấu hình Suricata

2.2.1. Cài đặt

Khi bạn đã đăng nhập vào phiên bản Ubuntu 16.04, hãy chạy lệnh sau để cập nhật hệ thống cơ sở của bạn với các gói có sẵn mới nhất.

```
apt-get update -y

apt-get upgrade -y
```

Cài đặt các phụ thuộc cần thiết:

Trước khi bắt đầu, cần cài đặt một số phụ thuộc theo yêu cầu của Suricata. Có thể cài đặt tất cả chúng bằng cách chạy lệnh sau:

```
apt-get install libpcre3-dbg libpcre3-devlibnet1-dev
libyaml-dev libjansson4 libcap-ng-dev libmagic-dev
libjansson-dev zlib1g-dev autoconf automake libnetfilter-
queue-dev libnetfilter-queue1 libnfnetlink-dev libtool
libpcap-dev -y
```

Sau khi tải hoàn tất các phụ thuộc cần thiết, bắt đầu cài đặt suricata.

Cài đặt suricata:

Đầu tiên, tải xuống phiên bản mới nhất của Suricata từ trang web chính thức của họ bằng lệnh sau:

```
wget https://www.openinfosecfoundation.org/doad/suricata-
4.0.5.tar.gz
```

Tiếp theo, giải nén tập tin đã tải xuống bằng lệnh sau:

```
tar -xvzf suricata-4.0.5.tar.gz
```

Sau đó, xây dựng Suricata bằng lệnh sau:

```
cd suricata-4.0.5

./configure --enable-nfqueue --prefix=/usr
--sysconfdir=/etc --localstatedir=/var
```

Output:

```
To build and install run 'make' and 'make install'.

You can run 'make install-conf' if you want to install
initial configuration
files to /etc/suricata/. Running 'make install-full' will
install configuration
and rules and provide you a ready-to-run suricata.

To install Suricata into /usr/bin/suricata, have the
config in
/etc/suricata and use /var/log/suricata as log dir, use:
./configure --prefix=/usr/ --sysconfdir=/etc/
--localstatedir=/var/
```

Tiếp theo, xây dựng Suricata bằng lệnh sau:

```
make
make install
```

Output:

```
Writing /usr/lib/python2.7/site-packages/suricatasc-0.9-
py2.7.egg-info
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/root/suricata-
4.0.5/scripts/suricatasc'
make[2]: Leaving directory '/root/suricata-
4.0.5/scripts/suricatasc'
make[2]: Entering directory '/root/suricata-4.0.5/scripts'
make[3]: Entering directory '/root/suricata-4.0.5/scripts'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/root/suricata-4.0.5/scripts'
make[2]: Leaving directory '/root/suricata-4.0.5/scripts'
make[1]: Leaving directory '/root/suricata-4.0.5/scripts'
Making install in etc
make[1]: Entering directory '/root/suricata-4.0.5/etc'
make[2]: Entering directory '/root/suricata-4.0.5/etc'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/root/suricata-4.0.5/etc'
make[1]: Leaving directory '/root/suricata-4.0.5/etc'
make[1]: Entering directory '/root/suricata-4.0.5'
make[2]: Entering directory '/root/suricata-4.0.5'
make[2]: Nothing to be done for 'install-exec-am'.
Run 'make install-conf' if you want to install initial
configuration files. Or 'make install-full' to install
configuration and rules
make[2]: Leaving directory '/root/suricata-4.0.5'
make[1]: Leaving directory '/root/suricata-4.0.5'
```

Cài đặt tệp cấu hình mặc định Suricata bằng lệnh sau:

```
make install-conf
```

Output:

```
install -d "/etc/suricata/"
install -d "/var/log/suricata/files"
install -d "/var/log/suricata/certs"
install -d "/var/run/"
install -m 770 -d "/var/run/suricata"
```

2.2.2. Cấu hình

Cần cài đặt bộ quy tắc Suricata IDS cho hệ thống của mình. Có thể cài đặt nó từ thư mục nguồn Suricata bằng lệnh sau:

```
cd suricata-4.0.5
make install-rules
```

Output:

```
install -d "/etc/suricata/rules"
/usr/bin/wget -qO -
https://rules.emergingthreats.net/open/suricata-
4.0/emerging.rules.tar.gz | tar -x -z -C "/etc/suricata/"
-f -
```

Lúc này có thể bắt đầu chạy suricata 1 cách trực tiếp:

```
/usr/bin/suricata -c /etc/suricata//suricata.yaml -i eth0
```

Các rules sẽ được lưu trong:

```
ls /etc/suricata/rules
```

Output:

```
app-layer-events.rules emerging-current_events.rules
emerging-netbios.rules emerging-voip.rules
botcc.portgrouped.rules emerging-deleted.rules emerging-
p2p.rules emerging-web_client.rules
botcc.rules emerging-dns.rules emerging-policy.rules
emerging-web_server.rules
BSD-License.txt emerging-dos.rules emerging-pop3.rules
emerging-web_specific_apps.rules
ciarmy.rules emerging-exploit.rules emerging-rpc.rules
emerging-worm.rules
compromised-ips.txt emerging-ftp.rules emerging-
scada.rules gpl-2.0.txt
compromised.rules emerging-games.rules emerging-scan.rules
http-events.rules
decoder-events.rules emerging-icmp_info.rules emerging-
shellcode.rules LICENSE
dnp3-events.rules emerging-icmp.rules emerging-smtp.rules
modbus-events.rules
dns-events.rules emerging-imap.rules emerging-snmp.rules
sid-msg.map
drop.rules emerging-inappropriate.rules emerging-sql.rules
smtp-events.rules
dshield.rules emerging-info.rules emerging-telnet.rules
stream-events.rules
emerging-activex.rules emerging-malware.rules emerging-
tftp.rules suricata-4.0-enhanced-open.txt
emerging-attack_response.rules emerging-misc.rules
emerging-trojan.rules tls-events.rules
emerging-chat.rules emerging-mobile_malware.rules
emerging-user_agents.rules tor.rules
```

Tiếp theo ta cần sửa đổi tệp suricata.yaml bằng câu lệnh:

```
nano /etc/suricata/suricata.yaml
```

Thực hiện các thay đổi sau theo yêu cầu của bạn:

```
HOME_NET: "[192.168.150.136]"
EXTERNAL_NET: "! $ HOME_NET"
```

Tạo một rule để kiểm tra

Tạo 1 bộ quy tắc riêng để kiểm tra hệ thống suricata. Quy tắc này sẽ tạo một cảnh báo trong tệp /var/log/suricata/fast.log khi ai đó cố gắng tấn công Ping, SSH hoặc DOS SYN FLOOD.

```
nano /etc/suricata/rules/my.rules
```

Sau đó thêm các dòng sau:

```
alert icmp any any -> $HOME_NET any (msg:" ICMP connection
attempt "; sid:1000002; rev:1;)
```

Lưu và đóng tệp tin.

Tiếp theo, cần xác định đường dẫn của tệp quy tắc này trong suricata.yaml:

```
nano /etc/suricata/suricata.yaml
```

Thêm luật `my.rules` vào trong tập `rule-files`:

```
- my.rules
```

Lưu và đóng tệp suricata.yaml.

Cần tắt bất kỳ tính năng giảm tải gói nào trên NIC mà Suricata đang nghe. Bằng câu lệnh:

```
ethtool -K eth0 tso off
ethtool -K eth0 tx off
ethtool -K eth0 gro off
```

Cuối cùng, chạy Suricata ở chế độ trực tiếp bằng lệnh sau:

```
/ usr / bin / suricata -D -c /etc/suricata/suricata.yaml
-i eth0
```

2.2.3. Thử nghiệm

Từ máy ảo windows 7 thực hiện ping sang máy ảo Ubuntu có cài đặt Suricata. Trên máy ảo Ubuntu ta gõ câu lệnh:

```
tail -f /var/log/suricata/fast.log
```

Nếu suricata đang hoạt động đúng thì sẽ phát hiện đc dạng như:

```
[**] [1:1000002:1] ICMP connection attempt [**]
[Classification: (null)] [Priority: 3] {ICMP}
192.168.150.145:8 -> 192.168.150.136:0
```

KẾT LUẬN

Kết quả đạt được:

Việc sử dụng một tường lửa thế hệ mới trong hệ thống mạng doanh nghiệp, nhà nước đang được khuyến khích và đẩy mạnh. Hiện nay, trên thế giới đã có rất nhiều doanh nghiệp, tổ chức nổi tiếng trong ngành đã đưa ra được các sản phẩm tường lửa thế hệ mới như Palo Alto, HP…Nhưng các sản phẩm đó là các sản phẩm thiết bị phần cứng, nên phần nào vẫn gây khó khăn cho các doanh nghiệp trong vấn đề chi phí, lắp đặt và hỗ trợ nâng cấp, phát triển. Việc tìm hiểu và đưa vào sử dụng thành công, hiệu quả một tường lửa mềm sẽ giúp họ khắc phục được các vấn đề khó khăn đó. Đặc biệt là các tường lửa mềm có mã nguồn mở đang nhận được sự quan tâm rất nhiều của giới công nghệ và doanh nghiệp, nhà nước.

Với tình hình thực tiễn như vậy, tường lửa thế hệ mới Suricata sẽ là một lựa chọn hàng đầu và mang lại hiệu quả cao cho các doanh nghiệp. Với những kiến thức nghiên cứu và phân tích chi tiết về Suricata trong bài báo cáo, nó sẽ là một tài liệu giúp cho việc tiếp cận ban đầu với Suricata của các doanh nghiệp, nhà nước trở nên dễ dàng và nhanh chóng nắm bắt được cấu trúc, hoạt động cũng như cách tích hợp và cài đặt nó trong hệ thống mạng của các công ty.

TÀI LIỆU THAM KHẢO

[1] http://www.openinfosecfoundation.org/index.php/download-suricata

[2] http://taosecurity.blogspot.com/2014/01/suricata-20beta2-as-ips-on-ubuntu-1204.html

[3] http://www.linux.org/threads/suricata-the-snort-replacer-part-3-rules.4363/

[4] http://suricata.readthedocs.io/en/latest/index.html

[5] https://github.com/OISF/suricata

[6] http://docs.nethserver.org/en/v7/suricata.html

[7] ***"New Open Source Intrusion Detector Suricata Released"***. Slashdot. 2009-12-31. Retrieved 2011-11-08.

[8] ***"Suricata Downloads"***. Open Security Information Foundation. Retrieved 2011-11-08.

[9] Một số bài tiểu luận và nguồn khác trên internet.