**VIETNAM CYBERSPACE SECURITY TECHNOLOGY**

**JSC**

**🙜🙞🕮🙜🙞**



**TÌM HIỂU CƠ CHẾ PARSING LOG VÀ CHUẨN HÓA DỮ LIỆU**

Người thực hiện: Mai Thành Thắng

Hà Nội 10/2019

Mục lục

[I. Cơ chế parsing log. 3](#_Toc24962044)

[1. Sourcetype 3](#_Toc24962045)

[2. Cấu hình souretype trong file prop.conf 3](#_Toc24962046)

[3. Dùng biểu thức chính quy 5](#_Toc24962047)

[4. Một số dạng biểu thức chính quy thường dùng: 7](#_Toc24962048)

[II. Chuẩn hóa dữ liệu: 7](#_Toc24962049)

# Cơ chế parsing log.

Parsing log hiểu đơng giản là sự phân tích cú pháp(cấu trúc) của nhật ký đưa ra l chuẩn dữ liệu theo 1 dạng nhất định.Vậy nên tìm hiểu phần này là trình bày về sourcetype( loại nguồn)

## Sourcetype

* Sourcetype là một trường mặc định trên Splunk được dùng để gán cho tất cả dữ liệu đầu vào. Nó phân loại những dữ liệu mà đẩy lên từ các agent hoặc chính bản thân nó. Có khả năng định dạng dữ liệu một cách thông minh trong quá trình lập chỉ mục.
* Sourcetype quản lý và tìm kiếm dữ liệu dễ ràng
* Sourcetype được sử dụng để gán các cấu hình như: nhận dạng dấu thời gian, ngắt sự kiện và trích xuất trường cho dữ liệu được lập chỉ mục bởi Splunk.
* Sourcetype định nghĩa dữ liệu đầu vào bằng cách là khi có dữ liệu đầu vào nó sẽ tự động lựa chọn và phân loại dữ liệu. Đối với những dữ liệu đặc biệt bạn phải định nghĩa loại sourcetype cho dữ liệu đầu vào.

## Cấu hình souretype trong file prop.conf

* Splunk sử dụng cấu hình trong props.conf trong khi lập chỉ mục cho log và để xử lý.
* Vị trí file prop.conf :

**/opt/splunk/etc/system/default/props.conf** ---> Không được sửa

**/opt/splunk/etc/system/local/props.conf** -----> file này dùng để sửa( nếu chưa có copy bên trong thư mục **/opt/splunk/etc/system/default/**)

* Có bảy thuộc tính phải đặt trong props.conf:
  + TIME\_PREFIX- Đây là thuộc tính **đầu tiên** được sử dụng để báo cho Splunk biết nơi bắt đầu tìm dấu thời gian trong sự kiện.
  + TIME\_FORMAT-Định dạng kiểu hiện thị dấu thời gian, ví dụ : 10/2/12 như October 2, 2012.
  + SHOULD\_LINEMERGE: gộp dòng sự kiện (vd: log của java)
  + LINE\_BREAKER ( tách dòng sự kiện trùng nhau)
  + TRUNCATE.- Được đo bằng byte, thuộc tính này giới hạn độ dài của một sự kiện (dòng dữ liệu) và sẽ bị phá vỡ khi đạt giới hạn. Giá trị mặc định là 10000 có thể sửa lớn hơn phù hợp với dữ liệu của bạn. Vì dữ liệu không bị chia thành các sự kiện đa dòng (SHOULD\_LINEMERGE = false), phải đảm bảo các sự kiện của mình không bị chia nhỏ một cách không chính xác.
  + TZ – Thuộc tính cài đặt múi giờ.
* Prop.conf dùng trong splunk :

+ Cấu hình thời gian theo quy chuyẩn

+ Định cấu hình ngắt dòng cho các sự kiện đa dòng

+ Thiết lập bộ mã hóa ký tự

+ Định nghĩa các trường bằng biểu thức chính quy

+ Cấu hình đoạn sự kiện

+ Cho phép sử lý nhị phân

+ Ghi đè các loại nguồn Splunk từng tồn tại  
+ Định nghĩa vị trí dùng để tạo lập bảng tra cứu.

* Định dạng mẫu của file Prop.conf:

[sourcetype name]

timestamp recognition configuration (Cấu hình đinh dạng thời gian)

linebreaking configuration (Cấu hình tách dòng)

binary check configuration (Cấu hình kiểm tra nhị phân)

Fieldalias configuration ( cấu hình tên trường)

regex1 for filed extarction (Dùng biểu thức chính quy để định dạng ra trường dữ liệu)

regex2 for filed extarction

Lookup table name reference ( Tên bảng tra cứu để kiểm tra các mục cần tra cứu)

* Ví dụ:

[perforce]

SHOULD\_LINEMERGE = false

LINE\_BREAKER = ([\n]+)(Perforce server info:|Perforce server error:)

EXTRACT-user = pid (?<pid>\d+) (?<user>\w+@[^\s]+) (?<IP\_addr>(?:[0-9]{1,3}\.){3}[0-9]{1,3})

EXTRACT-command = \[(?<client\_version>[^\[\]]+)\] '(?<command>[^']+)'

EXTRACT-lapse = lapse (?<lapse>\d\*\.?\d+)s

EXTRACT-usage = --- usage (?<usage\_user>\d+)\+(?<usage\_system>\d+)us (?<io\_read>\d+)\+(?<io\_write>\d+)io (?<IPC\_received>\d+)\+(?<IPC\_sent>\d+)net (?<memory>\d+)k (?<page\_faults>\d+)

EXTRACT-rpc = --- rpc msgs\/size in\+out (?<RPC\_received>\d+)\+(?<RPC\_sent>\d+)\/(?<RPC\_received\_size>\d+)mb\+(?<RPC\_sent\_size>\d+)mb himarks (?<RPC\_send\_window>\d+)\/(?<RPC\_rcv\_window>\d+) snd\/rcv (?<RPC\_send\_time>\d\*\.?\d+)s\/(?<RPC\_rcv\_time>\d\*\.?\d+)s

* Add data trên giao diện web splunk đến chỗ phần set sourcetye:

LINE\_BEAKER: {[\r\n]\*}<string> Tách dòng từ ký tự nào  
BEAK\_ONLY\_BEFORE: <STRING> tách dòng sự khiện theo thời gian được đính kèm

MUST\_BREAK\_AFTER:<String> vị trí kết thúc dòng sự kiện để tách với dòng sự kiện khác  
SHOULD\_LINEMERGE: có hợp dòng hay không

TIME\_PREFIX: khiểu múi giờ (GMT)

TIME\_FORMAT: ví dụ: %Y-%m-%dT%H:%M:%S.%3N( dạng thời gian)

MAX\_TIMESTAMP: độ dài phần biểu thị thời gian

MUST\_BREAK\_AFTER: ngắt dòng từ vị trí sau dòng nào<string>

MUST\_NOT\_BREAK\_BEFORE: không ngắt dòng phía trước dòng nào.<string>

BREAK\_ONLY\_BEFORE: < luôn ngắt dòng trước ký tự nào> vd: })] ( thực hiện ngắt dòng lần lượt khi gặp ký tự đó)

Luôn đi kèm với EVENT\_BREAK = regex

KV\_MODE = [none|auto|auto\_escaped|multi|json|xml]: Được sử dụng để trích xuất trường tìm kiếm theo thời gian.

TIME\_PREFIX: múi giờ( thường là GMT)

MAX\_TIMESTAMP\_LOOKAHEAD: định dạng độ dài thông tin thời gian

## Dùng biểu thức chính quy

Một số lưu ý khi dùng: ^[a-z0-9\_-]{3,15}$

 ^ Ký hiệu cho biết bắt đầu một dòng

 [a-z0-9\_-] Cho phép tên chứa ký tự a-z, số từ 0 - 9, ký tự -, ký tự \_

 {3,15} Tên dài 3 đến 15 ký tự

 $ Điểm kết thúc dòng

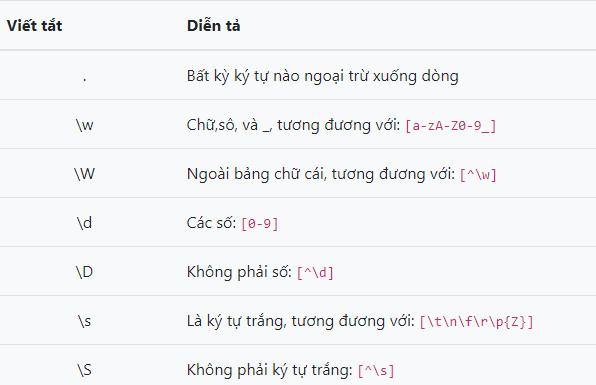
* return \r, newline \n.
* ký tự trắng \s

**Biểu thức chính quy Regex là phân biệt chữ hoa chữ thường**

Bảng ký tự biển diễn – meta



Bảng ký hiệu biểu diễn tập hợp:



- Biểu thức ?= lookahead để lọc kết quả trong một chuỗi:

Vi dụ: Ví dụ (T|t)he(?=\sfat) thì lookahead là (?=\sfat) - nghĩa là T hoặc t theo sau là he vậy tìm được 2 kết quả. Nhưng do có biểu thức lookahead, điều này thì kết quả phù hợp là chỉ lấy khi theo sau nó là chuỗi fat

(T|t)he(?=\sfat) => The fat cat sat on the mat.

- Biểu thức **?!** phủ định lookahead(tìm kiếm phủ định ):

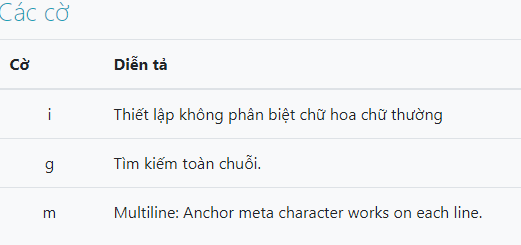
Ký hiệu là ?!, nghĩa là lấy kết quả mà đi sau nó không có chuỗi lookahead

(T|t)he(?!\sfat) => The fat cat sat on the mat.

-Biểu thức (?<=...) Lookbehind ( tìm kiếm trước)

Sử dụng để lấy các phù hợp mà đi trước là một mẫu cũ thể. (?<=(T|t)he\s)(fat|mat) có nghĩa lấy tất cả các từ fat hoặc mat sau các từ The hoặc the

(?<=(T|t)he\s)(fat|mat) => The fat cat sat on the mat.



Các cờ này được đưa vào mẫu theo dạng /RegExp/flags

"/The/gi" => The fat cat sat on the mat.

"/.(at)/" => The fat cat sat on the mat.

"/.(at)/g" => The fat cat sat on the mat.

"/.at(.)?$/gm" => The fat

cat sat

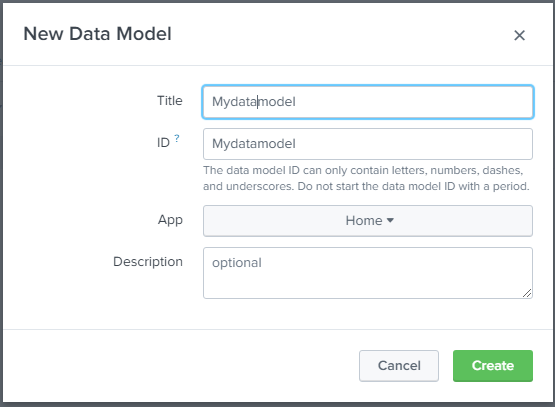
on the mat.

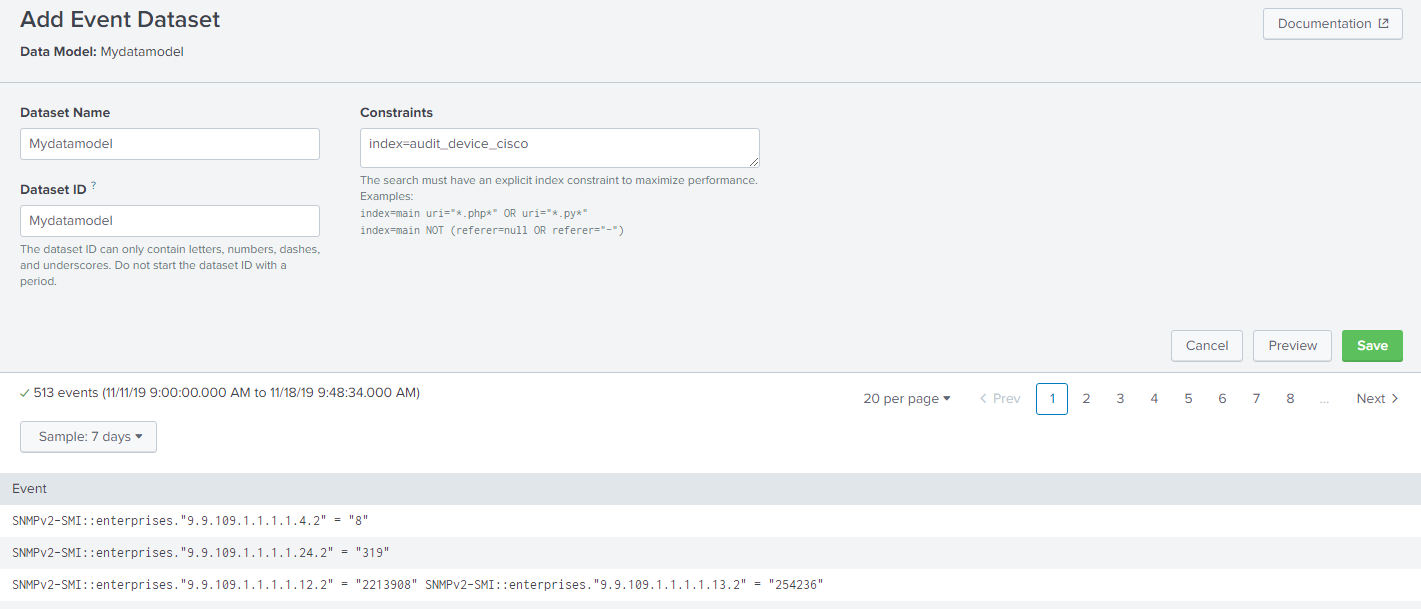
## Một số dạng biểu thức chính quy thường dùng:

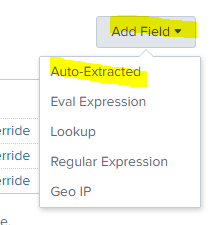
* Lọc ra chính xác một số trong ký tự nhiều chuỗi số: "(?<a>.\*12.\*)\"
* Lọc ra một chuỗi số là duy nhất: ^[^=\n]\*=\s+"(?P<used\_mem>\d+)[^=\n]\*=\s+"(?P<free\_mem>\d+)

# Chuẩn hóa dữ liệu:

* Là sự định nghĩa dữ liệu một cách nhất quán .
* Sử dụng data model để thực hiện công việc chuẩn hóa.
* Data model là một ánh xạ thời gian tìm kiếm có cấu trúc phân cấp về kiến ​​thức ngữ nghĩa về một hoặc nhiều bộ dữ liệu.
* Dùng data model: vào Setting-> data model-> new data model



* Add event tạo lập một dataset
* Add field đi kèm theo với dataset này



* Chọn chế độ trích xuất ra trường giữ liệu.

