検索例

結果のフィルタリング

RAWテキストにfailを含み、かつstatusが0のみのイベントに絞り込む	… | search fail status=0
ホスト値が同じ重複結果を排除	… | dedup host
"__raw"フィールドにプライベートアドレスClass A(10.0.0.0/8)のIPアドレスを含むイベントに絞り込む	… | regex _raw="(?<!\d)10.\d{1,3}\.\d{1,3}\.\d{1,3}(?!\d)"

結果をグループ化

検索結果をクラスタ化し、その"cluster _ count"値で並べ替えし、(データ量の)大きい順に20件のクラスタを返す。	… | cluster t=0.9 showcount=true | sort limit=20 -cluster_count
30秒以内に発生した同じ"host"と"cookie"を持ち、イベントとトランザクションの間隔が5秒以下の結果をグループ化する。	… | transaction host cookie maxspan=30s maxpause=5s
同一IPアドレス(clientip)を持つ結果のうち、最初の結果が"signon"を含み、最後の結果が"purchase"を含む結果をグループ化する。	… | transaction clientip startswith="signon" endswith="purchase"

結果の並び替え

結果のうち最初の20件を返す。	… | head 20
結果セットの順序を逆転する。	… | reverse
まず"ip"値で結果を昇順に並べ替えし、次に"url"値で降順に並べ替えする。	… | sort ip, -url
結果のうち最後の20件を (降順で) 返す。	… | tail 20

レポート

異常値を持つイベントを返す。	… | anomalousvalue action=filter pthresh=0.02
"size"が最大の"delay"を返す。ただし"size"は最高10個のバケットに等分する。	… | chart max(delay) by size bins=10
bar値でfooを分割した各値の最大 (delay) を返す。	… | chart max(delay) over foo by bar
fooの各値に対して最大(delay)を返す。	… | chart max(delay) over foo
範囲外の数値を全て除去する。	… | outlier
同じ"host"値を持つ重複結果を除去し、残った結果の総数を返す。	… | stats dc(host)
"lay"という文字列で終わる (例 delay、xdelay、relay等) 固有フィールドの毎時の平均を返す。	… | stats avg(*lay) by date_hour
各"host"の"cpu"の毎分の平均を計算する。	… | timechart span=1m avg(CPU) by host
"host"ごとの"web"ソース数のタイムチャートを作成する。	… | timechart count by host
"url"フィールドの最も共通な値20件を返す。	… | top limit=20 url
"url"フィールドの最も稀な値20件を返す。	… | rare url

フィールドの追加

距離／時間に速度を設定する。	… | eval velocity=distance/time
正規表現を使用して"from"と"to"フィールドを抽出する。生イベントに"From: Susan To: David"が含まれている場合はfrom=Susan, to=Davidとする。	… | rex field=_raw "From: (?<from>.*) To: (?<to>.*)"
"total _ count"フィールドの"count"のランニング合計を保存する。	… | accum count as total_count
"count"が存在する各イベントについて、countとその前回の値の差分を計算し、その計算結果を"countdiff"に保存する。	… | delta count as countdiff

フィールドのフィルタリング

"host"と"ip"フィールドを維持し、"host"、"ip"の順に表示する。	… | fields + host, ip
"host"と"ip"フィールドを削除する。	… | fields - host, ip

フィールドの加工

"_ ip"フィールドを"IPAddress"に名称変更する。	… | rename _ip as IPAddress
"localhost"で終わるホスト値を"mylocalhost"に変更する。	… | replace *localhost with mylocalhost in host

複数値フィールド

"recipient"フィールドの複数の値を一つの値にまとめる。	… | nomv recipients
"recipients"フィールドの値を複数のフィールド値に分割し、上位のrecipientを表示する。	… | makemv delim="," recipients | top recipients
"recipients"という複数値フィールドの各値ごとに新しい結果を作成する。	… | mvexpand recipients
RecordNumber以外全く同一の結果の各々に対し、その結果を合成し、様々な値が全て入った複数値フィールドをRecordNumberとして設定する。	… | fields EventCode, Category, RecordNumber | mvcombine delim="," RecordNumber
Recipient値の数を求める。	… | eval to_count = mvcount(recipients)
recipientフィールドの最初の電子メールアドレスを検索する。	… | eval recipient_first = mvindex(recipient,0)
.netまたは.orgで終わるrecipient値を検索する。	… | eval netorg_recipients = mvfilter(match(recipient, "\.net$") OR match(recipient, "\.org$"))
foo値、文字列bar、baz値の組合わせを求める。	… | eval newval = mvappend(foo, "bar", baz)
最初のrecipient値が"\.org$"に合致するインデクスを求める。	… | eval orgindex = mvfind(recipient, "\.org$")

ルックアップテーブル

ルックアップテーブルusertogroupの各イベント"user"フィールドの値をルックアップし、その"group"フィールドを抽出する。	… | lookup usertogroup user output group
ルックアップファイル"user.csv"に検索結果を書き込む。	… | outputlookup users.csv
ルックアップファイル"users.csv"から検索結果を読み込む。	… | inputlookup users.csv

正規表現 (REGEX)

正規表現は、検索コマンドrexや、eval関数のmatch()、replace()、フィールドの抽出などで、いろいろな使い方ができる。

正規表現	備考	例	説明
\s	余白スペース	\d\s\d	1桁数字・スペース・1桁数字
\S	非余白スペース	\d\S\d	1桁数字・非白スペース・1桁数字
\d	1桁数字	\d\d\d-\d\d-\d\d\d\d	SSN
\D	数字以外	\D\D\D	数字以外が3つ
\w	単語文字(アルファベット、数字、下線)	\w\w\w	単語文字3つ
\W	非単語文字	\W\W\W	非単語文字3つ
[...]	含まれる文字のいずれか	[a-z0-9#]	aからz、0から9、および#のいずれかの文字。
[^...]	以下の文字を含まない	[^xyx]	x、y、およびz以外のいずれかの文字
*	0以上	\w*	単語文字0以上
+	1以上	\d+	整数
?	0または1	\d\d\d-?\d\d-?\d\d\d\d	ダッシュはオプションのSSN
|	または	\w|\d	単語または1桁数字文字
(?P<var> ...)	命名した抽出値	(?P<ssn>\d\d\d-\d\d\-\d\d\d\d)	SSNを抽出し、"ssn"フィールドに割り当てる。
(?: ...)	論理グループ化	(?:\w|\d)|(?:\d|\w)	単語文字に数字が続くまたは数字に単語文字が続く
^	行の先頭	^\d+	最低数字1個で始まる行
$	行の最後	\d+$	最低数字1個で終わる行
{...}	反復回数	\d{3,5}	3から5桁の間
\	エスケープ	\[	escape the [char [文字をエスケープ

SPLUNK STRPTIMEフォーマット

Strp時間フォーマットはeval関数のstrftime()やstrptime( )、またイベントデータのタイムスタンプに便利である。

時刻	%H	24時制(00から23) (2桁表記)
	%I	12時制 (01 から12) (2桁表記)
	%M	分 (00から59)
	%S	秒 (00から61)
	%N	1秒未満 (%3N＝ミリ秒、%6N＝マイクロ秒、%9N＝ナノ秒)
	%p	午前または午後
	%Z	時間帯 (GMT)
	%s	1970 年1月1日(1308677092)からの秒数
日	%d	日 (01から31) (2桁表記)
	%j	ジュリアン日 (001から366)
	%w	曜日 (0から6)
	%a	曜日略語 (Sun)
	%A	曜日 (Sunday)
月	%b	月名略語 (Jan[1月])
	%B	月名 (January [1月])
	%m	暦月数字表記 (01から12)
年	%y	西暦年2桁表記 (00から99)
	%Y	西暦年4桁表記 (2008)
例	%Y-%m-%d	1998-12-31
	%y-%m-%d	98-12-31
	%b %d, %Y	Jan 24, 2003
	%B %d, %Y	January 24, 2003
	q|%d %b '%y = %Y-%m-%d|	q|25 Feb '03 = 2003-02-25|

splunk> クイック リファレンス ガイド

コンセプト

概要

インデックス時の処理: Splunkはホスト(例"my_machine")上のファイルやポートなどをソースとしてデータを読み取る際に、そのソースをソースタイプ(例:"syslog"、"access_combined"、"apache_error" など)に分類し、タイムスタンプを抽出し、ソースを個々のイベント(例:log events、alerts、など)に分解する。単行／複数行に表示された個々のイベントをディスク上にインデックス化することで後日検索可能とする。

検索時の処理: 検索を開始した際に、マッチしたイベントがディスクから読みこまれ、フィールド (例:code=404、user=david、...) がイベント内のテキストから抽出されてイベントタイプの定義に照合して分類される。検索が返してきたイベントをSplunkの検索言語に変換し作成したレポートはダッシュボード上に保存される。

イベント

イベントをデータの入力単位とする。下記のログファイルではウェブアクティビティログのイベントに相当する。

```
173.26.34.223 - - [01/Jul/2009:12:05:27 -0700] "GET
/trade/app?action=logout HTTP/1.1" 200 2953
```

イベントはあるタイムスタンプに紐付いた一連の値であるとも言える。イベントの多くはせいぜい1、2行の長さであるが、テキスト文書全部やコンフィギュレーションファイル、Javaスタックトレース全部などでは長くなることもある。検索結果を表示する際、Splunkは改行ルールをもとに複数のイベントをどこで区切るかを決めている。

ソース/ソースタイプ

例えば、/var/log/messages や UDP:514など、ファイルやストリームの名称や個々のイベントが発生した要素などがソースとなる。ソースは、access_combined(HTTPウェブサーバログ)などのよく知られたソースタイプに分類するか、もしくはSplunkがそれまでに見たことがないデータやフォーマットに遭遇した時に、都度、ソースタイプを作成しそこに分類する。イベントはソースが異なっても同じソースタイプに分類することができる。例えば、/var/log/messagesファイルからのイベントとudp:514に関するSyslogのインプットはどちらもlinux_syslogというソースタイプに分類できる。

ホスト

ホストはイベントが発生した物理デバイスまたは仮想デバイスの名称を指す。ホスト名を見ればあるデバイスから送出された全てのデータを簡単に検索することができる。

インデックス

Splunkにデータを追加すると、Splunkはそのデータを個々のイベントに分け、それぞれのイベントにタイムスタンプを付与し、インデックスに保存することで、後で検索、解析できるようにする。Splunkにフィードするデータは、"main"インデックスに保存するのがデフォルト設定であるが、入力データごとに異なるインデックス作成／指定することもできる。

フィールド

フィールドはイベントデータ内の名称と値を紐付けた、検索可能なペアを指す。インデックス時と検索時に、Splunkはイベントを処理する際フィールドを自動的に抽出する。インデックス時には、各イベントについてホスト、ソース、ソースタイプなど既定で指定されるいくつかのフィールドを抽出する。一方検索時には、ユーザが定義したパターンやuser_id=jdoeなど明らかにフィールド名／値とわかるペア等の様々なフィールドをイベントデータから抽出することができる。

タグ

タグはフィールド値のエイリアスである。例えば、同じコンピュータに2つのホスト名が付与されている場合、その両方に同じタグ(例:hal9000)をつけることができるが、hal9000で検索すると両方のホスト名値に関連するイベントが返ってくる

イベントタイプ

イベントタイプは検索時にイベントをカテゴリー分類するクロスレファレンスの検索を指す。例えば "error OR worn OR fatal OR fail"という検索定義に"problem"というイベントタイプを定義しておいた場合、検索結果にエラー、警告、失敗が含まれているイベントはeventtype=problemというイベントタイプのフィールド／値をもつことになる。もし、"login"で検索した場合、問題のあったログインにはeventtype=problem が付与されることになる。Eventtype はダイナミックタグのようなもので。Eventtype の検索定義に合致したイベントに紐付けられる。

レポート/ダッシュボード

図表などのフォーマット情報を伴う検索結果は正式名ではないが通常レポートを呼ばれており、複数のレポートをダッシュボードと呼ぶ共通ページに置くことができる。

Apps

splunkbase.com/apps からダウンロード可能

AppsはSplunkのコンフィギュレーション、オブジェクト、およびコードの集合体で、これを使ってSplunk上に様々な環境を構築することができる。

パーミッション/ユーザ/ロール

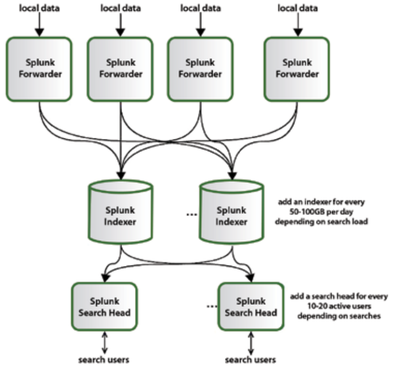
保存済み検索、イベントタイプ、レポート、タグのような保存したSplunkオブジェクトを利用するとデータが豊かになり検索や理解が容易になる。これらのオブジェクトにはユーザ定義可能なパーミッションがあり、データの追加やレポートの編集可否が制御できる。ただし無償ライセンス版のSplunkはユーザ認証をサポートしていない。

トランザクション

トランザクションは、解析を容易にするためにグループ化した一連のイベントを指す。例えば、オンラインショッピングをしているカスタマであれば、クリックごとに同じセッションIDを持つウェブアクセスイベントが作成される。このカスタマのイベントを全部グループ化して一つのトランザクションにまとめると便利である。一つのトランザクションイベントにグループ化しておけば、このカスタマが何分ショッピングしていたか、どの商品を何点購入したか等の統計値を返すことができる。

フォワーダ／インデクサ

フォワーダは、Splunkのインデクサ(単一または複数のインデクサ)にデータ転送するアプリケーションである。
インデクサは、ローカル／リモートいずれのデータでもインデックス化することができる。



検索言語

サーチは一連のコマンドと引数で、各コマンドや引数はその右側"|"(パイプ記号)を使って連結されている。

```
search-args | cmd1 cmd-args | cmd2 cmd-args | ...
```

サーチコマンドは、インデックス化されたデータをフィルタにかけ不要な情報を排除したり、より多くの情報の抽出、値の計算、フォーマット変換、統計解析等に使われる。インデックスからの検索結果はダイナミックに作成されたテーブルと見なすことができる。各検索コマンドで都度そのテーブルの形状が定義される。各行に個々のインデクス化されたイベントが割り当てられ、各フィールドの値がカラムに表示される。カラムには、そのデータに関する基本情報を表わすカラムの他に検索時に都度抽出されるカラムもある。

各検索の先頭にイベントのインデックスの検索を示唆するコマンドがあり、それを使ってキーワード検索(例:"error")、ブール値(例 "error OR failure) NOT success)やフレーズ(例 "database error")、ワイルドカード(例:"fail*"であればfail、fails、failure等に合致)、フィールド値(例:code=404)、不等(例:code!=404 or code>200)の他にフィールド値を意識しない検索(例:code=* や NOT code=*)を実行することができる。例えば:

```
sourcetype="access_combined" error | top 10 uri
```

であれば"error"という単語を含むディスクからインデクス化されたaccess_combinedイベントを検索(検索単語間はANDが想定される)し、検索されたイベントについて最も多く使われている上位10のURI値をレポートする。

サブサーチ

サブサーチとはコマンド自身の検索を起動するコマンド引数で、引数値をその検索結果として親コマンドに返す。サブサーチには角括弧がついている。例えば、前回ログインエラーを出したユーザのSysloグイベント全てを検索したい場合は:

```
sourcetype=syslog [ search login error
                    | return 1 user ]
```

となる。サブサーチ内の"return"コマンドは一つの値しか返してこないのがデフォルト設定であるため、この場合のサブサーチは1ユーザの値しか返してこないが、複数の値を返すためのオプションもある。(例:| return 1 user)

相対時間表記

ユーザインターフェイスにおいてカスタム時刻で時間幅を指定する以外に、検索されたイベントにlatest(直近の)やearliest(最初の)という検索条件を追加して、時間枠を指定することもできる。この相対時間は文字ストリングで時間量(整数と単位)を定義するほか、オプションで時間単位に対してスナップを行うこともできる。

```
[+|-]<time_integer><time_unit>@<snap_time_unit>
```

例えば、"error earliest=-1d@d latest=-1h@h"であれば、昨日(0:00にスナップ)から一時間前まで(○時台にスナップ)に発生した"error"という文字列を含むイベントが検索される。

時間単位:秒(s)、分(m)、時間(h)、日(d)、週(w)、月(mon)、四半期(q)、年(y)"time_integer"のデフォルト値は1である。(例:"m"は"1m"と同義)

スナップ: 指定した時間量を切り下げた値に最も近いもしくは最も直近の時刻を指す。スナップを使うと、指定時刻を越えないもっとも直近の時刻に切り下げられる。例えば、もし今現在が11時59分00秒で、時間@hにスナップすると、12時ではなく11時にスナップされる。@w0は日曜日、@w1は月曜日というように特定の曜日にスナップすることもできる。

よく使われる検索コマンド

コマンド名	説明
chart/timechart	チャート用/時系列チャート用に結果を表示
dedup	特定条件に合致する後続の結果を排除
eval	式の計算。(EVAL関数を参照のこと)
fields	検索結果からフィールドを削除
head/tail	最初/最後のN個の結果を返す
lookup	外部ソースからのフィールド値を追加
rename	特定フィールドの名称を変更。複数フィールドの指定にはワイルドカードを使用
replace	特定のフィールドの値を指定した値で置換
rex	フィールドを抽出するための名前をつけたグループの正規表現
search	結果にフィルターをかけ検索条件に合致するものに絞り込む
sort	検索結果を特定のフィールドで並べ替える
stats	統計値の提供。フィールドでグループ化する
top/rare	最も頻出/最も稀出のフィールド値
transaction	検索結果をトランザクションとしてグループ化

検索の最適化

検索性能向上にはディスクから読み込むデータ量を必要最小限に抑え、そのデータに検索の極力早い段階でフィルタをかけ、必要最小限のデータを対象に処理することが重要である。

複数のデータタイプを跨ぐ検索は滅多に実施しない検索、例えば、ウェブデータとファイヤーウォールデータはそれぞれ別なインデックスを設定するというように、データタイプのインデクスに分けておく。

- できる限り具体的、限定的な検索にする(例:*error*ではなく、fatal_error)
- 時間枠を必要な範囲に限定する(例:-1wではなく-1h)
- 検索のできるだけ早い段階で不用なフィールドはフィルタをかける。
- 計算前のできるだけ早い段階で検索結果にフィルタをかける。
- レポート出力する検索に関しては、タイムラインを計算するフラッシュタイムライン表示ではなく、アドバンスチャート表示を使う。
- フラッシュタイムラインでは、自動フィールド検出を必要としない場合にはそれをオフにしておく。
- サマリーインデックスを使って頻繁に使う値は事前に抽出しておく。
- ディスクI/Oは最速に設定する。

EVAL 関数

eval コマンドは、表現を計算しその結果をフィールドに書き込む。(例 "...| eval force = mass * acceleration")
下記のテーブルには、基本的な演算子 (+ - * / %) やストリングの結合 (例 '...| eval name = last . ", " . last')、ブール演算 (AND OR NOT XOR < > <= >= != = == LIKE)の他に、evalが理解できる関数がリストされている。

関数	説明	例
abs(X)	絶対値Xを返す。	abs(number)
case(X,"Y",…)	引数ペアX Yが設定されている場合、ブール演算子Xが真の時、対応する引数Yを返す。	case(error == 404, "Not found", error == 500,"Internal Server Error", error == 200, "OK")
ceil(X)	数字Xの最大値	ceil(1.9)
cidrmatch("X",Y)	ある特定のサブネットに属するIPアドレスを識別する。	cidrmatch("123.132.32.0/25",ip)
coalesce(X,…)	nullではない最初の値を返す。	coalesce(null(), "Returned val", null())
exact(X)	倍精度浮動小数点を使って表現Xを評価する。	exact(3.14*num)
exp(X)	exを返す	exp(3)
floor(X)	値Xの下限値を返す。	floor(1.9)
if(X,Y,Z)	Xが真の時は引数Yを結果とする。Xが偽の時は引数Zを結果とする。	if(error==200, "OK", "Error")
isbool(X)	Xがブール値の時は真を返す。	isbool(field)
isint(X)	Xが整数の時は真を返す。	isint(field)
isnotnull(X)	Xがnullでは無い時は、真を返す。	isnotnull(field)
isnull(X)	Xがnullの時は、真を返す。	isnull(field)
isnum(X)	Xが数字の時は、真を返す。	isnum(field)
isstr()	Xがストリングの時は、真を返す。	isstr(field)
len(X)	この関数はストリングXの文字数を返す。	len(field)
like(X,"Y")	XがYのSQLiteパターンと類似している時のみ、真を返す。	like(field, "foo%")
ln(X)	自身の自然対数を返す	ln(bytes)
log(X,Y)	引数Yを底とするXの対数を返す。Yのデフォルト値は10とする。	log(number,2)
lower(X)	小文字のXを返す。	lower(username)
ltrim(X,Y)	左からYをトリムしたXを返す。スペースとタブをYのデフォルトとする。	ltrim(" ZZZabcZZ ", " Z")
match(X,Y)	Xが正規表現Yと合致した場合に返す。	match(field, "^\d{1,3}\.\d$")
max(X,…)	最大値を返す。	max(delay, mydelay)
md5(X)	ストリング値XのMD5ハッシュを返す。	md5(field)
min(X,…)	最小値を返す	min(delay, mydelay)
mvcount(X)	Xの値の数を返す。	mvcount(multifield)
mvfilter(X)	ブール表現Xに基づき複数値フィールドをフィルタする。	mvfilter(match(email, "net$"))
mvindex(X,Y,Z)	複数値フィールドXの起点(0ベース)YからZ(オプショナル)までを返す。	mvindex(multifield, 2)
mvjoin(X,Y)	Xが複数値フィールド、Yがストリングデリミタである場合、Xの個々の値をYでジョインする。	mvjoin(foo, ";")
now()	Unix時で表示した現在時刻を返す。	now()
null()	この関数は引数を取らず、nullを返す。	null()
nullif(X,Y)	フィールドXとYを2つの引数とした場合、両者が異なる時はXを返し、それ以外の時はnullを返す。	nullif(fieldA, fieldB)
pi()	定数piを返す。	pi()
pow(X,Y)	XのY乗を返す。	pow(2,10)
random()	0から2147483647までの疑似乱数を返す。	random()
relative_time(X,Y)	Epochtime時間をX、相対時間指定子をYとした場合、Xに適用したepochtime値Yを返す。	relative_time(now(),"-1d@d")
replace(X,Y,Z)	ストリングX中に存在するストリングYの各々を代替ストリングZで置き換えたストリングを返す。	Returns date with the month and day numbers switched, so if the input was 1/12/2009 the return value would be 12/1/2009: replace(date, "^(\d{1,2})/(\d{1,2})/", "\2/\1/")
round(X,Y)	小数点以下Y桁で四捨五入したXを返す。四捨五入して整数にすることをデフォルトとする。	round(3.5)
rtrim(X,Y)	右からYをトリムした文字付きでXを返す。Yが指定されていない場合はスペースとタブをトリムする。	rtrim(" ZZZabcZZ ", " Z")

EVAL 関数 (続き)

関数	説明	例
searchmatch(X)	イベントが検索ストリングXに合致した場合、真を返す。	searchmatch("foo AND bar")
split(X,"Y")	デリミタYをデリミタとする複数値フィールドXを返す。	split(foo, ";")
sqrt(X)	Xの平方根を返す。	sqrt(9)
strftime(X,Y)	Yで定義されたフォーマットで表現されたエポック時刻値Xを返す。	strftime(_time, "%H:%M")
strptime(X,Y)	ストリングXが時刻を表わしている場合、ストリングYからパースされた値を返す。	strptime(timeStr, "%H:%M")
substr(X,Y,Z)	Xフィールドの、(1を底とする)起点Yから Z文字分(オプショナル)のストリングを返す。	substr("string", 1, 3) +substr("string", -3)
time()	百万分の一秒を分解能とするローカル時刻を返す。	time()
tonumber(X,Y)	入力ストリングXをYを底とする値に変換した値を返す。(Yはオプショナル。デフォルト値は10)	tonumber("0A4",16)
tostring(X,Y)	フィールド値Xをストリングとして返す。もしXが数字の場合、それをストリングにフォーマット変換する。Xがブール値の場合は"true"または"false"を返す。Xが数字の場合の引数Yはオプションで、"hex"(Xを16進数に変換)、"commas"(Xをカンマと小数点以下2桁で表示)"、"duration"(秒表現のXをHH:MM:SSという一般表現に変換)のいずれかを取る。	This example returns: foo=615 and foo2=00:10:15: … | eval foo=615 | eval foo2 = tostring(foo, "duration")
trim(X,Y)	両端からYトリムした文字をつけたXを返す。	trim(" ZZZZabcZZ ", " Z")
typeof(X)	データ種別をストリング表現で返す。	This example returns: "NumberStringBoolInvalid": typeof(12)+ typeof("string")+ typeof(1==2)+ typeof(badfield)
upper(X)	Xを大文字で返す。	upper(username)
urldecode(X)	XでデコードしたURLを返す。	urldecode("http%3A%2F%2Fwww.splunk.com%2Fdownload%3Fr%3Dheader")
validate(X,Y,…)	ブール値表現のXとストリングYという2つの引数がある場合、Xが偽の時はXに対応するストリングYを返す。全てが真の時はnullをデフォルトとする。	validate(isint(port), "ERROR: Port is not an integer", port >= 1 AND port <= 65535, "ERROR: Port is out of range")

STATS 関数

図表、統計解析、タイムチャートコマンドと併用する共通統計機能。フィールド名はワイルドカードが使用可能で、例えばavg(*delay)でdelayとxdelayフィールドの平均を計算することができる。

関数	説明
avg(X)	フィールドXの値の平均を返す。
count(X)	フィールドXの発生回数を返す。合致させる特定のフィールド値を指定するにはXをeval(field="value") と定義する。
dc(X)	フィールドXの異なる値の数を返す。
first(X)	フィールドXの最初の値を返す。通常、フィールドの最初の値は、そのフィールドの最も直近のインスタンスになっている。
last(X)	フィールドXの最後の値を返す。
list(X)	複数値を持つフィールドXの全ての値を列記したリストを返す。値は入力イベント順になっている。
max(X)	フィールドXの最大値を返す。Xの値が数字でない場合、最大値は辞書編纂式の順から求める。
median(X)	フィールドXの中央値を返す。
min(X)	フィールドXの最小値を返す。Xが数字でない場合、最小値は辞書編纂式の順から求める。
mode(X)	フィールドXの最も頻出する値を返す。
perc<X>(Y)	フィールドYのXパーセンタイル値を返す。例えば、perc5(total)は"total"というフィールドの5パーセンタイル値を返す。
range(X)	フィールドXの最大値と最小値の差分を返す。
stdev(X)	フィールドXのサンプル標準偏差を返す。
stdevp(X)	フィールドXの全数標準偏差を返す。
sum(X)	フィールドXの値の合計を返す。
sumsq(X)	フィールドXの値の自乗和を返す。
values(X)	複数値フィールドXの異なる値を全て列記したリストを返す。値は辞書編纂式の順。
var(X)	フィールドXのサンプル共分散を返す。