

Fake Data for Real Apps

SimData as a new Simulated Data Generator

David J. Cavuto, CISSP | Principal Product Manager, Data Ecosystem David Poncelow | Senior Software Engineer, Data Ecosystem

Date | Washington, DC

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2017 Splunk Inc. All rights reserved.

.screen?product_id=FL-DSH-01&JSE

Cavuto Bio

Bell Labs

Principal Engineer - Lucent VPN Firewall

► AT&T

Network security and analytics

Narus

- Product Manager Narus Cyber Analytics
- Splunk

3

- Sales Engineer, Security SME
- Principal Product Manager Splunk App for Stream
- Principal Product Manager Data Ecosystem Area
- David J. Cavuto dcavuto@splunk.com





History of Data Simulation at Splunk



splunk

conf2017

Purpose Of Data Simulation In General

Different personas see needs differently

Help user envision how Splunk might address their specific needs

- Simulation must create scenarios that look familiar
- Simulation must create impairments that mimic users impairments
- Help software testers QA apps and add-ons
 - Performance
 - Functional test
- Help software developers build and test apps and add-ons
 - Often, no access to original equipment

EventGen History

What EventGen Did / Does

- Sample-based
 - Replays "sample" events in original log file format
- Written in Python
- Written as a Modular Input
- Replace tokens based on rules



splunk

.conf2017

EventGen Limitations

Challenges users have with EventGen

- Realistic user cohorts was challenging
- Didn't really scale (for testing)
- Multiple scenarios required hacks
- Data correlation required specific scenarios artificially inserted into data

Next Steps in Data Simulation: SimData

SimData Concepts and Execution



SimData Design Goals

- Create Entity/Event based simulations
- Allow for multiple data outputs
- Maintain internal state of entities
- Scale to support load testing on multiple indexers
- Allow external control of running simulations
- Provide backwards-compatibility for EventGen configurations



Persona Targeted By SimData

Field Engineer

- A Splunk Field Engineer will be using SimData to demonstrate Splunk Core and Splunk App
- Simulation Author
 - A Simulation Author will create Simulations (primarily for Field Engineers)

Test/QA

- Test/QA engineer will use SimData to exercise elements of their App for testing purposes
- Test/QA engineer will use SimData to generate large amounts of load to test capacity and performance of distributed Splunk systems

App Developer

 A Splunk App Developer will use SimData to help populate dashboards in their App when a live data source is unavailable



DSL Overall Description

Domain-Specific Language designed specifically for creating simulation elements

- Development experience not required
- Intended that you specify the elements, connections between them, and the messages they pass
- Engine takes care of the rest



splunk

.conf2017

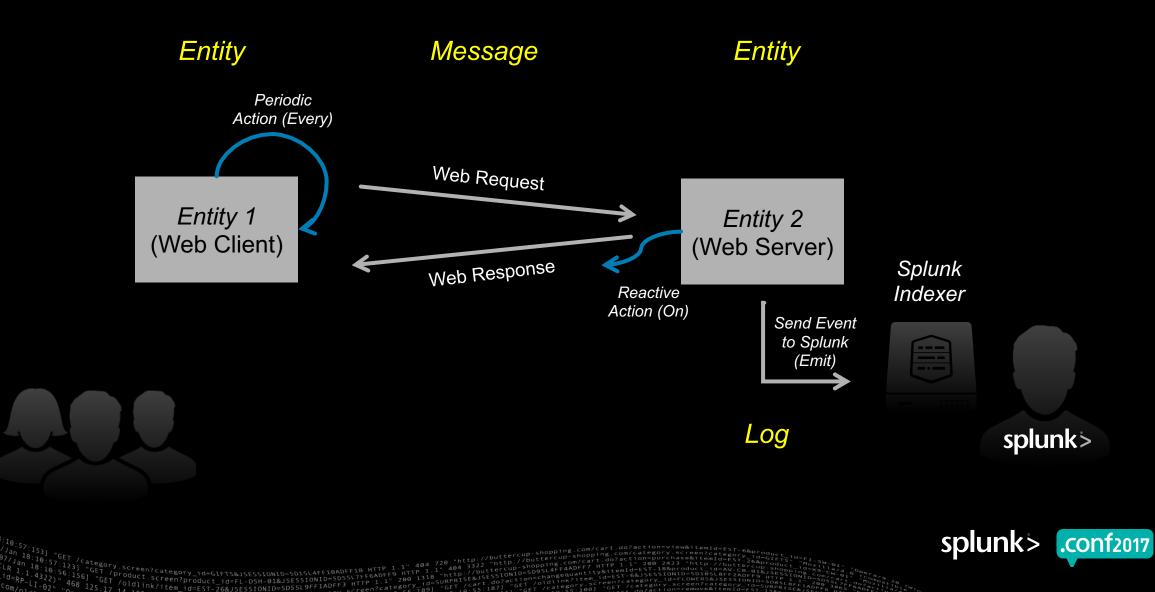
SimData Concepts

Basic concepts of a Data Simulation

- Scene Specifies the connections and messages between entities
- Simulation Specifies the entities and their behaviors
- Entity The functional element of any simulation
 - Init
 - Every
 - On
- Message How Entities communicate to each other
 - Send
 - Respond
- Log How Entities send data to Splunk
 - Emit

Simulation Structure Diagram

How Entities Connect and Communicate



DSL Syntax (v0.8) Yes, it's an eye chart

entity <entity-class-name> {

init { assignment; [assignment;] }; every (time) { statement }; on <message-class-name> { statement };

send <message-class-name> { parameters }
[in (time)];

respond { parameters } [in (time)]; emit <event-class-name> { format-parameters };

/* this defined a function for invoking as
self.method() */

/* and can use 'send', 'emit'. In the right
scope 'respond' too. */

method <method-class-name>([parameters]) {
 statement };
};

message <class-name> { message-parameters };

event <class-name> { statement };

debug(format-parameters);

}[, {

```
scene {
```

```
connect {
```

fromEntity: <entity-class-name>;
messageType: <message-class-name>;
toEntity: <entity-class-name>;

EKOry.screen?category_id=GIFTS&JSESSIONID=SDISL4FF10ADFF10 HTTP 1. ET /Product.screen?product_id=FL_DSH-01&JSESSIONID=SDSSL7FFADFF3 "GET /oldlink?item_id=EST-26&JSESSIONID=SD5SL9FF1ADFF3 HTTP 1.1" 5.17 14 usfreen?categi

```
fromEntity: <entity-class-name>;
messageType: <message-class-</pre>
```

toEntity: <entity-class-name>;

```
};
};
```

name>;

message-parameters: required: <field-name> [,<field-name>]; optional: <field-name>=<default-value> [,<field-name>=<default-value>];

format-parameters: (TBD)

```
statement:
    [statement;]
    [expression;]
```

expression:
 [assignment;]
 [<method-invocation>;]
 [<send-invocation>;]
 [<respond-invocation;]
 [<emit-invocation>;]
 [<debug-invocation;]
 [<conditional>;]

conditional:

if (<evaluation> <comparison> <evaluation>)
then { statement } [else { statement }]

```
assignments:
     <variable> = evaluation;
```

time:

numeric-evaluation (milliseconds | seconds
| minutes | hours)

evaluation:

expression that evaluates to a type object

numeric-evaluation expression that evaluates to a numeric scalar type

```
langauge builtins: (more TBD)
   random(<range>)
   now
   false
   true
   self
   if/else
   milliseconds
   seconds
   minutes
   hours
```

range: (low,high)
 numeric-evaluation, numeric-evaluation
 time, time



splunk

CONf2017

External Control Of Scenarios

- Change a running simulation to show how Splunk / App responds
- Create Sunny / Cloudy / Rainy-day scenarios
- Change parameters of systems
- Create impairments
 - Network outages or congestion
 - Enable / disable hosts
 - Create attack-based disruption

External Control Dashboard Example

Allows Real-time Control of Running Simulation

Welcome to the demo controller!

Basic Controls Advanced										
Alter Database Status. Enable All Databases Disable All Databases										
/user/scene/Database/\$b										
Vuser/scene/Database/\$a										
Alter WebServer Status.										
Set maxPendingRequests for all WebServers 100										
/user/scene/WebServer/\$b 1										
/user/scene/WebServer/\$a 1 100										



Examples and Demonstration

Some syntax and a live simulation



splunk>

.conf2017

Simulation Example 1 – Simulation File

```
message RequestPage {
                                                                             ip: ip;
  required: [requester, page, ip];
}
                                                                         every (3s) {
message PageResponse {
                                                                           send RequestPage {
                                                                             requester: self;
  required: page;
                                                                             page: "home";
                                                                             ip: ip;
event GotPage {
  required: [name, ip];
  template: "{{_time}} {{name}} here- got my page. I'm at
{{ip}}";
                                                                       entity Webserver {
                                                                         init {
event WebRequest {
                                                                           hostname = "webserver01";
  required: [page, server_ip, client_ip];
                                                                           ip = "5.5.5.5";
  template: "{{_time}} {{server_ip}} Received request for
'{{page}}' from {{client_ip}}";
                                                                         on RequestPage {
                                                                           emit WebRequest {
                                                                             page: message.page;
entity User {
                                                                             server_ip: ip;
  init {
                                                                             client_ip: message.ip;
    name = "joe";
    ip = "127.0.0.1";
                                                                           send message.requester PageResponse {
                                                                             page: message.page;
  on PageResponse {
    emit GotPage {
      name: name;
```

Simulation Example 1 – Scene File

Entirely in JSON format

```
"update_interval": 1,
"time_unit": "Seconds".
"entities": [
    "entity_name": "User",
    "initial_state": {
      "name": "Lucy",
      "ip": "185.19.32.1"
    },
    "count": 1
    "entity_name": "User",
    "initial_state": {
      "name": "Edward",
      "ip": "43.19.22.5"
    "count": 1
    "entity_name": "User",
    "initial_state": {
      "name": "Susan",
```

```
"ip": "39.182.16.4"
"count": 1
"entity_name": "User",
"initial_state": {
 "name": "Peter",
 "ip": "212.52.1.198"
"count": 1
"entity_name": "User",
"initial_state": {
 "name": "Mr. Tumnus"
"count": 1
"entity_name": "Webserver",
"initial_state": {
 "hostname": "webserver01"
"count": 1
```

```
}
],
"entity_wirings": [
    {
        "from": "User",
        "to": "Webserver",
        "message_type": "RequestPage",
        "wiring_type": "any",
        "filter": null
     }
],
"default_transport": "Text"
```



Sample Splunk Events

3 of 559,618 eve	ents matched	No Event Sampling	~	~ doL	' II	\$	٠	$\overline{\mathbf{T}}$	🍷 Smart Mode ∽
Events (3)	Patterns	Statistics	Visualization						
Format Timeline	ev −Zoom	Selection × Desele					1	10 milliseconds per column	
				1					

UET /category.screen?category_id=GIFTS&JSESSIONID=SDISL4FF10ADFF10 HTTP 1.1" 404 720 "http://buttercup-shopping.cc S6:156] "GET /product.screen?product_id=GIFTS&JSESSIONID=SDISL4FF10ADFF10 HTTP 1.1" 404 3322 "http://buttercup-" 468 125.17 id intp://buttercup.shopping.cc "no. 12" /oldlink?item_id=EI-268/JSESSIONID=SDISL9FF1ADFF3 HTTP 1.1" 200 1318 "http://buttercup.shopping.cc "no. 12" /oldlink?item_id=EI-268/JSESSIONID=SDISL9F1ADFF3 HTTP 1.1" 200 1318 "http://buttercup.shopping.cc "no. 12" /oldlink?item_id=EI-268/JSESSIONID=SDISL9F1ADFF3 HTTP 1.1" 200 1318 "for /cart.do?arc.do?

:≡ All Fields	i	Time	Event
	>	9/7/17	2017-09-07T15:35:01.6 userID=837 got response code=200 sessionID=c808e135-ea9c-4754-935b-148e4d251967
		3:35:01.000 PM	host = localhost:8088 source = user sourcetype = httpevent
	>	9/7/17	2017-09-07T15:35:01.2 Database Replied with status=ok connectionID=358d3b60-1792-4788-a95b-72ae619ad7f7
		3:35:01.000 PM	host = localhost:8088 source = webserver sourcetype = httpevent
	>	9/7/17	2017-09-07T15:35:00.8 Query Successful. connectionID=358d3b60-1792-4788-a95b-72ae619ad7f7 CPU_percent=0 disk_percent=0 duration=1
		3:35:00.000 PM	host = localhost:8088 source = database sourcetype = httpevent
	≔ All Fields	In the second s	See All Fields > 9/7/17 3:35:01.000 PM > 9/7/17 3:35:01.000 PM > 9/7/17 3:35:01.000 PM > 9/7/17

code 1

a connectionID 1

CPU_percent 1

disk_percent 1

duration 1

a index 1

linecount 1

a punct 3

a sessionID 1

a splunk server 1

splunk> .conf2017

Sample Dashboards



© 2017 SPLUNK INC.

Running Demo 1 Simple Splunk events



g=SURPRISE&JSESSIONIUSDDSSERFERAUTY NETFINIT 200 2423 "http://o 9] "GET //art.do?action=changequantity&itemId=EST-18&product_id=At 9] "GET //art.do?action=changequantityBitem_id=EST-6&JSESSIONIUS-SDIBSEA;



© 2017 SPLUNK INC.

Running Demo 2

dashboards + investigation + scenario changes



buttercup

ET /product.screen?category_id=GIFTS&JSESSIONID=SDISL4FF10ADFF10 HTTP 1.1" 404 3322 Http://buttercup-shopping.com/ "GET /oldlink?item_id=EL-DSH-01&JSESSIONID=SD5SL7FF6ADFf 5.17 Id.verger2category_d=Supping.com/desuping.com/desuping.com/desuping.com/desuping.com/desuping.com/desuping GET /oldlink?item_id=EST-26&JSESSIONID=SD5SL9FF1ADFF3 HTTP 1.1" 200 1318 "http://desuping.com/d



Q&A?

Questions from the Audience



Thank You

Don't forget to rate this session in the .conf2017 mobile app

