Making Sense of Web Fraud With Splunk Stream

An in-depth look at Stream use cases and customer success stories with a focus on stream:http

Jim Apger | Minister of Mayhem – Senior Security Architect
Matthew Joseff | Minster of Reality – Security Architect
Beau Morgan | Senior Sales Engineer

September 28, 2017 | 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.
Agenda

Making Sense of Web Fraud with Splunk Stream

▶ Introductions
▶ Stream Architecture
▶ Web Fraud Detection
▶ Customer Success
What is Stream?

What is rē(ə)l ˈtīm/?
Time is a Nonrenewable Resource

<table>
<thead>
<tr>
<th>Layer</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>7. Application</td>
<td>HTTP, SMTP</td>
</tr>
<tr>
<td>6. Presentation</td>
<td>TLS</td>
</tr>
<tr>
<td>5. Session</td>
<td>SCP</td>
</tr>
<tr>
<td>4. Transport</td>
<td>TCP, UDP</td>
</tr>
<tr>
<td>3. Network</td>
<td>IPv4, IPv6</td>
</tr>
<tr>
<td>2. Data Link</td>
<td>Ethernet</td>
</tr>
<tr>
<td>1. Physical</td>
<td>Ethernet, WiFi</td>
</tr>
</tbody>
</table>
Wire Data Collection / Metadata Generation

End Users → TAP or SPAN

Decryption (If Necessary) → Protocol Decoder

Request/Response → Events → Servers
### Example of Applications in Search

```bash
sourcetype=stream:* | stats count by app
```

<table>
<thead>
<tr>
<th>App</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon_aws</td>
<td>31</td>
</tr>
<tr>
<td>apple</td>
<td>5</td>
</tr>
<tr>
<td>apple_location</td>
<td>2</td>
</tr>
<tr>
<td>dhcp</td>
<td>6</td>
</tr>
<tr>
<td>facebook</td>
<td>6</td>
</tr>
<tr>
<td>flickr</td>
<td>1</td>
</tr>
<tr>
<td>google</td>
<td>58</td>
</tr>
<tr>
<td>google_analytics</td>
<td>4</td>
</tr>
<tr>
<td>google_gen</td>
<td>29</td>
</tr>
<tr>
<td>google_safebrowsing</td>
<td>8</td>
</tr>
<tr>
<td>google_tags</td>
<td>3</td>
</tr>
<tr>
<td>gstatic</td>
<td>11</td>
</tr>
<tr>
<td>http</td>
<td>7945</td>
</tr>
<tr>
<td>http2</td>
<td>11</td>
</tr>
<tr>
<td>https</td>
<td>214</td>
</tr>
<tr>
<td>icloud</td>
<td>8</td>
</tr>
<tr>
<td>imgur</td>
<td>9</td>
</tr>
<tr>
<td>krb5</td>
<td>30</td>
</tr>
<tr>
<td>live_hotmail</td>
<td>6</td>
</tr>
<tr>
<td>norton_update</td>
<td>5</td>
</tr>
<tr>
<td>ntp</td>
<td>2</td>
</tr>
<tr>
<td>ocsp</td>
<td>81</td>
</tr>
<tr>
<td>pinterest</td>
<td>1</td>
</tr>
<tr>
<td>skype</td>
<td>1411</td>
</tr>
<tr>
<td>smb</td>
<td>12</td>
</tr>
<tr>
<td>spdy</td>
<td>4</td>
</tr>
<tr>
<td>spotify</td>
<td>3</td>
</tr>
<tr>
<td>teredo</td>
<td>15</td>
</tr>
<tr>
<td>tumblr</td>
<td>28</td>
</tr>
<tr>
<td>twitter</td>
<td>11</td>
</tr>
<tr>
<td>yahoo</td>
<td>129</td>
</tr>
<tr>
<td>yahoo_search</td>
<td>1</td>
</tr>
<tr>
<td>ymsg_webmessenger</td>
<td>3</td>
</tr>
<tr>
<td>youtube</td>
<td>1</td>
</tr>
</tbody>
</table>
Prebuilt Reporting
What does real time mean to you?

Get visibility into applications performance and user experience

Understand database activity and performance without impacting database operation

Improve security and application intelligence with DNS analytics
Web Fraud Detection With Stream

Using Stream to “sessionize” clicks into complete web clickstreams
Web Fraud Detection

▶ “Sessionization” 101
▶ Use Case #1
  • Profile Change
  • Large Purchase
▶ Use Case #2
  • Multiple Shipments to Same Address
“Sessionizing” 101

- Platform
- Data (clicks)

- Enrich
- Add Context
- Sessionize

- Session Analytics
- History (1st time X)
- Outliers
- Risk
“Sessionizing” 101

From this mess of clicks To these nice clickstreams
“Sessionizing” 101

SessionID is changing!

Pre-Login Clicks
SessionID=1234

Login Page Click
SessionID=1234,5678
Username=Newman
LoginSuccess=1

Post Login Clicks
SessionID=5678,9012

CheckOut Clicks
SessionID=9012
OrderTotal=$435.23
CCNum=<hash>
Pro Tip: the "transaction" command will accept a multivalue field and will link on **any** of the specified field values.

<table>
<thead>
<tr>
<th>sessionID</th>
<th>1111</th>
<th>2222</th>
<th>3333</th>
<th>4444</th>
</tr>
</thead>
<tbody>
<tr>
<td>transaction sessionID</td>
<td>1111</td>
<td>2222</td>
<td>3333</td>
<td>4444</td>
</tr>
</tbody>
</table>
“Sessionizing” 101

```
rex field=cookie max_match=5 "fronten[^=]+=(?<splSessionid>w+)"
```
“Sessionizing” 101

| rex field=cookie max_match=5 "fronten[^=]+=(?<splSessionid>\w+)" | transaction splSessionid maxpause=10m maxspan=2h

```
splunk_web CSRF token: 10726947074221727; session_id=8000;7b0b5c7a397690cf03da2764458650335058967;
splunk_web=0d_BFD1ый73c_Mq8iteratedA50_dF1116_Ja8Mepq8Z3Pn7K3BVLY9789888b1Lb_v6C1qu8eJAD4FfEJVF4o9ju18b6X8n92Kt1j_TxPil+sdWb9s3hXpCgjyWbK3b0+4Oe58j6hOF;
frontend_cid=4XXX2zP8hbcB8hdf; external_no_cache=1; frontend=61d70i0ij6elsh6mgehkek13

splunk_web CSRF token: 10726947074221727; session_id=8000;7b0b5c7a397690cf03da2764458650335058967;
splunk_web=0d_BFD1ый73c_Mq8iteratedA50_dF1116_Ja8Mepq8Z3Pn7K3BVLY9789888b1Lb_v6C1qu8eJAD4FfEJVF4o9ju18b6X8n92Kt1j_TxPil+sdWb9s3hXpCgjyWbK3b0+4Oe58j6hOF;
frontend_cid=4XXX2zP8hbcB8hdf; external_no_cache=1; frontend=61d70i0ij6elsh6mgehkek13

splunk_web CSRF token: 10726947074221727; session_id=8000;7b0b5c7a397690cf03da2764458650335058967;
splunk_web=0d_BFD1ый73c_Mq8iteratedA50_dF1116_Ja8Mepq8Z3Pn7K3BVLY9789888b1Lb_v6C1qu8eJAD4FfEJVF4o9ju18b6X8n92Kt1j_TxPil+sdWb9s3hXpCgjyWbK3b0+4Oe58j6hOF;
frontend_cid=4XXX2zP8hbcB8hdf; external_no_cache=1; frontend=61d70i0ij6elsh6mgehkek13
```
Use Case: Profile Change + Large Purchase

Single Stream with Profile Edit and Large Purchase

<table>
<thead>
<tr>
<th>cookie</th>
<th>session_id</th>
<th>url_path</th>
<th>time</th>
<th>sp1_username</th>
<th>sp1_password</th>
<th>sp2_loginTime</th>
<th>sp2_profileEditTime</th>
<th>sp2_purchaseTime</th>
<th>sp2_fraudScore</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/login/</td>
<td>2017-09-08 08:42:07</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/loginPost/</td>
<td>2017-09-08 08:42:57</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/logout/</td>
<td>2017-09-08 08:43:53</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/logout/post/</td>
<td>2017-09-08 08:44:09</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/wishlist/</td>
<td>2017-09-08 08:44:15</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/wishlist/post/</td>
<td>2017-09-08 08:44:46</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/edit/</td>
<td>2017-09-08 08:44:87</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/edit/post/</td>
<td>2017-09-08 08:45:04</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/edit/success/</td>
<td>2017-09-08 08:45:22</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/edit/success/post/</td>
<td>2017-09-08 08:45:46</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/logout/</td>
<td>2017-09-08 08:45:46</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/logout/post/</td>
<td>2017-09-08 08:45:46</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/wishlist/</td>
<td>2017-09-08 08:44:15</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/wishlist/post/</td>
<td>2017-09-08 08:44:46</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/edit/</td>
<td>2017-09-08 08:44:87</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/edit/post/</td>
<td>2017-09-08 08:45:04</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/edit/success/</td>
<td>2017-09-08 08:45:22</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/index.php/customer/account/edit/success/post/</td>
<td>2017-09-08 08:45:46</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

18022016180.045793
18022016180.045793
Use Case: Profile Change + Large Purchase

GOAL: Find a profile change occurring before a large purchase

```sn Musical
index=wiredata dest_port=80 OR dest_port=443  http_content_type="text/html; charset=UTF-8"
[search index=wiredata sourcetype=stream:http dest_port=80 OR dest_port=443 splGrandtotal="**" http_content_type="text/html; charset=UTF-8" |fields splGrandtotal src_ip|rex field=splGrandtotal mode=sed "s/\//\"|where splGrandtotal > 500|stats values(src_ip) as src_ip]
|reverse
|table _time cookie uri_path splGrandtotal splUsername
|rex field=cookie max_match=5 "fronten[^=]+=(?<splSessionid>\w+)"
|eval splLoginTime=if(uri_path="/index.php/customer/account/loginPost/",_time,NIL)
|eval splPurchaseTime=if(uri_path="/index.php/checkout/onepage/success/",_time,NIL)
|eval splProfileEditTime=if(uri_path="/index.php/customer/account/editPost/",_time,NIL)
|rex field=splGrandtotal mode=sed "s/\//\"
|transaction splSessionid maxpause=20m maxspan=2h
|where splProfileEditTime < splPurchaseTime AND splGrandtotal > 500
|eval PurchaseTime=strftime(splPurchaseTime,%F %T"
|eval LoginTime=strftime(splLoginTime,%F %T"
|eval ProfileEditTime=strftime(splProfileEditTime,%F %T"
|table splUsername duration PurchaseTime LoginTime ProfileEditTime splGrandtotal splSessionid uri_path
```
Use Case: Profile Change + Large Purchase

**GOAL:** Find a profile change occurring before a large purchase

```splunk
index=wiredata dest_port=80 OR dest_port=443 http_content_type="text/html; charset=UTF-8"
[search index=wiredata sourcetype=stream:http dest_port=80 OR dest_port=443 splGrandtotal="*"
http_content_type="text/html; charset=UTF-8" ]fields splGrandtotal src_ip|rex field=splGrandtotal mode=sed "s/\//\n"
| where splGrandtotal > 500|stats values(src_ip) as src_ip]
|reverse
|table _time cookie uri_path splGrandtotal splUsername
|rex field=cookie max_match=5 "fronten[^=]+=(?<splSessionid>\w+)"
|eval splLoginTime=if(uri_path="/index.php/customer/account/loginPost/",_time,NULL)
|eval splPurchaseTime=if(uri_path="/index.php/checkout/onepage/success/",_time,NULL)
|eval splProfileEditTime=if(uri_path="/index.php/customer/account/editPost/",_time,NULL)
|rex field=splGrandtotal mode=sed "s/\n/\""
|transaction splSessionid maxpause=20m maxspan=2h
|where splProfileEditTime < splPurchaseTime AND splGrandtotal > 500
|eval PurchaseTime=strftime(splPurchaseTime,"%F %T")
|eval LoginTime=strftime(splLoginTime,"%F %T")
|eval ProfileEditTime=strftime(splProfileEditTime,"%F %T")
|table splUsername duration PurchaseTime LoginTime ProfileEditTime splGrandtotal splSessionid uri_path
```
Use Case: Profile Change + Large Purchase

GOAL: Find a profile change occurring before a large purchase

index=wiredata dest_port=80 OR dest_port=443 http_content_type="text/html; charset=UTF-8"
## Use Case: Profile Change + Large Purchase

### Remove the 'where' constraint

### Limit to 5 URIs to make pretty
Use Case: Multiple Accounts -> Same Address

How does this happen?

- Account Take-Over
- Account Harvesting
- Often preceded by Account Peeking

Same Address
Unsuspecting Mule
Use Case: Multiple Accounts -> Same Address

```
index=wiredata sourcetype="stream:http" dest_port=80 OR dest_port=443
[search index=wiredata sourcetype=stream:http dest_port=80 OR dest_port=443 splShippingStreet="*"
  http_content_type="text/html; charset=UTF-8" |stats values(src_ip) as src_ip]
|fields splPurchaseTime splUsername splPassword splSessionid splGrandtotal splShipping* cookie uri_path dest_content
|rex field=cookie max_match=5 "fronten[^=]+=(?<splSessionid\\w+)"
|eval splPurchaseTime=if(uri_path="/index.php/checkout/onepage/success/",_time,.NULL)
|rex field=dest_content "Your order # is:[^>]\+[<]?<splOrderNumber>[^>]\+[<]"
|reverse
|transaction splSessionid maxpause=10m maxspan=2h
|eval PurchaseTime=strftime(splPurchaseTime,"%F %T")
|stats list(splOrderNumber) as OrderNum list(splUsername) as Username list(splPassword) as Password
list(splShippingFirstname) as ShippingFirstname list(splShippingLastname) as ShippingLastname list(splShippingPhone) as ShippingPhone list(PurchaseTime) as PurchaseTime count by splShippingStreet splShippingCity splShippingZip
splShippingCountry
|where count > 2
```
Use Case: Multiple Accounts -> Same Address

index=wiredata sourcetype="stream:http" dest_port=80 OR dest_port=443
[search index=wiredata sourcetype=stream:http dest_port=80 OR dest_port=443 splShippingStreet="*"
 http_content_type="text/html; charset=UTF-8" |stats values(src_ip) as src_ip]
|fields splPurchaseTime splUsername splPassword splSessionid splGrandtotal splShipping* cookie uri_path dest_content
|rex field=cookie max_match=5 "frontend[^=]+=(?<splSessionid>\w+)"
|eval splPurchaseTime=if(uri_path="/index.php/checkout/onepage/success/",_time,NULL)
|rex field=dest_content "Your order # is:[^>]+>(?<splOrderNumber>[^>]+)<" |reverse
|transaction splSessionid maxpause=10m maxspan=2h
|eval PurchaseTime=strftime(splPurchaseTime,"%F %T")
|stats list(splOrderNumber) as OrderNum list(splUsername) as Username list(splPassword) as Password
 list(splShippingFirstname) as ShippingFirstname list(splShippingLastname) as ShippingLstname list(splShippingPhone) as ShippingPhone list(PurchaseTime) as PurchaseTime count by splShippingStreet splShippingCity splShippingZip splShippingCountry
|where count > 2
Use Case: Multiple Accounts -> Same Address

```
index=wiredata sourcetype="stream:http" dest_port=80 OR dest_port=443
|search index=wiredata sourcetype=stream:http dest_port=80 OR dest_port=443 splShippingStreet="*"
|http_content_type="text/html; charset=UTF-8" |stats values(src_ip) as src_ip
|fields splPurchaseTime splUsername splPassword splSessionid splGrandtotal splShipping* cookie uri_path dest_content
|rex field=cookie max_match=5 "fronten[^=]+=(?<splSessionid>\w+)"
|eval splPurchaseTime=if(uri_path="/index.php/checkout/onepage/success/",_time,NULL)
|rex field=dest_content "Your order # is:[^>]+=([^>]*)"((?<splOrderNumber>[^>]+)<")
|reverse
|transaction splSessionid maxpause=10m maxspan=2h
|eval PurchaseTime=strftime(splPurchaseTime,"%F %T")
|stats list(splOrderNumber) as OrderNum list(splUsername) as Username list(splPassword) as Password
list(splShippingFirstname) as ShippingFirstname list(splShippingLastname) as ShippingLstname list(splShippingPhone) as ShippingPhone list(PurchaseTime) as PurchaseTime count by splShippingStreet splShippingCity splShippingZip splShippingCountry
|where count > 2
```
Use Case: Multiple Accounts -> Same Address

```
index=wiredata sourcetype="stream:http" dest_port=80 OR dest_port=443
[search index=wiredata sourcetype=stream:http dest_port=80 OR dest_port=443 splShippingStreet="**"
http_content_type="text/html; charset=UTF-8" |stats values(src_ip) as src_ip]
|fields splPurchaseTime splUsername splPassword splSessionId splGrandtotal splShipping* cookie uri_path dest_content
|rex field=cookie max_match=5 "fronten[^=]+=(?<splSessionId>\w+))"
|eval splPurchaseTime=if(uri_path="/index.php/checkout/onepage/success/", _time, NULL)
|rex field=dest_content "Your order # is:[^>]+>(?<splOrderNumber>[^>]+)"
|reverse
|transaction splSessionId maxpause=10m maxspan=2h
|eval PurchaseTime=strftime(splPurchaseTime,"%F %T")
|stats list(splOrderNumber) as OrderNum list(splUsername) as Username list(splPassword) as Password
list(splShippingFirstname) as ShippingFirstname list(splShippingLastname) as ShippingLastname list(splShippingPhone) as ShippingPhone list(PurchaseTime) as PurchaseTime count by splShippingStreet splShippingCity splShippingZip
|where count > 2
```
Customer Success With Stream

From Fraud to DevOps
Use Cases

It all started with EASY fraud detection use cases…

- Single IP address using multiple user IDs
- Single user ID being accessed by < 3 IP addresses
- Multiple user IDs using the same CC number (hashed) for purchases
- Single user ID with multiple shipping addresses used
- Multiple user IDs using same complex password (hashed)
- User IDs created with email accounts with high shannon entropy scores
Stream: HTTP

Everything you need in one sourcetype

- HTTP headers
- HTTP arguments
- Form data
- Cookie data
- HTML content
Customer Architecture

Deploying Stream with Splunk Cloud

- Used Deployment Server to push out Splunk_TA_stream to 120+ webservers
- Configure Stream app on Deployment server for pushing stream configs
- Install Stream app on Cloud for dashboards
- Gathering data in minutes
DevOps Use Cases
Enter DevOps and stream:tds (Tabular Data Stream)

- Troubleshoot application calls to internal and 3rd party systems
- Real-time capacity planning for calls to 3rd party systems where license is usage based
- Provide visibility in transition to microservices architecture
Eliminated lost revenue associated to outages which previously happened 4-5 times a year

Saved 100 FTE hours a month in productivity

40% reduction in API/SaaS usage by showing real-time application tuning effectiveness instead of monthly reports from SaaS provider

Rapid transition to microservices architecture and speed of DevOps CI/CD

80% reduction in DevOps lifecycle

DevOps Success Metrics

“Stream is essential to any organization migrating to microservices”
What Now?

Call to Action

- Visit the Fraud Detection Booth in the source=*Pavilion
- Contact your Splunk representative for a **free** Fraud Workshop
- Attend the next Fraud talk: “Using Splunk for Retail Banking Cross Channel Fraud Analysis, Detection & Investigation” tomorrow, Thursday at 1135
- Schedule a BOTS 2.0 engagement: Now with Fraud use cases!
Thank You

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