



splunk>

How Did the Timesheet Catch the Spy?

Joke or the downfall of the insider threat

Ben Lovley | Network Forensics - Ministry of Defence (MoD)



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. © 2018 Splunk Inc. All rights reserved.

Our Speakers

**BEN LOVLEY**

Network Forensics, Incident Response and Enterprise Threat Hunt

**ANDREW 'MAC' MCALLISTER**

Customer Success - UK Public Sector

MODERATED BY GREEN TRACKSUIT

Ben Lovely

MoD Network Forensics & Hunt analyst

- ▶ 2,5 Years lead Forensic analyst
 - Covering Network, Mobile forensics as well as Enterprise Threat Hunt
 - ▶ 10+ Years working in MoD cyber
 - ▶ 3 Years active Splunk use from analysis through to Architecture
 - ▶ IoT enthusiast
 - ▶ Keen motorcyclist!





Andrew 'Mac' McAllister

- ▶ Splunk Customer Success
 - Management - UK Public Sector
- ▶ 10+ years in the UK military
- ▶ Weapons Engineering
- ▶ TOGAF-Certified Enterprise Architect
- ▶ Chartered Engineer

Slide in the middle

Centre of Excellence and Innovation



splunk® listen to your data™

Slide in the middle

Centre of Excellence and Innovation



- ▶ Shorten the Feedback Loop
- ▶ Decrease Wasted Cycles
- ▶ Increase Innovation
- ▶ Increase Trust
- ▶ Become a hero!

...Which Has Enabled New Thinking

Within a single customer

- ▶ The next iteration: cross-correlation with timesheets, logins

- ▶ Redundant network ports
 - ▶ Cooling Fan Failures
 - ▶ Enterprise Power and Environmental Analytics
 - ▶ Splunk in Your Pocket
 - ▶ Raspberry Pi Desk Utilization

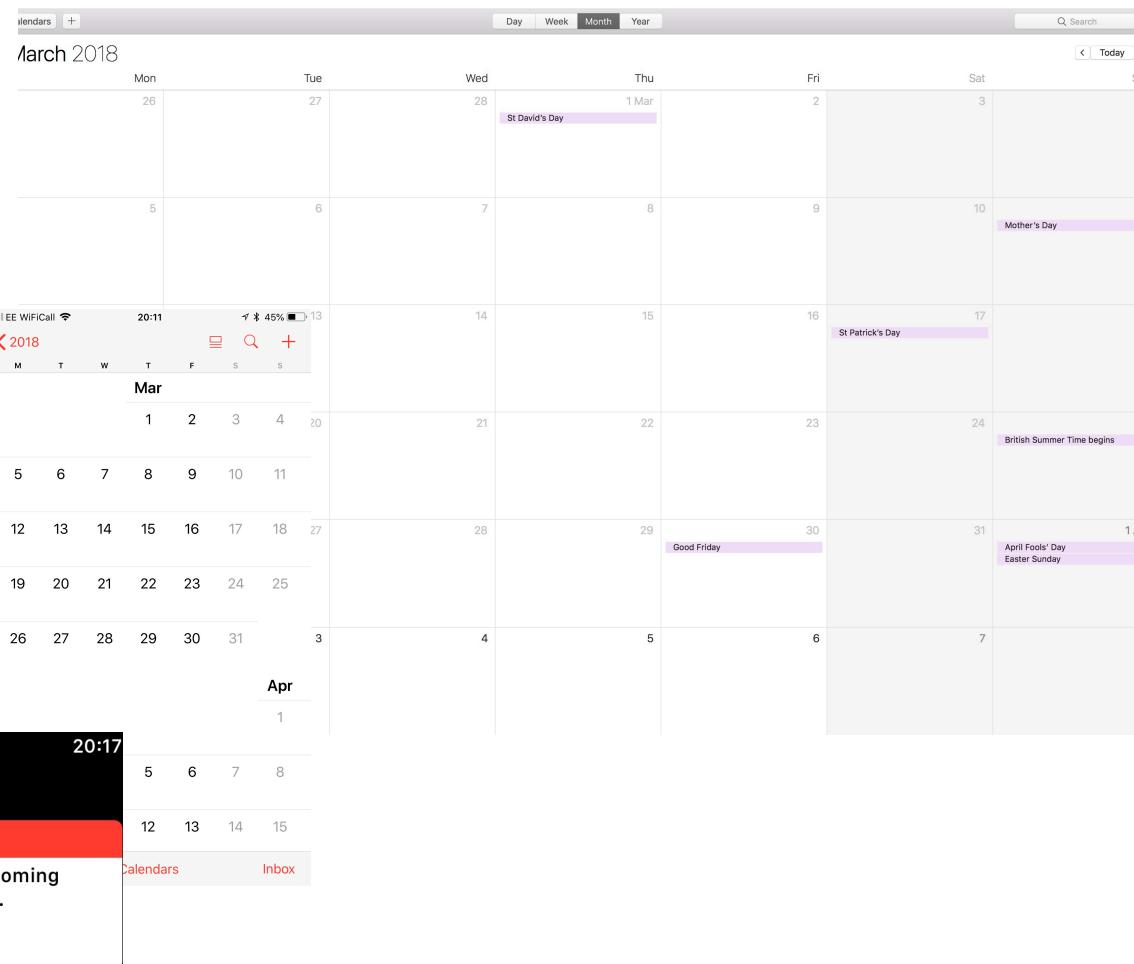
Time Recording & Security

The basic setup

How Did the Timesheet Catch the Spy?

Joke or the downfall of the insider threat

- ▶ Nearly all enterprises from big to small require employees to account for their time for a magnitude of reasons
- ▶ The objective is to utilise current hardware deployments (CYOD/BYOD) to capture this information, push it to Splunk and transform it into intelligence
- ▶ Meaning no more filling in time sheets manually!



The Big Idea

Using hardware the user already has



Capture data from deployed hardware



Push the data to Splunk



Splunk Enterprise
indexes data and
generates
intelligence



Intelligence used to capture the inside threat. Operations can continue

Why This...

Unlike other projects, its “not just because I can”

▶ Personnel Security

- Real-time geo-fencing data is ideal to aid in protection of personnel and the threat towards them
 - E-privacy

▶ Time Management

- Overtime, annual leave, sickness etc. these can all be calculated without the user interacting with time sheets
 - Anomalies in behaviour can be examined deeper

► Money Saving

- See what rooms are being used and where the power can be saved

A Royal Navy submariner who was disgruntled after being denied a promotion has admitted gathering military secrets to pass to the Russians.



Hardware and Software

The basic setup



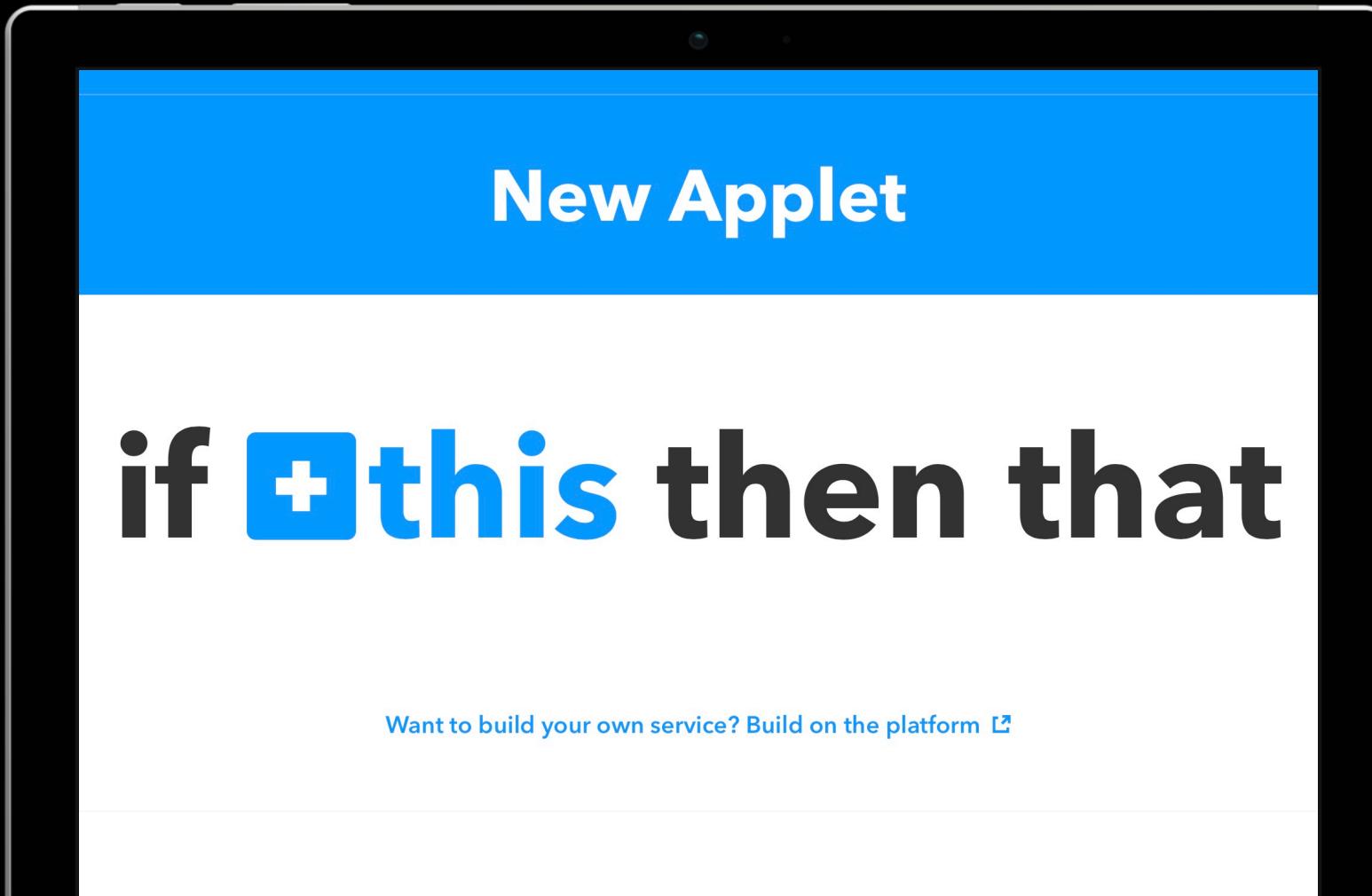
Hardware

Smartwatches
Smartphones
Tablets

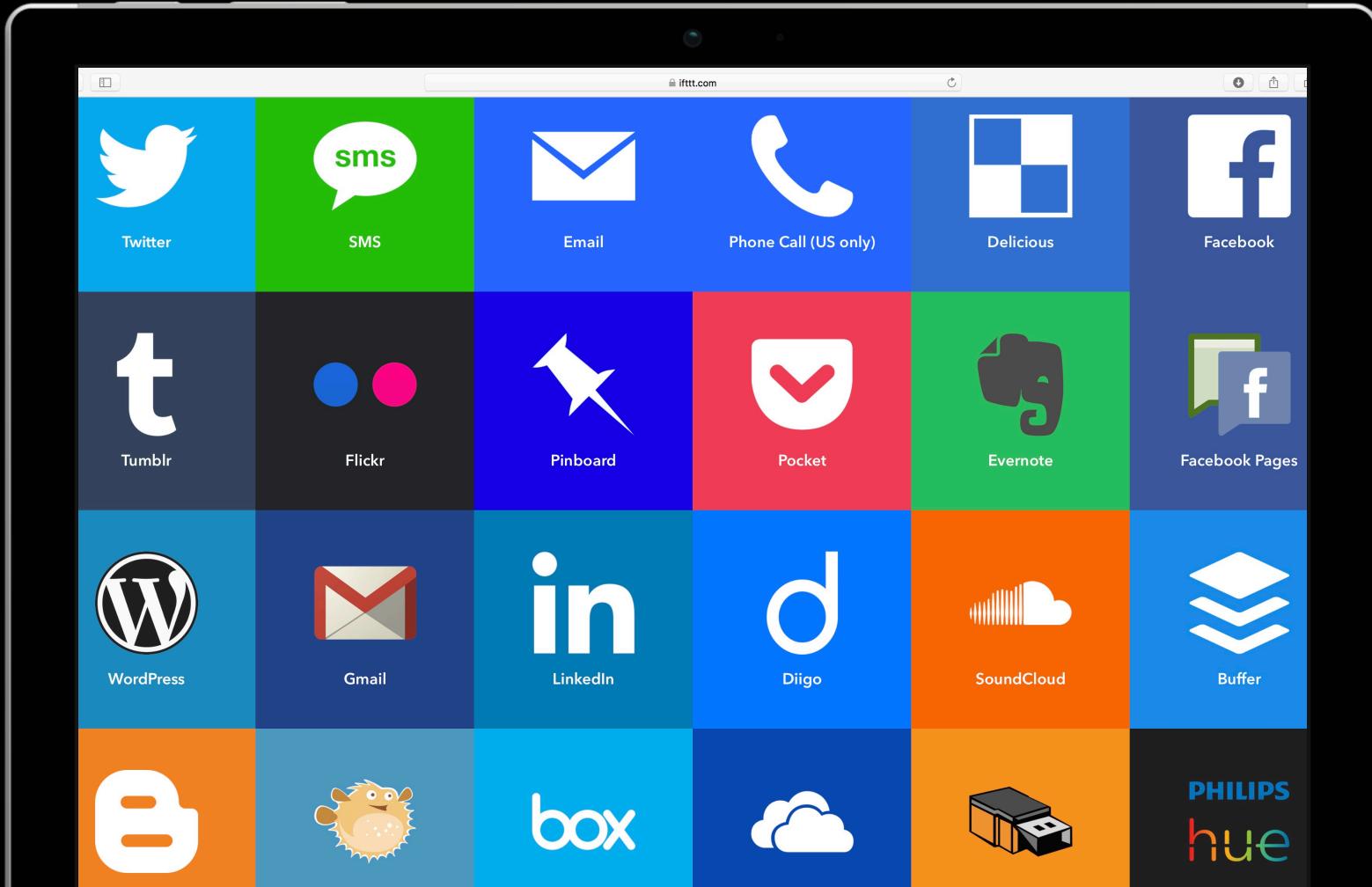


Software / Application

The smart part



- ▶ Not strictly Software, IFTTT is an app
 - If This Then That
- ▶ Create custom chains of conditional statements
 - Automate tasks
 - Such as Geo fence locations
- ▶ First thing to do is choose If This



- ▶ Many apps to choose from
- ▶ Can use device functions
 - Location, SMS etc

- ▶ Choose location

The image shows a mobile application interface. At the top, the title "Choose a service" is displayed in a large, bold, dark font. Below it, the text "Step 1 of 6" is shown in a smaller, gray font. A search bar is present, containing the text "location" with a magnifying glass icon. Below the search bar is a blue rectangular button with a white location pin icon and the word "Location" in white text. The background of the app is white, and the overall design is clean and modern.

[Back](#)

Choose trigger

Step 2 of 6

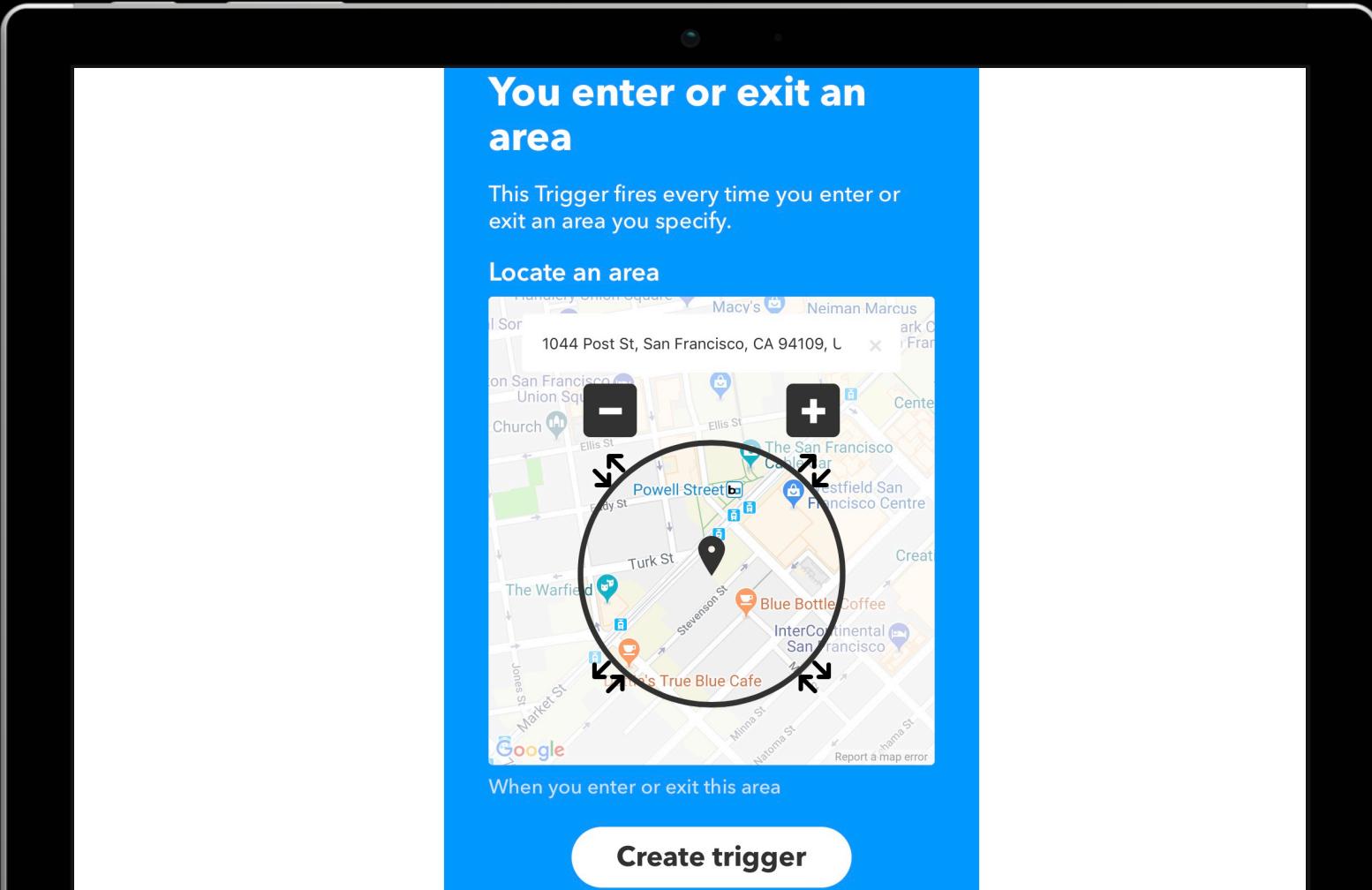
You enter an area
This Trigger fires every time you enter an area you specify.

You exit an area
This Trigger fires every time you exit an area you specify.

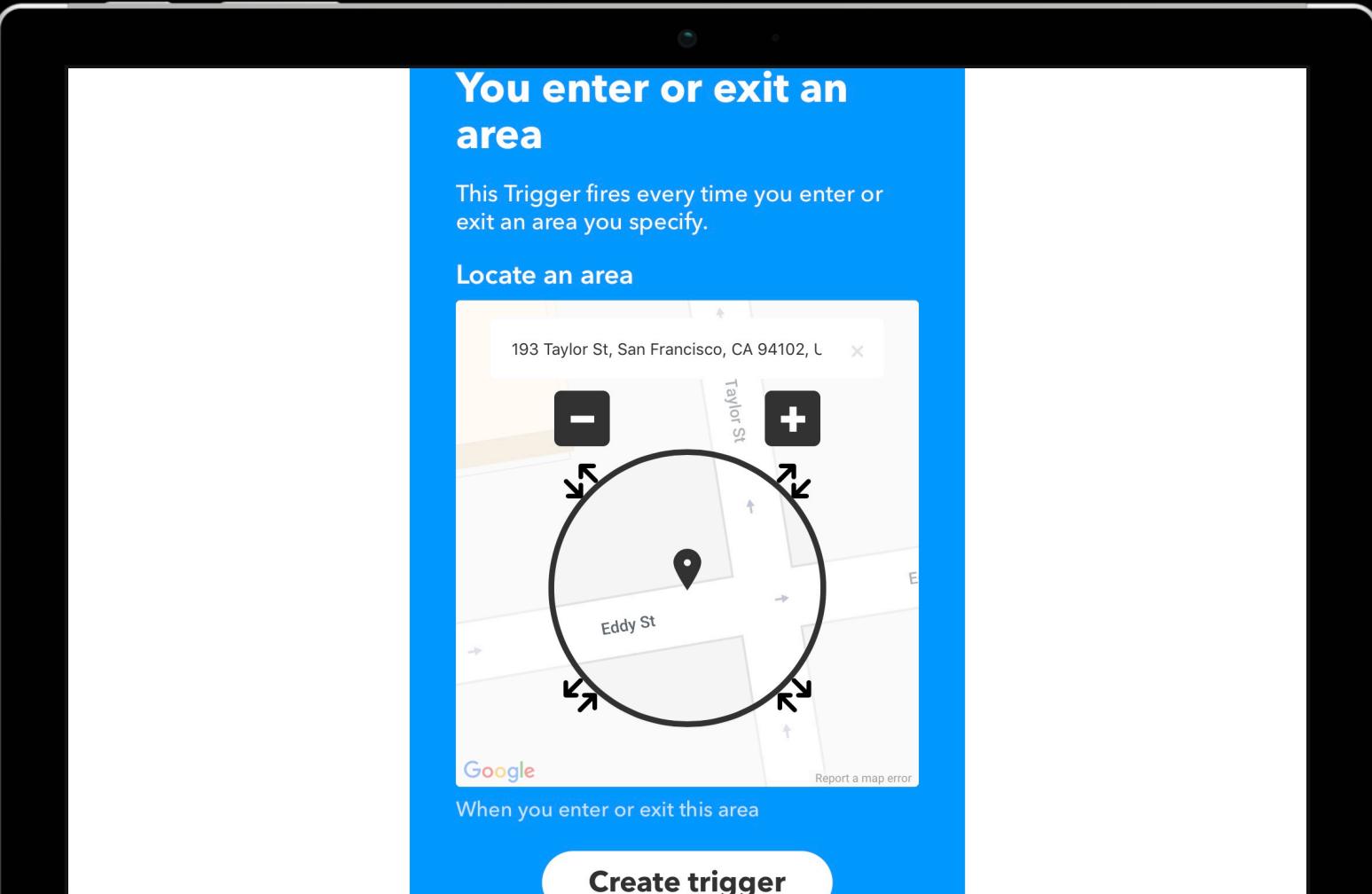
You enter or exit an area
This Trigger fires every time you enter or exit an area you specify.

Don't see what you're looking for? [Suggest a new trigger](#)

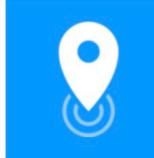
- ▶ 3 options
 - Enter
 - Exit
 - Enter or Exit



- ▶ Select area to use as geo fence



- ▶ Can zoom to good scale for finer area

if  then that

- ▶ Now this is done its time to choose that
 - The action for IFTTT to take

Choose action service

Step 3 of 6

sheet

 Google Sheets

- ▶ Choose Google Sheet
 - Details of Google drive need to be entered

Choose action

Step 4 of 6

Add row to spreadsheet

This action will add a single row to the bottom of the first worksheet of a spreadsheet you specify. Note: a new spreadsheet is created after 2000 rows.

Update cell in spreadsheet

This action will update a single cell in the first worksheet of a spreadsheet you specify. Note: a new spreadsheet is created if the file doesn't exist.

Don't see what you're looking for? Suggest a new action

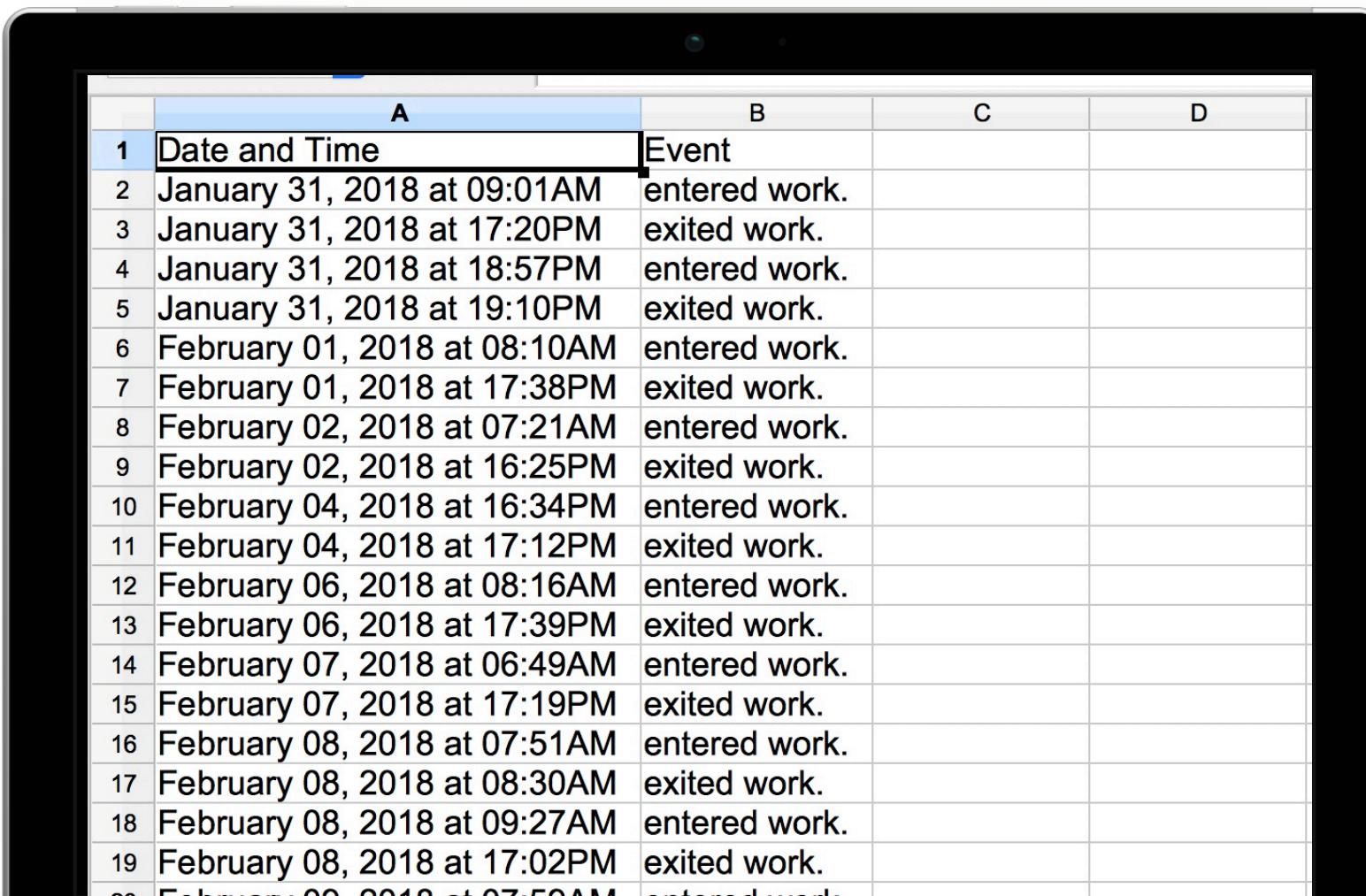
- ▶ 2 options
 - Add row
 - Update cell



- ▶ Customise the Google sheet
 - Name of document
 - Details to be entered by IFTTT
 - Location of document

The Output

CSV with the chosen data



	A	B	C	D
1	Date and Time	Event		
2	January 31, 2018 at 09:01AM	entered work.		
3	January 31, 2018 at 17:20PM	exited work.		
4	January 31, 2018 at 18:57PM	entered work.		
5	January 31, 2018 at 19:10PM	exited work.		
6	February 01, 2018 at 08:10AM	entered work.		
7	February 01, 2018 at 17:38PM	exited work.		
8	February 02, 2018 at 07:21AM	entered work.		
9	February 02, 2018 at 16:25PM	exited work.		
10	February 04, 2018 at 16:34PM	entered work.		
11	February 04, 2018 at 17:12PM	exited work.		
12	February 06, 2018 at 08:16AM	entered work.		
13	February 06, 2018 at 17:39PM	exited work.		
14	February 07, 2018 at 06:49AM	entered work.		
15	February 07, 2018 at 17:19PM	exited work.		
16	February 08, 2018 at 07:51AM	entered work.		
17	February 08, 2018 at 08:30AM	exited work.		
18	February 08, 2018 at 09:27AM	entered work.		
19	February 08, 2018 at 17:02PM	exited work.		
20	February 09, 2018 at 07:50AM	entered work.		

- ▶ Using the settings previously, this is what you would get
 - 2 columns
 - Date and Time
 - Event

Now the Splunk magic

Indexing, Formating and Analysing



The Splunk Part

All sorts of intelligence can be generated with the simple CSV generated

- ▶ The data from the CSV can be manipulated to give you so much detailed data, such as:
 - Entry time
 - Exit time
 - Overtime calculations
 - Average working hours
 - Unusual work times
 - Visually stunning graphics for the hierarchy

The main point is....Unusual working hours!

I will show you how this simple CSV can highlight unusual behaviour and can be alerted on for further analysis

The Basics

Google sheet is indexed as CSV

The screenshot shows the Splunk web interface with a search bar containing 'index=work-hours'. Below the search bar, it says '46 events (29/05/2018 00:00:00.000 to 28/06/2018 11:52:54.000) No Event Sampling'. The 'Events (46)' tab is selected. The main area displays a table with columns: Time, Event, and several source and sourcetype details. The table shows entries for work hours, such as 'exited work.' and 'entered work.' at specific dates and times. On the left, there are sections for 'SELECTED FIELDS' and 'INTERESTING FIELDS', both listing various date-related fields like 'host', 'source', and 'sourcetype'.

Time	Event	host = local	source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv	sourcetype = work-hours
27/06/2018 18:24:00.000	June 27, 2018 at 17:24PM exited work.			
27/06/2018 09:21:00.000	June 27, 2018 at 08:21AM entered work.			
26/06/2018 19:35:00.000	June 26, 2018 at 18:35PM exited work.			
26/06/2018 12:42:45.000	Date and Time Event 12:42:45.000	host = local	source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv	sourcetype = work-hours
26/06/2018 12:40:28.000	Date and Time Event 12:40:28.000	host = local	source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv	sourcetype = work-hours
26/06/2018 09:23:00.000	June 26, 2018 at 08:23AM entered work.	host = local	source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv	sourcetype = work-hours
25/06/2018 18:17:00.000	June 25, 2018 at 17:17PM exited work.	host = local	source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv	sourcetype = work-hours

- ▶ Data added as CSV using standard 'add data'
- ▶ Simple 2 columns

Step 1

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** index=work-hours sourcetype=work-hours Event="exited work." OR Event="entered work." | transaction 'Event' startswith="Event=entered*" endswith="Event=exited*" keepevicted=true
- Results Summary:** 22 events (29/05/2018 00:00:00.000 to 28/06/2018 11:57:24.000) No Event Sampling ▾
- Event List:** Events (22) Patterns Statistics Visualization
- Timeline:** Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect
- Event Data:**

	Time	Event
>	27/06/2018 09:21:00.000	June 27, 2018 at 08:21AM entered work. June 27, 2018 at 17:24PM exited work. host = local source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv sourcetype = work-hours
>	26/06/2018 09:23:00.000	June 26, 2018 at 08:23AM entered work. June 26, 2018 at 18:35PM exited work. host = local source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv sourcetype = work-hours
>	25/06/2018 09:35:00.000	June 25, 2018 at 08:35AM entered work. June 25, 2018 at 17:17PM exited work. host = local source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv sourcetype = work-hours
>	22/06/2018 10:45:00.000	June 22, 2018 at 09:45AM entered work. June 22, 2018 at 14:22PM exited work. host = local source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv sourcetype = work-hours
>	21/06/2018 09:23:00.000	June 21, 2018 at 08:23AM entered work. June 21, 2018 at 17:16PM exited work. host = local source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv sourcetype = work-hours
>	20/06/2018 09:22:00.000	June 20, 2018 at 08:23AM entered work. June 20, 2018 at 17:05PM exited work. host = local source = /Users/forensics/Desktop/splunk-test-data/Work IO.csv sourcetype = work-hours
- Fields Panel:**
 - SELECTED FIELDS: host 1, source 1, sourcetype 1
 - INTERESTING FIELDS: closed_txn 1, duration 17, Event 2, eventcount 1, field_match_sum 1, index 1, linecount 1, splunk_server 1
 - + Extract New Fields

- ▶ First thing to do is transaction the events
 - Enter work
 - Exit work
 - = one transaction

Step 2

The screenshot shows a Splunk search interface with the following details:

- Search Command:**

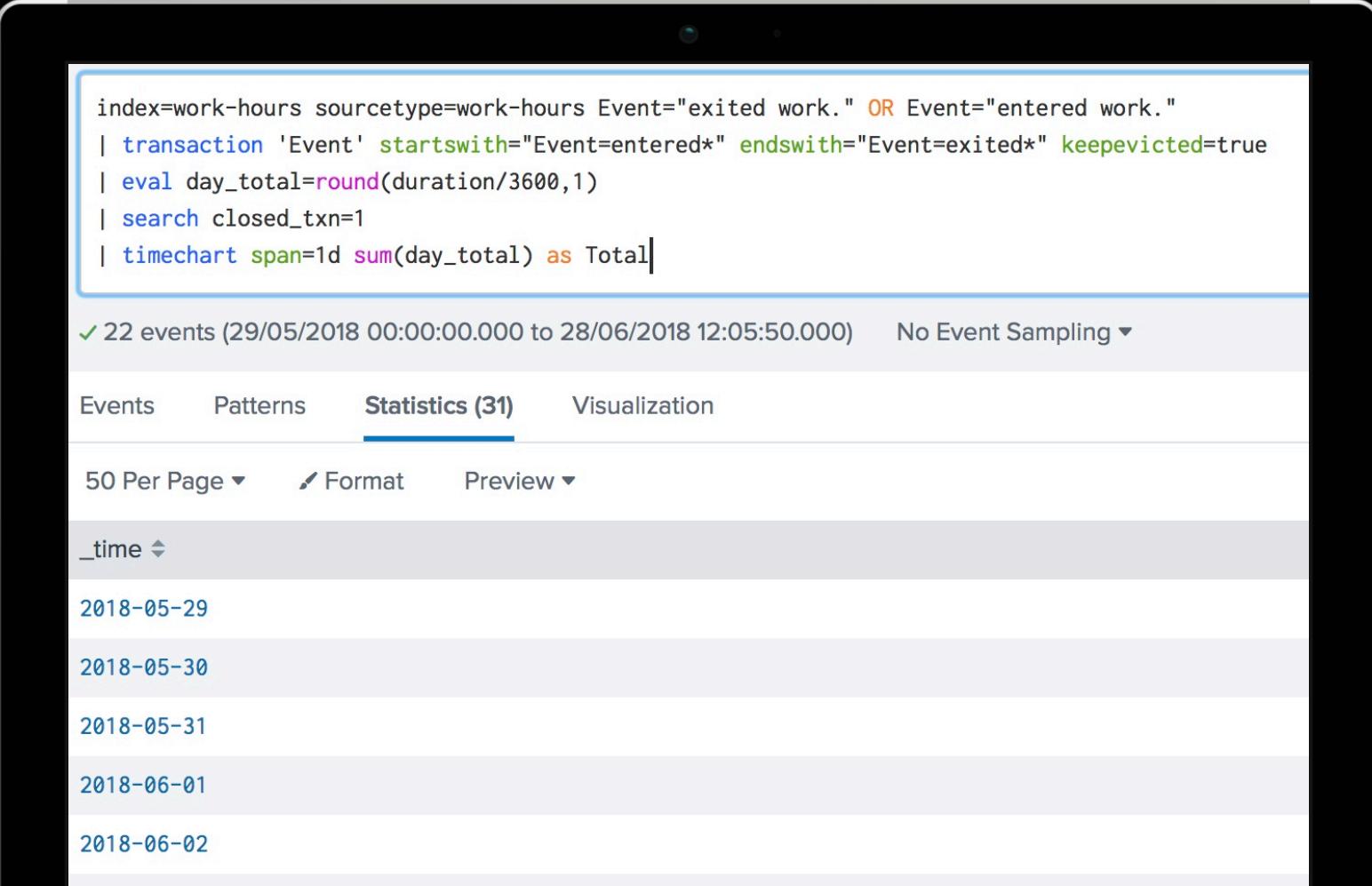
```
index=work-hours sourcetype=work-hours Event="exited work." OR Event="entered work."
| transaction 'Event' startswith="Event=entered*" endswith="Event=exited*" keepevicted=true
| eval day_total=round(duration/3600,1)
```
- Results Summary:** 22 events (29/05/2018 00:00:00.000 to 28/06/2018 12:01:10.000) No Event Sampling ▾
- Event Timeline:** Format Timeline ▾, - Zoom Out, + Zoom to Selection, X Deselect. The timeline shows several green bars representing event durations.
- Event View:**

	Time	Event
27/06/2018 09:21:00.000	June 27, 2018 at 08:21AM	entered work.
	June 27, 2018 at 17:24PM	exited work.
- Selected Fields:** host 1, source 1, sourcetype 1
- Interesting Fields:** closed_txn 1, day_total 12, duration 17, Event 2, eventcount 1, field_match_sum 1, index 1, linecount 1, splunk_server 1
- Event Actions:** A table showing event details:

Type	Field	Value
Selected	host	local
	source	/Users/forensics/Desktop/sp
	sourcetype	work-hours
Event	Event	entered work. exited work.
	closed_txn	1
	day_total	9.1
	duration	32580

- ▶ Work out total hours per transaction
 - Create new field of day_total
 - Uses duration divided by seconds in a day

Step 3



The screenshot shows a Splunk search interface. The search bar contains the following SPL command:

```
index=work-hours sourcetype=work-hours Event="exited work." OR Event="entered work."  
| transaction 'Event' startswith="Event=entered*" endswith="Event=exited*" keepevicted=true  
| eval day_total=round(duration/3600,1)  
| search closed_txn=1  
| timechart span=1d sum(day_total) as Total
```

Below the search bar, the results summary is displayed: "✓ 22 events (29/05/2018 00:00:00.000 to 28/06/2018 12:05:50.000) No Event Sampling ▾".

The navigation tabs at the top are "Events", "Patterns", "Statistics (31)" (which is underlined, indicating it is selected), and "Visualization".

Below the tabs, there are filters: "50 Per Page ▾", "Format", and "Preview ▾".

The main results table has a header labeled "_time" with a sorting arrow. The data rows show dates from "2018-05-29" to "2018-06-02".

- ▶ Ensure good transactions are shown
 - closed_txn=1
- ▶ 1 day = sum of day_total and renamed to Total

Step 4

The screenshot shows a Splunk search interface. At the top, there is a search command:

```
index=work-hours sourcetype=work-hours Event="exited work." OR Event="entered work."
| transaction 'Event' startswith="Event=entered*" endswith="Event=exited*" keepevicted=true
| eval day_total=round(duration/3600,1)
| search closed_txrn=1
| timechart span=1d sum(day_total) as Total
| eventstats avg(Total) as splunk_without_zeros_average sum(Total) as sum_total count as d_count
```

Below the command, it says "22 events (29/05/2018 00:00:00.000 to 28/06/2018 12:09:03.000)" and "No Event Sampling".

The interface has tabs: Events, Patterns, Statistics (31), and Visualization. The Statistics tab is selected, indicated by a blue underline.

Below the tabs, there are filters: "50 Per Page", "Format", and "Preview".

A table is displayed with two columns: "_time" and "Total". The data is as follows:

_time	Total
2018-05-29	8.8
2018-05-30	9.4
2018-05-31	8.9
2018-06-01	9.1
2018-06-02	
2018-06-03	

- ▶ Calculate the last 30 days total
- ▶ Work out average amount of hours
 - Average can be used to get overall view of time present in a set period

Step 5

```
index=work-hours sourcetype=work-hours Event="exited work." OR Event="entered work."
| transaction 'Event' startswith="Event=entered*" endswith="Event=exited*" keepevicted=true
| eval day_total=round(duration/3600,1)
| search closed_txn=1
| timechart span=1d sum(day_total) as Total
| eventstats avg(Total) as splunk_without_zeros_average sum(Total) as sum_total count as d_count
| eval splunk_without_zeros_average=round(splunk_without_zeros_average,1)
| eval zero_day;if@Total>=0,"1","0")|
```

✓ 22 events (29/05/2018 00:00:00.000 to 28/06/2018 12:14:44.000) No Event Sampling ▾

Events Patterns Statistics (31) Visualization

50 Per Page ▾ Format Preview ▾

_time	Total	d_count
2018-05-29	8.8	31
2018-05-30	9.4	31
2018-05-31	8.9	31
2018-06-01	9.1	31
2018-06-02		31

- ▶ Tidy up by rounding the numbers
- ▶ Work out if they have been in or not (weekends)

Step 6

The screenshot shows a Splunk search interface. At the top, there is a search command:

```
index=work-hours sourcetype=work-hours Event="exited work." OR Event="entered work."
| transaction 'Event' startswith="Event=entered*" endswith="Event=exited*" keepevicted=true
| eval day_total=round(duration/3600,1)
| search closed_txn=1
| timechart span=1d sum(day_total) as Total
| eventstats avg(Total) as splunk_without_zeros_average sum(Total) as sum_total count as d_count
| eval splunk_without_zeros_average=round(splunk_without_zeros_average,1)
| eval zero_day;if(Total>=0,"1","0")
| eventstats sum(zero_day) as work_day
| eval "actual_average"=round(sum_total/d_count,1)
```

Below the command, it says "22 events (29/05/2018 00:00:00.000 to 28/06/2018 12:21:27.000)" and "No Event Sampling".

The interface includes tabs for "Events", "Patterns", "Statistics (31)", and "Visualization". The "Statistics (31)" tab is selected. Below the tabs, there are filters: "50 Per Page", "Format", and "Preview".

_time	Total	actual_average	d_count
2018-05-29	8.8	6.1	
2018-05-30	9.4	6.1	
2018-05-31	8.9	6.1	
2018-06-01	9.1	6.1	

- ▶ Calculate average of worked time
 - Not including weekends and leave etc

Step 7

```
index=work-hours sourcetype=work-hours Event="exited work." OR Event="entered work."
| transaction 'Event' startswith="Event=entered*" endswith="Event=exited*" keepevicted=true
| eval day_total=round(duration/3600,1)
| search closed_txn=1
| timechart span=1d sum(day_total) as Total
| eventstats avg(Total) as splunk_without_zeros_average sum(Total) as sum_total count as d_count
| eval splunk_without_zeros_average=round(splunk_without_zeros_average,1)
| eval zero_day;if(Total>0,"1","0")
| eventstats sum(zero_day) as work_day
| eval "actual_average"=round(sum_total/d_count,1)
| where Total>2
| eval average=round(average,1)|
```

✓ 22 events (29/05/2018 00:00:00.000 to 28/06/2018 12:23:30.000) No Event Sampling ▾

Events Patterns Statistics (22) Visualization

50 Per Page ▾ Format Preview ▾

_time	Total	actual_average	d_count
2018-05-29	8.8	6.1	31
2018-05-30	9.4	6.1	31
2018-05-31	8.9	6.1	31

- ▶ Add condition to only class a work day as more than two hours
 - Personal setting for my time but can be changed to anything
- ▶ Then round the average to a useable number

Step 8

The screenshot shows a Splunk search interface. The search bar contains the following command:

```
index=work-hours sourcetype=work-hours Event="exited work." OR Event="entered work."
| transaction 'Event' startswith="Event=entered*" endswith="Event=exited*" keepevicted=true
| eval day_total=round(duration/3600,1)
| search closed_txn=1
| timechart span=1d sum(day_total) as Total
| eventstats avg(Total) as splunk_without_zeros_average sum(Total) as sum_total count as d_count
| eval splunk_without_zeros_average=round(splunk_without_zeros_average,1)
| eval zero_day;if(Total>=0,"1","0")
| eventstats sum(zero_day) as work_day
| eval "actual_average"=round(sum_total/d_count,1)
| where Total>2
| eval average=round(actual_average,1)
| fillnull value=0
| rename _time as date actual_average as "Average/All days" d_count as "Number of Days" splunk_without_zeros_average as "Total Hours Worked per Day"
```

Below the search bar, the results summary is shown: 22 events (29/05/2018 00:00:00.000 to 28/06/2018 12:30:01.000) and No Event Sampling. The Statistics tab is selected, showing the following table:

	Total Hours Worked per Day	Average/All days
date	50 Per Page	Format
	Preview	

- ▶ Beautify the results with some renames

Results

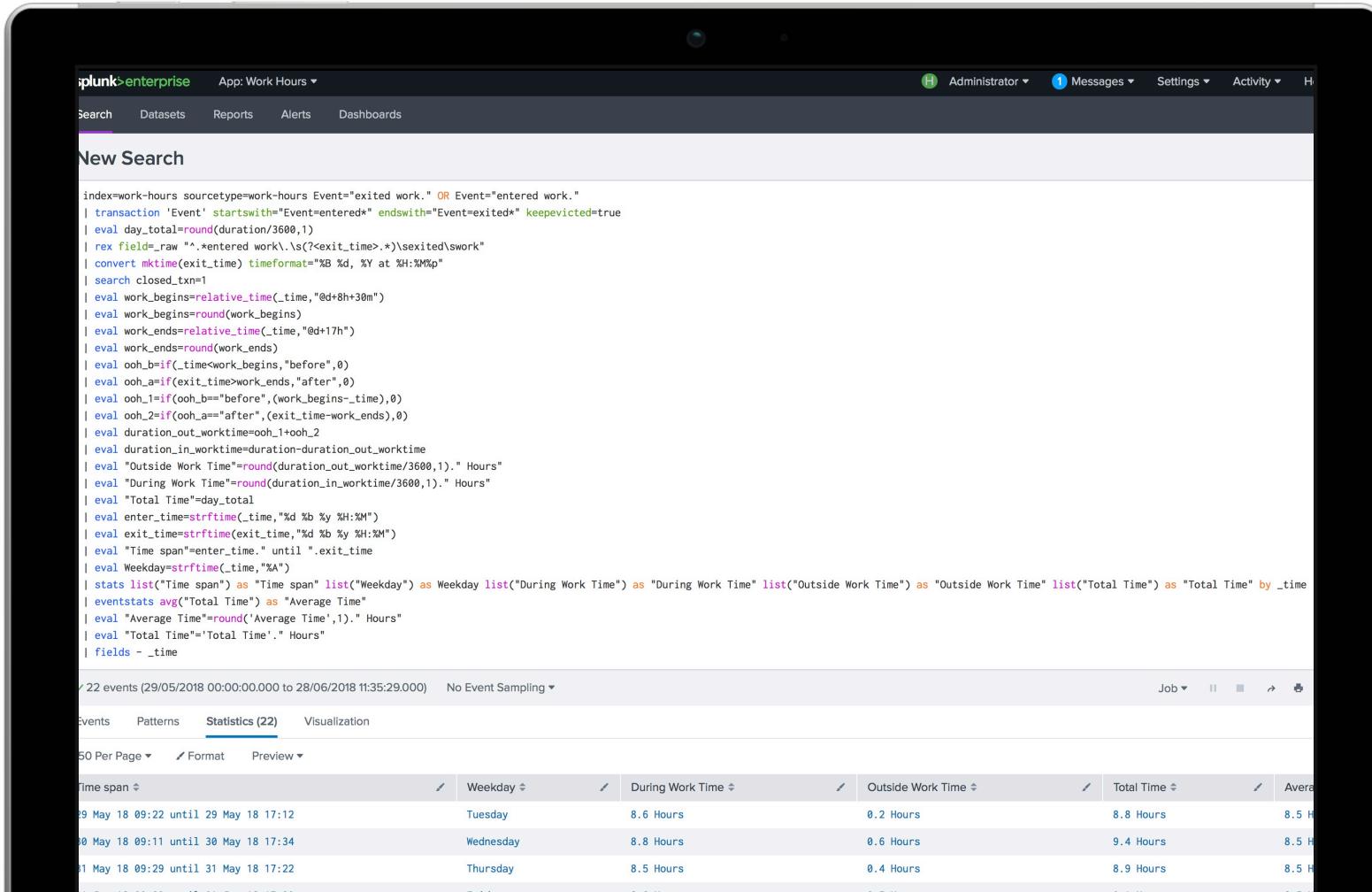
The screenshot shows a Splunk search interface with the following details:

- Search bar: `| eval date=strftime(date,"%d %b %y") | rename date as Date`
- Statistics summary: 22 events (29/05/2018 00:00:00.000 to 28/06/2018 12:38:36.000) | No Event Sampling
- Job status: Job ▾ II ■
- Table Headers:
 - Date
 - Total Hours Worked per Day
 - Average/All days
 - Average/Workdays
 - Days Worked
 - Number of Days
- Table Data:

Date	Total Hours Worked per Day	Average/All days	Average/Workdays	Days Worked	Number of Days
29 May 18	8.8	6.1	8.5	22	31
30 May 18	9.4	6.1	8.5	22	31
31 May 18	8.9	6.1	8.5	22	31
01 Jun 18	9.1	6.1	8.5	22	31
04 Jun 18	8.7	6.1	8.5	22	31
05 Jun 18	9.2	6.1	8.5	22	31
06 Jun 18	9.1	6.1	8.5	22	31
07 Jun 18	8.8	6.1	8.5	22	31
08 Jun 18	5.0	6.1	8.5	22	31
11 Jun 18	9.0	6.1	8.5	22	31
12 Jun 18	9.1	6.1	8.5	22	31
13 Jun 18	9.2	6.1	8.5	22	31
14 Jun 18	9.0	6.1	8.5	22	31
15 Jun 18	6.0	6.1	8.5	22	31
18 Jun 18	9.3	6.1	8.5	22	31
19 Jun 18	8.8	6.1	8.5	22	31
20 Jun 18	9.0	6.1	8.5	22	31
21 Jun 18	8.9	6.1	8.5	22	31
22 Jun 18	4.6	6.1	8.5	22	31
25 Jun 18	8.7	6.1	8.5	22	31
26 Jun 18	10.2	6.1	8.5	22	31
27 Jun 18	9.1	6.1	8.5	22	31

- ▶ Now we have useable data for working stats

Why Stop There!



The screenshot shows the Splunk Enterprise search interface. The top navigation bar includes 'plunk>enterprise' and 'App: Work Hours'. Below the bar are links for 'Search', 'Datasets', 'Reports', 'Alerts', and 'Dashboards'. The main area is titled 'New Search' and contains the following search command:

```

index=work-hours sourcetype=work-hours Event="exited work." OR Event="entered work."
| transaction 'Event' startswith="Event=entered*" endswith="Event=exited*" keepevicted=true
| eval day_total=round(duration/3600,1)
| rex field=_raw ".\x0d\x0a.*\x0d\x0a.\x0d\x0aEvent\x0d\x0a"
| convert mktimedate exit_time timeformat=%B %d, %Y at %H:%M%p
| search closed_txn=1
| eval work_begins=relative_time(_time,"@d+8h30m")
| eval work_begins=round(work_begins)
| eval work_ends=relative_time(_time,"@d+17h")
| eval work_ends=round(work_ends)
| eval ooh_b=if(_time<work_begins,"before",0)
| eval ooh_a=if(exit_time>work_ends,"after",0)
| eval ooh_1=if(ooh_b=="before", (work_begins-_time),0)
| eval ooh_2=if(ooh_b=="after", (exit_time-work_ends),0)
| eval duration_out_worktime=ooh_1+ooh_2
| eval duration_in_worktime=duration_out_worktime
| eval "Outside Work Time"=round(duration_out_worktime/3600,1). " Hours"
| eval "During Work Time"=round(duration_in_worktime/3600,1). " Hours"
| eval "Total Time"=day_total
| eval enter_time=strftime(_time,"%d %B %Y %H:%M")
| eval exit_time=strftime(exit_time,"%d %B %Y %H:%M")
| eval "Time span"=enter_time..exit_time
| eval Weekday=strftime(_time,"%A")
| stats list("Time span") as "Time span" list("Weekday") as Weekday list("During Work Time") as "During Work Time" list("Outside Work Time") as "Outside Work Time" list("Total Time") as "Total Time" by _time
| eventstats avg("Total Time") as "Average Time"
| eval "Average Time"=round(Average Time,1). " Hours"
| eval "Total Time"="Total Time". " Hours"
| fields - _time

```

Below the search command, it says '22 events (29/05/2018 00:00:00.000 to 28/06/2018 11:35:29.000) No Event Sampling'. The results table has columns: 'Time span', 'Weekday', 'During Work Time', 'Outside Work Time', 'Total Time', and 'Average'. The first three rows of the table are:

Time span	Weekday	During Work Time	Outside Work Time	Total Time	Average
29 May 18 09:22 until 29 May 18 17:12	Tuesday	8.6 Hours	0.2 Hours	8.8 Hours	8.5 H
30 May 18 09:11 until 30 May 18 17:34	Wednesday	8.8 Hours	0.6 Hours	9.4 Hours	8.5 H
31 May 18 09:29 until 31 May 18 17:22	Thursday	8.5 Hours	0.4 Hours	8.9 Hours	8.5 H

- ▶ Using the same data and a mix of other Splunk commands
 - If functions
 - Time formatting
 - Avg
 - list
- ▶ A more detailed view can be achieved

More Detail...

The screenshot shows a Splunk search interface with the following details:

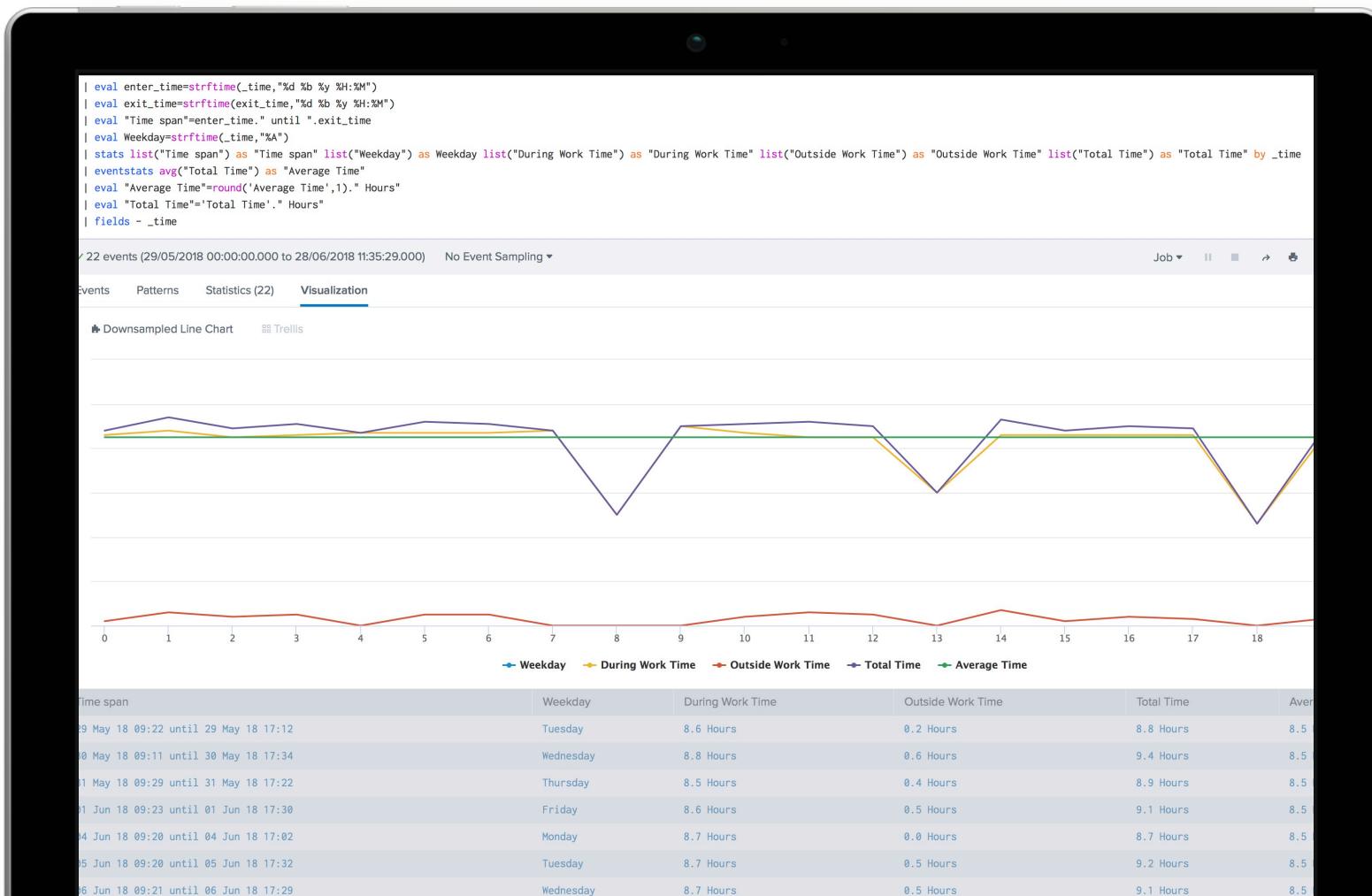
- Search Bar:** Work Hours | Splunk 7.1.1


```
| stats list("Time span") as "Time span" list("Weekday") as Weekday list("During Work Time") as "During Work Time" list("Outside Work Time") as "Outside Work Time" list("Total Time") as "Total Time" by _time
| eventstats avg("Total Time") as "Average Time"
| eval "Average Time"=round("Average Time",1)." Hours"
| eval "Total Time"='Total Time'." Hours"
| fields - _time
```
- Event Count:** ✓ 22 events (29/05/2018 00:00:00.000 to 28/06/2018 11:35:29.000) No Event Sampling ▾
- Statistics View:** Statistics (22) selected. Other tabs include Events, Patterns, Visualization.
- Table Headers:** Time span, Weekday, During Work Time, Outside Work Time, Total Time, Average T
- Table Data:** A list of 22 rows, each representing a time span and its corresponding weekday, work times, total time, and average time. The data spans from May 29 to June 27.

Time span	Weekday	During Work Time	Outside Work Time	Total Time	Average T
29 May 18 09:22 until 29 May 18 17:12	Tuesday	8.6 Hours	0.2 Hours	8.8 Hours	8.5 Hours
30 May 18 09:11 until 30 May 18 17:34	Wednesday	8.8 Hours	0.6 Hours	9.4 Hours	8.5 Hours
31 May 18 09:29 until 31 May 18 17:22	Thursday	8.5 Hours	0.4 Hours	8.9 Hours	8.5 Hours
01 Jun 18 09:23 until 01 Jun 18 17:30	Friday	8.6 Hours	0.5 Hours	9.1 Hours	8.5 Hours
04 Jun 18 09:28 until 04 Jun 18 17:02	Monday	8.7 Hours	0.0 Hours	8.7 Hours	8.5 Hours
05 Jun 18 09:20 until 05 Jun 18 17:32	Tuesday	8.7 Hours	0.5 Hours	9.2 Hours	8.5 Hours
06 Jun 18 09:21 until 06 Jun 18 17:29	Wednesday	8.7 Hours	0.5 Hours	9.1 Hours	8.5 Hours
07 Jun 18 09:15 until 07 Jun 18 17:01	Thursday	8.8 Hours	0.0 Hours	8.8 Hours	8.5 Hours
08 Jun 18 10:20 until 08 Jun 18 14:22	Friday	5.0 Hours	0.0 Hours	5.0 Hours	8.5 Hours
11 Jun 18 08:50 until 11 Jun 18 16:50	Monday	9.0 Hours	0.0 Hours	9.0 Hours	8.5 Hours
12 Jun 18 09:20 until 12 Jun 18 17:23	Tuesday	8.7 Hours	0.4 Hours	9.1 Hours	8.5 Hours
13 Jun 18 09:28 until 13 Jun 18 17:37	Wednesday	8.5 Hours	0.6 Hours	9.2 Hours	8.5 Hours
14 Jun 18 09:30 until 14 Jun 18 17:30	Thursday	8.5 Hours	0.5 Hours	9.0 Hours	8.5 Hours
15 Jun 18 11:02 until 15 Jun 18 16:02	Friday	6.0 Hours	0.0 Hours	6.0 Hours	8.5 Hours
18 Jun 18 09:22 until 18 Jun 18 17:39	Monday	8.6 Hours	0.7 Hours	9.3 Hours	8.5 Hours
19 Jun 18 09:24 until 19 Jun 18 17:14	Tuesday	8.6 Hours	0.2 Hours	8.8 Hours	8.5 Hours
20 Jun 18 09:23 until 20 Jun 18 17:25	Wednesday	8.6 Hours	0.4 Hours	9.0 Hours	8.5 Hours
21 Jun 18 09:23 until 21 Jun 18 17:16	Thursday	8.6 Hours	0.3 Hours	8.9 Hours	8.5 Hours
22 Jun 18 10:45 until 22 Jun 18 14:22	Friday	4.6 Hours	0.0 Hours	4.6 Hours	8.5 Hours
25 Jun 18 09:35 until 25 Jun 18 17:17	Monday	8.4 Hours	0.3 Hours	8.7 Hours	8.5 Hours
26 Jun 18 09:23 until 26 Jun 18 18:35	Tuesday	8.6 Hours	1.6 Hours	10.2 Hours	8.5 Hours
27 Jun 18 09:21 until 27 Jun 18 17:24	Wednesday	8.7 Hours	0.4 Hours	9.1 Hours	8.5 Hours

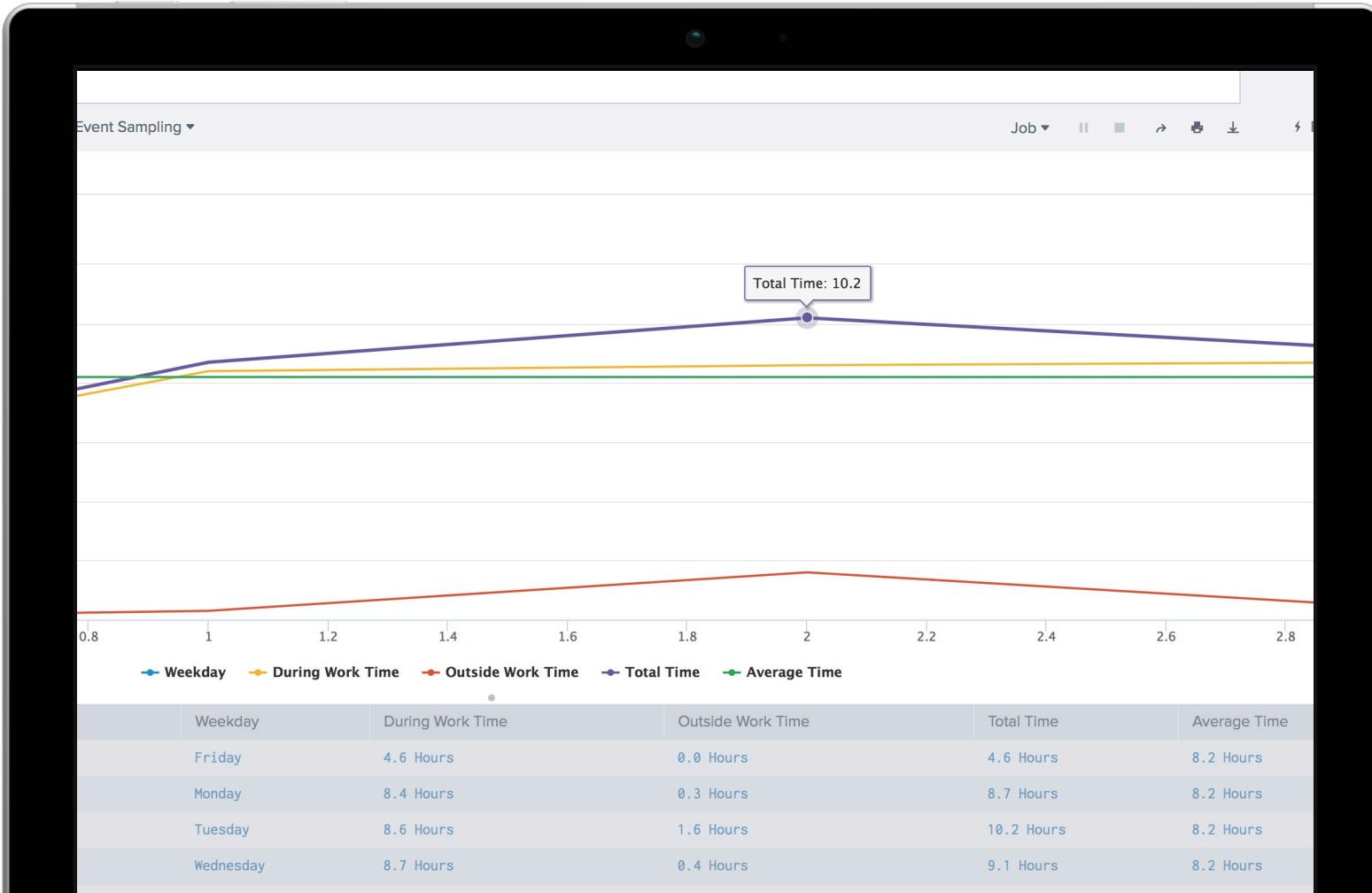
- ▶ A more detailed table can be created

Visually Better



- ▶ Leading to a more visually pleasing chart of the users working hours

Easier To Understand



- ▶ Not only is it visually better
- ▶ It can also be used to highlight data you may want to set alerts for
- ▶ For example, we have an above average time with a visible peak in out of hours presence

The So What...

- ▶ This peak shows that the user was present in work out side of the times set for the users normal working day
 - What was the user doing
 - Should they be in at that time
 - Why are they in 2 hours longer than normal
 - Insider threat???
 - ▶ How about:
 - Exfil
 - Log deletion
 - Implant

All is not lost! You can alert to this potential threat potentially before it happens

Alerts



- ▶ Looking at behaviour of their normal working hours
- ▶ Alerts can be generated for anything out of the norm

What About Way Points

- ▶ Using the IFTTT and the recording setup, its also possible to set way points of a route taken.
 - ▶ Think about:
 - Personnel at risk of kidnap
 - Security vans
 - Routine movement of company property

Future Plans

Going forward with the project

- ▶ In its current version, time recording is done with IFTTT, the plan to go forward is:
 - Create separate application for iOS and Android and place in App Stores
 - Feed data directly in to Splunk, not via CSV monitor
 - Link alerting to mail server
 - Manager app
 - See live data for manager perspective
 - Health and safety - Fire evacuation, who is where
- ▶ Create a TA for Splunk to partner with the App from the app store, the benefits of this would be:
 - Pre built dashboards displaying key information
 - Simple alerting so key security concerns can be received as soon as possible

Cross industry utility

Where else can this be used...

- ▶ The time recording app and TA would not only benefit security teams, but what about...
 - Automotive - Where are the vehicles = Where should new charging points be
 - Hours in hospital for surgeons - tired surgeon, error in surgery
 - Professional drivers - Leave the depot, are they resting?
 - Education - Are students attending lectures?
 - Staff of interest - Is kidnap a risk at your organisation
 - Basically anything with a GPS signal....

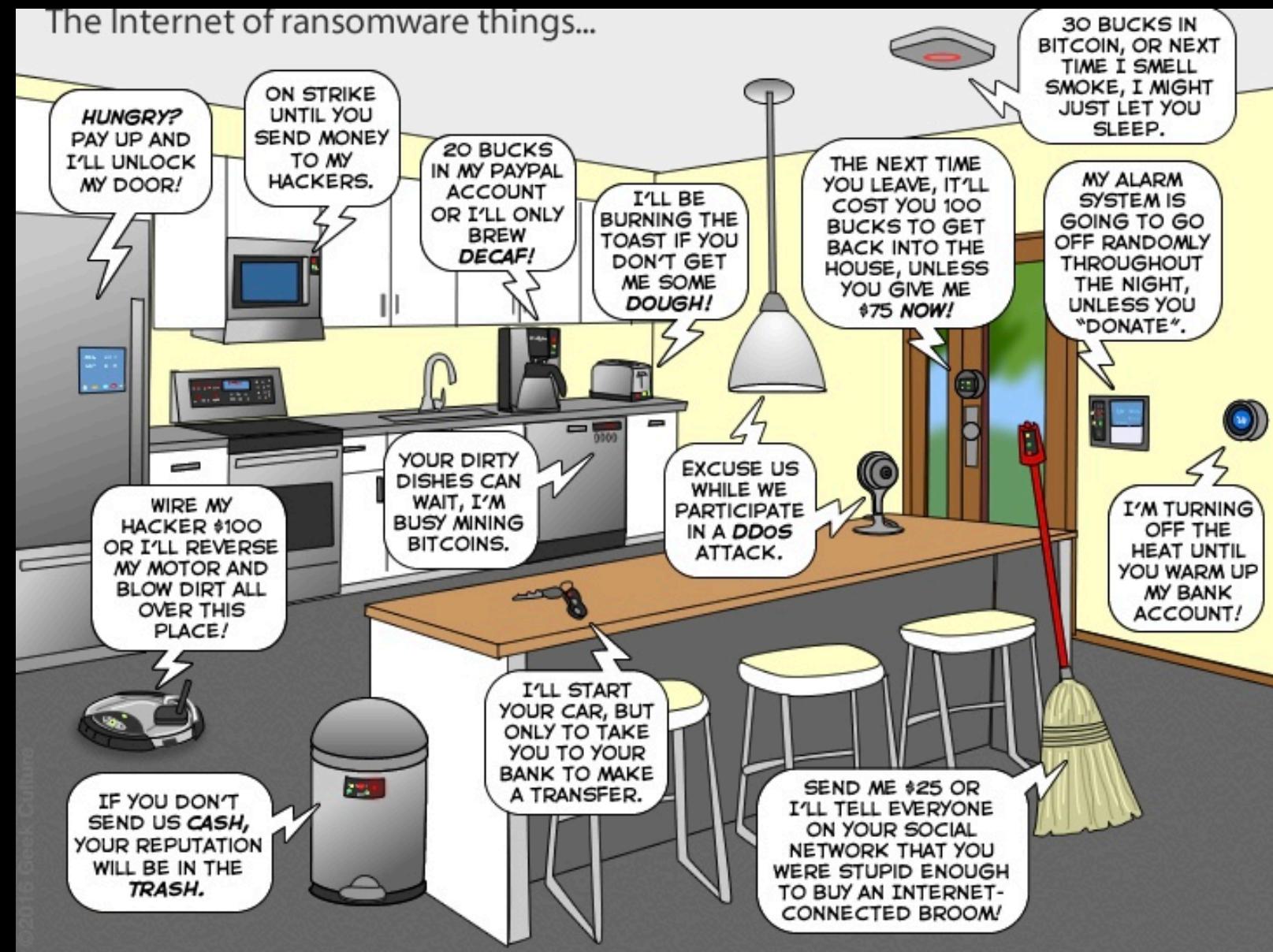
The possibilities are endless for location based monitoring in Splunk, if it isn't normal behaviour, what is it??

Making machine data accessible, usable and valuable to everyone.

A background grid of log entries from the Buttercup Shopping application, showing various user interactions like product views, purchases, and category browses, overlaid with the main title.

Any Questions?

Thank You



Thank You

Don't forget to rate this session
in the .conf18 mobile app



splunk>

