FORENSIC ARTIFACT CORRELATION VIA ELASTIC

MATTHEW SEYER & DAVID COWEN

OSDFCON 2015

PROCESS OVERVIEW

- Two step process… (but not the Texas two step)
- Step 1: Indexing – Collecting Artifact Information
- Step 2: Correlating – Connecting the dots to see the big picture

THE ISSUES

- No one tool does it all
- Wide variety of specific tools for the job
- Each tool can have multiple output types and multiple formats for each type (txt[csv,tsv], json, xml)
- No way to link the multitude of reports together for the overall picture
- Why? Large variety of tools for large variety of artifacts…

THAT'S A LOT OF ARTIFACTS!

LogFile

Shell Bag

MFT

Prefetch

USN

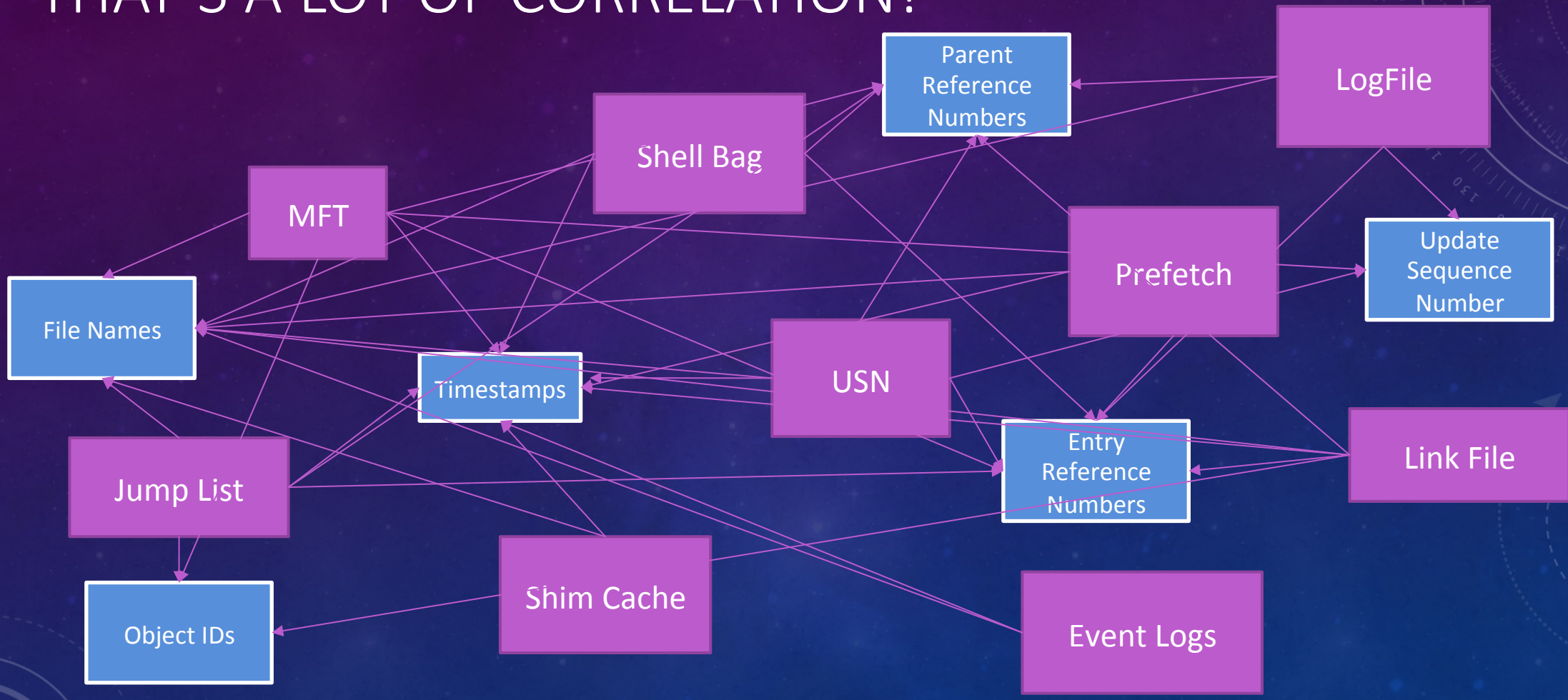
Link File

Jump List

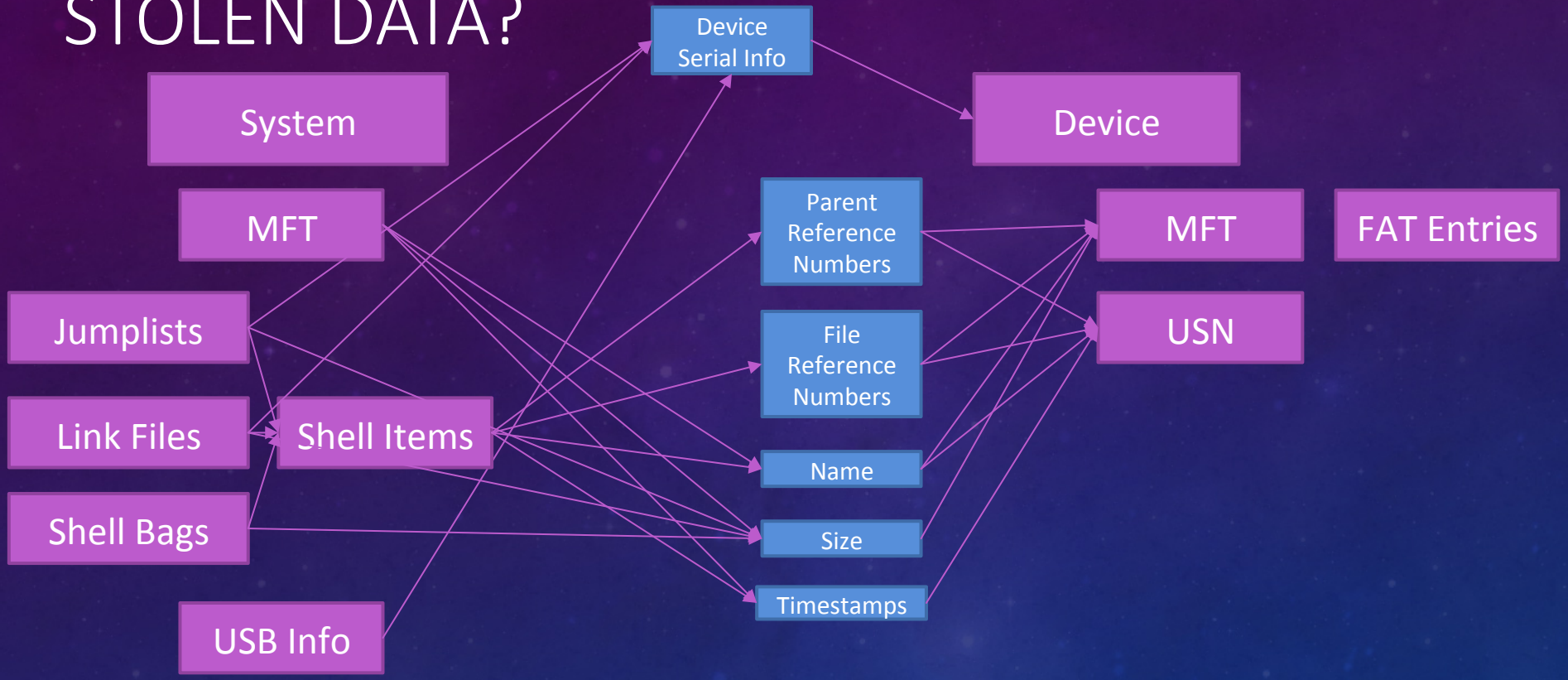
Shim Cache

Event Logs

THAT'S A LOT OF CORRELATION!



STOLEN DATA?


System
MFT
Jumplists
Link Files
Shell Bags
USB Info
Shell Items
Device Serial Info
Parent Reference Numbers
File Reference Numbers
Name
Size
Timestamps
Device
MFT
USN
FAT Entries

WHY ELASTIC?

- Pros:
  - Easy to use
  - Extremely fast retrieving data
  - Scales easily
  - Has a nice Kibana interface
  - No need to predefine fields or column types
- Cons:
  - Does not handle relational data (artifact correlation = relational)
  - Not many interfaces for viewing data (No SQLite type browsers)
  - Only has Kibana

WHY NOT KIBANA

- Kibana is timestamp driven (Great for log analysis)
- Kibana is not dynamic enough to deal with needs of forensics
- What is needed?
- Dynamic Interface
  - User can configure “Artifacts” via mappings and json
- Defined Correlations
  - User specifies what “Correlates” Artifacts based on normalized data stored in Elastic Search

STEP 1: NORMALIZING YOUR TOOL OUTPUT

- You have multiple tools
- Each tool can name fields differently
- Most tools support some type of csv/tsv output
- Using ElasticHandler we can remap columns
- Normalizing field types and names for inter-report correlation

EXAMPLE JSON –TZWORKS JUMPLIST PARSER

```
{
    "delimiter":"|",
    "start_line":"7",
    "type":"tz_linkstruct",
    "map_file":"etc\\tz_linkstruct.mapping",
    "columns":[
        "Source Location",
        "Source Type",
        "AppId",
        "MRU MFU",
        "Stream",
        "MRU Datetime UTC",
        "File Modify Datetime UTC",
        "File Access Datetime UTC",
        "File Create Datetime UTC",
        "Target Modify Datetime UTC",
        "Target Access Datetime UTC",
        "Target Create Datetime UTC",
        "ObjID Datetime UTC",
        "Target Attrib",
        "Inode",
        "Seq",
        "File Size",
        "Target Name",
        "Idlist Extra Info",
        "Volume Type",
        "Volume Serial",
        "Volume Label",
        "Local Path",
        "Common Path",
        "Network and Device Info",
        "Extra Info",
        "Netbios Name",
        "Volume Id",
        "Object Id",
        "MAC Addr"
    ],
"add_columns":{
    "EntryNames":{
        "type":"get_from_path",
        "options":{
            "sep":"\\"
        },
        "source":[
            "{Local Path}",
            "{Target Name}",
            "{Common Path}"
        ]
    },
    "FileName":{
        "type":"get_filename",
        "options":{
            "sep":"\\"
        },
        "source":[
            "{Local Path}",
            "{Target Name}",
            "{Common Path}"
        ]
    },
    "EntryReferences":{
        "type":"append",
        "source":[
            "{Inode}-{Seq}"
        ]
    }
},
    },
    "sub_record_columns":[
        "Extra Info"
    ]
}
```

EXAMPLE MAPPING

```
{
    "mappings": {
        "tz_linkstruct": {
            "properties": {
                "index_timestamp":{
                    "type": "date",
                    "format": "MM/dd/yyyy HH:mm:ss.SSSSSS||MM/dd/yyyy HH:mm:ss||yyyy-MM-dd HH:mm:ss"
                },
                "FileRef":{
                    "type": "string",
                    "index": "not_analyzed"
                },
                "FileName":{
                    "type": "string",
                    "index": "not_analyzed"
                },
                "FileExt":{
                    "type": "string",
                    "index": "not_analyzed"
                },
                "EntryNames":{
                    "type": "string",
                    "index": "not_analyzed"
                },
                "EntryReferences":{
                    "type": "string",
                    "index": "not_analyzed"
                },
                "FullFileName":{
                    "type": "string",
                    "index": "not_analyzed"
                },
                "BaseFileName":{
                    "type": "string",
                    "index": "not_analyzed"
                },
                "ParentName":{
                    "type": "string",
                    "index": "not_analyzed"
                },
```

```
"PointerLocation":{
    "type": "string",
    "index": "not_analyzed"
},
"Source Location": {
    "type": "string",
    "index": "not_analyzed"
},
"Source Type": {
    "type": "string",
    "index": "not_analyzed"
},
"AppId": {
    "type": "string",
    "index": "not_analyzed"
},
"MRU MFU": {
    "type": "string",
    "index": "not_analyzed"
},
"Stream": {
    "type": "string",
    "index": "not_analyzed"
},
"MRU Datetime UTC": {
    "type": "date",
    "format": "MM/dd/yyyy HH:mm:ss.SSS||MM/dd/yyyy HH:mm:ss"
},
"File Modify Datetime UTC": {
    "type": "date",
    "format": "MM/dd/yyyy HH:mm:ss.SSS||MM/dd/yyyy HH:mm:ss"
},
"File Access Datetime UTC": {
    "type": "date",
    "format": "MM/dd/yyyy HH:mm:ss.SSS||MM/dd/yyyy HH:mm:ss"
},
```

DEMO

- Running ElasticHandler
- Source code review

CORRELATING OUTSIDE ELASTIC

- Normalizing value names allow correlation
- Elastic can map and sort
- Elastic cannot make relational queries
- We can make our correlations using ES data

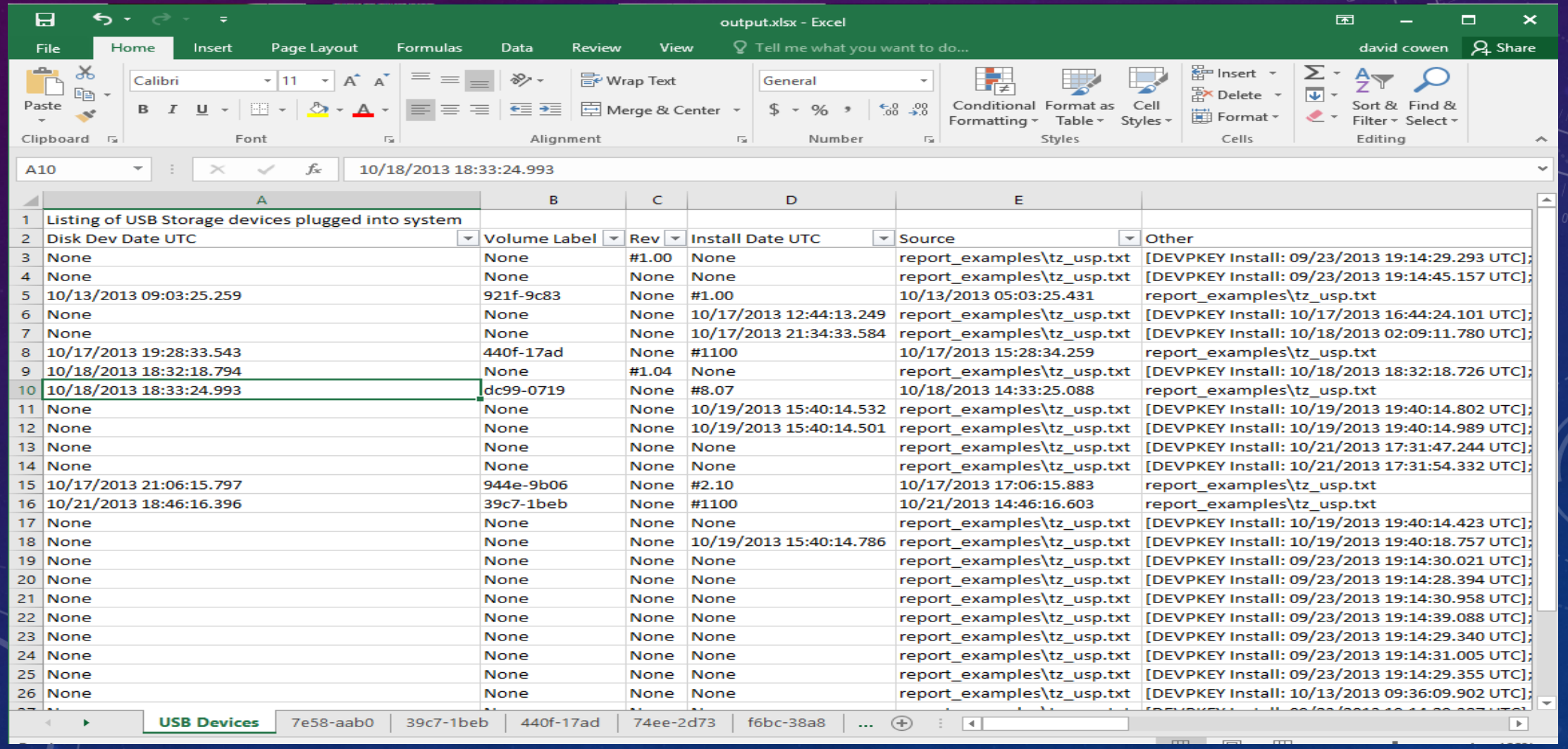
CUSTOM CORRELATIONS

- Define your relations
- Parse your reports
- Learn your query syntax
- Build your output
- Automate the boring parts of the job!

DEMO

- An example python script that builds a spreadsheet of data known taken to USB devices
- Saves hours of work
- Short code review

EXAMPLE REPORT



output.xlsx - Excel

	A	B	C	D	E	
1	Listing of USB Storage devices plugged into system					
2	Disk Dev Date UTC	Volume Label	Rev	Install Date UTC	Source	Other
3	None	None	#1.00	None	report_examples\tz_usp.txt	[DEVPKEY Install: 09/23/2013 19:14:29.293 UTC];
4	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 09/23/2013 19:14:45.157 UTC];
5	10/13/2013 09:03:25.259	921f-9c83	None	#1.00	10/13/2013 05:03:25.431	report_examples\tz_usp.txt
6	None	None	None	10/17/2013 12:44:13.249	report_examples\tz_usp.txt	[DEVPKEY Install: 10/17/2013 16:44:24.101 UTC];
7	None	None	None	10/17/2013 21:34:33.584	report_examples\tz_usp.txt	[DEVPKEY Install: 10/18/2013 02:09:11.780 UTC];
8	10/17/2013 19:28:33.543	440f-17ad	None	#1100	10/17/2013 15:28:34.259	report_examples\tz_usp.txt
9	10/18/2013 18:32:18.794	None	#1.04	None	report_examples\tz_usp.txt	[DEVPKEY Install: 10/18/2013 18:32:18.726 UTC];
10	10/18/2013 18:33:24.993	dc99-0719	None	#8.07	10/18/2013 14:33:25.088	report_examples\tz_usp.txt
11	None	None	None	10/19/2013 15:40:14.532	report_examples\tz_usp.txt	[DEVPKEY Install: 10/19/2013 19:40:14.802 UTC];
12	None	None	None	10/19/2013 15:40:14.501	report_examples\tz_usp.txt	[DEVPKEY Install: 10/19/2013 19:40:14.989 UTC];
13	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 10/21/2013 17:31:47.244 UTC];
14	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 10/21/2013 17:31:54.332 UTC];
15	10/17/2013 21:06:15.797	944e-9b06	None	#2.10	10/17/2013 17:06:15.883	report_examples\tz_usp.txt
16	10/21/2013 18:46:16.396	39c7-1beb	None	#1100	10/21/2013 14:46:16.603	report_examples\tz_usp.txt
17	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 10/19/2013 19:40:14.423 UTC];
18	None	None	None	10/19/2013 15:40:14.786	report_examples\tz_usp.txt	[DEVPKEY Install: 10/19/2013 19:40:18.757 UTC];
19	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 09/23/2013 19:14:30.021 UTC];
20	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 09/23/2013 19:14:28.394 UTC];
21	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 09/23/2013 19:14:30.958 UTC];
22	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 09/23/2013 19:14:39.088 UTC];
23	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 09/23/2013 19:14:29.340 UTC];
24	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 09/23/2013 19:14:31.005 UTC];
25	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 09/23/2013 19:14:29.355 UTC];
26	None	None	None	None	report_examples\tz_usp.txt	[DEVPKEY Install: 10/13/2013 09:36:09.902 UTC];

USB Devices | 7e58-aab0 | 39c7-1beb | 440f-17ad | 74ee-2d73 | f6bc-38a8 | ...

QUESTIONS?

- Email us:
  - Matt: mseyer@g-cpartners.com
  - Dave: dcowen@g-cpartners.com
- Tweet us:
  - @forensic_matt
  - @hecfblog
- Get the code!
  - https://github.com/devgc/ElasticHandler