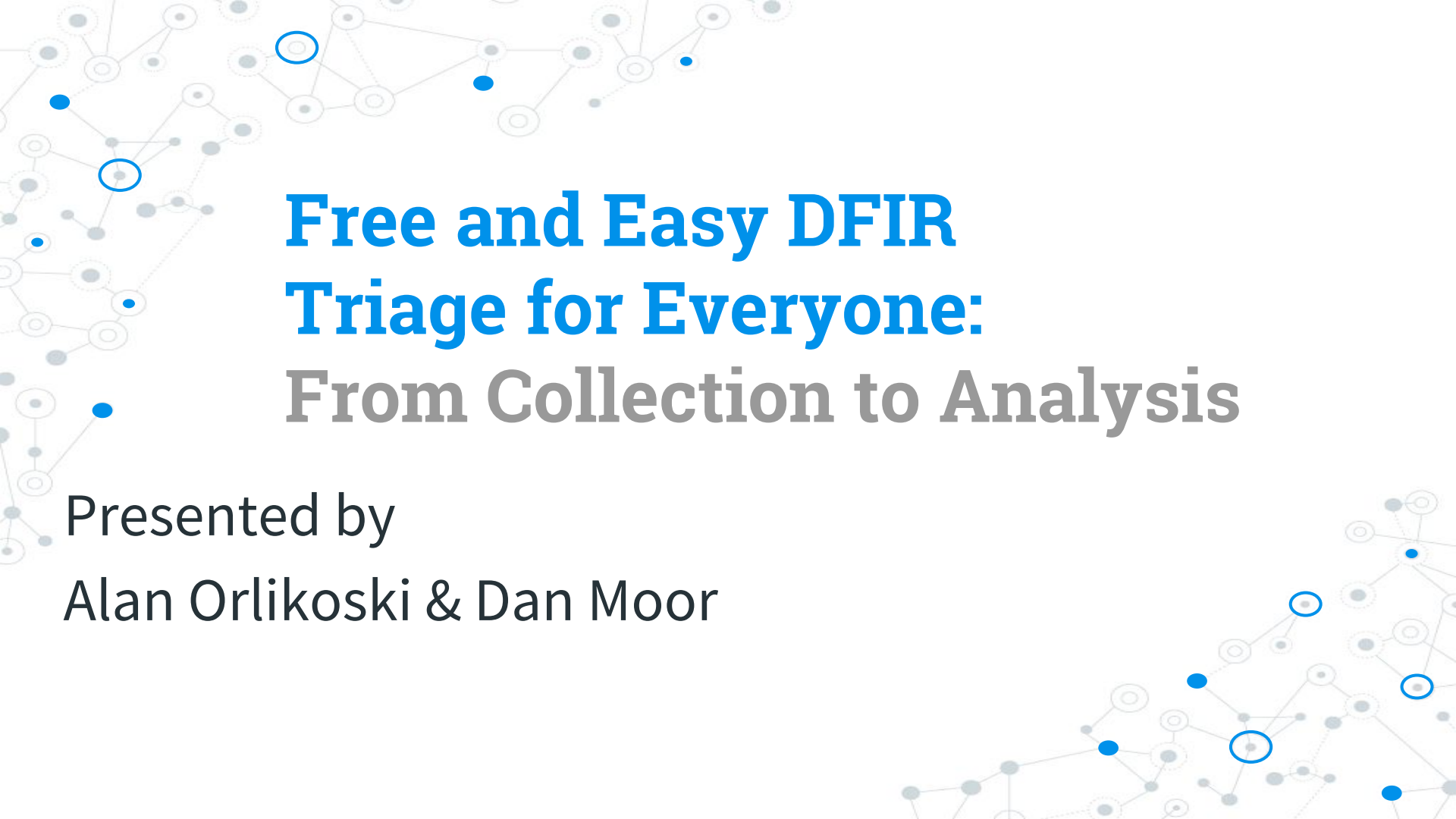


The background is a vibrant, cyberpunk-style illustration. In the center is a character's face with a serious expression, featuring glowing blue eyes and a blue beard. The face is framed by two large, circular, green, mechanical-looking structures that resemble stylized DNA helices or gears. Overlaid on the character's forehead is a large, glowing 'X' logo composed of 3D, isometric blocks in shades of red, pink, and cyan. The overall color palette is dominated by deep blues, purples, and greens, with high-contrast lighting effects.

DEFCON

WORKSHOPS



Free and Easy DFIR Triage for Everyone: From Collection to Analysis

Presented by
Alan Orlikoski & Dan Moor

Who we are

Alan Orlikoski

Security Engineer, Square

@alanorlikoski

- ◎ Over 11 years of Cyber Security Project Management experience
- ◎ Over 13 years of experience working with SOC's
- ◎ Over 17 years of experience working in Cyber Security
- ◎ Author of the Cold Disk Quick Response (CDQR), CyLR and CCF-VM forensics tools (<https://github.com/rough007>)

Dan Moor

Technical Lead, Manager - Incident Response, FusionX

@1EyeInDFIR

- ◎ Bunches of years doing Digital Forensics and Incident Response in the Enterprise space
- ◎ Investigations from HR “find the naughty pictures” to full, global breaches
- ◎ Contributing author of patents relating to Threat Intelligence sharing (software patents for the meh!)



Workshop Sections

01: Intro to CCF-VM

02: Workflows and Triage

03: Data Collection

04: Data Processing

05: Analysis Methods

06: Final Chapter



Section 01

Intro to CCF-VM



S01 - Lesson 01: What is the CCF-VM?

- ◎ CyLR, CDQR - Forensic VM (CCF-VM)
- ◎ Created in 2016 by
 - Alan Orlikoski (<https://github.com/rough007>)
- ◎ Purpose
 - Get the right data in front of the analyst quickly and accurately
 - Make the process:
 - ◎ Easy to use
 - ◎ Scalable
 - ◎ Affordable
 - Tools designed to work together
 - ◎ Ease “utility sprawl”
 - ◎ Extensible

S01 - Lesson 02: The components

- ◎ CyLR - Collect Artifacts
- ◎ CDQR - Process Artifacts / Images
 - Plaso - Parsing back-end
 - Organizing and optimizing reports and database
- ◎ CCF-VM
 - CyLR
 - CDQR
 - ElasticSearch
 - Cerebro
 - Kibana & TimeSketch

S01 - Lesson 03: Installing CCF-VM

Option 01:

- Download CCF-VM 2.0

- <https://drive.google.com/file/d/0B5z7g7P2BWJAeXdPYXVtUWJLQWM>

- Download Target-VM

- <https://drive.google.com/open?id=0B0YdQ-EW7-1cRWpuQnVYdjEwNUk>

Option 2:

- Use the USB Drive

Deploy both VM's to local machine

S01 - Lesson 03: Configuring CCF-VM and Virtualbox

Configure and Validate CCF-VM in Virtualbox

- Set network interface to 'Bridged'
- **ifconfig -a** (get NIC name)
- **tail -4 /etc/network/interfaces** (get existing interface name)
- **sudo sed -i 's/ens32/enp0s3/g' /etc/network/interfaces**
- **sudo service networking restart**
- **ifconfig -a** (confirm you have an IP address)

- ### You should now be able to open a local browser and connect to SSH, Kibana, and Cerebro via that address

```
cdqr@CCF_VM:~$ ifconfig -a
enp0s3: Link encap:Ethernet HWaddr 08:00:27:41:71:39
        BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo:    Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:1568 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1568 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:197157 (197.1 KB)  TX bytes:197157 (197.1 KB)

cdqr@CCF_VM:~$ tail -4 /etc/network/interfaces

# The primary network interface
auto ens32
iface ens32 inet dhcp
cdqr@CCF_VM:~$ sudo sed -i 's/ens32/enp0s3/g' /etc/network/interfaces
[sudo] password for cdqr:
cdqr@CCF_VM:~$ sudo service networking restart
cdqr@CCF_VM:~$ _
```

```
cdqr@CCF_VM:~$ ifconfig -a
enp0s3: Link encap:Ethernet HWaddr 08:00:27:41:71:39
        inet addr:192.168.1.76  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe41:7139/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:14 errors:0 dropped:0 overruns:0 frame:0
```

S01 - Practical: CCF-VM Setup

- ◎ **Note your IP address:**
- ◎ Validate Cerebro: `http://<IP address>:9000`
 - Bookmark link
- ◎ Validate Kibana: `http://<IP address>:5601`
 - Bookmark link
- ◎ Validate TimeSketch: `http://<IP address>:5000`
 - Bookmark link

For reference

```
cdqr@CCF_VM:~$ ifconfig -a
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:0b:81:65
        BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:19896 errors:0 dropped:0 overruns:0 frame:0
        TX packets:19896 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:2949270 (2.9 MB)  TX bytes:2949270 (2.9 MB)

cdqr@CCF_VM:~$ tail -4 /etc/network/interfaces

# The primary network interface
auto ens32
iface ens32 inet dhcp
cdqr@CCF_VM:~$ sudo sed -i 's/ens32/enp0s3/g' /etc/network/interfaces
cdqr@CCF_VM:~$ sudo ifconfig enp0s3 up
cdqr@CCF_VM:~$
```

Section 02

Workflows and Triage



S02 - Lesson 01: Live Response vs Disk Image

Live Response:

Small collection of critical artifacts

- ⊙ Enables rapid investigation
- ⊙ Initial set of artifacts
 - Requires subsequent collection of suspicious files
- ⊙ “Sub-optimal”™ collection technique

Cold Disk/ Full Disk Image:

Bit copy of full disk structure

- ⊙ Introduces a delay before investigation
- ⊙ Collects all disk data on the target host reducing the need for subsequent collections
- ⊙ Required for forensically sound investigations

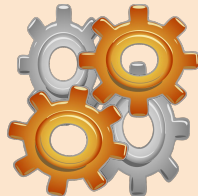
S02 - Lesson 01: The components

Collect



CyLR
(+ SFTP Server)

Process



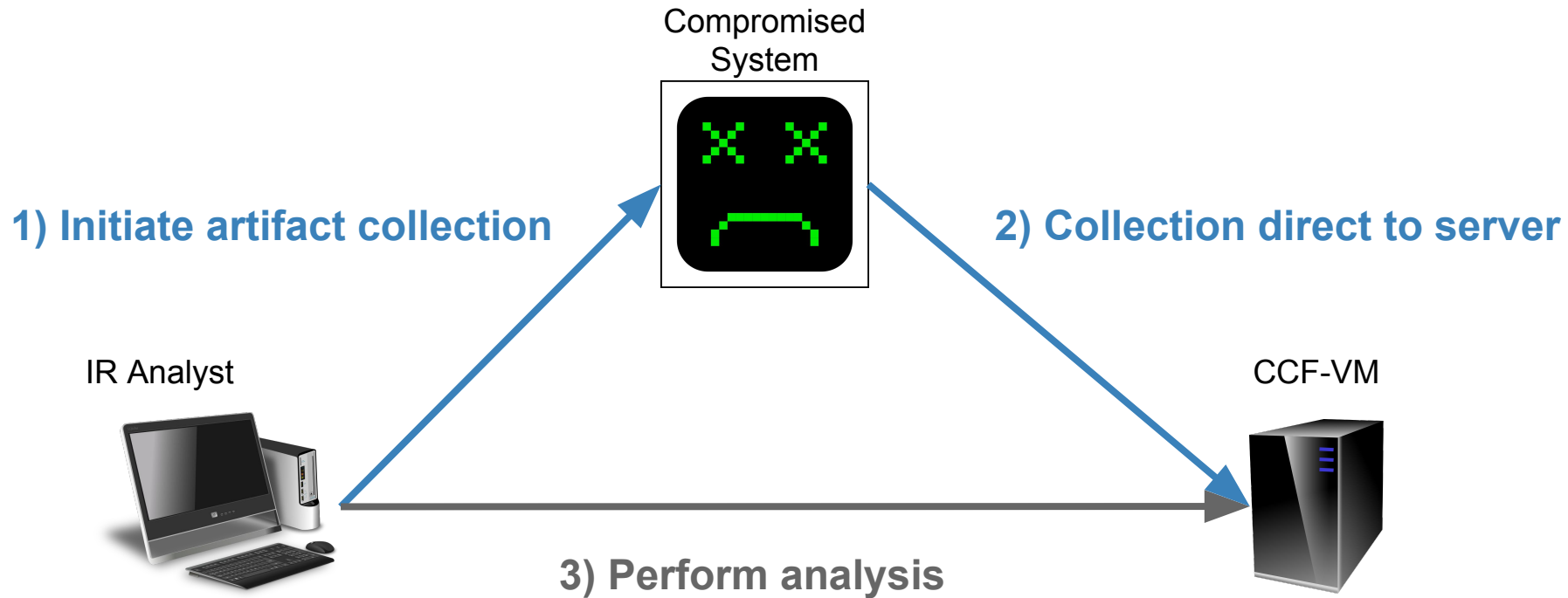
CDQR

Investigate



CCF-VM

S02 - Lesson 02: Logical Workflow



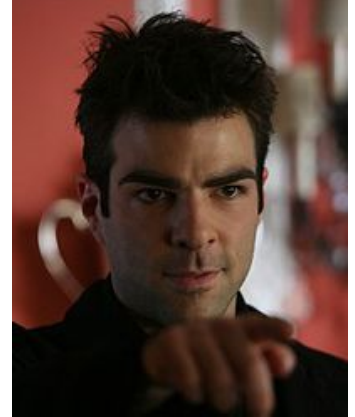
Section 03

Data Collection



S03 - Lesson 01: What is CyLR

- ◎ C# Live Response (CyLR)Tool
- ◎ Created in 2016 by
 - Alan Orlikoski (<https://github.com/rough007>)
 - Jason Yegge (<https://github.com/Lansatac>)
- ◎ Current capabilities
 - Quick collection (it's really fast)
 - Raw file collection process does not use Windows API
 - Optimized to store the collected artifacts in memory*
 - Built in SFTP capability



S03 - Lesson 02: CyLR Options

- ◎ Run as Admin for full utility
 - Non-administrative privileges defaults to use of Windows API
- ◎ Command options
 - -od (Output Directory name)
 - -of (Output File name)
 - -c (custom lists)
 - -u Username for SFTP
 - -p Password for SFTP
 - -s SFTP server IP address
 - --force-native
 - -zp

S03 - Lesson 03: Collection List Options

Default items collected

- ◎ "%SYSTEMROOT%\System32\drivers\etc\hosts"
- ◎ "%SYSTEMROOT%\SchedLgU.Txt"
- ◎ "%PROGRAMDATA%\Microsoft\Windows\Start Menu\Programs\Startup"
- ◎ "%SYSTEMROOT%\System32\config"
- ◎ "%SYSTEMROOT%\System32\winevt\logs"
- ◎ "%SYSTEMROOT%\Prefetch"
- ◎ "%SYSTEMROOT%\Tasks"
- ◎ "%SYSTEMROOT%\System32\LogFiles\W3SVC1"
- ◎ "%SystemDrive%\\$MFT"

Custom Collections

◎ Lists of files/folders

- more collection.txt | CyLR.exe
- CyLR.exe -c Collection.txt

◎ List of arguments

- CyLR.exe "%SYSTEMROOT%\Tasks" ...

Sample collection.txt contents (one item per line)

```
"%SystemDrive%\$MFT"  
"D:\$MFT"  
"%SYSTEMROOT%\Prefetch"  
"D:\Temp"
```

S03 - Lesson 03: Default Collection Demo

```
erational.evtx  
Collecting File: C:\Windows\System32\winevt\logs\Microsoft-Windows-WorkFolders%4W  
C.evtx  
Collecting File: C:\Windows\System32\winevt\logs\Microsoft-Windows-Workplace Join%  
4Admin.evtx  
Collecting File: C:\Windows\System32\winevt\logs\Microsoft-Windows-WPD-ClassInstal  
ler%4Operational.evtx  
Collecting File: C:\Windows\System32\winevt\logs\Microsoft-Windows-WPD-CompositeC  
lassDriver%4Operational.evtx  
Collecting File: C:\Windows\System32\winevt\logs\Microsoft-Windows-WPD-MTPClassDr  
ver%4Operational.evtx  
Collecting File: C:\Windows\System32\winevt\logs\Microsoft-Windows-WWAN-SVC-Events  
%4Operational.evtx  
Collecting File: C:\Windows\System32\winevt\logs\Microsoft-WindowsPhone-Connectiv  
ty-WiFiConnSvc-Channel.evtx  
Collecting File: C:\Windows\System32\winevt\logs\OAlerts.evtx  
Collecting File: C:\Windows\System32\winevt\logs\PreEmptive.evtx  
Collecting File: C:\Windows\System32\winevt\logs\Security.evtx
```

S03 - Lesson 03: Custom Collection Demo

```
Collecting File: C:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTEventLog-System.etl
Collecting File: C:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTMuroc System Trace
.etl
Collecting File: C:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTSteam Event Tracin
g.etl
Collecting File: C:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTUBPM.etl
Collecting File: C:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTWFP-IPsec Diagnost
ics.etl
Collecting File: C:\Windows\System32\LogFiles\WMI\LwtNetLog.etl
Collecting File: C:\Windows\System32\LogFiles\WMI\NtfsLog.etl
Collecting File: C:\Windows\System32\LogFiles\WMI\SpoolerLogger.etl.001
Collecting File: C:\Windows\System32\LogFiles\WMI\SpoolerLogger.etl.002
Collecting File: C:\Windows\System32\LogFiles\WMI\Wifi.etl
Extraction complete. 0:00:00.2915224 elapsed

C:\Users\asor1\temp
λ ssh cdqr@192.168.232.128
cdqr@192.168.232.128's password: |
```

S03 - Lesson 04: Victim-PC Information

Name	username	password
Mr. Anderson (administrator)	Anderson	<blank>
C Level	c-user	ABC
Keypunching Monkey	kpm	ioatft
Windows Subsystem for Linux	nixon	aroo

A default Windows 10 VM with accounts noted above. No significant modification of system beyond the installation of Windows Subsystem for Linux

S03 - Practical: Collecting Data with CyLR

- © Log into victim-pc
- © Open command prompt (run as Administrator)
- © Default collection on victim-pc with output file named defcon.zip and send to CCF-VM
- © Collect “C:\Windows\System32\config\SYSTEM” and send to CCF-VM
- © Validate files were collected correctly

Section 04

Data Processing



S04 - Lesson 01: What is CDQR

- ◎ Cold Disk Quick Response (CDQR) Tool
- ◎ Created in 2015 by
 - Alan Orlikoski (<https://github.com/rough007>)
- ◎ CDQR provides
 - Decreased time required to process data
 - Simplified command set
 - Parsed and optimized outputs
 - Enables seamless output to external tools



S04 - Lesson 02: CDQR Options

- ⦿ Available Python or Windows Stand-alone Executable version
 - cdqr.py
 - cdqr.exe
- ⦿ -p (Pick a parser)
- ⦿ --nohash (Disable file hashing)
- ⦿ --max_cpu (Use all CPU threads)
- ⦿ --export (force export into json format)
- ⦿ --es (ElasticSearch output)
- ⦿ -z (Read input from ZIP archive)

S04 - Lesson 03: Input Types

- ◎ **Single Artifacts**
 - ~/artifacts/SYSTEM
- ◎ **Folder of Artifacts**
 - ~/artifacts
- ◎ **Zip Archive of Artifacts**
 - ~/config_folder.zip
- ◎ **Forensic Images**
 - ~/artifacts/sample.E01
 - ~/artifacts/sample.dd
- ◎ **Virtual Disks**
 - ~/artifacts/sample.vmdk
- ◎ **Mounted Disks**
 - /mnt/windows_mount/

S04 - Lesson 04: All about the parsers



Name: win

- Windows - 62 parsers

Appcompatcache,bagmru,binary_cookies,ccleaner,chrome_cache,chrome_cookies,chrome_extension_activity,chrome_history,chrome_preferences,explorer_mountpoints2,explorer_programscache,filestat,firefox_cache,firefox_cache2,firefox_cookies,firefox_downloads,firefox_history,google_drive,java_idx,mcafee_protection,mft,mrulist_shell_item_list,mrulist_string,mrulistex_shell_item_list,mrulistex_string,mrulistex_string_and_shell_item,mrulistex_string_and_shell_item_list,msie_zone,msiecf,mstsc_rdp,mstsc_rdp_mru,network_drives,opera_global,opera_typed_history,prefetch,recycle_bin,recycle_bin_info2,rplog,safari_history,symantec_scanlog,userassist,usnrnl,windows_boot_execute,windows_boot_verify,windows_run,windows_sam_users,windows_services,windows_shutdown,windows_task_cache,windows_timezone,windows_typed_urls,windows_usb_devices,windows_usbstor_devices,windows_version,winevt,winevtx,winfirewall,winjob,winlogon,winrar_mru,winreg,winreg_default



Name: lin

- Linux - 31 parsers

Binary_cookies,bsm_log,chrome_cache,chrome_cookies,chrome_extension_activity,chrome_history,chrome_preferences,cron,dockerjson,dpkg,filestat,firefox_cache,firefox_cache2,firefox_cookies,firefox_downloads,firefox_history,google_drive,imessage,java_idx,mac_appfirewall_log,mcafee_protection,opera_global,opera_typed_history,popularity_contest,safari_history,selinux,ssh,symantec_scanlog,utmp,utmpx,zsh_extended_history

Source: <https://github.com/rough007/CDQR/blob/master/docs/Parser%20Traceability%20Matrix.xlsx>

S04 - Lesson 04: All about the parsers



Name: mac

- MacOS - 46 parsers

airport,apple_id,appusage,binary_cookies,chrome_cache,chrome_cookies,chrome_extension_activity,chrome_history,chrome_preferences,cron,dockerjson,dpkg,filestat,firefox_cache,firefox_cache2,firefox_cookies,firefox_downloads,firefox_history,google_drive,imessage,ipod_device,java_idx,mac_appfirewall_log,mac_keychain,mac_securityd,mackeeper_cache,macosx_bluetooth,macosx_install_history,mactime,macuser,maxos_software_update,mcafee_protection,opera_global,opera_typed_history,plist,plist_default,popularity_contest,safari_history,spotlight,spotlight_volume,ssh,symantec_scanlog,time_machine,utmp,utmpx,zsh_extended_history



Name: datt

- Do All The Things - 129 parsers



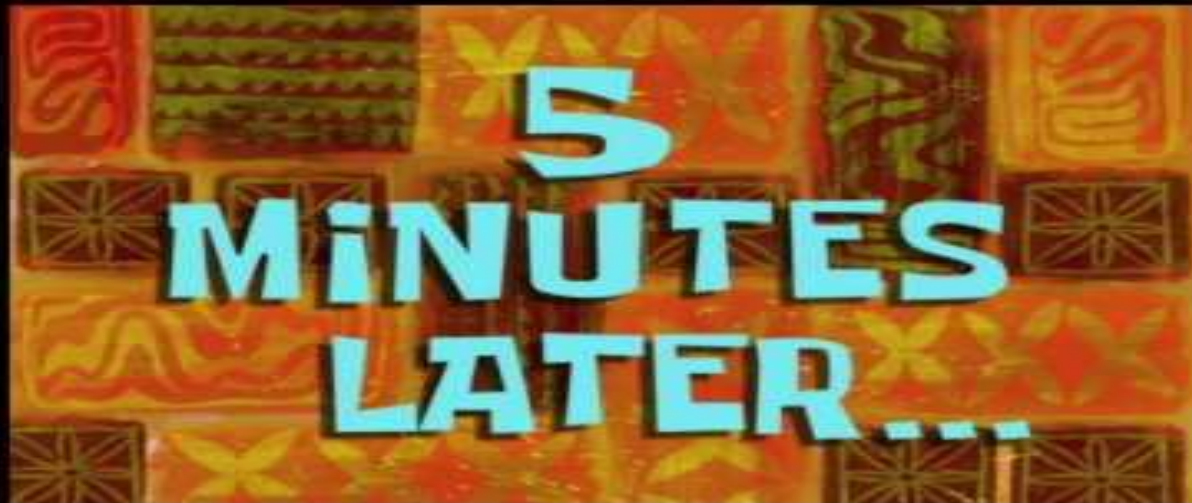
airport,android_app_usage,android_calls,android_sms,appcompatcache,apple_id,appusage,asl_log,bagmru,bencode,bencode_transmission,bencode_utorrent,binary_cookies,bsm_log,ccleaner,chrome_cache,chrome_cookies,chrome_extension_activity,chrome_history,chrome_preferences,cron,cups_ipp,custom_destinations,dockerjson,dpkg,esedb,esedb_file_history,explorer_mountpoints2,explorer_programscache,filestat,firefox_cache,firefox_cache2,firefox_cookies,firefox_downloads,firefox_history,google_drive,imessage,ipod_device,java_idx,kik_messenger,lnk,ls_quarantine,mac_appfirewall_log,mac_document_versions,mac_keychain,mac_securityd,mackeeper_cache,macosx_bluetooth,macosx_install_history,mactime,macuser,macwifi,maxos_software_update,mcafee_protection,mft,microsoft_office_mru,microsoft_outlook_mru,mrulist_shell_item_list,mrulist_string,mrulistex_shell_item_list,mrulistex_string,mrulistex_string_and_shell_item,mrulistex_string_and_shell_item_list,msie_webcache,msie_zone,msiecf,ms_tsc_rdp,mstsc_rdp_mru,network_drives,olecf,olecf_automatic_destinations,olecf_default,olecf_document_summary,olecf_summary,openxml,opera_global,opera_typed_history,pe,plist,plist_default,pls_recall,popularity_contest,prefetch,recycle_bin,recycle_bin_info2,rplog,safari_history,sccm,selinux,skydrive_log,skydrive_log_old,skype,spotlight,spotlight_volume,sqlite,ssh,symantec_scanlog,syslog,time_machine,twitter_ios,userassist,usnrnl,utmp,utmpx,windows_boot_execute,windows_boot_verify,windows_ru_n,windows_sam_users,windows_services,windows_shutdown,windows_task_cache,windows_timezone,windows_typed_urls,windows_usb_devices,windows_usbstor_devices,windows_version,winevt,winevtx,winfirewall,winiis,winjob,winlogon,winrar_mru,winreg,winreg_default,xchatlog,xchatscrollback,zeitgeist,zsh_extended_history

S04 - Lesson 05: Output options

- ⦿ Fixed
 - Plaso database file (*.db)
- ⦿ Default: CSV Reports
 - SuperTimeline (all data in one bucket)
 - Up to 16 Reports that group related data sets together
 - Additional parsing
 - ⦿ 560+ Event ID conversions (you're welcome)
 - ⦿ Additional columns of useful data
- ⦿ Optional: ElasticSearch (ES)
 - Entered into the ES database



S04 - Lesson 06: CDQR to CSV Reports



S04 - Lesson 06: CDQR to ElasticSearch



**A LITTLE LONGER
THAN A FEW
MINUTES LATER**

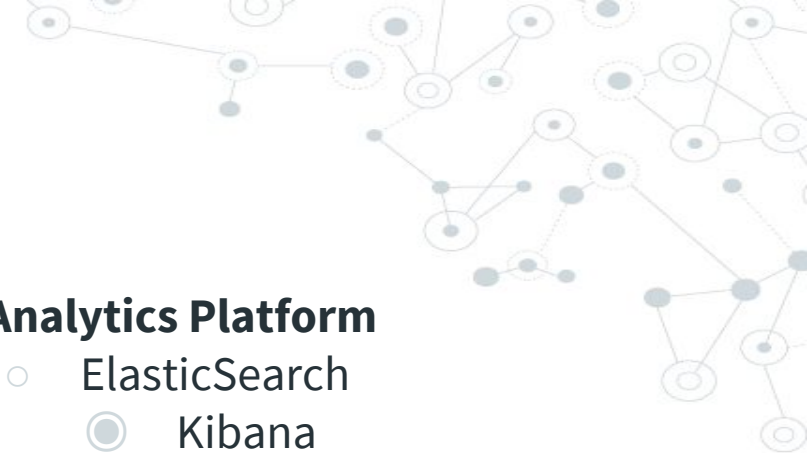
S04 - Practical: Process some data

- ⦿ Process *defcon.zip* using default parsers and output to CSV
- ⦿ Process *defcon.zip* using datt parser and output to CSV
- ⦿ Process *defcon.zip* using windows parsers, maximize CPU threads, and output to ES with index name, *defcon*
- ⦿ Write down command to process *sample_linux.vmdk* and output to CSV
- ⦿ Write down command to process *sample_folder* and output to CSV

Section 05

Analysis Methods





S05 - Lesson 01: Analysis Options

🎯 Flat Files (.csv)

- SuperTimeline
- Special Reports

🎯 Analytics Platform

- ElasticSearch
 - 🎯 Kibana
 - 🎯 Elasticsearch API
 - 🎯 TimeSketch



S05 - Lesson 02: Flat Files (.csv)

⦿ Advantages

- Widely used, standardized format
- Special reports for similar data sets
- No knowledge of databases or Kibana required

⦿ Disadvantages

- Harder to correlate data

⦿ Report Types - 16

- Appcompat, Login, Event Logs, File System, MFT, UsnJrnl, Internet History, Prefetch, Registry, Scheduled Tasks, Persistence, System Information, AntiVirus, Firewall, Mac, and Linux

S05 - Lesson 02: Flat Files - Optimizations

🎯 Optimized Reports

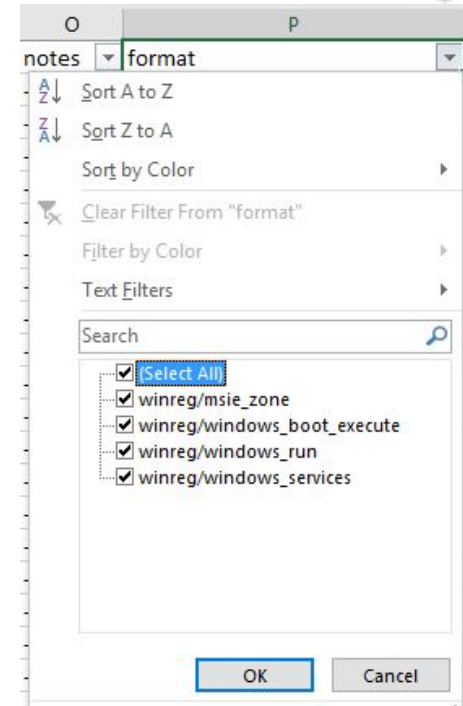
- Prefetch
- Appcompat
- Event Logs
- MFT
- Scheduled Tasks
- File System

🎯 Benefits

- Desc and extra fields parsed to dedicated columns
- Event ID is converted to english
- Event ID is parsed to dedicated column

S05 - Lesson 02: Flat Files - Best Practices

- Filter the “Format” column to determine parsers are available
- Date and Time fields should always be converted to UTC
- MACB - Modified, Accessed, Created, Born
- “Source” column states what type of artifact it came from
- “Desc” and “Extra” columns may contain multiple values, usually with colon delimited (and also in the values)
- “Filename” and “MD5” are the values of the artifact collected, not the content of the parser
- “Xml_string” column should contain the raw xml



S05 - Practical 01: Flat files

- ① Write down how many parsers are shown in the **Reports/Persistence Report.csv**?
- ① Write down how many rows were blank for column **EID_desc** in the **Reports/Event Log Report.csv**?
- ① How many entries doesC/Windows/System32/winevt have in the **Reports/Event Log Report.csv**? And why?
- ① How many unique file hashes are in the **Reports/MFT Report.csv**?
- ① How many rows have the word **logon** in the **desc** column in the **Reports/Registry Report.csv**?

S05 - Lesson 02: Analytics Platform

⦿ Advantages

- Multi-user platform
- Repeatable, fast searches
- Pivoting through data
- Custom Dashboards/Visualizations/Searches
- Sharing what works with community (Everyone gets better/faster/stronger)

⦿ Single Index (single target system)

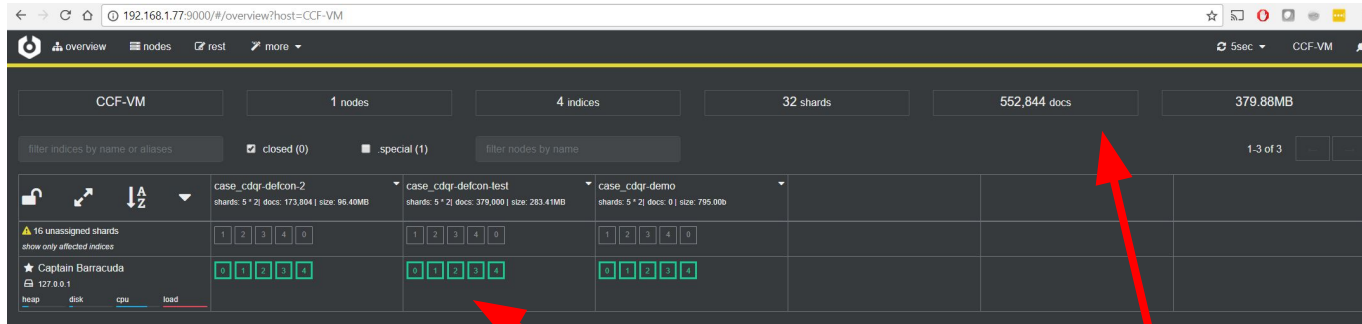
- Easy to limit searches on shared platform to one index

⦿ Stacking Indices

- Combine/compare sets of artifacts
- Search across multiple indices simultaneously

S05 - Lesson 03: Database Management (Cerebro)

Cerebro (<http://<your CCF-VM IP Address>:9000>) and similar plugins provide an easier (and visual) means of monitoring and modifying the data stores of Elasticsearch.



System resources

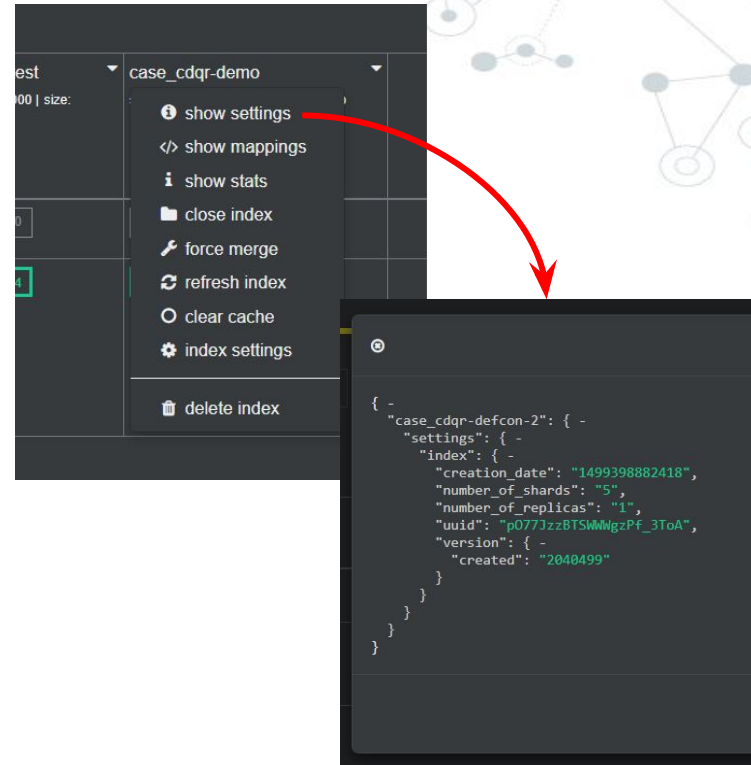
Index details

System summary

S05 - Lesson 03: Database Management (Cerebro)

Cerebro can be used to show the details of a given index, modify the index, or delete the index.

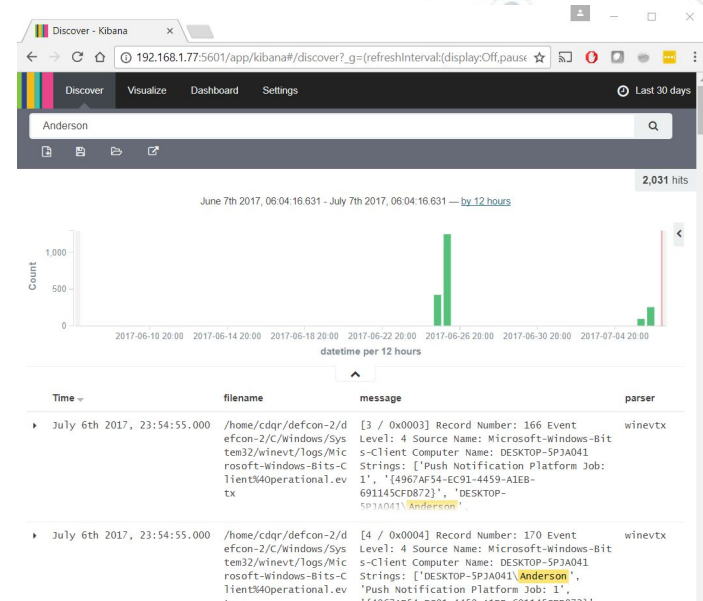
- ⦿ Settings details the index created as a result of the “--es <index>” option in CDQR
- ⦿ Deleting an index is easy and fast (you have been warned!)
- ⦿ Is, by default, an unauthenticated means of access to your data!



S05 - Lesson 04: Kibana

Kibana (<http://<your CCF-VM IP Address>:5601>) is the 'default' data front end in the ELK stack and provides a fast, highly extensible means of access to the volumes of data DFIR analysts produce. Of significant importance to us is the ability to 'record' data pivots and summaries of interest to us and create dashboards from them.

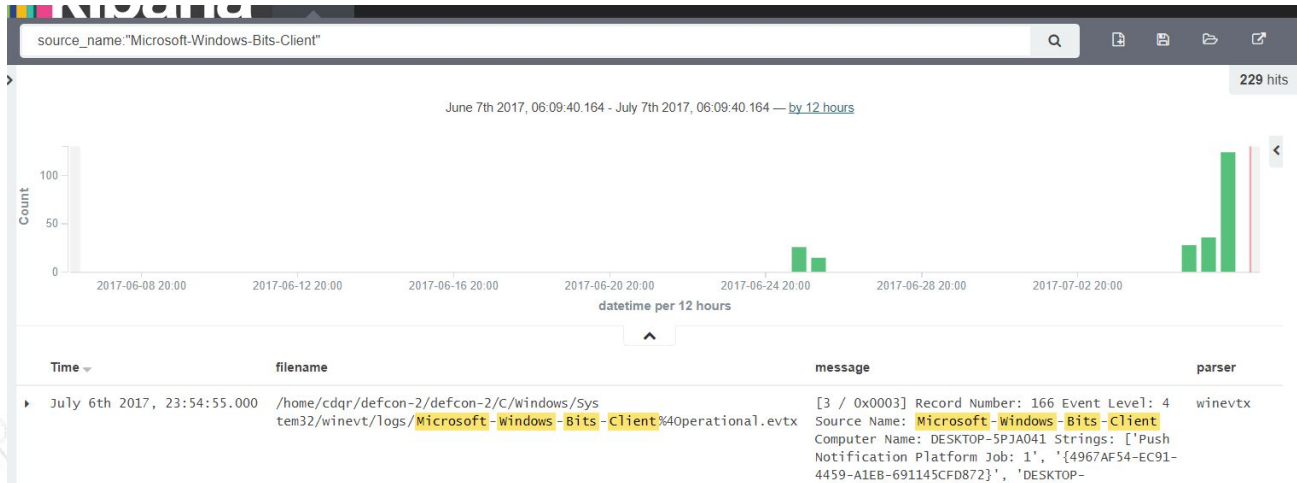
<https://www.elastic.co/products/kibana>



S05 - Lesson 04: Kibana, Searching

Kibana utilizes the Lucene search engine/query language. You can quickly search across all data, filter by index, or target specific data fields.

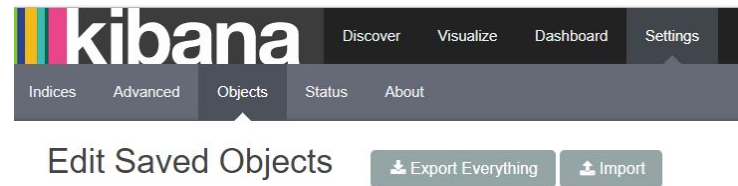
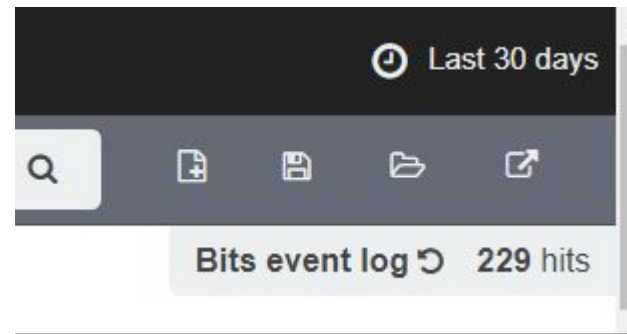
<https://www.elastic.co/guide/en/kibana/current/search.html>



S05 - Lesson 04: Kibana, Searching

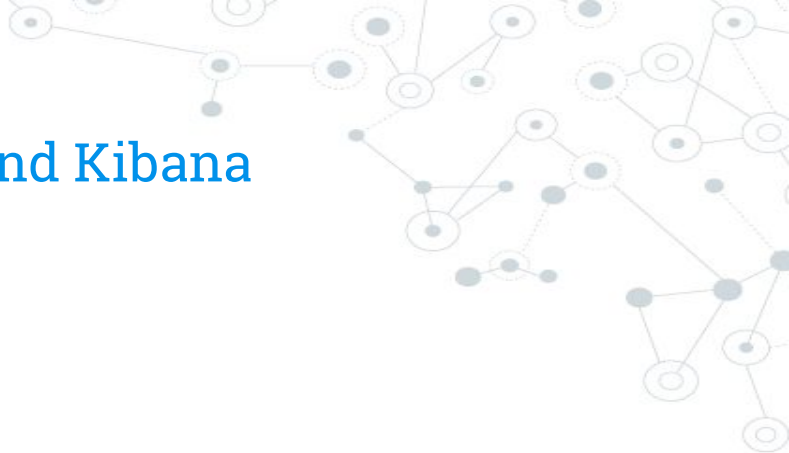
You can save your searches for later use

- ① Click the “Save” icon
- ① Find your previous saved searches under the Discovery tab and select the “Open” icon. Search for the name of your saved search
- ① You can also share a saved search for troubleshooting or exporting purposes
- ① Are managed through “Settings -> Objects -> Searches”



S05 - Lesson 04: Kibana, Searching

- Field:<value>
 - message:notepad
 - source_name:(Bits OR security)
 - source_name:(Windows OR security)
- Wild cards: ? * ~
- Booleans: + - AND && OR || NOT !
- Grouping: (notepad OR notepad++) AND parser:mft
- Ranges: [1 TO 20] {5 TO 56}
- Comparisons: >, >=, =, <=, <
- Reserved chars: + - = && || > < ! () { } [] ^ " ~ * ? : \ /
- Escape char: \



S05 - Practical: Using Elasticsearch and Kibana

- ◎ Loading data into Elasticsearch
 - CDQR
 - Manually
 - ◎ Reviewing results
 - ◎ Eliminating known values (hash sets)
- 

S05 - Practical: Using ElasticSearch and Kibana

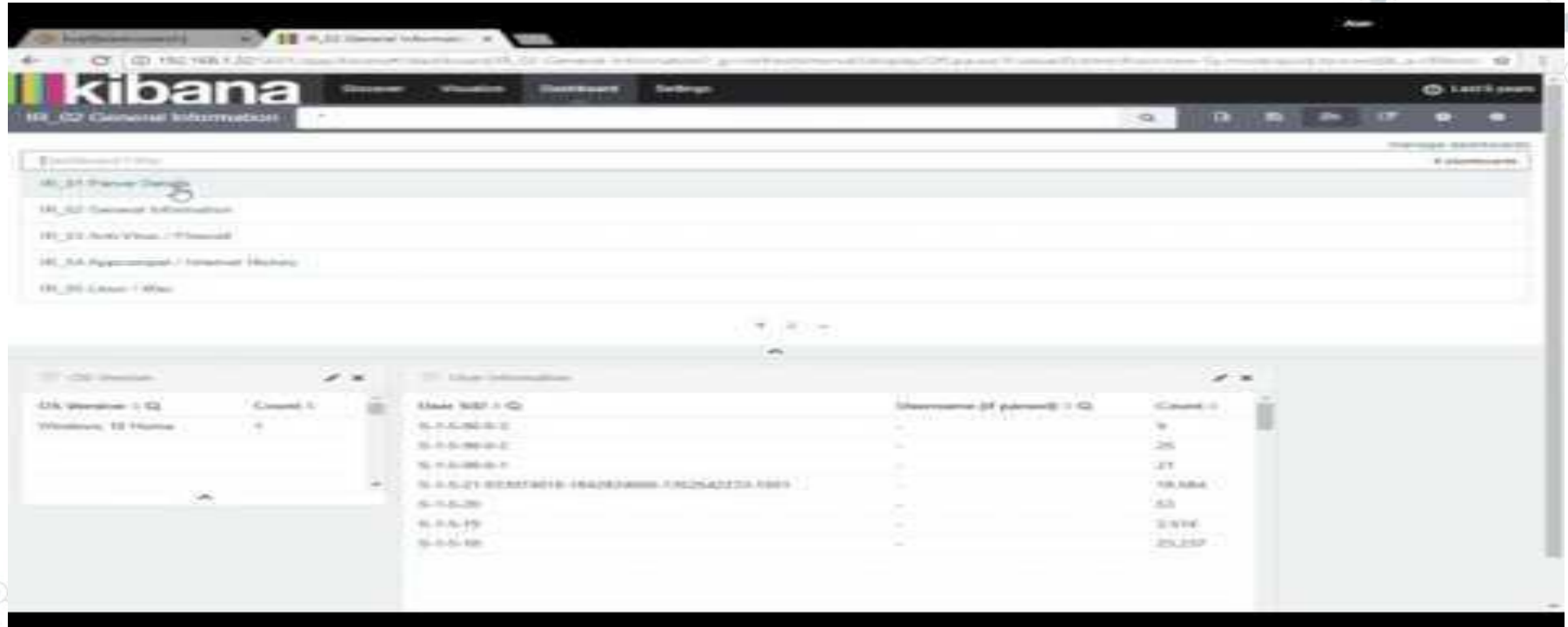
- ◎ Loading data into ElasticSearch
 - CDQR
 - ◎ `cdqr.py -p win -z defcon-test.zip --max_cpu --es defcon`
 - ◎ 'defcon' will be the final part of the index name this data is loaded under resulting in an index named 'case_cdqr-defcon'
 - Manually
 - ◎ `psort.py -o elastic --raw_fields --index_name case_cdqr-demo demo.db`
 - ◎ This will result in an index named 'case_cdqr-defcon'
- ◎ All CCF-VM dashboards default to using any index matching 'case_cdqr*'

S05 - Practical: Using Elasticsearch and Kibana

© Reviewing results

- Perform an open search for **system32**
 - What fields were matched? (from the left, select filename, message, and parser to help with views)
 - Is it case sensitive?
 - How many results did you get?
- Perform a search for **filename:"System32"**
- Perform a search for **filename.raw:"System32"**
- Why are the results above different?

S05 - Lesson 05: Using Kibana Dashboards



Credit: Open Source DFIR Made Easy (Alan Orlikoski & Stephen Hinck)

<https://www.youtube.com/watch?v=S6-ZM1SjFcY>

S05 - Practical: Using Kibana Dashboards

© Dashboards

- Open the dashboard tab
- Review the supplied dashboards (Load saved Dashboards)
 - Parser Details
 - General Information
 - Ant-Virus / Firewall
 - Appompat / Internet History
 - Linux /Mac
 - Persistence / Prefetch

S05 - Practical: Modifying Dashboards

- ◎ Open the Parser Details dashboard
 - Click the edit icon
 - Review the options
 - Change some of the values and watch the changes on the right
 - Change 'Order By' value to 'metric: Number of Records' and click the 'Play icon'
 - Save the new visualization
 - Did this change the dashboard?

The screenshot shows the Kibana interface with the 'Parser Details' dashboard. The top navigation bar includes 'Discover', 'Visualize', 'Dashboard', and 'Settings'. The dashboard title is 'IR_01 Parser Details'. Below the title, there is a 'Parser Results' section with a table showing the number of records for different parsers. A red circle highlights the edit icon (pencil) in the top right corner of this section. A red arrow points from this icon to the 'Order By' dropdown menu in the 'buckets' section of the 'case_cdqr-' visualization, which is currently set to 'Term'. The 'buckets' section also shows 'Split Rows' and 'Aggregation' options. The 'metrics' section shows a table with 'Parser Results' and 'Number of Records'.

Parser Results	Number of Records
winreg/winreg_default	259,162
winreg/winlogon	4

Parser Results	Number of Records
winreg/winreg_default	259,162
winreg/winlogon	4
winreg/windows_version	3
winreg/windows_task_cache	407
winreg/windows_run	2
winreg/msie_zone	24
winevtx	17,685
mtt	1,730,715
filestat	441

S05 - Lesson 06: Using Elasticsearch - API

Like everything else out there these days, Elasticsearch supports a very robust API. The API exposes more of the features and capabilities of the tools than Kibana and should be considered for automation or advanced queries.

- ◎ From the command line in CCF-VM
 - `sudo apt-get install jq`
 - `curl -s -XGET 'localhost:9200/_search?q="notepad"&pretty'|jq '.hits.hits[]._source.message'`

```
cdqr@CCF_VM:~/Results$ curl -s -XGET 'localhost:9200/_search?q="notepad"&pretty'|jq '.hits.hits[]._source.message'
"[HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Notepad\\DefaultFonts] iPointSize: [REG_DWORD_LE] 110 lfFaceName: [REG_SZ] Consolas"
"[HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Notepad] Value: No values stored in key."
"[HKEY_LOCAL_MACHINE\\Software\\WOW6432Node\\Microsoft\\Notepad] Value: No values stored in key."
"[28115 / 0x6dd3] Record Number: 57 Event Level: 4 Source Name: Microsoft-Windows-Shell-Core Computer Name: DESKTOP-5PJA041 Strings: ['Notepad', '{1A', '56'}]"
"[28115 / 0x6dd3] Record Number: 257 Event Level: 4 Source Name: Microsoft-Windows-Shell-Core Computer Name: DESKTOP-5PJA041 Strings: ['Notepad', '{1e', '56'}]"
cdqr@CCF_VM:~/Results$
```

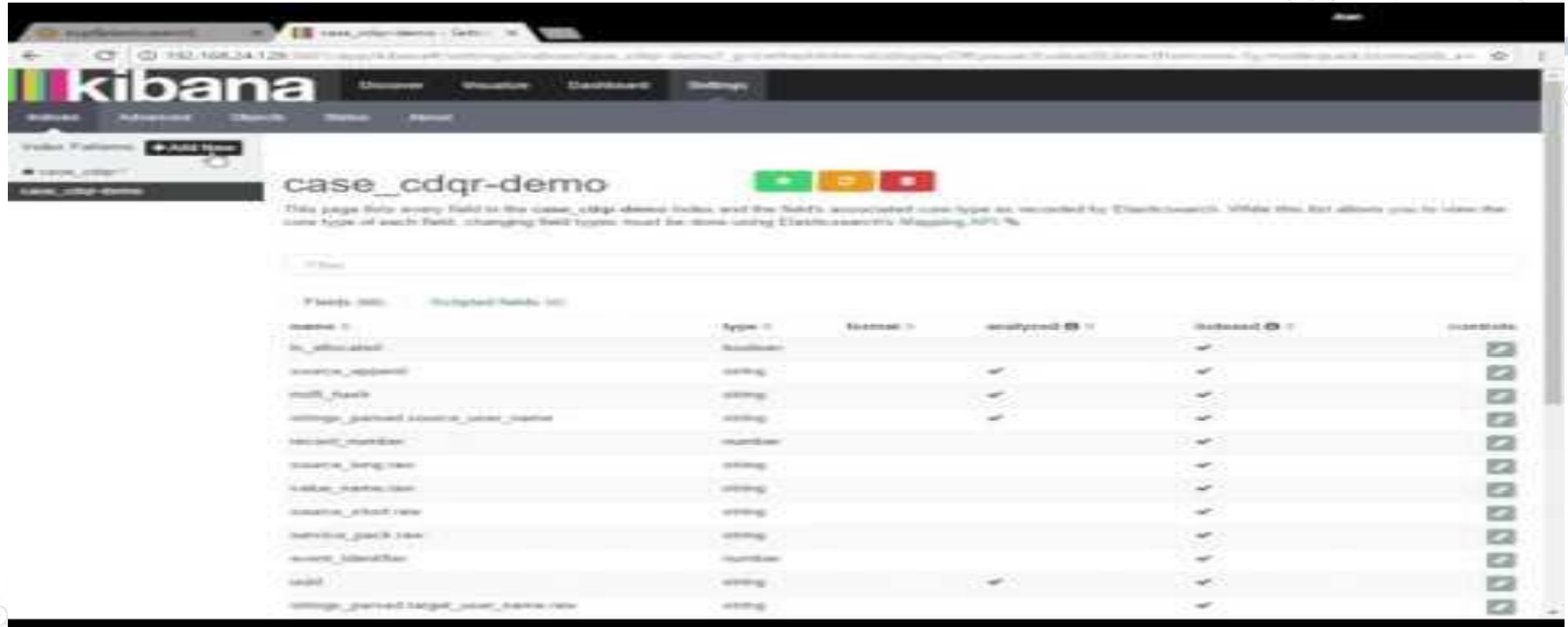
S05 - Lesson 06: Using Elasticsearch - API

A real world IR search: Use of BITS for data movement

```
curl -s -XPOST 'localhost:9200/case_cdqr-defcon-2/_search?pretty' -d  
'{"query": { "match_phrase": { "source_name":  
"Microsoft-Windows-Bits-Client" } } }'|jq '.hits.hits[]._source.message' | cut -d,  
-f4
```

```
cdqr@CCF_VM:~/Results$ curl -s -XPOST 'localhost:9200/case_cdqr-defcon-2/_search?pretty  
rce.message' | cut -d, -f4  
'http://7.tlu.dl.delivery.mp.microsoft.com/filestreamingservice/files/2c617786-3573-43  
'https://oneclient.sfx.ms/Win/Prod/17.3.6917.0607/OneDriveSetup.exe'  
  
'https://fs.microsoft.com/fs/windows/config.json'  
'https://fs.microsoft.com/fs/windows/fontset-2017-04.json'  
'http://img-s-msn-com.akamaized.net/tenant/amp/entityid/BBDMAf6.img?w=100&h=100&m=6&t  
'https://msftspeechmodelsprod.azureedge.net/SR/SV10-EV100/en-us-n/MV101/11033.ini'  
'https://msftspeechmodelsprod.azureedge.net/SR/SV10-EV100/en-us-n/MV101/11033.smp'  
'https://msftspeechmodelsprod.azureedge.net/SR/SV10-EV100/en-us-n/MV101/11033.wwd'  
cdqr@CCF_VM:~/Results$
```

S05 - Lesson 07: Working with Indices (stacking)

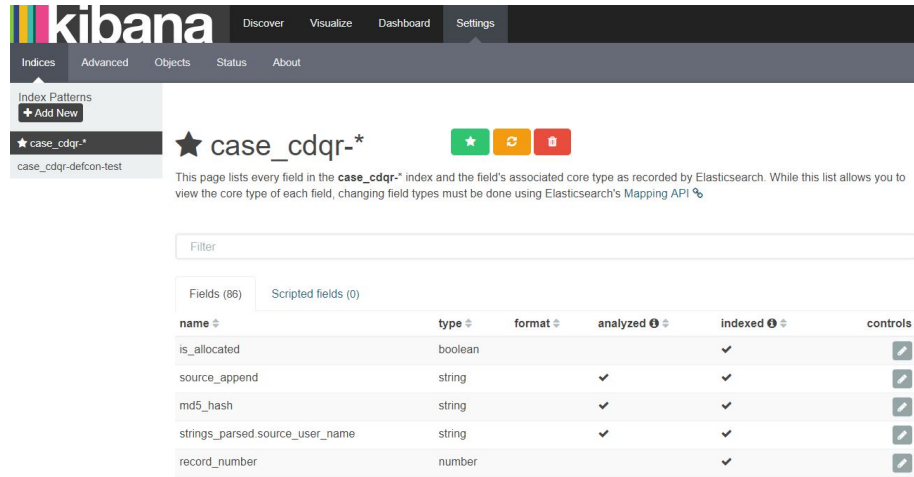


The screenshot shows the Kibana interface for the 'case_cdqr-demo' index. The top navigation bar includes 'Discover', 'Visualize', 'Dashboard', and 'Settings'. The left sidebar shows the 'case_cdqr-demo' index pattern. The main content area displays a list of fields with their types and whether they are indexed or analyzed.

Field	Type	Indexed	Analyzed	Example
to_address	string	✓	✓	
source_address	string	✓	✓	
mail_base	string	✓	✓	
message_parsed.source_user_name	string	✓	✓	
message_number	number	✓	✓	
message_lang_raw	string	✓	✓	
value_message_text	string	✓	✓	
message_subject_raw	string	✓	✓	
message_park_raw	string	✓	✓	
event_id	number	✓	✓	
id	string	✓	✓	
message_parsed.target_user_name	string	✓	✓	

S05 - Practical: Kibana Working with Indices

Indices are the basic data bundles that Elasticsearch/Kibana use. By default, CCF-VM uses a combined index for all case data by creating an index pattern of **case_cdqr-***. CDQR automatically prefixes each --es entry into ELK with this string.



The screenshot shows the Kibana interface with the 'Settings' tab selected. The left sidebar shows the 'Indices' section with the 'case_cdqr-*' index pattern selected. The main content area displays the settings for this index pattern, including a list of fields and their associated core types.

★ case_cdqr-*

This page lists every field in the **case_cdqr-*** index and the field's associated core type as recorded by Elasticsearch. While this list allows you to view the core type of each field, changing field types must be done using Elasticsearch's Mapping API.

Filter

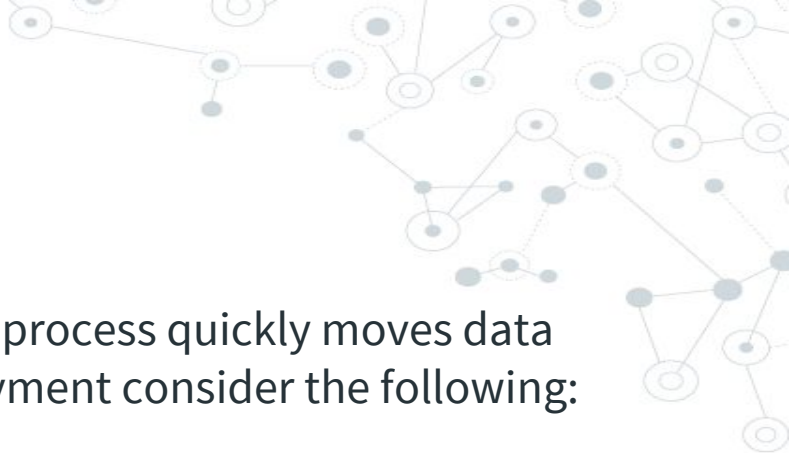
Fields (86) Scripted fields (0)

name	type	format	analyzed	indexed	controls
is_allocated	boolean			✓	✓
source_append	string		✓	✓	✓
md5_hash	string		✓	✓	✓
strings_parsed source_user_name	string		✓	✓	✓
record_number	number			✓	✓

Section 06

Final Chapter

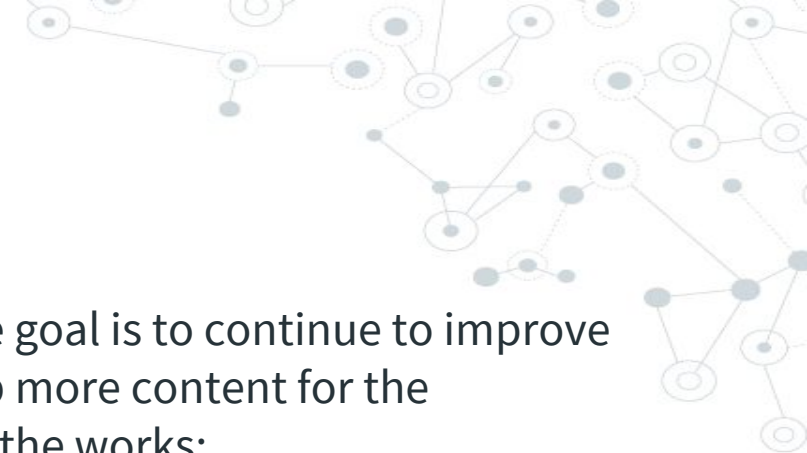




S06 - Lesson 01: Going forward

The core ELK platform is highly extensible. This process quickly moves data from endpoints to ELK for you. For a real deployment consider the following:

- ◎ Bare-iron or dedicated installation
 - ◎ HA and ELK performance optimizations
 - ◎ Authentication (X-Pack)
 - ◎ Data protection standards for your organization
- 



S06 - Lesson 02: Future Work

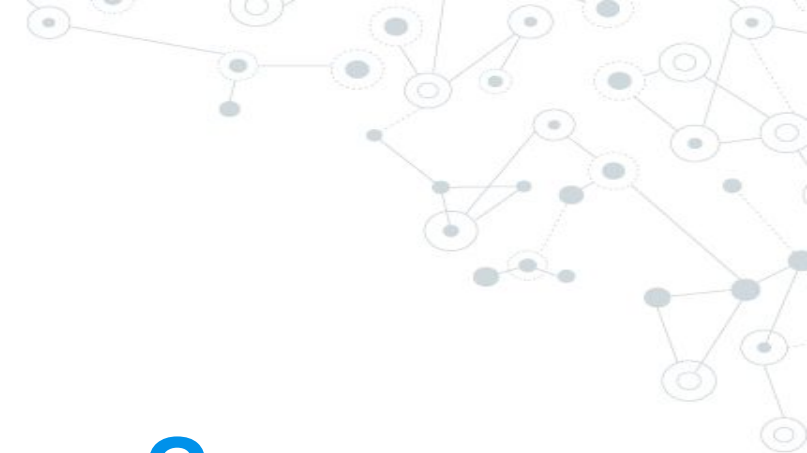
CCF-VM is being (semi) actively developed. The goal is to continue to improve its performance and feature set. As we develop more content for the dashboards they will be added. Some items in the works:

🎯 **Tool Improvements**

- Updates to CyLR
- Updates to CDQR

🎯 **Updates to CCF-VM**

- Better dashboards
 - Better timeline presentation (Dan)
 - CDQR integration for Timesketch
- 



Questions?