

**XIII
JORNADAS
STIC
CCN-CERT**



**ThreatHunting
práctico: no tengas
miedo a Sigma y EQL**

**COMUNIDAD Y CONFIANZA,
BASES DE NUESTRA CIBERSEGURIDAD**

#XIIIJORNADASCNCERT



David Barroso Berrueta

CounterCraft

dbarroso@countercraft.eu

Índice

1. Gamification of Threat Hunting
2. MITRE ATT&CK, EQL and Sigma
3. Theory
4. Practice



Home



Explore



Notifications



Messages



Bookmarks



Lists



Profile



More

Tweet



Tweet



Samir @SBousseaden · Oct 15

example of [#threathunting](#) query (not real-time alert!) for some "suspicious" rar|7z cmdline params (-dw & -sdel for deletion of compressed stuff, -ta & -tb is for compressing files with mcao time after/before YYYYMMDDHHMMSS & -p is for pwd, u can also add cond 4 renamed process.

```
process where opcode==1 and
wildcard(original_file_name,"7z*.exe","rar.exe","*winrar*"
,"*Command*line*RAR*") and
wildcard(command_line,"* -p*" "* -ta*" "* -tb*" "* -sdel*" "*
-dw") and parent_process_path!="c:\\program
files*\\*"
```



5



37



99



Samir Retweeted



Florian Roth
@cyb3rops

Replying to @SBousseaden

I'd write the rule like that :

```
1 title: Suspicious Archiver Parameters
2 status: experimental
3 description: Detects programs running in suspicious files system locations
4 references:
5   - https://twitter.com/SBousseaden/status/1184067445612535811
6 tags:
7   - attack.exfiltration
8   - attack.t1020
9   - attack.t1002
10 author: Florian Roth, Samir Bousseaden
11 date: 2019/10/15
12 logsource:
13   category: process_creation
```



Search Twitter

Relevant people



Florian Roth
@cyb3rops

Following

Nextron Systems [#DFIR](#) [#YARA](#)
[#ThreatIntel](#) | Creator of
[@thor_scanner](#), Valhalla YARA rule feed,
Sigma, LOKI, yarGen & much more



Samir
@SBous... Follows you

Following

Security Researcher [#DFIR](#)
[#ThreatHunting](#)

Trends for you



Trending in Spain

Cataluña

29.5K Tweets

Trending in Spain

#OzunaEH

Trending in Spain

NO LIGUES

4,972 Tweets

Trending in Spain

Cepeda

13.2K Tweets

Trending in Spain

Arsenal

164K Tweets

Detection Process

- Use ATT&CK to identify common behaviors, instead of just tools
- Explore the mind of the attacker
- Understand your data and visibility
- Express detection logic for *your* platform
- Continuously create, test, and refine analytics

MITRE ATT&CK

Tactics

ID	Name	Description
TA0001	Initial Access	The adversary is trying to get into your network.
TA0002	Execution	The adversary is trying to run malicious code.
TA0003	Persistence	The adversary is trying to maintain their foothold.
TA0004	Privilege Escalation	The adversary is trying to gain higher-level permissions.
TA0005	Defense Evasion	The adversary is trying to avoid being detected.
TA0006	Credential Access	The adversary is trying to steal account names and passwords.
TA0007	Discovery	The adversary is trying to figure out your environment.
TA0008	Lateral Movement	The adversary is trying to move through your environment.
TA0009	Collection	The adversary is trying to gather data of interest to their goal.
TA0011	Command and Control	The adversary is trying to communicate with compromised systems to control them.
TA0010	Exfiltration	The adversary is trying to steal data.
TA0040	Impact	The adversary is trying to manipulate, interrupt, or destroy your systems and data.

MITRE ATT&CK

Techniques

MITRE ATT&CK™

[Matrices](#)[Tactics ▾](#)[Techniques ▾](#)[Mitigations ▾](#)[Groups](#)[Software](#)[Resources ▾](#)[Blog ↗](#)[Contribute](#)

ENTERPRISE ▾

TECHNIQUES

[Overview](#)[Initial Access ▾](#)[Execution ▾](#)[Persistence ▴](#)[.bash_profile and .bashrc](#)[Accessibility Features](#)[Account Manipulation](#)[AppCert DLLs](#)[AppInit DLLs](#)[Application Shimming](#)[Authentication Package](#)[BITS Jobs](#)[Bootkit](#)[Browser Extensions](#)[Change Default File Association](#)[Home](#) > [Techniques](#) > [Enterprise](#)

Enterprise Techniques

Enterprise Techniques: 266

ID	Name	Description
T1156	.bash_profile and .bashrc	<code>~/.bash_profile</code> and <code>~/.bashrc</code> are shell scripts that contain shell commands. These files are executed in a user's context when a new shell opens or when a user logs in so that their environment is set correctly. <code>~/.bash_profile</code> is executed for login shells and <code>~/.bashrc</code> is executed for interactive non-login shells. This means that when a user logs in (via username and password) to the console (either locally or remotely via something like SSH), the <code>~/.bash_profile</code> script is executed before the initial command prompt is returned to the user. After that, every time a new shell is opened, the <code>~/.bashrc</code> script is executed. This allows users more fine-grained control over when they want certain commands executed. These shell scripts are meant to be written to by the local user to configure their own environment.
T1134	Access Token Manipulation	Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token. For example, Microsoft promotes the use of access tokens as a security best practice. Administrators should log in as a standard user but run their tools with administrator privileges using the built-in access token manipulation command <code>runas</code> .
T1015	Accessibility Features	Windows contains accessibility features that may be launched with a key combination before a user has logged in (for example, when the user is on the Windows logon screen). An adversary can modify the way these programs are launched to get a command prompt or backdoor without logging in to the system.
T1531	Account Access Removal	Adversaries may interrupt availability of system and network resources by inhibiting access to accounts utilized by legitimate users. Accounts may be deleted, locked, or manipulated (ex: changed credentials) to remove access to accounts.

MITRE Cyber Analytics Repository

Implementations – Example CAR-2019-08-001: Credential Dumping via Windows Task Manager

Procdump - File Create (Pseudocode)

This base pseudocode looks for file create events where a file with a name similar to lsass.dmp is created by the Windows task manager process.

```
files = search File:Create
lsass_dump = filter files where (
  file_name = "lsass*.dmp" and
  image_path = "C:\Windows\*\taskmgr.exe")
output lsass_dump
```

Procdump - File Create (Splunk, Sysmon native)

A Splunk/Sysmon version of the above pseudocode.

```
index=__your_sysmon_index__ EventCode=11 TargetFilename="*lsass*.dmp" Image="C:\\Windows\\*\\taskmgr.exe"
```

Procdump - File Create (EqL, EQL native)

An EQL version of the above pseudocode.

```
file where file_name == "lsass*.dmp" and process_name == "taskmgr.exe"
```

Unit Tests

Test Case 1

1. Open Windows Task Manager as Administrator
2. Select lsass.exe
3. Right-click on lsass.exe and select "Create dump file".

MITRE Cyber Analytics Repository

Implementations

We usually have the following languages:

- Pseudocode
- Sysmon / Splunk
- Sigma
- EQL – Endgame Query Language

My Recommendation

First focus on suspicious and/or anomalous activity (EQL is great for that)

Then, focus on how threat actors are abusing the system (Sigma is great for that)

EQL (2018)

Implementations

- Event Query Language is simple and concise
- Schema-independent and OS-agnostic
- Designed for real-time detection with stream processing
- Supports multi-event behaviors, stacking and sifting through data
- Function syntax instead of keyword explosion (e.g. length(field))



EQL (2018)

Simples queries

- Boolean and comparison logic and or not< <= == != >=>
- Wildcard matching with * carácter
- String comparisons are case-insensitive



process where process_name == "svchost.exe" **and** (command_line != "* -k *" **or** parent_process_name != "services.exe")

EQL (2018)

Sequences

- Multi-event behaviors with ordering
- Match properties between events with by syntax
- Time limits maxspan=1 hr
- Sequences can be expired with an until condition

sequence with maxspan=5m

[**file where** file_name == "*.exe"] **by** user_name, file_path [**process where** true] **by** user_name, process_path



EQL (2018)

Joins

- Match events specified, without time limits
- Supports by and until syntax for additional matching or state
- Unlike SQL, it finds adjacent pairs instead of cross-products



join

```
[file where file_path == "*\\System32\\Tasks\\h4x0r.xml"] [registry where registry_path ==  
"\\runonce\\h4xor"]
```

EQL (2018)

Joins

- Match events specified, without time limits
- Supports by and until syntax for additional matching or state
- Unlike SQL, it finds adjacent pairs instead of cross-products



```
join by source_ip, destination_ip  
[network where destination_port == 3389] // RDP [network where destination_port == 135] //  
RPC [network where destination_port == 445] // SMB
```

EQL (2018)

Pipes and filter

- Pipes can be used to transform or reduce output
- Combine in various ways to perform stacking or reduce data set
- **count filter head sort tail unique unique_count**

```
process where true
```

```
// Remove duplicate pairs
```

```
| unique process_name, command_line
```

```
// Count per process_name to get unique # of commands
```

```
| count process_name | filter count < 5
```



EQL (2018)

Parents and children

- Natively tracks process lineage by monitoring create/terminate events.
- Supports **descendant of**, **child of**, and **event of**
- Combine with other boolean logic



network where process_name == "powershell.exe"

and not descendant of

[process where process_name == "explorer.exe"]

EQL (2018)

Parents and children

- Natively tracks process lineage by monitoring create/terminate events.
- Supports **descendant of**, **child of**, and **event of**
- Combine with other boolean logic



```
file where file_name == "*.exe"  
  and event of [process where child of  
    [process where process_name == "powershell.exe"]]
```

EQL (2018)

Examples

Technique: Spearphishing Attachment (T1193) PowerShell (T1059)



process where

```
parent_process_name in ("winword.exe", "excel.exe", "powerpnt.exe")  
and process_name in ("powershell.exe", "cscript.exe",  
"wscript.exe", "cmd.exe")
```

EQL (2018)

Examples

Technique: Spearphishing Attachment (T1193)



```
sequence with maxspan=5m  
  [file where file_name == "*.exe"  
    and process_name in ("winword.exe", "excel.exe", "powerpnt.exe") ] by file_path  
  [process where true] by process_path
```

EQL (2018)

Examples

- Technique: Scheduled Task (T1053)

```
process where process_name == "schtask.exe"  
    and user_name != "SYSTEM"  
    and (command_line == "* /ru system" or  
        command_line == '* /ru "nt authority\\")  
| unique user_name, command_line
```



EQL (2018)

Examples

- Technique: Masquerading (T1096)



```
process where process_name in (  
    "csrss.exe", "dllhost.exe", "lsass.exe",  
    "lsm.exe", "services.exe", "winlogon.exe",  
    /* etc */  
    ) and not (process_path == "C:\\windows\\system32\\*" and  
                process_path != "C:\\windows\\system32\\*\\")
```

EQL (2018)

Examples

- Technique: Access Sensitive Data or Credentials in Files (T



```
process where process_name == "findstr.exe"  
and command_line == "*password*" | unique parent_process, command_line
```

EQL (2018)

Examples

- Technique: Account Discovery (T1087) Remote System Discovery (T1033) Discovery (T1033)



join by user_name

[process where process_name in

("ipconfig.exe", "hostname.exe", "whoami.exe")] **[process** where process_name == "net.exe" and

(**command_line** == "*group*" or **command_line** == "* user*")]

[process where **process_name** in ("tasklist.exe", "qprocess.exe", "sc.exe")]

| unique **user_name**

EQL (2018)

Examples

- Technique: Data Staged (T1074)
Data Compressed (T1072) Data Encrypted (T1022)



```
sequence by unique_pid with maxspan=5m  
  [process where command_line == "* -hp*" or command_line == "* /hp*"]  
  [file where file_name == "*.rar"]  
| unique events[0].process_path, events[1].file_name
```

EQL (2018)

Examples

- Technique: Inhibit System Recovery (T1490)



process where

(process_name == "vssadmin.exe" **and**
command_line == "*delete*") **or** (process_name == "wmic.exe" **and**
command_line == "*shadow*delete*") **or**
(process_name == "wevtutil.exe" **and** command_line == "* cl *")

CONTENTS

Getting Started

Analytics

Access of Outlook Email Archives

Account Discovery via Built-In Tools

AD Dumping via Ntdsutil.exe

Adding the Hidden File Attribute with via attrib.exe

AppCert DLLs Registry Modification

Audio Capture via PowerShell

Audio Capture via SoundRecorder

Bypass UAC via CMSTP

Bypass UAC via Fodhelper.exe

Bypass UAC via Fodhelper.exe

Bypass UAC via WSReset.exe

Change Default File Association

Clearing Windows Event Logs with wevtutil

COM Hijack via Script Object

Command-Line Creation of a RAR file

Control Panel Items

Creation of an Archive with

Analytics

Analytic	Contributors	Updated	Tactics	Techniques
Access of Outlook Email Archives	Endgame	7/26/2019	Collection	T1114 Email Collection
Account Discovery via Built-In Tools	Endgame	7/26/2019	Discovery	T1087 Account Discovery
AD Dumping via Ntdsutil.exe	Tony Lambert	01/07/2019	Credential Access	T1003 Credential Dumping
Adding the Hidden File Attribute with via attrib.exe	Endgame	7/26/2019	Defense Evasion Persistence	T1158 Hidden Files and Dir
AppCert DLLs Registry Modification	Endgame	7/26/2019	Privilege Escalation Persistence	T1182 AppCert DLLs
Audio Capture via PowerShell	Endgame	11/30/2018	Collection	T1123 Audio Capture
Audio Capture via SoundRecorder	Endgame	11/30/2018	Collection	T1123 Audio Capture
Bypass UAC via CMSTP	Endgame	11/30/2018	Defense Evasion Execution	T1191 CMSTP T1088 Bypass User Accoi
Bypass UAC via Fodhelper.exe	Tony Lambert	05/17/2019	Privilege Escalation	T1088 Bypass User Account
Bypass UAC via Fodhelper.exe	Tony Lambert	05/17/2019	Privilege Escalation	T1088 Bypass User Account
Bypass UAC via WSReset.exe	Tony Lambert	05/17/2019	Privilege Escalation	T1088 Bypass User Account
Change Default File Association	Endgame	11/30/2018	Persistence	T1042 Change Default File /
Clearing Windows Event Logs with wevtutil	Endgame	11/30/2018	Defense Evasion	T1070 Indicator Removal on
COM Hijack via Script Object	Endgame	11/30/2018	Persistence Defense Evasion	T1122 Component Object M
Command-Line Creation of a RAR file	Endgame	11/30/2018	Exfiltration	T1002 Data Compressed

SIGMA

The contender



Generic Signature Format for SIEM Systems

SIGMA

The contender

- <https://github.com/Neo23x0/sigma/wiki>

title id [optional] related [optional] - type {type-identifier} id {rule-id} status [optional] description [optional] author [optional] references [optional] logsource category [optional] product [optional] service [optional] definition [optional] ... detection {search-identifier} [optional] {string-list} [optional] {field: value} [optional] ... timeframe [optional] condition fields [optional] falsepositives [optional] level [optional] tags [optional] ... [arbitrary custom fields]

SIGMA

The contender

- <https://github.com/Neo23x0/sigma/wiki>
- <https://github.com/Neo23x0/sigma/wiki/Fields:-Processes>

Fields: Processes

Florian Roth edited this page on 13 Aug · 6 revisions

APLHA - used for discussions. No production use yet.

See page "[Taxonomy](#)" for information on the current schemes.

Field	Type	Description
process_action	string	Endpoint solution's action on process (e.g. 'allow', 'block')
process_authentihash	string	Process image Authentihash (e.g. 'cfc5937b4db3a1d5718fe144b621d50b0a337854ab3f5e2')
process_authentihash	string	Process image Authentihash (e.g. 'cfc5937b4db3a1d5718fe144b621d50b0a337854ab3f5e2')
process_cmd	string	Full process command line (e.g. 'ssh -l user 10.0.0.16')
process_cpu_load	float	CPU load consumed by the process (in percent) (e.g. '98.5')
process_cwd	string	The process' working directory (e.g. 'C:\temp')
process_image	string	Process executable (e.g. 'C:\Windows\System32\calc.exe')
process_image_company	string	Process image file copyright (e.g. 'Microsoft Corporation')
process_image_description	string	Process image file description (e.g. 'Microsoft Management Console')
process_image_file_version	string	Process image file version (e.g. '6.1.7600.16385 (win7_rtm.060917.0148)')
process_image_product	string	Process image file product (e.g. 'Microsoft® Windows® Operating System')
process_imphash	string	Process image Imphash (e.g. 'ceefb55f764020cc5c5f8f23')
process_imphash	string	Process image Imphash (e.g. 'ceefb55f764020cc5c5f8f23')

```
3 status: experimental
4 description: Detects WannaCry ransomware activity
5 references:
6   - https://www.hybrid-analysis.com/sample/ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa?environmentId=100
7 author: Florian Roth (rule), Tom U. @c_APT_ure (collection)
8 logsource:
9   category: process_creation
10  product: windows
11 detection:
12   selection1:
13     Image:
14       - '*\tasksche.exe'
15       - '*\mssecsvc.exe'
16       - '*\taskdl.exe'
17       - '*\@WanaDecryptor@*'
18       - '*\WanaDecryptor*'
19       - '*\taskhsvc.exe'
20       - '*\taskse.exe'
21       - '*\111.exe'
22       - '*\lhdfgui.exe'
23       - '*\diskpart.exe'
24       - '*\linuxnew.exe'
25       - '*\wannacry.exe'
26   selection2:
27     CommandLine:
28       - '*icaccls * /grant Everyone:F /T /C /Q*'
29       - '*bcdedit /set {default} recoveryenabled no*'
30       - '*wbadmin delete catalog -quiet*'
31       - '*@Please_Read_Me@.txt*'
32   condition: 1 of them
33 fields:
34   - CommandLine
35   - ParentCommandLine
36 falsepositives:
37   - Diskpart.exe usage to manage partitions on the local hard drive
38 level: critical
```

```
1  title: Ransomware Deletes Volume Shadow Copies
2  id: 4eebe114-4b24-4a9d-9a6c-c7bd7c8eaa61
3  status: experimental
4  description: Detects commands that delete all local volume shadow copies as used by different Ransomware families
5  references:
6      - https://www.bleepingcomputer.com/news/security/why-everyone-should-disable-vssadmin-exe-now/
7      - https://www.hybrid-analysis.com/sample/ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa?environmentId=100
8  author: Florian Roth
9  date: 2019/06/01
10 logsources:
11     category: process_creation
12     product: windows
13  detection:
14     selection:
15         CommandLine:
16             - '*vssadmin delete shadows*'
17             - '*wmic SHADOWCOPY DELETE*'
18     condition: selection
19  fields:
20     - CommandLine
21     - ParentCommandLine
22  falsepositives:
23     - Administrative scripts - e.g. to prepare image for golden image creation
24  level: critical
```

```
1 title: Data Compressed
2 id: 6f3e2987-db24-4c78-a860-b4f4095a7095
3 status: experimental
4 description: An adversary may compress data (e.g., sensitive documents) that is collected prior to exfiltration in order to make
5   of data sent over the network
6 author: Timur Zinniatullin, oscd.community
7 date: 2019/10/21
8 modified: 2019/11/04
9 references:
10   - https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1002/T1002.yaml
11 logsource:
12   category: process_creation
13   product: windows
14 detection:
15   selection:
16     Image|endswith: '\rar.exe'
17     CommandLine|contains|all:
18       - ' a '
19       - '-r'
20   condition: selection
21 fields:
22   - Image
23   - CommandLine
24   - User
25   - LogonGuid
26   - Hashes
27   - ParentProcessGuid
28   - ParentCommandLine
29 falsepositives:
30   - highly likely if rar is default archiver in the monitored environment
31 level: low
32 tags:
33   - attack.exfiltration
34   - attack.t1002
```

```
1 title: Net.exe Execution
2 id: 183e7ea8-ac4b-4c23-9aec-b3dac4e401ac
3 status: experimental
4 description: Detects execution of Net.exe, whether suspicious or benign.
5 references:
6   - https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/
7 author: Michael Haag, Mark Woan (improvements)
8 tags:
9   - attack.s0039
10  - attack.lateral_movement
11  - attack.discovery
12 logsource:
13   category: process_creation
14   product: windows
15 detection:
16   selection:
17     Image:
18       - '*\net.exe'
19       - '*\net1.exe'
20     CommandLine:
21       - '* group*'
22       - '* localgroup*'
23       - '* user*'
24       - '* view*'
25       - '* share'
26       - '* accounts*'
27       - '* use*'
28       - '* stop *'
29   condition: selection
30 fields:
31   - CommandLine
32   - ParentCommandLine
33 falsepositives:
34   - Will need to be tuned. If using Splunk, I recommend | stats count by Computer,CommandLine following the search for easy h
35 level: low
```

```
1  title: CobaltStrike Process Injection
2  id: 6309645e-122d-4c5b-bb2b-22e4f9c2fa42
3  description: Detects a possible remote threat creation with certain characteristics which are typical for Cobalt Strike beacons
4  references:
5      - https://medium.com/@olafhartong/cobalt-strike-remote-threads-detection-206372d11d0f
6      - https://blog.cobaltstrike.com/2018/04/09/cobalt-strike-3-11-the-snake-that-eats-its-tail/
7  tags:
8      - attack.defense_evasion
9      - attack.t1055
10 status: experimental
11 author: Olaf Hartong, Florian Roth, Aleksey Potapov, oscd.community
12 date: 2018/11/30
13 modified: 2019/11/08
14 logsource:
15     product: windows
16     service: sysmon
17 detection:
18     selection:
19         EventID: 8
20         TargetProcessAddress|endswith:
21             - '0B80'
22             - '0C7C'
23             - '0C88'
24     condition: selection
25 falsepositives:
26     - unknown
27 level: high
28
```

```
10 tags:
11   - attack.persistence
12   - attack.t1100
13 level: critical
14 logsource:
15   product: windows
16   service: sysmon
17 detection:
18   selection_1:
19     EventID: 11
20   selection_2:
21     TargetFilename|contains: '\inetpub\wwwroot\'
22   selection_3:
23     TargetFilename|contains:
24       - '.asp'
25       - '.ashx'
26       - '.ph'
27   selection_4:
28     TargetFilename|contains:
29       - '\www\'
30       - '\htdocs\'
31       - '\html\'
32   selection_5:
33     TargetFilename|contains: '.ph'
34   selection_6:
35     - TargetFilename|contains|all:
36       - '\'
37       - '.jsp'
38     - TargetFilename|contains|all:
39       - '\cgi-bin\'
40       - '.pl'
41   condition: selection_1 and ( selection_2 and selection_3 ) or
42     selection_1 and ( selection_4 and selection_5 ) or
43     selection_1 and selection_6
44 falsepositives:
45   - Legitimate administrator or developer creating legitimate executable files in a web application folder
```

```
1  title: Turla Group Named Pipes
2  id: 739915e4-1e70-4778-8b8a-17db02f66db1
3  status: experimental
4  description: Detects a named pipe used by Turla group samples
5  references:
6      - Internal Research
7  date: 2017/11/06
8  tags:
9      - attack.g0010
10 author: Markus Neis
11 logsource:
12     product: windows
13     service: sysmon
14     definition: 'Note that you have to configure logging for PipeEvents in Symson config'
15 detection:
16     selection:
17         EventID:
18             - 17
19             - 18
20         PipeName:
21             - '\atctl' # https://www.virustotal.com/#/file/a4ddb2664a6c87a1d3c5da5a5a32a5df9a0b0c8f2e951811bd1ec1d44d42ccf1/det
22             - '\userpipe' # ruag apt case
23             - '\iehelper' # ruag apt case
24             - '\sdlrpc' # project cobra https://www.gdatasoftware.com/blog/2015/01/23926-analysis-of-project-cobra
25             - '\comnap' # https://www.gdatasoftware.com/blog/2015/01/23926-analysis-of-project-cobra
26             # - '\rpc' # may cause too many false positives : http://kb.palisade.com/index.php?pg=kb.page&id=483
27     condition: selection
28 falsepositives:
29     - Unkown
30 level: critical
31
```

Summary

- Use ATT&CK to identify common behaviors, instead of just tools
- Explore the mind of the attacker
- Understand your data and visibility
- Express detection logic for *your* platform
- Continuously create, test, and refine analytics
- Choose your preferred language, and start coding!

First focus on suspicious and/or anomalous activity (EQL is great for that)

Then, focus on how threat actors are abusing the system (Sigma is great for that)

COMUNIDAD Y CONFIANZA,
BASES DE NUESTRA CIBERSEGURIDAD

#XIIIJORNADASCCNCERT

OC.CCN.CNI.ES

WWW.CCN.CNI.ES

WWW.CCN-CERT.CNI.ES

**XIII
JORNADAS
STIC
CCN-CERT**

ccn-cert
centro criptológico nacional