MB SECURE

Sirius Security

DETT&CT: MAPPING YOUR BLUE TEAM TO ATT&CK®

WHOAREWE

RUBEN BOUMAN

- Freelance Cyber Defense Expert
- Co-owner Sirius Security
- Roots in development
- Nine years of experience in Info Security
- Co-developer of the DeTT&CT framework

@RubenB_2

MARCUS BAKKER

- Freelance Cyber Defense Expert
- Nine years of experience in Info Security
- Co-developer of the DeTT&CT framework
- Co-author of the TaHiTI Threat Hunting Methodology

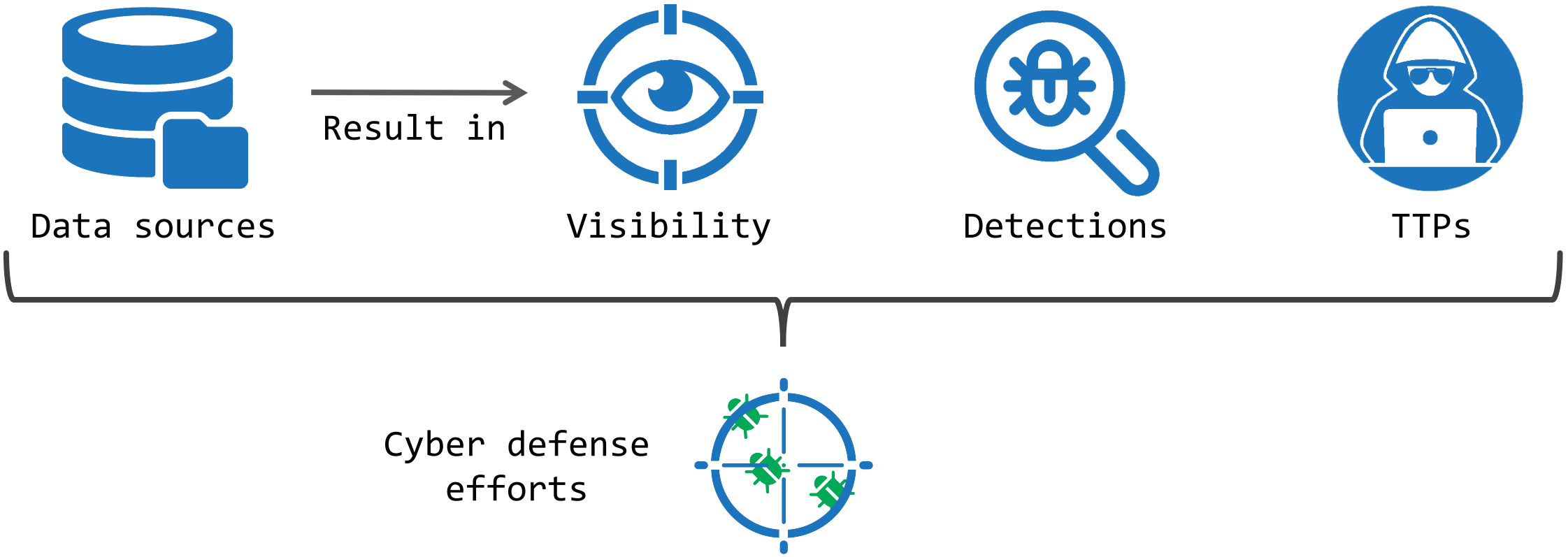
@Bakk3rM

AGENDA

- Introduction on DeTT&CT
- New features
- Roadmap

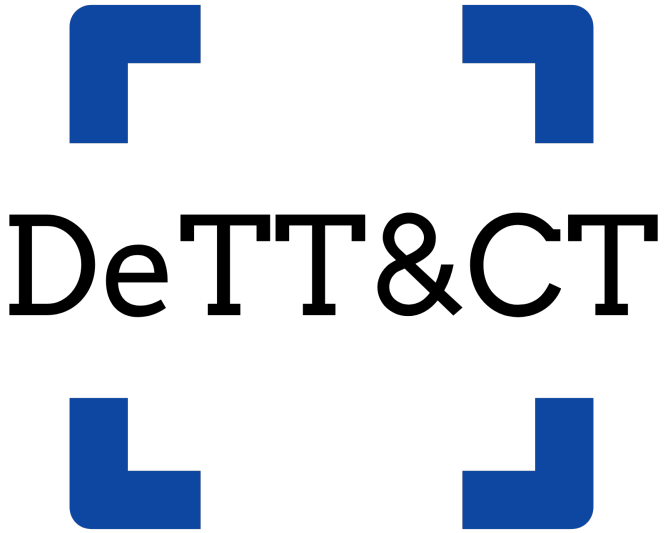
CHALLENGE: WHERE DO WE START OUR CYBER DEFENSE EFFORTS

- Intelligence-driven approach with a focus on TTPs



DETECT TACTICS, TECHNIQUES & COMBAT THREATS

- Framework to administrate, score and compare:
  - Data source quality
  - Visibility
  - Detections
  - Threat actor behaviours
- Result: where do you focus on
  - Which techniques?
  - Where to improve visibility?
- Scoring tables to guide you
- Administration = YAML files



`github.com/rabobank-cdc/DeTTECT`

DISCLAIMER

All shown data and visualisation regarding
data quality, visibility, detection and
threat actor groups are based on sample data.

DATA SOURCES: WHAT DO WE LOG

- Identify data sources

  Process injection

  ID: T1055

  Tactic: Defense Evasion, Privilege Escalation

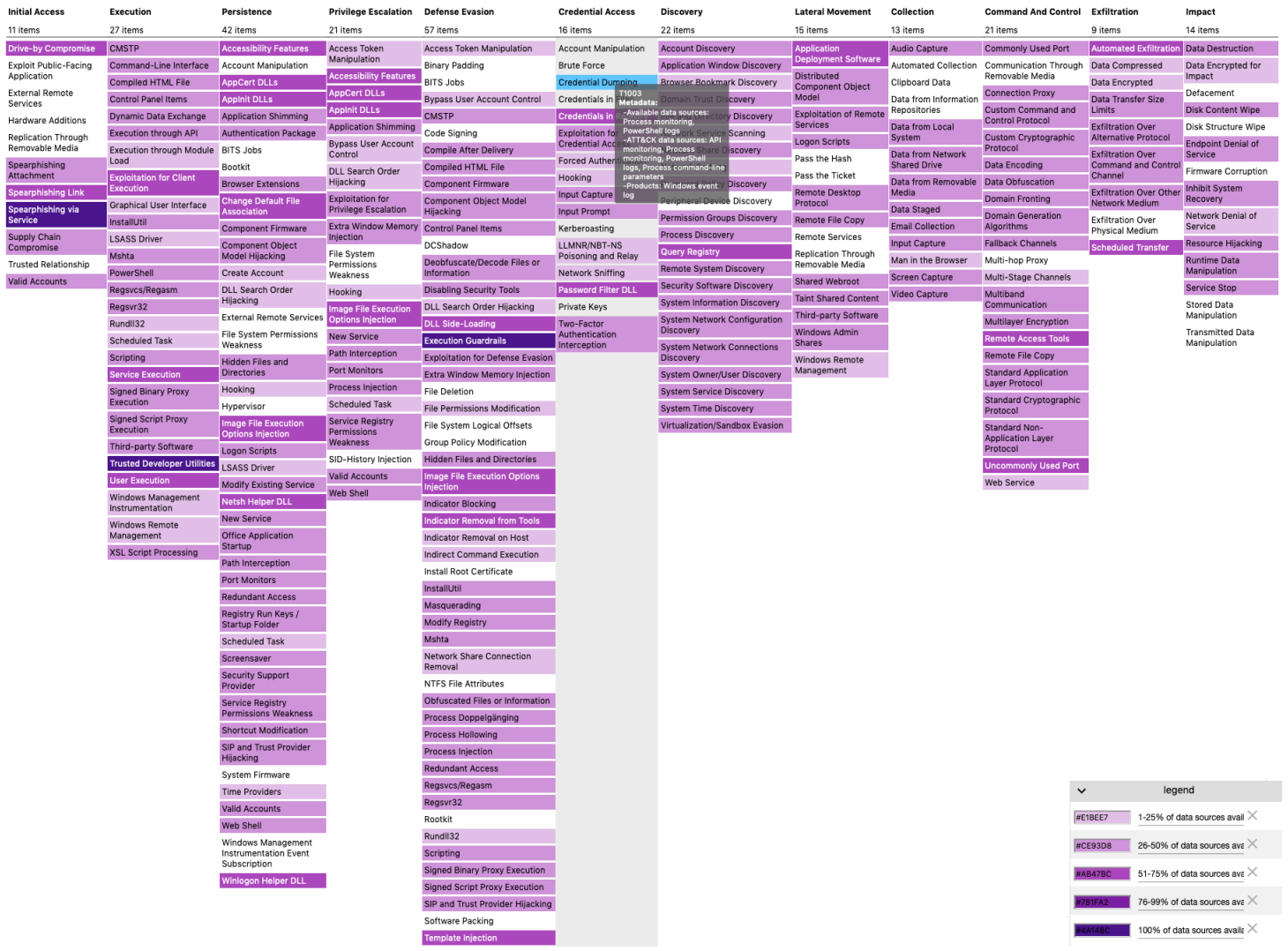
  Platform: Linux, macOS, Windows

  Permissions Required: User, Administrator, SYSTEM, root

  Effective Permissions: User, Administrator, SYSTEM, root

  Data Sources: API monitoring, Windows Registry, File monitoring, DLL monitoring, Process monitoring, Named Pipes

- Score data quality (DQ)
- Visualise in the ATT&CK Navigator
- Export to Excel



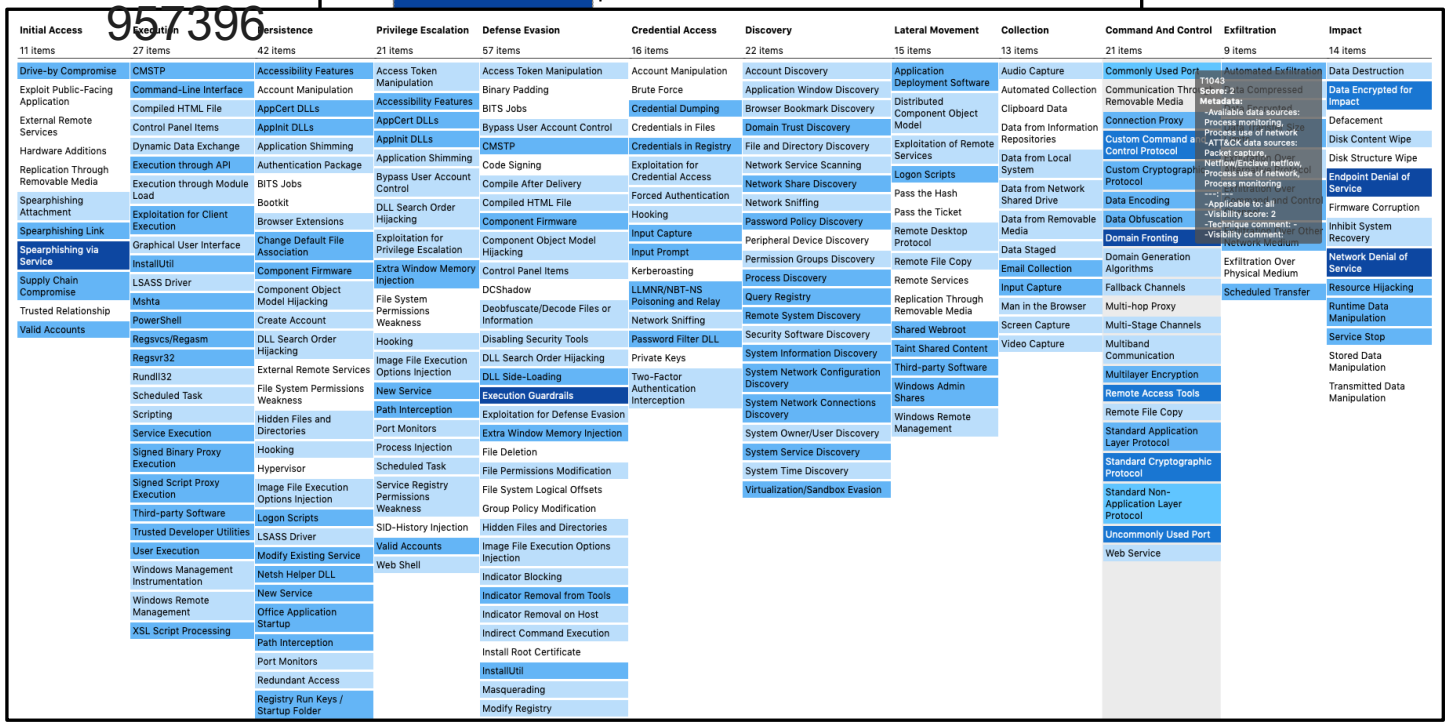
VISIBILITY: WHAT CAN WE SEE?

- Manual score visibility
  - Allows you to be exact
  - One source is more important than the other
  - Minimal set of data sources to have useful visibility
- Administrate in YAML file
- Visualise in the ATT&CK Navigator
- Export to Excel

Visibility scores

Score	Score name	Description
0	None	No visibility at all.
1	Minimal	Sufficient data sources with sufficient quality available to be able to see one aspect of the technique's procedures.
2	Medium	Sufficient data sources with sufficient quality available to be able to see more aspects of the technique's procedures compared to "1/Minimal".
3	Good	Sufficient data sources with sufficient quality available to be able to see almost all known aspects of the technique's procedures.
4	Excellent	All data sources and required data quality necessary to be able to see all known aspects of the technique's procedures are available.

957396

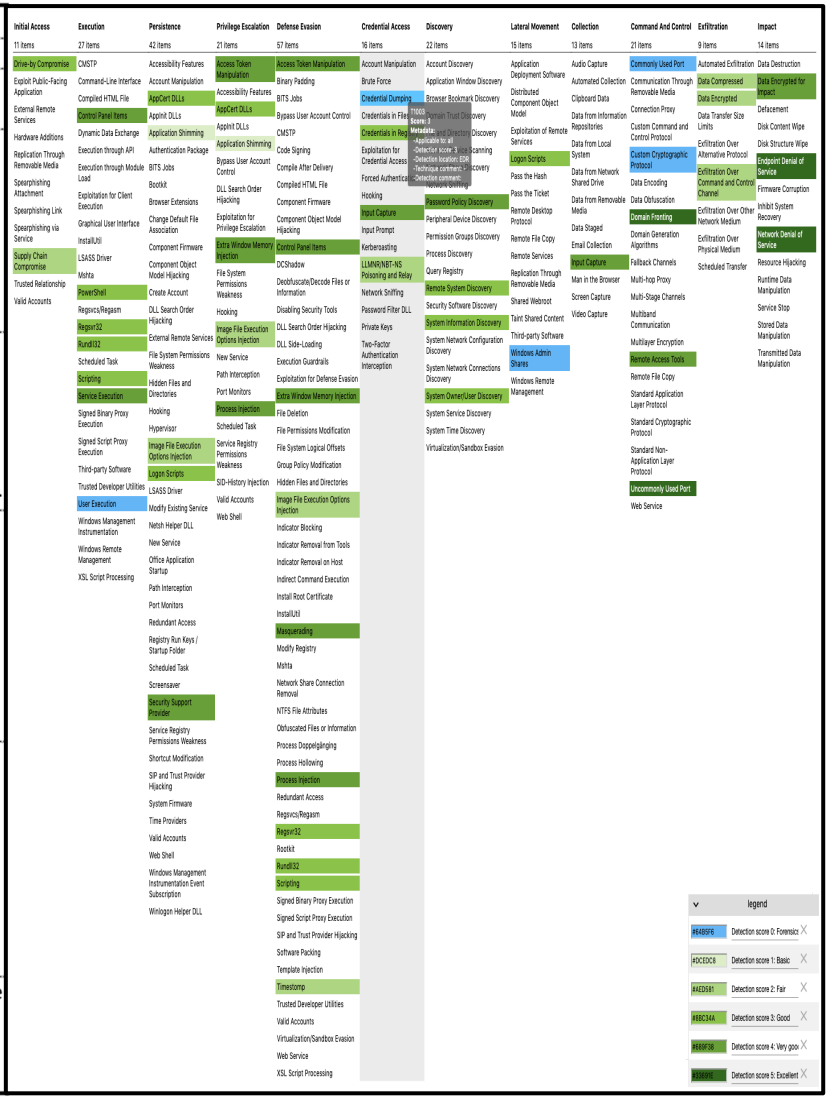


DETECTION: WHAT IS OUR DETECTION COVERAGE?

- Manual score detection
- Administrated in the same YAML file as visibility
- Visualise in the ATT&CK Navigator
- Export to Excel

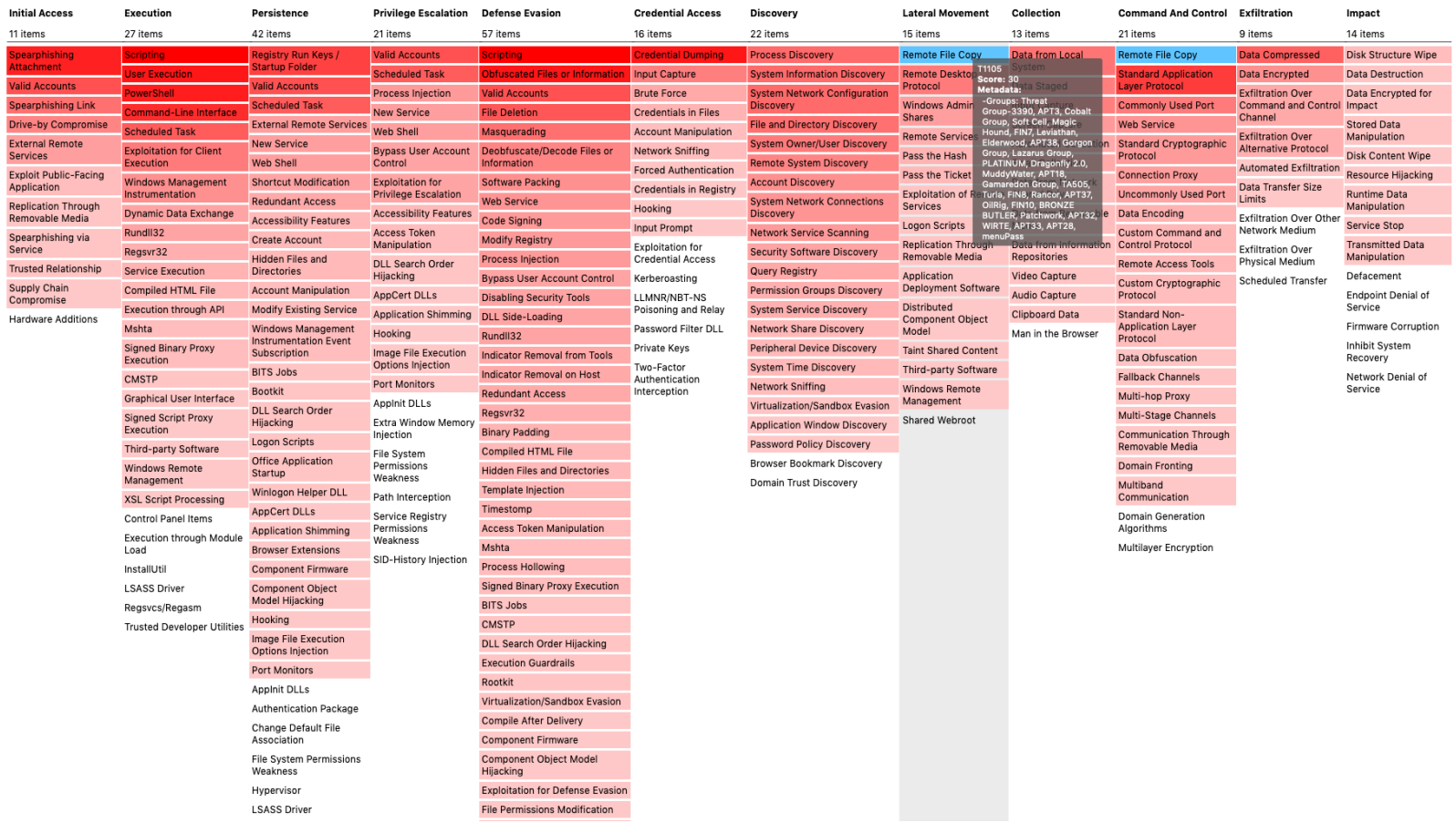
Detection scores

Score	Score name	Description
-1	None	No detection.
0	Forensics / context	No detection, but the technique is being logged for forensic purposes and can be used to provide context.
1	Basic	Detection is in place using a basic signature to detect a specific part(s) of the technique's procedures. Therefore, only a very small number of aspects of the technique are covered. Hence number of false negatives is high and possible (but not necessarily) a high false positive rate. Detection is possibly not real time.
2	Fair	The detection no longer only relies on a basic signature but makes use of a (correlation) rule to cover more aspects of the technique's procedures. Therefore, the number of false negatives is lower compared to "1/Poor" but may still be significant. False positives may still be present. Detection is possibly not real time.
3	Good	Effective in detecting malicious use of the technique by making use of more complex analytics. Many known aspects of the technique's procedures are covered. Bypassing detection by means of evasion and obfuscation could be possible. False negatives are present. False positives may still be present but are easy to recognize and can possibly be filtered out. Detection is real time.
4	Very good	Very effective in detecting malicious use of the technique in real time by covering almost all known aspects of the technique's procedures. Bypassing detection by means of evasion and obfuscation methods is harder compared to level "3/good". The number of false negatives is low but could be present. False positives may still be present but are easy to recognize and can possibly be filtered out.
5	Excellent	Same level of detection as level "4/very good" with one exception: all known aspects of the technique's procedures are covered. Therefore, the number of false negatives is lower compared to level "4/very good".



GROUPS: WHAT ARE ATTACKERS DOING?

- Generate heat maps
  - Threat actor group data from ATT&CK
  - Own intel stored in a group YAML file
  - Threat actor data from third parties *1
- Compare threat actors

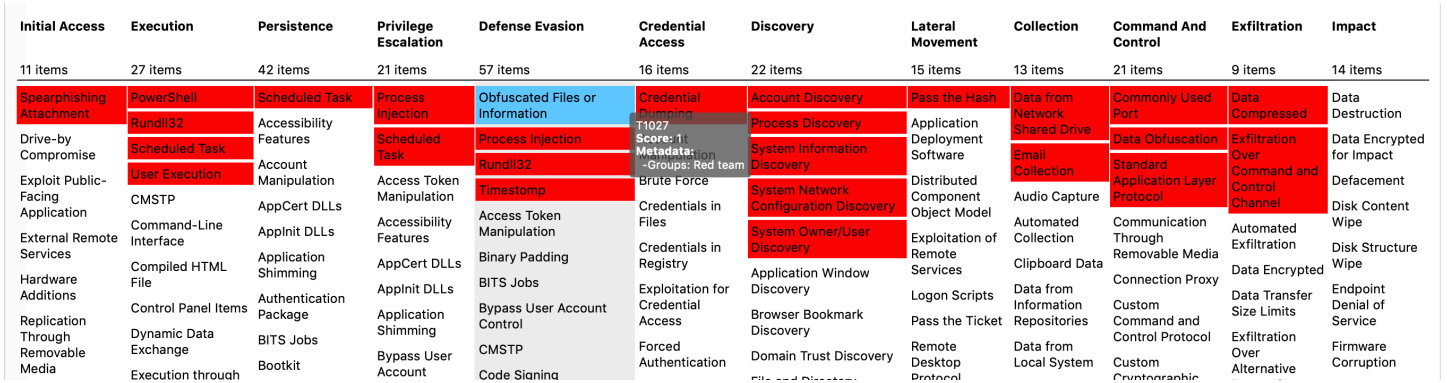


*1 https://github.com/rabobank-cdc/DeTTECT/tree/master/threat-actor-data

GROUPS: WHAT ARE ATTACKERS DOING?

- Generate heat maps
  - Threat actor group data from ATT&CK
  - Own intel stored in a group YAML file
  - Threat actor data from third parties *1
- Compare threat actors

```yaml
%YAML 1.2
---
version: 1.0
file_type: group-administration
groups:
  -
    group_name:      Red team
    campaign:        Scenario 1
    technique_id:    [T1086, T1053, T1193, T1204, T1003, T1055,
                      T1027, T1085, T1099, T1082, T1016, T1033,
                      T1087, T1075, T1057, T1039, T1041, T1071,
                      T1043, T1001, T1114, T1002]
    software_id:     [S0002] # Mimikatz
    enabled:         True
```

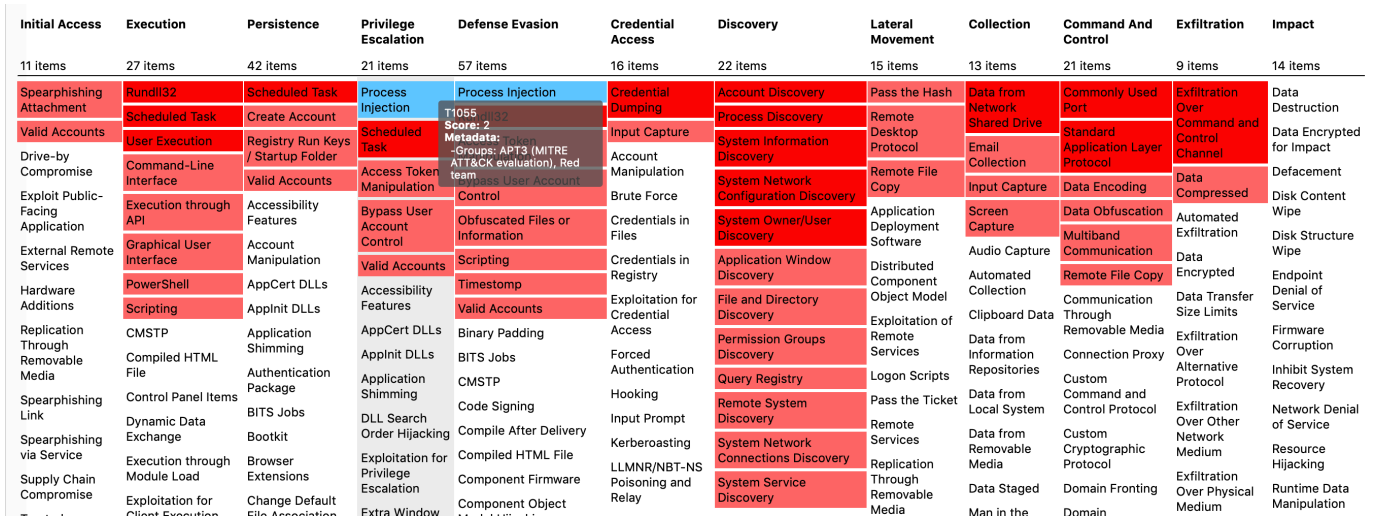


*1 https://github.com/rabobank-cdc/DeTTECT/tree/master/threat-actor-data

GROUPS: WHAT ARE ATTACKERS DOING?

- Generate heat maps
  - Threat actor group data from ATT&CK
  - Own intel stored in a group YAML file
  - Threat actor data from third parties *1
- Compare threat actors

```yaml
groups:
  -
    group_name:     Red team
    campaign:       Scenario 1
    technique_id:   [T1086, T1053, T1193, T1204, T1003, T1055,
                     T1027, T1085, T1099, T1082, T1016, T1033,
                     T1087, T1075, T1057, T1039, T1041, T1071,
                     T1043, T1001, T1114, T1002]
    software_id:    [S0002]
    enabled:        True
  -
    group_name:     APT3 (MITRE ATT&CK evaluation)
    campaign:       First Scenario
    technique_id:   [T1204, T1064, T1085, T1060, T1043, T1071,
                     T1132, T1016, T1059, T1033, T1057, T1106,
                     T1007, T1082, T1069, T1087, T1012, T1088,
                     T1134, T1055, T1018, T1049, T1003, T1026,
                     T1076, T1136, T1061, T1105, T1053, T1083,
                     T1056, T1010, T1113, T1039, T1041, T1078]
    software_id:    [S0154]
    enabled:        True
```



*1 https://github.com/rabobank-cdc/DeTTECT/tree/master/threat-actor-data

PRIORITISE YOUR CYBER DEFENSE EFFORTS

- Intelligence-driven approach with a focus on TTPs

Legend

- The technique only present in the group
- We have some level of detection
- We have detection and used by the group

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	27 items	42 items	21 items	57 items	16 items	22 items	15 items	13 items	21 items	9 items	14 items
Drive-by Compromise	Control Panel Items	Security Support Provider	Access Token Manipulation	Access Token Manipulation	Input Capture	Password Policy Discovery	Logon Scripts	Input Capture	Domain Fronting	Data Compressed	Endpoint Denial of Service
Supply Chain Compromise	Service Execution	AppCert DLLs	Extra Window Memory Injection	Control Panel Items	Credential Dumping	Remote System Discovery	Pass the Hash	Data from Network Shared Drive	Uncommonly Used Port	Data Encrypted	Network Denial of Service
Spearphishing Attachment	PowerShell	Logon Scripts	Process Injection	Extra Window Memory Injection	Credentials in Registry	System Information Discovery	Application Deployment Software	Email Collection	Remote Access Tools	Exfiltration Over Command and Control Channel	Data Encrypted for Impact
Exploit Public-Facing Application	Regsvr32	Image File Execution Options Injection	AppCert DLLs	Masquerading	LLMNR/NBT-NS Poisoning and Relay	System Owner/User Discovery	Distributed Component Object Model	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
External Remote Services	Rundll32	Application Shimming	Image File Execution Options Injection	Process Injection	Account Manipulation	Account Discovery	Exploitation of Remote Services	Automated Collection	Data Obfuscation	Data Transfer Size Limits	Defacement
Hardware Additions	Scripting	Scheduled Task	Application Shimming	Regsvr32	Brute Force	Process Discovery	Pass the Ticket	Clipboard Data	Standard Application Layer Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Replication Through Removable Media	Scheduled Task	Accessibility Features	Scheduled Task	Rundll32	Credentials in Files	System Network Configuration Discovery	Remote Desktop Protocol	Data from Information Repositories	Communication Through Removable Media	Exfiltration Over Other Network Medium	Disk Structure Wipe
Spearphishing Link	User Execution	Account Manipulation	Accessibility Features	Scripting	Exploitation for Credential Access	Application Window Discovery	Remote File Copy	Data from Local System	Connection Proxy	Exfiltration Over Physical Medium	Firmware Corruption
Spearphishing via Service	CMSTP	AppInit DLLs	AppInit DLLs	Image File Execution Options Injection	Forced Authentication	Browser Bookmark Discovery	Remote Services	Data from Removable Media	Custom Command and Control Protocol		Inhibit System Recovery
	Command-Line Interface	Authentication Package		Timestomp	Hooking	Domain Trust Discovery	Replication Through	Data Staged			Resource
	Compiled HTML File			Obfuscated Files or Information		File and Directory					
	Dynamic Data Exchange			Binary Padding							
	Execution through API										

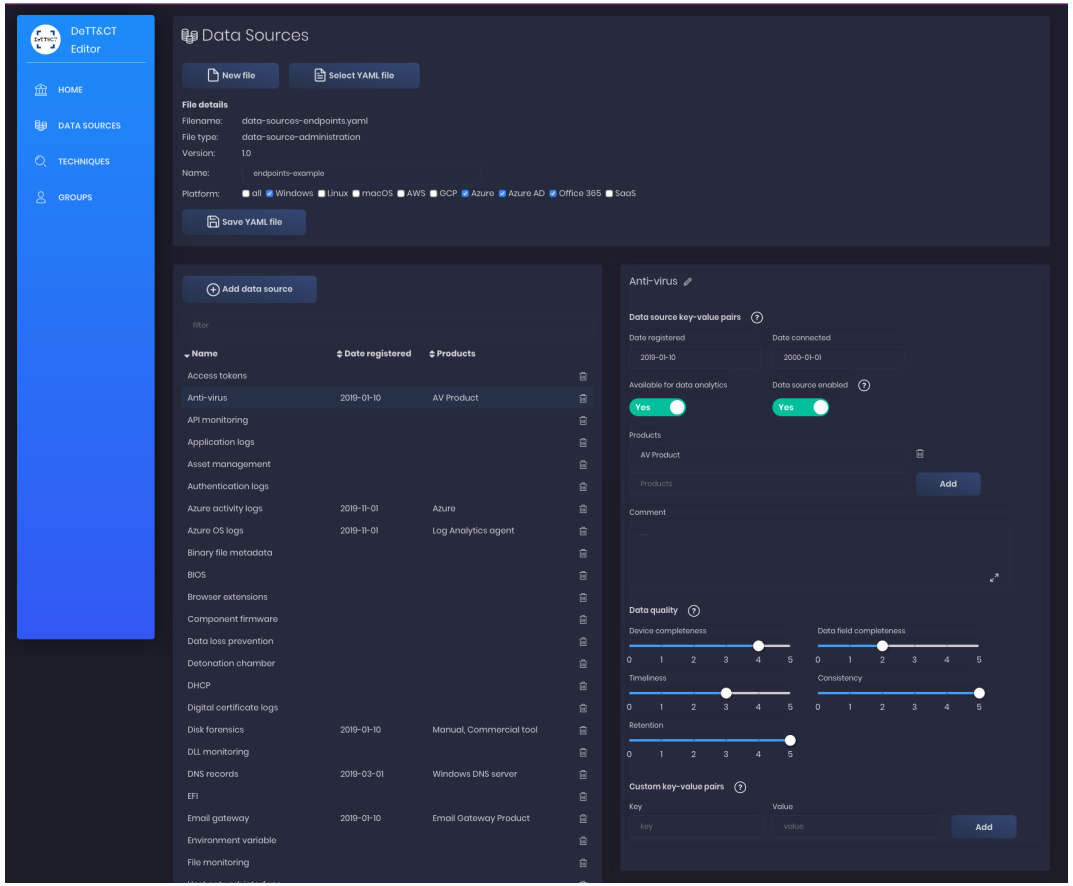
NEW FEATURES

- DeTT&CT Editor
- Group YAML files with a count per technique
- Added support for cloud platforms
- EQL integration
- Score logbook for keeping track of historic data
- Updating visibility scores based on a updated data source admin. file

DETT&CT EDITOR

- Editing big YAML files by hand is cumbersome and introduces errors
- Use it online: https://rabobank-cdc.github.io/dettect-editor
  - Its fully client side and developed in Vue JS
- Run it locally:

```
python dettect.py editor
```



GROUP YAML FILES WITH A COUNT

```yaml
%YAML 1.2
---
# Source: https://www.crowdstrike.com/resources/reports/2020-crowdstrike-global-threat-report/
version: 1.0
file_type: group-administration
platform:
  - Windows
  - Linux
  - macOS
groups:
  - group_name:      CrowdStrike Global Threat Report 2020
    campaign:
    technique_id:
      T1059 : 70
      T1078 : 65
      T1064 : 56
      T1016 : 54
      T1105 : 53
      T1003 : 52
      T1086 : 52
      T1033 : 51
      T1087 : 50
      T1076 : 48
      T1082 : 48
      T1049 : 46
      T1018 : 45
      T1057 : 43
      T1083 : 42
      T1089 : 40
      T1047 : 39
      T1043 : 37
      T1027 : 34
      T1100 : 33
      T1486 : 32
```

GROUP YAML FILES WITH A COUNT

Red Canary Threat Detection Report 2020

Top techniques 2019

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	34 items	62 items	32 items	69 items	21 items	23 items	18 items	13 items	22 items	9 items	16 items
Spearphishing Attachment	Scheduled Task	Scheduled Task	Process Injection	Process Injection	Credential Dumping	Domain Trust Discovery	Windows Admin Shares	Audio Capture	Remote File Copy	Automated Exfiltration	Account Access Removal
Drive-by Compromise	PowerShell	DLL Search Order Hijacking	Scheduled Task	Masquerading	Account Manipulation	Account Discovery	Remote File Copy	Automated Collection	Commonly Used Port	Data Compressed	Data Destruction
Exploit Public-Facing Application	Scripting	Accessibility Features	DLL Search Order Hijacking	Disabling Security Tools	Bash History	Application Window Discovery	AppleScript	Clipboard Data	Communication Through Removable Media	Data Encrypted	Data Encrypted for Impact
External Remote Services	Service Execution	Local Job Scheduling	Accessibility Features	DLL Search Order Hijacking	Brute Force	Browser Bookmark Discovery	Application Deployment Software	Data from Information Repositories	Connection Proxy	Data Transfer Size Limits	Defacement
Hardware Additions	Windows Management Instrumentation	.bash_profile and .bashrc	Access Token Manipulation	Scripting	Credentials from Web Browsers	File and Directory Discovery	Component Object Model and Distributed COM	Data from Local System	Custom Command and Control Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Replication Through Removable Media	Rundll32	Account Manipulation	AppCert DLLs	Rundll32	Credentials in Files	Network Service Scanning	Exploitation of Remote Services	Data from Network Shared Drive	Custom Cryptographic Protocol	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Local Job Scheduling	AppCert DLLs	AppInit DLLs	Deobfuscate/Decode Files or Information	Credentials in Registry	Network Share Discovery	Internal Spearphishing	Data from Removable Media	Data Encoding	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Mshta	AppInit DLLs	Application Shimming	Mshta	Exploitation for Credential Access	Network Sniffing	Logon Scripts	Data Staged	Data Obfuscation		Firmware Corruption
	AppleScript	Application Shimming	Bypass User Account Control	Process Hollowing	Forced Authentication	Password Policy Discovery		Email Collection	Domain Fronting		Inhibit System Recovery
	CMSTP	Authentication Package		Access Token Manipulation		Peripheral Device Discovery			Domain		Network Denial of Service
	Command-Line Interface			Binary Padding							
	Compiled HTML			BITS Jobs							

GROUP YAML FILES WITH A COUNT

CrowdStrike Global Threat Report 2020

Techniques OverWatch observed in targeted attacks in 2019

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	34 items	62 items	32 items	69 items	21 items	23 items	18 items	13 items	22 items	9 items	16 items
Valid Accounts	Command-Line Interface	Valid Accounts	Valid Accounts	Valid Accounts	Credential Dumping	System Network Configuration Discovery	Remote File Copy	Data from Local System	Remote File Copy	Data Compressed	Data Encrypted for Impact
Exploit Public-Facing Application	Scripting	Web Shell	Web Shell	Scripting	Brute Force	System Owner/User Discovery	Remote Desktop Protocol	Data Staged	Commonly Used Port	Exfiltration Over Command and Control Channel	Inhibit System Recovery
Spearphishing Attachment	PowerShell	Create Account	Scheduled Task	Disabling Security Tools	Credentials in Files	Account Discovery	Windows Admin Shares	Automated Collection	Uncommonly Used Port	Exfiltration Over Alternative Protocol	Service Stop
External Remote Services	Windows Management Instrumentation	Scheduled Task	New Service	Obfuscated Files or Information	Account Manipulation	System Information Discovery	Remote Services	Data from Network Shared Drive	Standard Application Layer Protocol	Automated Exfiltration	Resource Hijacking
Replication Through Removable Media	Rundll32	New Service	Process Injection	Masquerading	Bash History	System Network Connections Discovery	Replication Through Removable Media	Data from Information Repositories	Remote Access Tools	Data Encrypted	System Shutdown/Reboot
Spearphishing Link	Graphical User Interface	Account Manipulation	Accessibility Features	Modify Registry	Credentials in Registry	Remote System Discovery	Windows Remote Management	Input Capture	Connection Proxy	Data Transfer Size Limits	Runtime Data Manipulation
Trusted Relationship	Scheduled Task	Modify Existing Service	Image File Execution Options Injection	Rundll32	Input Capture	Process Discovery	Logon Scripts	Clipboard Data	Data Obfuscation	Exfiltration Over Other Network Medium	Account Access Removal
Hardware Additions	Service Execution	Registry Run Keys / Startup Folder	Bypass User Account Control	Indicator Removal on Host	Private Keys	File and Directory Discovery	Third-party Software	Email Collection	Standard Cryptographic Protocol	Exfiltration Over Physical Medium	Data Destruction
Drive-by Compromise	Regsvr32	Accessibility Features	Exploitation for Privilege Escalation	File Deletion	Exploitation for Credential Access	Network Service Scanning	AppleScript	Screen Capture	Web Service	Scheduled Transfer	Defacement
Spearphishing via Service	Mshta	Image File Execution Options Injection	Access Token Manipulation	Process Injection	Kerberoasting	Network Share Discovery	Application Deployment Software	Audio Capture	Custom Command and Control Protocol		Disk Content Wipe
Supply Chain Compromise	Exploitation for Client Execution	Redundant Access	DLL Search Order Hijacking	Deobfuscate/Decode Files or Information	Network Sniffing	Permission Groups Discovery	Component Object Model and Distributed COM	Data from Removable Media	Custom Cryptographic Protocol		Disk Structure Wipe
	Local Job Scheduling	External Remote Services	Setuid and Setgid	File and Directory Permissions Modification	Two-Factor Authentication Interception	Domain Trust Discovery	Exploitation of Remote Services	Man in the Browser	Data Encoding		Endpoint Denial of Service
	Windows Remote Management	BITS Jobs	Sudo	Connection Proxy	Credentials from Web Browsers	Query Registry	Internal Spearphishing	Video Capture	Standard Non-Application Layer Protocol		Firmware Corruption
	CMSTP	DLL Search Order Hijacking	AppInit DLLs	DLL Side-Loading	Forced Authentication	System Service Discovery	Pass the Hash		Multiband Communication		Network Denial of Service
	Control Panel Items	Hidden Files and Directories	Service Registry Permissions Weakness	Regsvr32	Hooking	System Time Discovery	Pass the Ticket		Multilayer Encryption		Stored Data Manipulation
	Execution through API	Local Job Scheduling	Startup Items	Image File Execution Options Injection	Input Prompt	Security Software Discovery	Shared Webroot		Communication Through Removable Media		Transmitted Data Manipulation
	InstallUtil	Setuid and Setgid	AppCert DLLs	Mshta	Keychain	Network Sniffing	SSH Hijacking		Domain Fronting		
	Third-party Software	Windows Management Instrumentation Event Subscription	Application Shimming	Redundant Access	LLMNR/NBT-NS Poisoning and Relay	Password Policy Discovery	Taint Shared		Domain Generation		
	Trusted Developer Utilities	AppInit DLLs		Bypass User Account Control	Password Filter DLL	Peripheral Device Discovery					
	User Execution			Clear Command History		Software Discovery					
	XSL Script Processing			Timestomp							
				Web Service							
				Access Token Manipulation							

CLOUD PLATFORMS

- Allow the use of cloud platforms

- Mapping data sources to platforms
  - https://github.com/rabobank-cdc/DeTTECT/wiki/Data-sources-per-platform

ENDGAME'S EQL INTEGRATION IN DETT&CT

- Use EQL to filter your YAML data in DeTT&CT

- Example use case: how did our detection coverage look like X time ago?

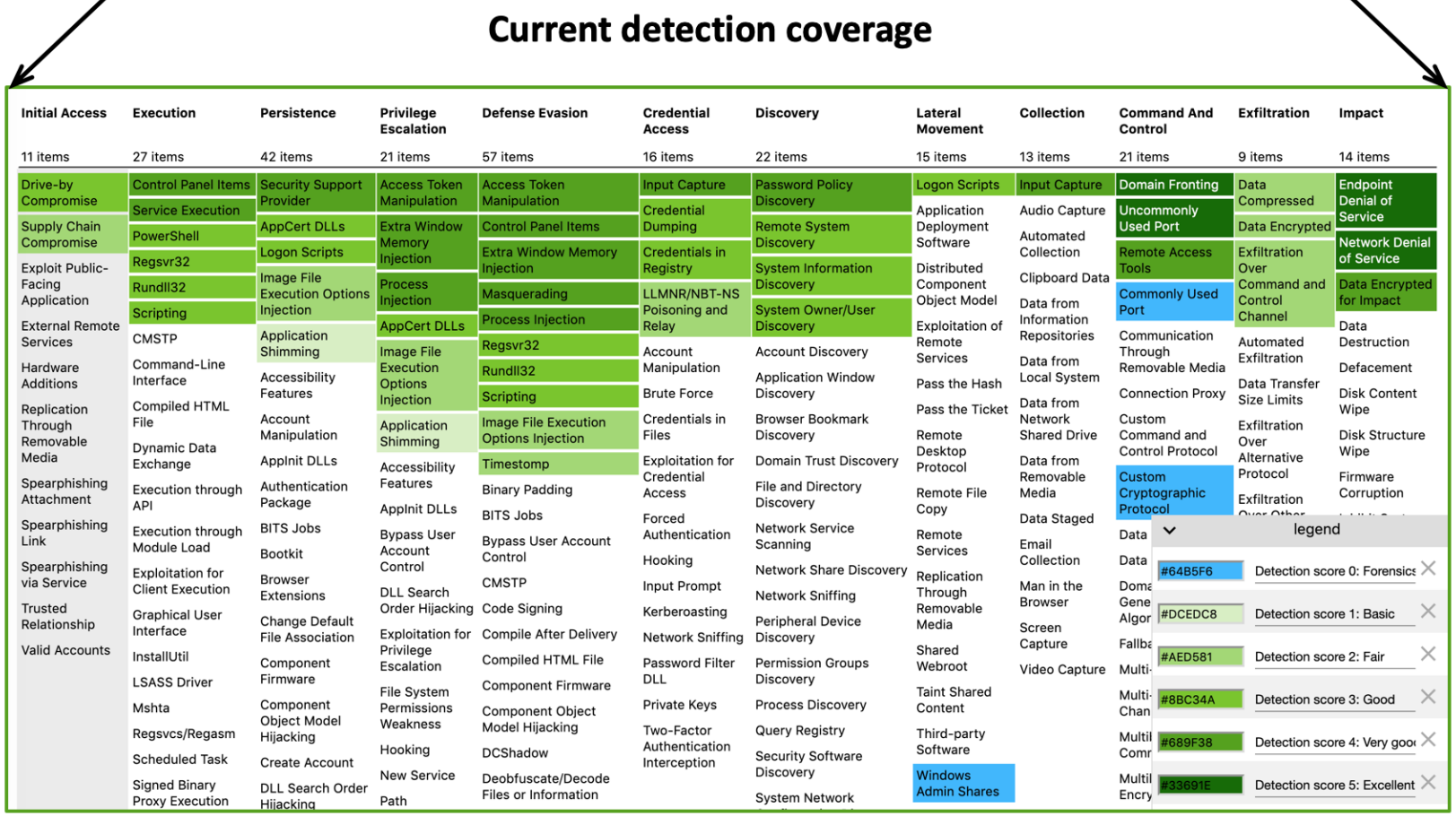
```
python dettect.py d -ft sample-data/techniques-admin.yaml --layer
--search-detection "techniques where detection.score_logbook.date <
'2017-11-01'" --all-scores
```

- Only include data sources which can be used in data analytics:

```
python dettect.py ds -fd sample-data/data-sources-endpoints.yaml -l
--search "data_sources where available_for_data_analytics = true"
```

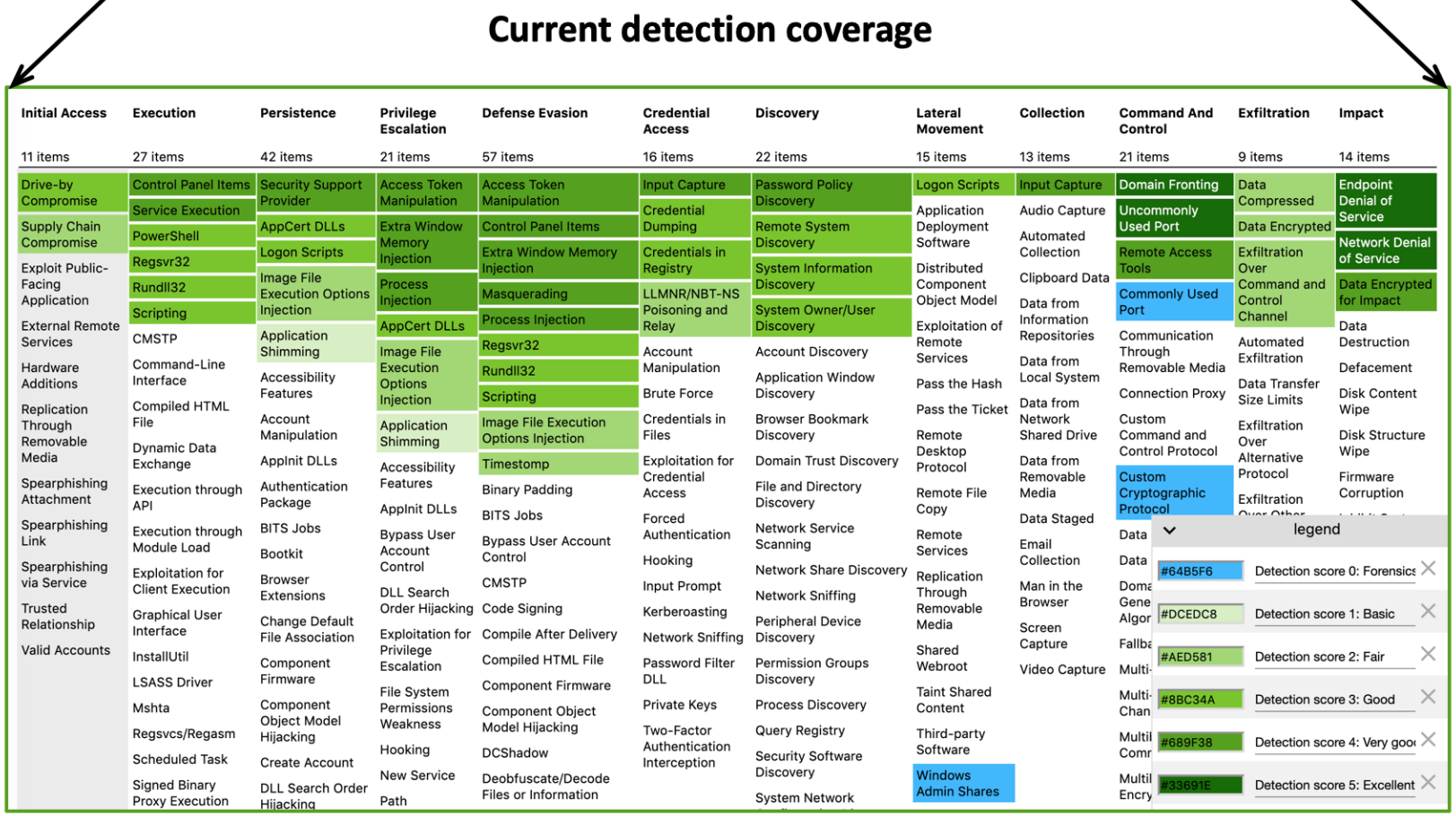
- Detection coverage over time

```
unknown:DeTTECT mb$ python dettect.py d -ft sample-data/techniques-administration-endpoints.yaml --layer
File written:   output/detection_example.json
```

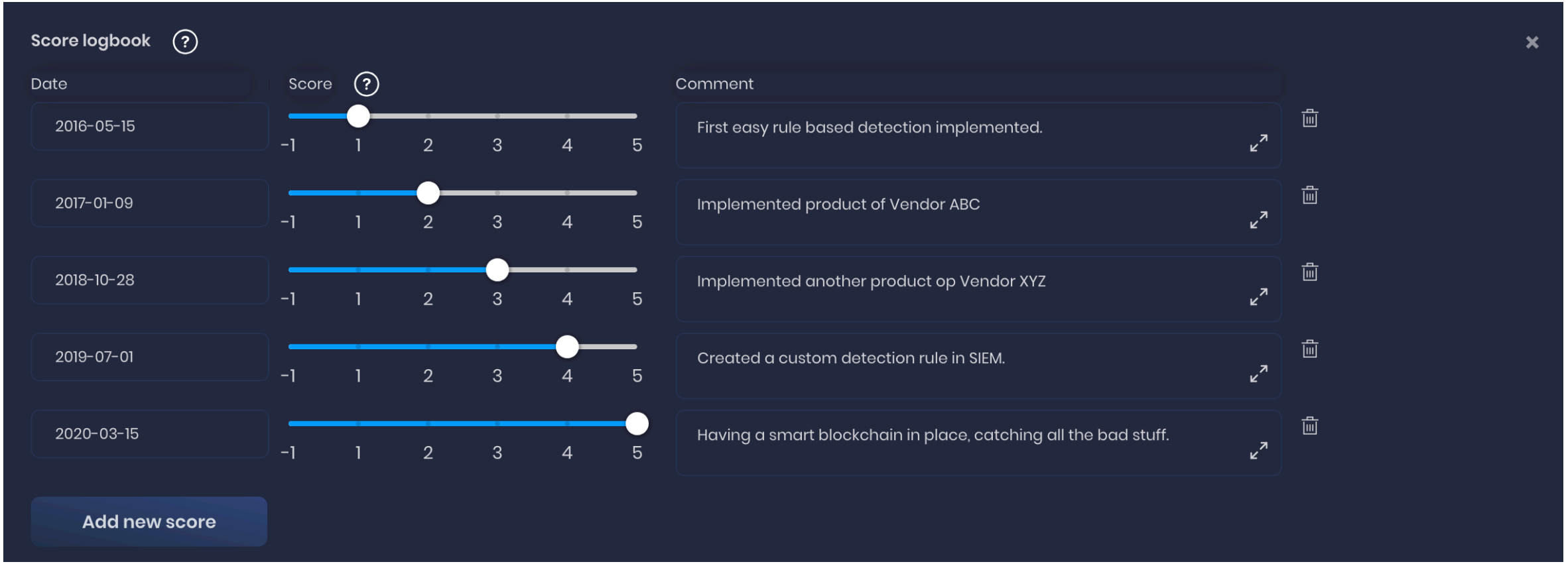


- Detection coverage over time

```
unknown:DeTTECT mb$ python dettect.py d -ft sample-data/techniques-administration-endpoints.yaml --layer
File written:   output/detection_example.json
```



SCORE LOGBOOK



UPDATING VISIBILITY SCORES

- Update visibility scores (automatically) based on changes in your data sources file
- Update when MITRE released a new ATT&CK version

UPDATING VISIBILITY SCORES

```
python dettect.py ds -fd ~/Downloads/data-sources-new.yaml -u -ft output/techniques-administration-
myexample-windows.yaml

Do you want to fill in the visibility comment for the updated scores?
 >>   y(yes) / n(no): y

 >>   Visibility comment for in the new 'score' object: Update Q2
```

UPDATING VISIBILITY SCORES

```
The following new technique IDs are added to the technique administration file with a visibility score
derived from the nr. of data sources:
 - T1192
 - T1194
 - T1195
 - T1534

A total of 1 visibility scores are eligible for an update.

For all most recent visibility score objects that are eligible for an update, the key-value pair 'auto-
generated' is set to 'true'.
This implies that these scores are auto-generated. How do you want to proceed?:
  1) Update all visibility scores that have changed.
  2) Decide per visibility score, that has changed if you want to update or not.
     Both the current and new visibility score will be printed.
  3) Cancel.

 >>   2
```

UPDATING VISIBILITY SCORES

```
------------------------------------------------------------------------
                                                    [updates remaining: 1]
Visibility object:
 - ATT&CK ID/name      T1189 / Credential Dumping
 - Applicable to:      all
 - Technique  comment:

OLD score object:
 - Date:               2020-05-15
 - Score:              0
 - Visibility comment:
 - Auto generated:     True
NEW score object:
 - Date:               2020-05-15
 - Score:              1
 - Visibility comment: Update Q2
 - Auto generated:     True

Update the score?
 >>   y(yes) / n(no): yes
```

UPDATING VISIBILITY SCORES

```
 - Updated a score in technique ID: T1189   (applicable to: all)

Written backup file:   output/techniques-administration-myexample-windows_backup_3.yaml

File written:   output/techniques-administration-myexample-windows.yaml
```

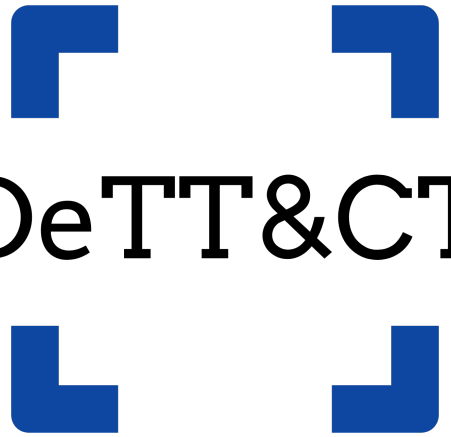
FUTURE PLANS

- Support for sub-techniques
  - Including conversion function
- DeTT&CT:
  - Score logbook for data sources
  - Applicable to field for data sources
- DeTT&CT Editor improvements:
  - Better searching
  - Edit YAML comment and exeptions section
  - Better navigation through data sources / techniques
  - Multiple UX improvements

The complete list: https://github.com/rabobank-cdc/DeTTECT/wiki/Future-dev

MB SECURE

Sirius Security



github.com/rabobank-cdc/DeTTECT

Questions?

@Bakk3rM

@RubenB_2