# Full Circle Detection

From Hunting to Actionable Detection

# Bio

Mathieu Saulnier

Tech Director Advanced Security at Syntax

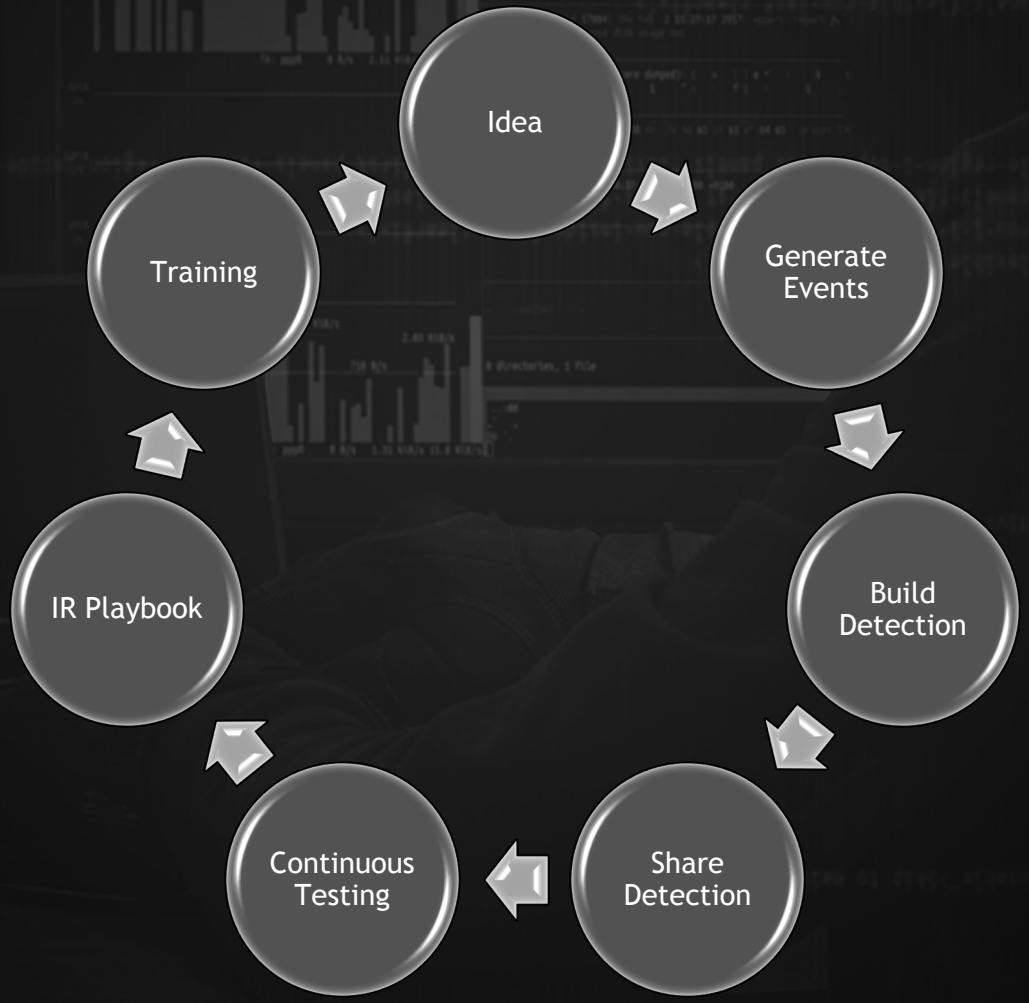
Core Mentor @BlueTeamVillage

Threat Hunting
Adversary Detection

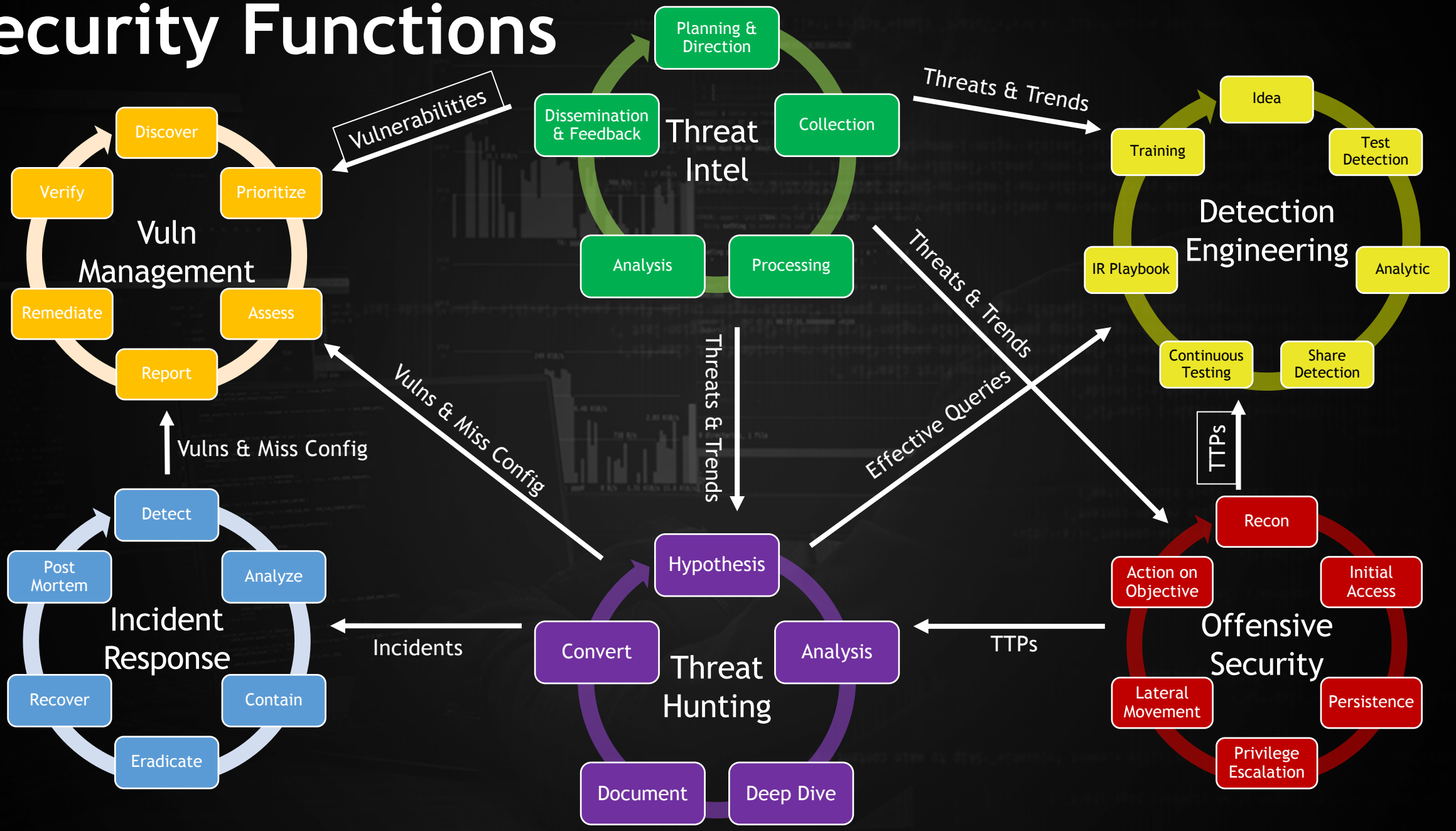
Talks at : Derbycon, BTV, NorthSec, SecTor & BSides

@ScoubiMtl

# The Process


Idea
Generate Events
Build Detection
Share Detection
Continuous Testing
IR Playbook
Training

# Security Functions



# The Idea / Hypothesis

There is a ton of places to get ideas for detection

- Twitter
- Mitre ATT&CK
- Slack
  - BloodHound
  - ThreatHunting
- Discord
  - TrustedSec
  - BHIS
  - BTV
- Infosec News Site
- Threat Intel Team

# The Idea / Hypothesis

MDSEC

**A Fresh Outlook on Mail Based Persistence**

https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/

```
Option Explicit

Private WithEvents olInboxItems As Items

Private Sub Application_Startup()
    Set olInboxItems = Session.GetDefaultFolder(olFolderInbox).Items
End Sub

Private Sub olInboxItems_ItemAdd(ByVal Item As Object)
    On Error Resume Next
    Dim olMailItem As MailItem
    If TypeOf Item Is MailItem Then
       If InStr(olMailItem.Subject, "MDSec") > 0 Then
            MsgBox "Hack The Planet"
            Shell "calc.exe"
            olMailItem.Delete
        End If
    End If
    Set Item = Nothing
```

# Detection

Monitoring of creation/modification events (Sysmon event ID 11) for the

%APPDATA%\Roaming\Microsoft\Outlook\VbaProject.OTM

file.

Monitoring for creation/changes events (Sysmon event ID 12) for the

HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Security

key and value Level.

# What We Did

**Group Hunt**

Book an hour learning meeting

Read the article

Look for the artifacts

**Planted Evidence**

Modify the registry

Edited VBAProject.OTM

Sent a mail

Hunt again


GOOD FOR HEART
GOOD FOR EDUCATION

# Generate the Event(s)

Did you find any events?

How can we easily generate events?

Atomic Red Team

github.com/redcanaryco/atomic-red-team

# How To Build an ART?

```yaml
attack_technique: T1137.002
display_name: 'Office Application Startup: Office Test'
atomic_tests:
- name: Office Apllication Startup Test Persistence
  auto_generated_guid: c3e35b58-fe1c-480b-b540-7600fb612563
  description: |
    Office Test Registry location exists that allows a user to specify an arbitrary DLL that will be executed every time an Office
    application is started. Key is used for debugging purposes. Not created by default & exist in HKCU & HKLM hives.
  supported_platforms:
  - windows
  input_arguments:
    thing_to_execute:
      description: Thing to Run
      type: Path
      default: C:\Path\AtomicRedTeam.dll
  executor:
    command: |
      reg add "HKEY_CURRENT_USER\Software\Microsoft\Office test\Special\Perf" /t REG_SZ /d "#{thing_to_execute}"
    cleanup_command: |
      reg delete "HKEY_CURRENT_USER\Software\Microsoft\Office test\Special\Perf"
    name: command_prompt
```

# How To Build an ART?

```yaml
attack_technique: T1137
display_name: Office Application Startup
atomic_tests:
- name: Office Application Startup - Outlook as a C2
  auto_generated_guid: bfe6ac15-c50b-4c4f-a186-0fc6b8ba936c
  description: |
    As outlined in MDSEC's Blog post
    https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
    it is possible to use Outlook Macro as a way to achieve persistance and execute
    arbitrary commands. This transform Outlook into a C2.
    Too achieve this two things must happened on the syste
    - The macro security registry value must be set to '4'
    - A file called VbaProject.OTM must be created in the Outlook Folder.
  supported_platforms:
  - windows
  executor:
    command: |
      reg add "HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Security\Level" /t REG_DWORD /d 1 /f
      mkdir  %APPDATA%\Microsoft\Outlook\ >nul 2>&1
      echo "Atomic Red Team TEST" > %APPDATA%\Microsoft\Outlook\VbaProject.OTM
    cleanup_command: |
      reg delete "HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Security\Level" /f
      del %APPDATA%\Microsoft\Outlook\VbaProject.OTM
    name: command_prompt
```

# How To Build an ART?

```yaml
attack_technique: T1137
display_name: Office Application Startup
atomic_tests:
- name: Office Application Startup - Outlook as a C2
  auto_generated_guid: bfe6ac15-c50b-4c4f-a186-0fc6b8ba936c
  description: |
    As outlined in MDSEC's Blog post
    https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
    it is possible to use Outlook Macro as a way to achieve persistance and execute
    arbitrary commands. This transform Outlook into a C2.
    Too achieve this two things must happened on the syste
    - The macro security registry value must be set to '4'
    - A file called VbaProject.OTM must be created in the Outlook Folder.
  supported_platforms:
  - windows
  executor:
    command: |
      reg add "HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Security\Level" /t REG_DWORD /d 1 /f
      mkdir  %APPDATA%\Microsoft\Outlook\ >nul 2>&1
      echo "Atomic Red Team TEST" > %APPDATA%\Microsoft\Outlook\VbaProject.OTM
    cleanup_command: |
      reg delete "HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Security\Level" /f
      del %APPDATA%\Microsoft\Outlook\VbaProject.OTM
    name: command_prompt
```

# How To Build an ART?

```yaml
attack_technique: T1137
display_name: Office Application Startup
atomic_tests:
- name: Office Application Startup - Outlook as a C2
  auto_generated_guid: bfe6ac15-c50b-4c4f-a186-0fc6b8ba936c
  description: |
    As outlined in MDSEC's Blog post
    https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
    it is possible to use Outlook Macro as a way to achieve persistance and execute
    arbitrary commands. This transform Outlook into a C2.
    Too achieve this two things must happened on the syste
    - The macro security registry value must be set to '4'
    - A file called VbaProject.OTM must be created in the Outlook Folder.
  supported_platforms:
  - windows
  executor:
    command: |
      reg add "HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Security\Level" /t REG_DWORD /d 1 /f
      mkdir  %APPDATA%\Microsoft\Outlook\ >nul 2>&1
      echo "Atomic Red Team TEST" > %APPDATA%\Microsoft\Outlook\VbaProject.OTM
    cleanup_command: |
      reg delete "HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Security\Level" /f
      del %APPDATA%\Microsoft\Outlook\VbaProject.OTM
    name: command_prompt
```

# Convert Hunt to Detection


- Refine/Optimize Query
- Select Output
  - Alert / Ticket
  - Report
  - Dashboard

# Share Detection with Community

- Sigma
  - github.com/SigmaHQ/sigma
- Share the Detection Logic
  - Known OS Exclusion
  - Not Org specific Exclusion

# Create Sigma Rule

## Existing Rule

```yaml
title: Office Application Startup - Office Test
id: 3d27f6dd-1c74-4687-b4fa-ca849d128d1c
status: experimental
description: Detects the addition of office test registry that allows a user to
    specify an arbitrary DLL that will be executed everytime an Office application
    is started
references:
    - https://attack.mitre.org/techniques/T1137/002/
author: omkar72
tags:
    - attack.persistence
    - attack.t1137.002
date: 2020/10/25
logsource:
    category: registry_event
    product: windows
detection:
    selection_registry:
        TargetObject:
            - 'HKEY_CURRENT_USER\Software\Microsoft\Office test\Special\Perf'
            - 'HKEY_LOCAL_MACHINE\Software\Microsoft\Office test\Special\Perf'
    condition: selection_registry
falsepositives:
    - Unlikely
level: medium
```

## Our Rule

```yaml
title: Outlook C2 Registry Key
id: e3b50fa5-3c3f-444e-937b-0a99d33731cd
status: experimental
description: Detects the modification of Outlook Security Setting to allow unprompted
    execution. Goes with win_outlook_c2_macro_creation.yml and is particularly
    interesting if both events occur near to each other.
references:
    - https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
author: '@ScoubiMtl'
tags:
    - attack.persistence
    - attack.command_and_control
    - attack.t1137
    - attack.t1008
    - attack.t1546
date: 2021/04/05
modified: 2021/09/13
logsource:
    category: registry_event
    product: windows
detection:
    selection_registry:
        TargetObject: 'HKCU\Software\Microsoft\Office\16.0\Outlook\Security\Level'
        Details|contains: '0x00000001'
    condition: selection_registry
falsepositives:
    - Unlikely
level: medium
```

```yaml
title: Outlook C2 Registry Key
id: e3b50fa5-3c3f-444e-937b-0a99d33731cd
status: experimental
description: Detects the modification of Outlook Security Setting to allow unprompted
    execution. Goes with win_outlook_c2_macro_creation.yml and is particularly
    interesting if both events occur near to each other.
references:
    - https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
author: '@ScoubiMtl'
tags:
    - attack.persistence
    - attack.command_and_control
    - attack.t1137
    - attack.t1008
    - attack.t1546
date: 2021/04/05
modified: 2021/09/13
logsource:
    category: registry_event
    product: windows
detection:
    selection_registry:
        TargetObject: 'HKCU\Software\Microsoft\Office\16.0\Outlook\Security\Level'
        Details|contains: '0x00000001'
    condition: selection_registry
falsepositives:
    - Unlikely
level: medium
```

```yaml
title: Outlook C2 Registry Key
id: e3b50fa5-3c3f-444e-937b-0a99d33731cd
status: experimental
description: Detects the modification of Outlook Security Setting to allow unprompted
    execution. Goes with win_outlook_c2_macro_creation.yml and is particularly
    interesting if both events occur near to each other.
references:
    - https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
author: '@ScoubiMtl'
tags:
    - attack.persistence
    - attack.command_and_control
    - attack.t1137
    - attack.t1008
    - attack.t1546
date: 2021/04/05
modified: 2021/09/13
logsource:
    category: registry_event
    product: windows
detection:
    selection_registry:
        TargetObject: 'HKCU\Software\Microsoft\Office\16.0\Outlook\Security\Level'
        Details|contains: '0x00000001'
    condition: selection_registry
falsepositives:
    - Unlikely
level: medium
```

```yaml
title: Outlook C2 Registry Key
id: e3b50fa5-3c3f-444e-937b-0a99d33731cd
status: experimental
description: Detects the modification of Outlook Security Setting to allow unprompted
    execution. Goes with win_outlook_c2_macro_creation.yml and is particularly
    interesting if both events occur near to each other.
references:
    - https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
author: '@ScoubiMtl'
tags:
    - attack.persistence
    - attack.command_and_control
    - attack.t1137
    - attack.t1008
    - attack.t1546
date: 2021/04/05
modified: 2021/09/13
logsource:
    category: registry_event
    product: windows
detection:
    selection_registry:
        TargetObject: 'HKCU\Software\Microsoft\Office\16.0\Outlook\Security\Level'
        Details|contains: '0x00000001'
    condition: selection_registry
falsepositives:
    - Unlikely
level: medium
```

# Create Sigma Rule

## Existing Rule

```yaml
title: Detection of SafetyKatz
id: e074832a-eada-4fd7-94a1-10642b130e16
status: experimental
description: Detects possible SafetyKatz Behaviour
references:
    - https://github.com/GhostPack/SafetyKatz
tags:
    - attack.credential_access
    - attack.t1003          # an old one
    - attack.t1003.001
author: Markus Neis
date: 2018/07/24
modified: 2020/08/23
logsource:
    category: file_event
    product: windows
detection:
    selection:
        TargetFilename|endswith: '\Temp\debug.bin'
    condition: selection
falsepositives:
    - Unknown
level: high
```

## Our Rule

```yaml
title: Outlook C2 Macro Creation
id: 8c31f563-f9a7-450c-bfa8-35f8f32f1f61
status: experimental
description: Detects the creation of a macro file for Outlook. Goes with
win_outlook_c2_registry_key. VbaProject.OTM is explicitly mentioned in T1137.
Particularly interesting if both events Registry & File Creation happens at
the same time.
references:
    - https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
author: '@ScoubiMtl'
tags:
    - attack.persistence
    - command_and_control
    - attack.t1137
    - attack.t1008
    - attack.t1546
date: 2021/04/05
logsource:
    category: file_event
    product: windows
detection:
    selection:
        TargetFilename|endswith: '\Microsoft\Outlook\VbaProject.OTM'
    condition: selection
falsepositives:
    - User genuinly creates a VB Macro for their email
level: medium
```

```yaml
title: Outlook C2 Macro Creation
id: 8c31f563-f9a7-450c-bfa8-35f8f32f1f61
status: experimental
description: Detects the creation of a macro file for Outlook. Goes with
win_outlook_c2_registry_key. VbaProject.OTM is explicitly mentioned in T1137.
Particularly interesting if both events Registry & File Creation happens at
the same time.
references:
    - https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
author: '@ScoubiMtl'
tags:
    - attack.persistence
    - command_and_control
    - attack.t1137
    - attack.t1008
    - attack.t1546
date: 2021/04/05
logsource:
    category: file_event
    product: windows
detection:
    selection:
        TargetFilename|endswith: '\Microsoft\Outlook\VbaProject.OTM'
    condition: selection
falsepositives:
    - User genuinly creates a VB Macro for their email
level: medium
```

```yaml
title: Outlook C2 Macro Creation
id: 8c31f563-f9a7-450c-bfa8-35f8f32f1f61
status: experimental
description: Detects the creation of a macro file for Outlook. Goes with
win_outlook_c2_registry_key. VbaProject.OTM is explicitly mentioned in T1137.
Particularly interesting if both events Registry & File Creation happens at
the same time.
references:
    - https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
author: '@ScoubiMtl'
tags:
    - attack.persistence
    - command_and_control
    - attack.t1137
    - attack.t1008
    - attack.t1546
date: 2021/04/05
logsource:
    category: file_event
    product: windows
detection:
    selection:
        TargetFilename|endswith: '\Microsoft\Outlook\VbaProject.OTM'
    condition: selection
falsepositives:
    - User genuinly creates a VB Macro for their email
level: medium
```

# Chris & Josh

https://drive.google.com/file/d/1nayvP3m8GD8cxV_nrk6459mHDV2xaqFB/view

## Sigma – Detection Expression

The detection expression is made up of two components – Search Identifiers & the Condition Expression. Search Identifiers are the fields and values the detection is targeting, while the Condition Expression ties those fields together and dictates how the detection tool will process each field in relation to the others.

```
18  detection:
19      rdp_outbound:
20          DestinationPort: 3389      # FIELD: MAP VALUE
21          Initiated: 'true'          # FIELD: MAP VALUE
22      filter:
23          Image|endswith:            # FIELD | VALUE MODIFIER
24              - '\mstsc.exe'         # LIST VALUE
25              - '\RTSApp.exe'        # LIST VALUE
26              - '\RTS2App.exe'       # LIST VALUE
27      condition: rdp_outbound and not filter
```

Search Identifiers (lines 19–26)

Condition Expression (line 27)

This rule contains a map that looks for logs that have *3389* in the **DestinationPort** field AND *true* in the **Initiated** field.

The condition tells the detection engine to match both items in the **rdp_outbound** map, but to exclude any log whose **Image** field contains any of the values in the **filter** list.

In summary, generate an alert for outbound network connections with a destination port of 3389 (RDP) if the connection was not generated by a known legitimate process.

### General Principles for the Detection Expression

*Adapted from https://github.com/SigmaHQ/sigma/wiki/Specification*

- YAML rules apply
- All values are case-insensitive strings with wildcards: "" and '?'
- Wildcards can be escaped with \, e.g. *. If a wildcard after a backslash should be searched, the backslash has to be escaped: *
- Empty value ''
- Null value is defined with 'null'



**LearnSigmaRules.com**

AND

# Pro Tip

uncoder.io

Sigma | ArcSight Rule | Azure Sentinel Query

```yaml
title: BEC - Outlook C2 Macro Creation
id: 8c31f563-f9a7-450c-bfa8-35f8f32f1f61
status: experimental
description: Detects the creation of a macro file for Outlook
references:
  - https://www.mdsec.co.uk/2020/11/a-fresh-outlook-on-mail-based-persistence/
author: '@ScoubiMtl'
tags:
  - attack.persistence
  - command_and_control
  - attack.t1008
  - attack.t1546
date: 2021/04/05
logsource:
  category: file_event
  product: windows
detection:
  selection:
    TargetFilename|endswith:
      '\Microsoft\Outlook\VbaProject.OTM'
  condition: selection
falsepositives:
  - User genuinly creates a VB Macro for their email
level: medium
```

658 / 5000

Elastic Query | QRadar | Splunk

Translate

```
file.path:*\Microsoft\Outlook\VbaProject.OTM
```

Suggest translation | Copy

Translating to: Elastic Query

# Detection Pipeline

- Make sure the whole chain is working
- Run your ART often
- Scheduled Tasks / Cron
- Docker
- Put the source system in Allow/Ignore list
- Send the ticket to a test queue
- blog.3coresec.com/2021/08/detection-as-code-dac-challenges.html

# Incident Response Playbook

Process for SOC Analysts

Business E-Mail Compromise (BEC)

# Two Open Frameworks

**RE&CT**

atc-project.github.io/atc-react/

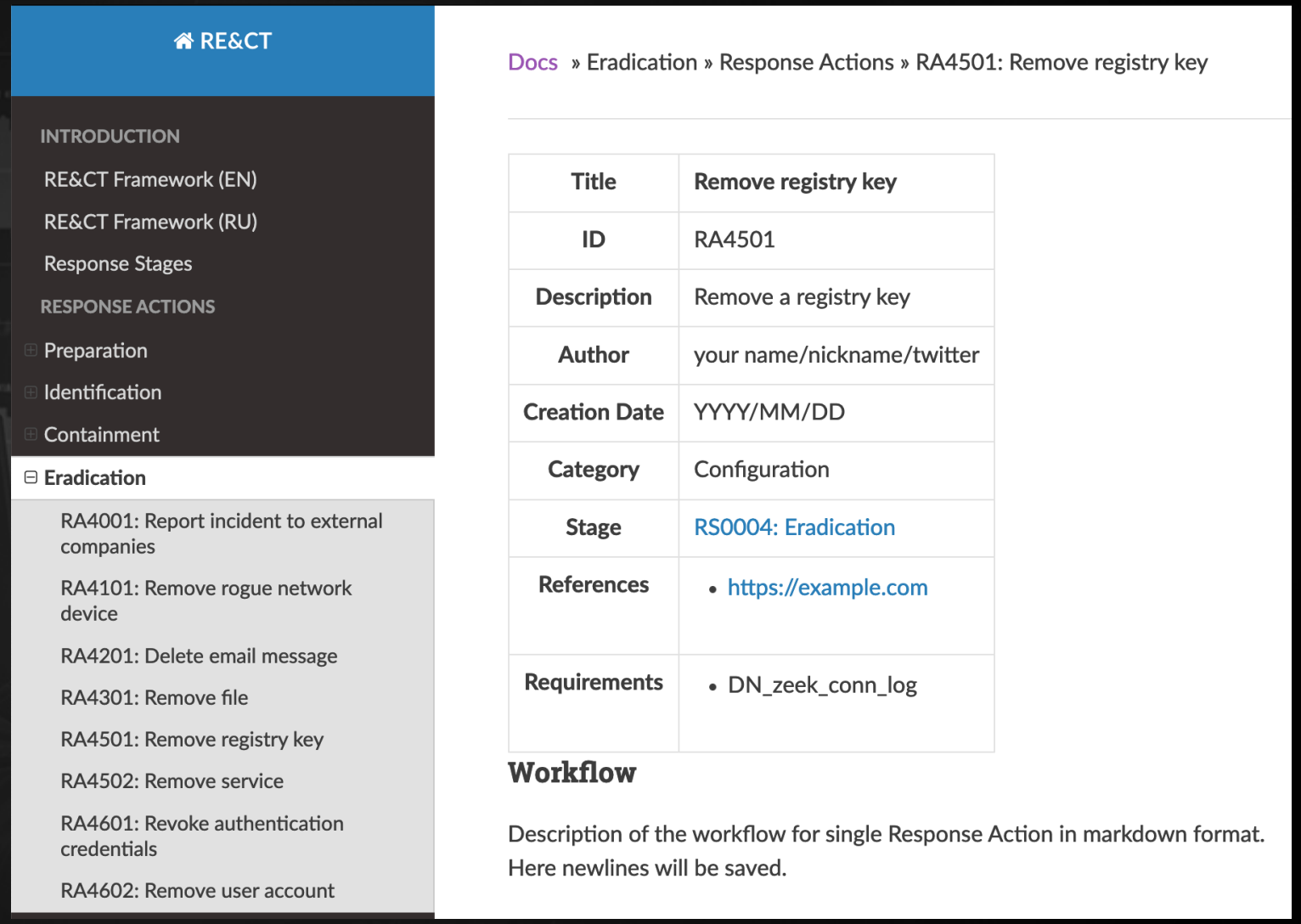
**Syntax**

gitlab.com/syntax-ir/playbooks/

Released in July @SANS DFIR

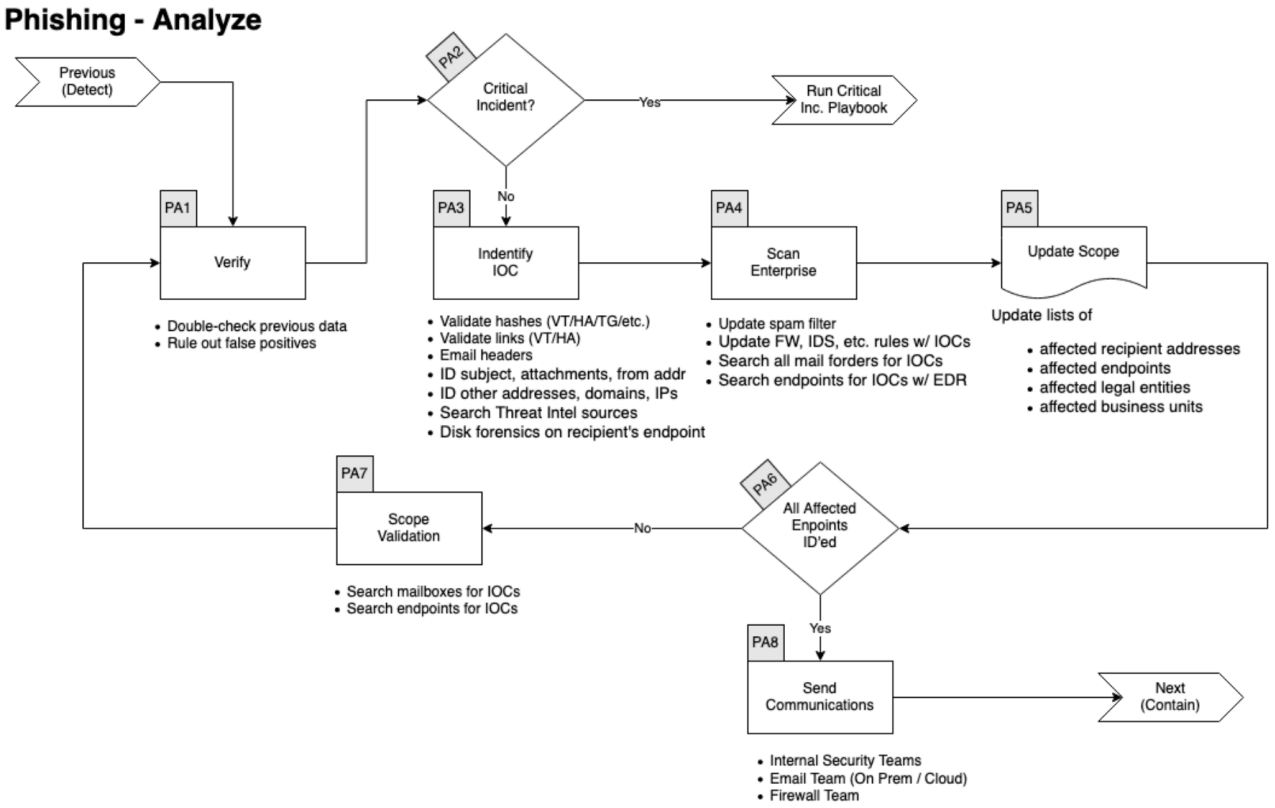
# RE&CT

@atc_project
Modular
Open Source
Framework
WIP



# Syntax Playbooks

Git
Markdown / Draw.io
Open & Free
Easy to modify




### Verify

▼ Expand/Colapse
In conjonction with a senior member of the ISOC

- Double check previous datsa
- Rule out False Positive

### Identify IOCs

▼ Expand/Colapse

- Validate hashes
  - VirusTotal
  - Hybrid Analysis
- Validate links
  - VirusTotal
  - Hybrid Analysis
  - URLScan

# Training

For current and future staff

Easy to Consume

- Video
- Powerpoint
- Wiki

# Bonus

- Automate!
  - SOAR
    - Shuffle
    - Walkoff
  - VirusTotal
  - Greynoise
  - Spamhaus

# Thank You!

Mathieu Saulnier / @ScoubiMtl