# Hunting Malicious Office Macros

SANS Threathunting Summit 2021

https://github.com/Antonlovesdnb/SANSTHS2021

# About Me

- Adversarial Collaboration Engineer - Lares
  - Purple / Blue Team
- Log, SIEM, query, detection fan
- Twitter: @Antonlovesdnb
- Email: aovrutsky@lares.com

# What is this about?

- Malicious Office Macro Baselining
- Threat hunting + Alerting Techniques
- Focus on:
  - Word and Excel
  - Endpoint Telemetry

# T1024.002 - User Execution: Malicious File

122 Groups total - ATT&CK
**59** Utilize “T1024.002”

# Why Macros?

## Case Summary

We assess with medium confidence that the initial threat vector for this intrusion was a password protected archive, delivered via malspam campaigns. The zip attachment would likely contain a Word or Excel document with macros, which upon execution, would start a Trickbot infection.

## Initial Access

Initial access for this intrusion was via a malicious attachment “order 06.21.doc”. The attachment was a Microsoft Word document that drops a malicious HTA file “textboxNameNamespace.hta”.

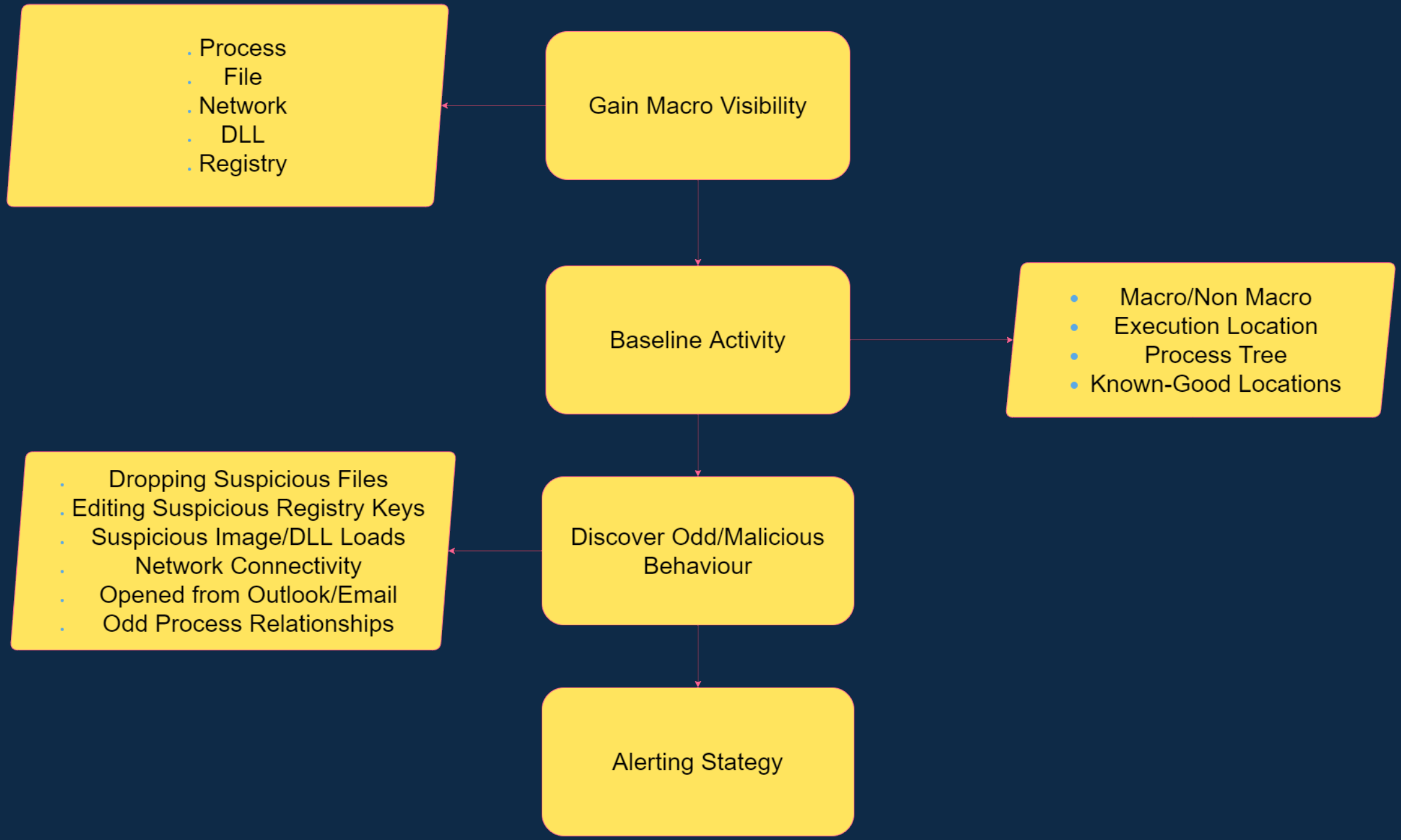
## Case Summary

We assess, with moderate confidence, the Trickbot DLL that we executed was originally delivered via a malicious Office document.

https://thedfirreport.com/

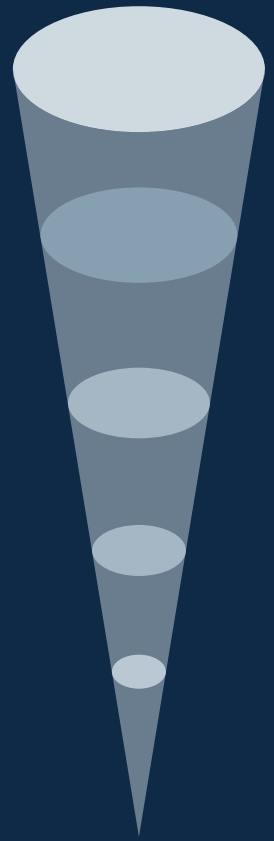
# “Our Users Use Macros”

# Breaking it Down


. Process
. File
. Network
. DLL
. Registry
Gain Macro Visibility
Baseline Activity
• Macro/Non Macro
• Execution Location
• Process Tree
• Known-Good Locations
. Dropping Suspicious Files
. Editing Suspicious Registry Keys
. Suspicious Image/DLL Loads
. Network Connectivity
. Opened from Outlook/Email
. Odd Process Relationships
Discover Odd/Malicious Behaviour
Alerting Stategy

# Baselining Activity

- **OfficeWatch.xml**
  - “Show me everything Word and Excel are doing”
- **OfficeShush.xml**
  - “Filter out the activity I saw above”
- **OfficeSus.xml**
  - “Bubble up suspicious activity that does not meet baseline”

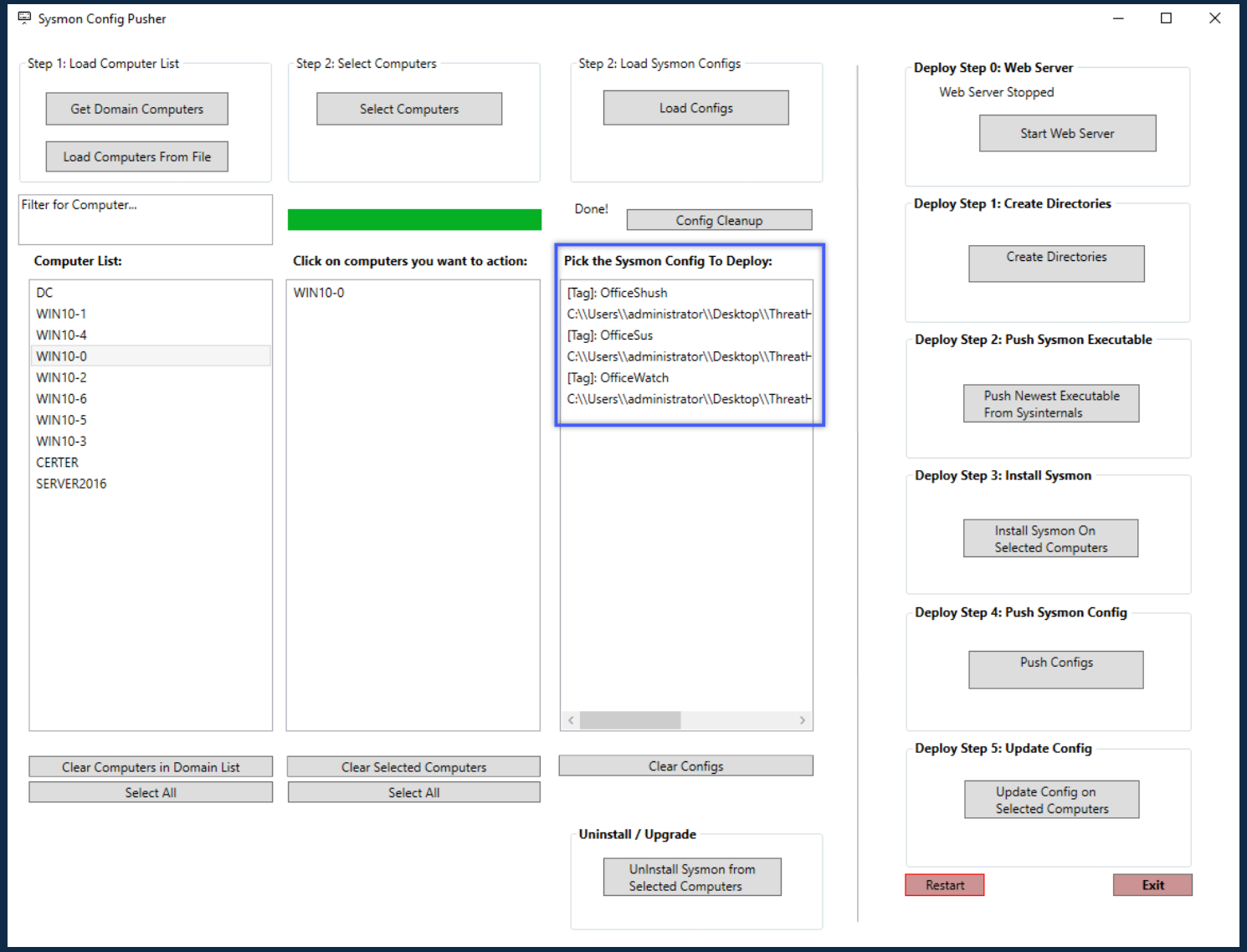


# Baselining Activity

- Use *OfficeWatch.xml* to see all Office Activity
- Use *OfficeShush.xml* to filter out the noisy events
- Add suspicious or abnormal activity to *OfficeSus.xml*
  - *Or* Add activity to main Sysmon configuration file / use data for hunting with own telemetry sources

# Shameless...

https://github.com/LaresLLC/SysmonConfigPusher




Sysmon Config Pusher makes flipping back and forth between Sysmon configuration files relatively easy

# The Fruits of Our Baseline Labor

Image loaded:
RuleName: -
UtcTime: 2021-09-04 16:41:42.216
ProcessGuid: {26d732db-a1c4-6133-b408-000000005300}
ProcessId: 7024
Image: C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE
ImageLoaded: C:\Program Files\Microsoft Office\root\vfs\ProgramFilesCommonX64\Microsoft Shared\VBA\VBA7.1\1033\VBE7INTL.DLL
FileVersion: 7.01.1091
Description: Visual Basic Environment International Resources
Product: Visual Basic Environment
Company: Microsoft Corporation
OriginalFileName: -
Hashes: MD5=CDA3EA478C604783B76964E88FD7030D

Registry value set:
RuleName: -
EventType: SetValue
UtcTime: 2021-09-04 16:41:41.744
ProcessGuid: {26d732db-a1c4-6133-b408-000000005300}
ProcessId: 7024
Image: C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE
TargetObject: HKU\S-1-5-21-1782144875-2244134600-1088407481-500\SOFTWARE\Microsoft\Office\16.0\Word\Security\Trusted Documents\TrustRecords\%USERPROFILE%/Desktop/Tests/Calc.doc
Details: Binary Data

Image loaded:
RuleName: -
UtcTime: 2021-09-04 16:41:42.258
ProcessGuid: {26d732db-a1c4-6133-b408-000000005300}
ProcessId: 7024
Image: C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE
ImageLoaded: C:\Windows\System32\wshom.ocx
FileVersion: 5.812.10240.16384
Description: Windows Script Host Runtime Library
Product: Microsoft ® Windows Script Host Runtime Library
Company: Microsoft Corporation
OriginalFileName: wshom.ocx

Process accessed:
RuleName: -
UtcTime: 2021-09-04 16:41:42.287
SourceProcessGUID: {26d732db-a1c4-6133-b408-000000005300}
SourceProcessId: 7024
SourceThreadId: 7080
SourceImage: C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE
TargetProcessGUID: {26d732db-a1c6-6133-b808-000000005300}
TargetProcessId: 9404
TargetImage: C:\Windows\SYSTEM32\calc.exe
GrantedAccess: 0x1FFFFF
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9e664|C:\Windows\System32\KERNELBASE.dll+8e73|C:\Windows\System32\KERNELBASE.dll+71a6|C:\Windows\System32\KERNEL32.DLL+1cbb4|C:\Program Files\Microsoft Office\Root\Office16\AppVIsvSubsystems64.dll+d9437|C:\Program Files\Microsoft Office\Root\Office16\AppVIsvSubsystems64.dll+d848f|C:\Program Files\Microsoft Office\Root\Office16\AppVIsvSubsystems64.dll+d8ef8|C:\Program Files\Microsoft Office\Root\Office16\AppVIsvSubsystems64.dll+d192e|C:\Program Files\Microsoft Office\Root\Office16\AppVIsvSubsystems64.dll+d24c7|C:\Windows\System32\wshom.ocx+c39d|C:\Windows\System32\wshom.ocx+c8a0|C:\Windows\System32\OLEAUT32.dll+1fd0f|C:\Windows\System32\OLEAUT32.dll+129b6|C:\Windows\System32\OLEAUT32.dll+fd65|C:\Windows\System32\wshom.ocx+e069|C:\Windows\System32\wshom.ocx+26ad|C:\Program Files\Common Files\Microsoft Shared\VBA\VBA7.1\VBE7.DLL+2d7020|C:\Program Files\Common Files\Microsoft Shared\VBA\VBA7.1\VBE7.DLL+38f6eb|C:\Program Files\Common Files\Microsoft Shared\VBA\VBA7.1\VBE7.DLL+390dca|C:\Program Files\Common Files\Microsoft Shared\VBA\VBA7.1\VBE7.DLL+383a44|C:\Windows\System32\OLEAUT32.dll+1fd0f|C:\Windows\System32\OLEAUT32.dll+129b6|C:\Program Files\Common Files\Microsoft Shared\VBA\VBA7.1\VBE7.DLL+1108ca|C:\Program Files\Common Files\Microsoft Shared\VBA\VBA7.1\VBE7.DLL+1d2975

# The Fruits of Our Baseline Labor

```
2021-09-05 10:39:55 "C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE" /n "C:\Users\administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\9BKPDZ49\Calc

2021-09-05 10:40:02 calc
2021-09-05 10:39:55 "C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE"  /Embedding

2021-09-05 10:39:35 "C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE" /h "\\dc\MyFileShare\Calc.doc" /o ""
2021-09-05 10:39:38 calc
```

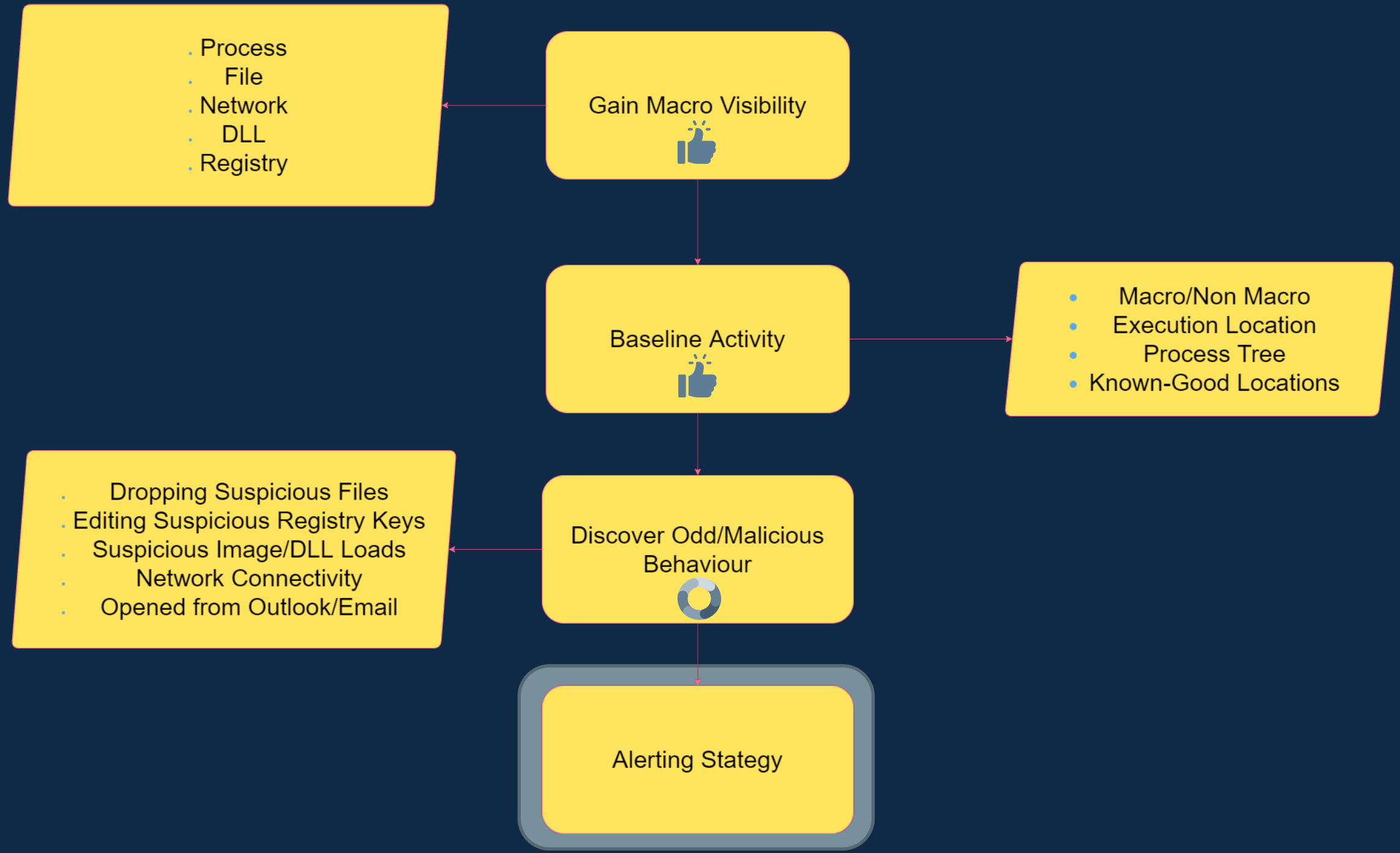
```
OUTLOOK.EXE (9036)
  |--- WINWORD.EXE (10028)
  ``  |--- calc.exe (9352)
  ``  |--- WINWORD.EXE (1084)

explorer.exe (1432)
  |--- WINWORD.EXE (232)
  ``  |--- calc.exe (7268)
```

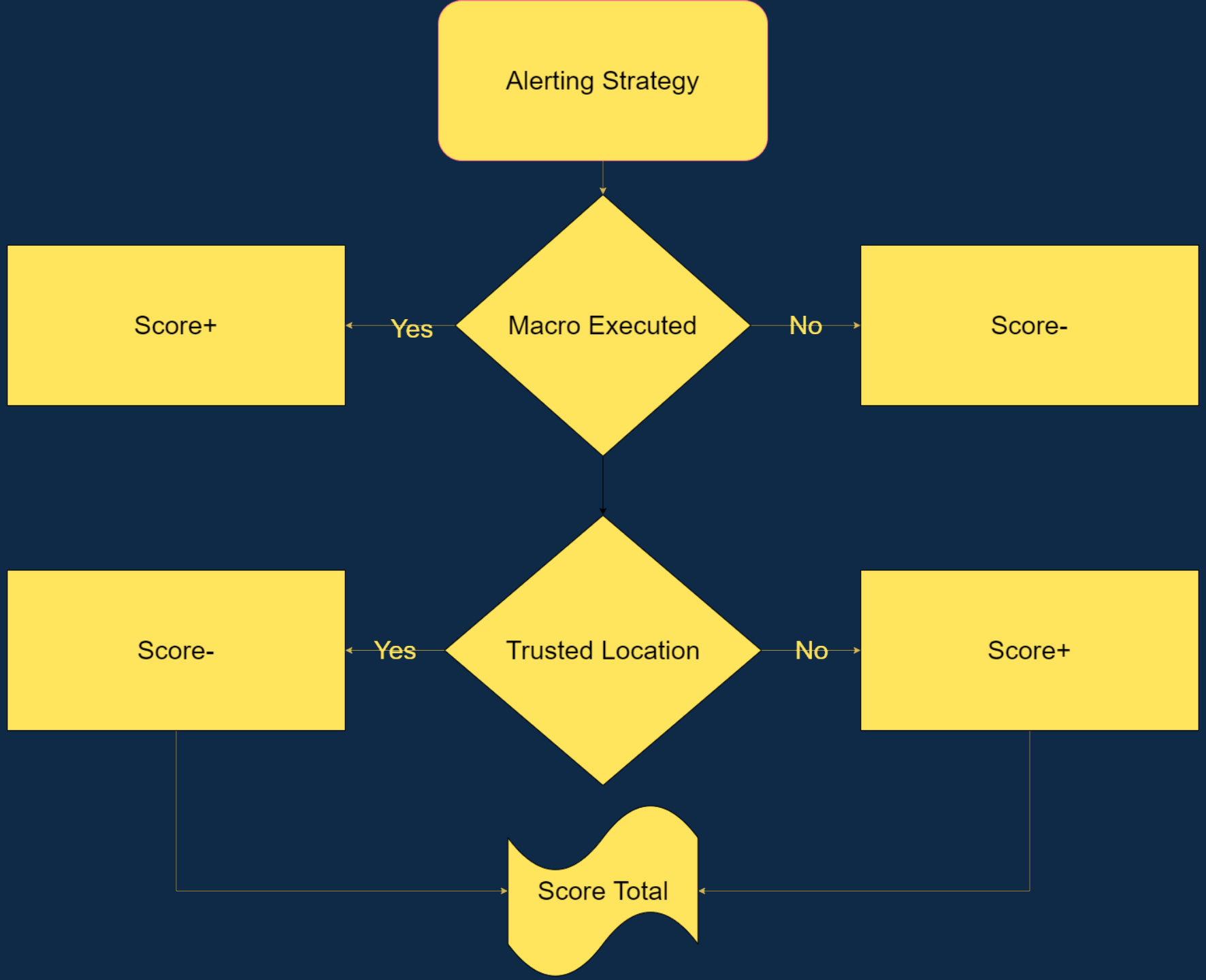
https://github.com/murchisd/splunk_pstree_app/

https://twitter.com/donaldmurchison

# Knowledge Check

- Who is using documents with macro functionality
- Where these documents are executing from
- Whether these documents are:
  - Dropping odd executables or scripts
  - Talking to Azure/Cloudflare/DO or other strange domains
  - Loading DLL files necessary for COM, WMI or .NET functionality


. Process
. File
. Network
. DLL
. Registry
Gain Macro Visibility
Baseline Activity
Macro/Non Macro
Execution Location
Process Tree
Known-Good Locations
. Dropping Suspicious Files
. Editing Suspicious Registry Keys
. Suspicious Image/DLL Loads
. Network Connectivity
. Opened from Outlook/Email
Discover Odd/Malicious Behaviour
Alerting Stategy


Alerting Strategy
Macro Executed
Yes
Score+
No
Score-
Trusted Location
Yes
Score-
No
Score+
Score Total

# In Action – The Query

```
index=sysmon Image=*WINWORD.EXE* OR ParentImage = *WINWORD.EXE*
| bin _time span=5m
| eval ProcessGuid=coalesce(ProcessGuid,SourceProcessGUID)
| eval qualifiers=if(match(ParentImage,"OUTLOOK.EXE"), mvappend(qualifiers,"Outlook as Parent # score: 2"),qualifiers)
| eval qualifiers=if(match(ImageLoaded,"VBE"),mvappend(qualifiers,"VBE DLL Loaded # score: 3"),qualifiers)
| eval qualifiers=if(match(TargetObject,"Trusted Documents"),mvappend(qualifiers,"Trust Record Modification # score: 3"),qualifiers)
| eval qualifiers=if(match(CommandLine,"MyFileShare"),mvappend(qualifiers,"File Opened From Trusted Source # score: -3"),qualifiers)
| eval qualifiers=if(match(GrantedAccess,"0x1fffff"),mvappend(qualifiers,"RWX Granted Access in CallTrace # score: 2"),qualifiers)
| eval qualifiers=if(match(Image,"powershell"),mvappend(qualifiers,"PowerShell spawned from Office Product # score: 10"),qualifiers)
| rex field=qualifiers "(?<=score: )(?<score>(.*)(?=))"
| eventstats sum(score) as score_total by host,_time
| search qualifiers=*
| stats values(qualifiers),values(score_total) BY host,_time
```

https://gist.github.com/MHaggis/11b24e40ef56a4f02049182b8e5b05dc - **@M_haggis**

https://ateixei.medium.com/siem-hyper-queries-introduction-current -detection-methods-part-i-ii-13330b5137df **@ateixei**

# In Action – The Results

| | host | _time | values(qualifiers) | values(score_total) |
|---|---|---|---|---|
| 1 | WIN10-0 | 2021-09-09 15:17:00 | PowerShell spawned from Office Product # score: 10<br>Trust Record Modification # score: 3<br>VBE DLL Loaded # score: 3 | 19 |
| 2 | WIN10-0 | 2021-09-09 15:18:00 | File Opened From Trusted Source # score: -3<br>Trust Record Modification # score: 3<br>VBE DLL Loaded # score: 3 | 6 |

- You have control over the “levers”
- Macro execution not always malicious
- Context matters
- PowerShell as a child of Word, when the document was sent via email = higher score
- Macro execution, no process spawned, from a trusted source = lower score

# Atomic Red Team

## T1204.002 - Malicious File – Atomic Test 1

| | | | | |
|---|---|---|---|---|
| 3 | WIN10-0 | 2021-09-09 15:26:00 | Cscript spawned from Office Product # score: 10<br>Suspicious JSE File Created # score: 10<br>Suspicious WMI ImageLoad # score: 10<br>VBE DLL Loaded # score: 3 | 56 |

## T1204.002 - Malicious File – Atomic Test 3

| | host | _time | values(qualifiers) | values(score_total) |
|---|---|---|---|---|
| 1 | WIN10-0 | 2021-09-09 15:30:00 | Command Prompt spawned from Office Product # score: 10<br>Command Prompt with suspicious parameters spawned from Office Product # score: 15<br>Suspicious WMI ImageLoad # score: 10<br>VBE DLL Loaded # score: 3 | 61 |

## T1204.002 - Malicious File – Atomic Test 6

| | host | _time | values(qualifiers) | values(score_total) |
|---|---|---|---|---|
| 1 | WIN10-0 | 2021-09-09 15:32:00 | RWX Granted Access in CallTrace # score: 2<br>Suspicious VBS File Created # score: 10<br>Suspicious WMI ImageLoad # score: 10<br>VBE DLL Loaded # score: 3 | 60 |

# Windows Management Instrumentation (WMI)

Breaks Parent/Child Process Detections ☹

```
WmiPrvSE.exe (4236)
|--- calc.exe (8500)                              2021-09-05 12:55:50 calc

explorer.exe (9268)
|--- WINWORD.EXE (2836)                           2021-09-05 12:55:49 "C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE" /n
"C:\Users\administrator\Desktop\Tests\calc_vba_wmi.doc" /o ""
```

BUT ☺

- Function Calls via Event ID 10 (https://www.lares.com/blog/hunting-in-the-sysmon-call-trace/)
- WMI ImageLoad Events

| host | _time | values(qualifiers) | values(score_total) |
|---|---|---|---|
| WIN10-0 | 2021-09-05 12:55:00 | RWX Granted Access # score: 2<br>Suspicious WMI Function # score: 10<br>Suspicious WMI ImageLoad # score: 10<br>Trust Record Modification # score: 3<br>VBE DLL Loaded # score: 3 | 61 |

# PPID Spoofing

**Goal**: Explorer → PowerShell → Calc
**NOT**: WinWord → PowerShell → Calc

SourceImage: C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE
TargetProcessGUID: {26d732db-653b-613a-9503-000000005800}
TargetProcessId: 6640
TargetImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
GrantedAccess: 0x1FFFFF

| | host | _time | values(qualifiers) | values(score_total) |
|---|---|---|---|---|
| 1 | WIN10-0 | 2021-09-09 15:49:00 | RWX Granted Access in CallTrace # score: 2<br>Suspicious TargetImage (PowerShell) # score: 10<br>Suspicious WMI ImageLoad # score: 10<br>Trust Record Modification # score: 3<br>VBE DLL Loaded # score: 3 | 51 |

https://blog.christophetd.fr/building-an-office-macro-to-spoof-process-parent-and-command-line/

# .NET – Gadget2Jscript

https://github.com/med0x2e/GadgetToJScript

- Approach:
  - Build test payload
  - Use OfficeShush.xml – compare normal macro to .NET code macro
  - Add differences to OfficeSus.xml (or existing tooling)
  - Add qualifiers to alert
- Results:
  - Clr.dll Loaded by Word/Excel
  - .NET Native Images Loaded by Word/Excel → *C:\Windows\assembly*

```
| eval qualifiers=if(match(ImageLoaded,"clr.dll"),mvappend(qualifiers,"DotNet Office Load # score: 10"),qualifiers)
| eval qualifiers=if(match(ImageLoaded,"assembly"),mvappend(qualifiers,"DotNet Native Image Office Load # score: 10"),qualifiers)
```

# Scoring Our .NET Macro

| values(qualifiers) | values(score_total) |
| --- | --- |
| DotNet Native Image Office Load # score: 10<br>DotNet Office Load # score: 10<br>Suspicious WMI ImageLoad # score: 10<br>Trust Record Modification # score: 3<br>VBE DLL Loaded # score: 3 | 129 |
| File Opened From Trusted Source # score: -3<br>RWX Granted Access in CallTrace # score: 2<br>Trust Record Modification # score: 3 | 2 |

**Dotnet Macro**

**Normal Macro Opened from file share**

# The Bigger Picture

Phishing Email

User

**We are here**

Attacker Infrastructure

*Phish Sent*

*Malicious Attachment Opened*

*Command and Control*

# Goals

- Office is a massive attack surface
  - Telemetry is not great, think PowerShell telemetry vs Macro telemetry
- Impossible to keep up detections for macro tradecraft
  - *Google -> #maldoc site:https://twitter.com/Sbousseaden*
- Long/medium term goal of baselining and alerting strategy should be prevention – disabling of macros, hardening Office products
- Very Difficult to do, but worth doing


https://twitter.com/JohnLaTwC/status/682351869641949185

# Thank You! / Questions ?