

Forum:[Hack Forums](#)

Thread: [Anti-Analysis - prevent analysis on your file](#)

debug - December 15, 2016, 11:32 am

Hi,

Here is a project I was working on during my coding stream. It is unfinished, but has good potential.

About

This project is designed to help you prevent unwanted reversing and otherwise, analysis, on your binary. This is useful in many cases.

Current Features
On-detection method: Kill process, display fake error message, and consume resources.

- Anti Sandboxie

- Anti Virtual Machine

- Anti Emulation

Download

Github: <https://github.com/gubed/Anti-Analysis>

Disclaimer

This project was created as a proof of concept and for educational purposes only. It helps prevent reverse engineering and analysis by blocking access on Virtual Machines and Sandboxie.

Zilog80 - December 15, 2016, 11:58 am

Wow! Really nice

I'm trying to figure what Emulation.cs does

Can you explain us what that 5 for-loop does?

debug - December 15, 2016, 12:02 pm

(Dec 15, 2016 12:58 PM)Zilog80 Wrote: [\[CITE\]](#) Wow! Really nice

I'm trying to figure what Emulation.cs does

Can you explain us what that 5 for-loop does?Oops, forgot to mention that one. Basically, most analyzing programs such as firewalls (Comodo), anti viruses, debuggers, etc, can simply step over traditional methods to block them from monitoring your process. So, for instance, if you do Thread.Sleep(60000) (sleep 60 seconds) at the start of your application to prevent other applications from seeing what yours is doing, they can simply ignore that bit of code.

What the emulation does, is force the application to 'sleep' for about 60 seconds. By generating a random string and getting the md5, the program will then proceed to crack the md5 hash, which will

take about 1 minute. This prevents analyzing agents from simply stepping over that code. Most agents stop 'pre-monitoring' after about 15-45 seconds.

Voidlight - December 20, 2016, 9:46 pm

(Dec 15, 2016 01:02 PM)debug Wrote: [\[CITE\]](#) Oops, forgot to mention that one. Basically, most analyzing programs such as firewalls (Comodo), anti viruses, debuggers, etc, can simply step over traditional methods to block them from monitoring your process. So, for instance, if you do Thread.Sleep(60000) (sleep 60 seconds) at the start of your application to prevent other applications from seeing what yours is doing, they can simply ignore that bit of code.

What the emulation does, is force the application to 'sleep' for about 60 seconds. By generating a random string and getting the md5, the program will then proceed to crack the md5 hash, which will take about 1 minute. This prevents analyzing agents from simply stepping over that code. Most agents stop 'pre-monitoring' after about 15-45 seconds.this is brilliant, nice job OP.

Malum™ - December 21, 2016, 1:54 am

These are really cool! I know the anti-sandboxie one has been around awhile but haven't seen these other ones. I'll definitely save them for later!

rbcks - December 22, 2016, 4:08 pm

how i can add it to my project?

Malum™ - December 23, 2016, 3:34 am

(Dec 22, 2016 05:08 PM)rbcks Wrote: [\[CITE\]](#) how i can add it to my project?

You add one of the classes from the [modules](#) and then basically copy [this](#) code and modify it to work with your project.

Man16 - December 23, 2016, 7:33 am

(Dec 15, 2016 01:02 PM)debug Wrote: [\[CITE\]](#) Oops, forgot to mention that one. Basically, most analyzing programs such as firewalls (Comodo), anti viruses, debuggers, etc, can simply step over traditional methods to block them from monitoring your process. So, for instance, if you do Thread.Sleep(60000) (sleep 60 seconds) at the start of your application to prevent other applications from seeing what yours is doing, they can simply ignore that bit of code.

What the emulation does, is force the application to 'sleep' for about 60 seconds. By generating a random string and getting the md5, the program will then proceed to crack the md5 hash, which will take about 1 minute. This prevents analyzing agents from simply stepping over that code. Most agents stop 'pre-monitoring' after about 15-45 seconds.

So programs such as anti viruses etc that monitor what programs do at startup only work for about 15-45 seconds. So in your own program to avoid them, you can make your application sleep for about a minute?

Zilog80 - December 29, 2016, 11:17 am

(Dec 29, 2016 11:18 AM)Salmoneus2 Wrote: [\[CITE\]](#) debug u are still a noob lool can u pls pm me Why @debug should PM you when you have only 3 post and you will not be able to reply him due post number cap? ?

Salmoneus2 - December 29, 2016, 1:52 pm

(Dec 29, 2016 12:17 PM)Zilog80 Wrote: [\[CITE\]](#) Why @debug should PM you when you have only 3 post and you will not be able to reply him due post number cap? ?
because i need his skype so i can roast him lol

kozit - January 1, 2017, 7:55 am

this looks great but look like it might be slower on older machines

debug - *January 1, 2017, 11:02 am*

(Jan 1, 2017 08:55 AM)kozit Wrote: [\[CITE\]](#) this looks great but look like it might be slower on older machinesYes, WMI is very slow. I did take a few measures to increase the speed, as you can see commented out version that took about 2 extra seconds to query the information.