Malware Analysis 101 - Sandboxing

Aditya Anand Follow
Sep 29 · 7 min read

> *This article is a continuation of my previous write-up "Malware Analysis 101- Basic Static Analysis", do give it a read before going ahead with this one to have a better understanding of the things that I will be explaining here.*

I wrote my previous article on Basic Static Analysis of malware and the next article I had in mind was the Basic Dynamic Analysis of malware. Before I start with the dynamic analysis of malware I need to explain the pros and cons of dynamic analysis of malware and how to protect yourself best from the cons.

Let's begin!

Malware Analysis is broadly divided into two groups Static Analysis & Dynamic Analysis. We can describe static analysis to be all those examinations of the malware where we don't actually execute the malware but try to figure out what the malware is trying to do and the commands it is attempting to execute. Dynamic analysis, on the other hand, is all those examinations that you carry out when you actually execute the malware most preferably in a sandboxed environment and then try to figure out the functionality of the malware.

Sandbox

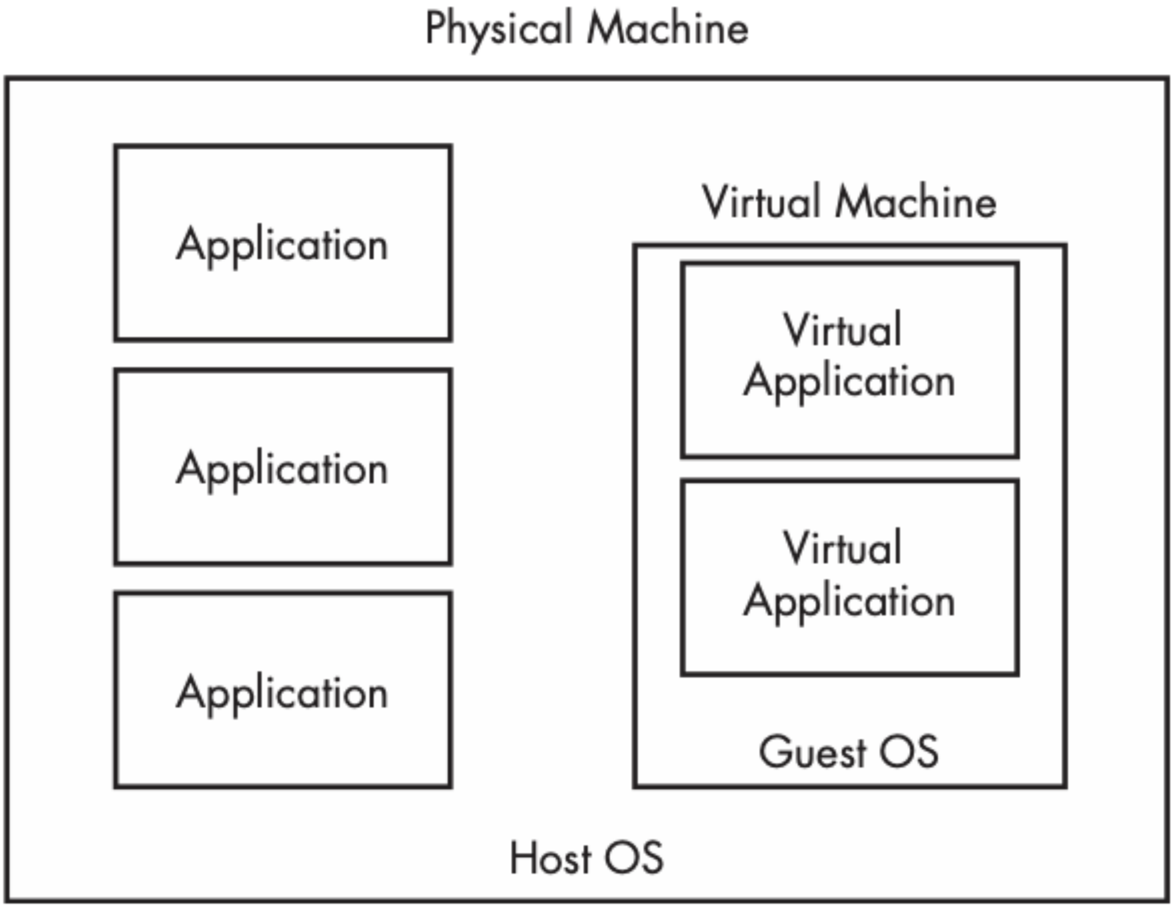
Now, we know that while carrying out the dynamic analysis of malware we need to execute it and the one thing we know for sure is that we should never let any malware enter our system let alone execute it. These two lines are contradictory so then how will we ever carry out dynamic analysis and therein comes the idea of sandboxing.

This technique helps us to protect our systems from malicious software while at the same time it allows us to carry dynamic analysis to better understand the working of the malware.

To be true I understood this theoretical explanation while I was in high school but never really understood what it actually is for a long time so let’s look in-depth about what it means to set up your own sandbox environment.

There are various tools including GFI Sandbox, Norman SandBox, Anubis Joe Sandbox, ThreatExpert, BitBlaze, and Comodo Instant Malware Analysis which are used to set up a sandbox environment, but I will focus on configuring sandbox on your own system.

An oversimplification of what a sandbox is that it is a virtual machine. So to understand how a virtual machine acts as a sandbox you need to look at this image.



This depicts, how the virtual machine even though it resides on your system is still independent of its host machine and so “whatever happens on your virtual machine stays on your virtual machine”.

Pros of using a VM

There are various reasons for using a virtual machine as a sandboxed environment to test out the malware rather than using an actual system. One of the clear cut reason is that using actual machines are way too costly when compared too using a virtual machine.

The other most apparent reason for using a virtual machine over a bare-metal machine is that when a malware infects a virtual machine and goes out of control we can just delete the malware and start all over again but that is not the case with bare-metal machines as the malware can still linger on in the machines where they were being executed on and removing the malware 100% from the system might be a bit difficult.

Virtual Machines provide us with the feature of “snapshot” which is a huge advantage and I can’t stress that enough.

The functionality of the snapshot option is to preserve a state of the virtual machine at a certain point of time so that whenever the malware messes up stuff he can always go back to the last snapshot they took and continue from there. To understand this in even more layman terms consider this saving your progress while playing a super long game.

Cons of using a VM

While this is all good and great but there is a hurdle in this and that is whenever you run malware in a virtual machine there are chances that the malware will detect it being executed inside a virtual machine and will not carry out the set of steps it would if it was in an actual system.

The malware can detect this in a lot of different methodologies whether it is being executed in a virtual machine environment or in an actual system:-

Problem 1: One of them being the amount of space on the hard drive. For example, a normal machine would at least have 256 GB of storage in SSD or around 512GB if HDD but if the malware detects that the hard disk storage is only 20GB or so then it will consider that it is being executed in a virtual environment and chances are it is being examined by a malware analyst.

Solution: To tackle this we have to assign at least 50GB+ hard disk space to this machine so that the malware doesn’t outright reject the machine.

Problem 2: Another example could be a test for internet connectivity. Generally, virtual machines in which we test out malware are kept disconnected from the internet as we don’t know what the malware might attempt to do if it finds an active internet connection. It might try to propagate to other machines via the internet and so we can’t allow that hence most of the time when we are doing malware analysis we keep the virtual machine in an internal network rather than going for bridged or NAT.

Solution: The solution to this problem is the use of tools like ApateDNS, InetSIM, Netcat, etc. These tools are used to set up a fake internet connection so that if the malware tries to figure out if the machine has an internet connection or not it will actually fall prey to these tools and consider that there is a connection even though there isn’t one in reality.

Problem 3: There is another problem that arises on a virtual machine when the malware searches for another device on the network and if it doesn't find any it acts as if it is being run on a virtual machine as any network with only one device is treated mostly as a virtual machine.

Solution: The best way to solve this problem is to set up various virtual machines and then make them part of a same network using internal networking and then set up fake internet connectivity oon each of these devices. In this way, the malware is tricked into thinking it is being executed in a real-life environment and then we can monitor the traffic using tools like Wireshark, to understand exactly what the malware is trying to do and is there any command and control server it is trying to reach out to or is it trying to spread the malware on the network, etc.

It is always a game of cat and mouse trying to find different ways to outdo the hacker and trying to find ways to analyze their malware while the attacker is trying to do the same thing.

Next Step: Basic Dynamic Analysis

Now that you know what a sandbox is and setting it up is similar to configuring a virtual box, I will now mention the techniques used for basic dynamic analysis in my upcoming article. Keep this in mind while configuring the sandbox environment in the virtual machine mostly go by the default configuration except the networking and hard disk space unless you have some specific requirements.

Edit: Most of the stuff mentioned in the article and the screenshots are taken from the book - Practical Malware Analysis. You should definitely consider buying the book, link here. I have summarised my notes here and a few points have been directly picked from the book as I considered them to very well explained and it didn't make sense to rephrase or edit them.

If you enjoyed it please do clap & let's collaborate. Get, Set, Hack!

Website : aditya12anand.com | Donate : paypal.me/aditya12anand

Telegram : https://t.me/aditya12anand

Twitter : twitter.com/aditya12anand

LinkedIn : linkedin.com/in/aditya12anand/

E-mail : aditya12anand@protonmail.com

. . .

Follow Infosec Write-ups for more such awesome write-ups.

InfoSec Write-ups

A collection of write-ups from the best hackers in the world on topics ranging from bug bounties and CTFs to

medium.com