

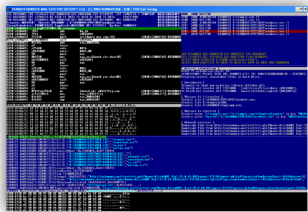
## Norman SandBox Analyzer PRO

Analyzing malware can be a cumbersome and time consuming task, involving multiple applications for code analysis as well as a network of computers. Each one of the applications is meant to perform its specific task and most of the time you need to combine the result of several of these to find the true actions and intent of the malware.

With Norman's new SandBox Malware Analyzer product line, the complexity, speed and infrastructure needed to analyze files have been dramatically reduced. This will give you a quick return on investment.

[www.malwareanalyzer.com](http://www.malwareanalyzer.com) • [www.virusanalyzer.com](http://www.virusanalyzer.com)

The SandBox Analyzer PRO is a console GUI application designed to analyze WIN 32 PE executables.



### How can Norman SandBox Malware Analyzers help?

#### Save time

- The average response time to a new threat is normally 6 – 24 hours.
- Get a head start with knowledge of what the sample is trying to do.

#### Save money

- A growing number of viruses to analyze require a high number of analyst efforts.
- Finding the right people to analyze malware is a difficult, time-consuming, and costly task.

#### Save the day

- You have been in the situation where something needed to be analyzed yesterday and now you have access to the tools to make it happen.

### Product description

The Norman SandBox Analyzer PRO (NSAP) is an application designed to perform deep analysis of any Win 32 PE executable file more effectively than any other analyzing tool. The SandBox Analyzer PRO is unique in letting the users analyze malicious code such as viruses, worms, trojans, keyloggers, etc. When performing in-depth analysis of files and their behavior and actions, you can look at loaded libraries, running threads, created sockets etc. You can even set breakpoints and enter commands. There is disassembly view, register view, memory dump, API log view, command input view and more.

NSAP is a console GUI application designed to analyze WIN 32 PE executables. When working with NSAP you can use an extensive list of parameters enabling you to analyze and manipulate the emulation, extend emulation cycles, etc. You will also be able to explore files and changes made to the simulated SandBox OS to get the full view of the impact execution the respective would have had if it was run on a “real computer”.

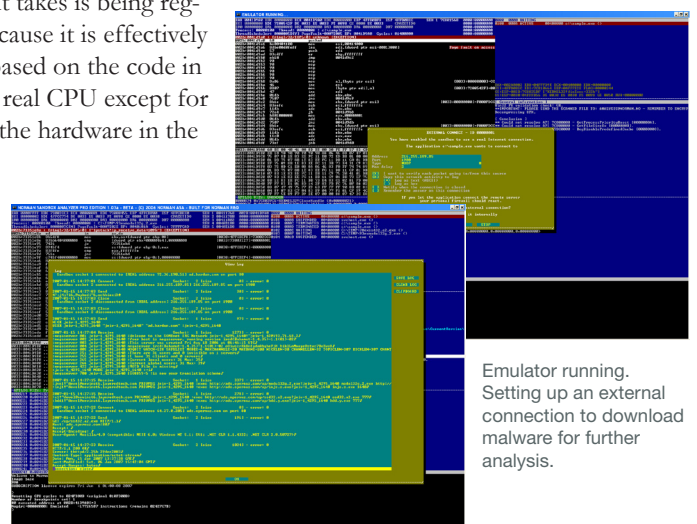
Norman SandBox is the core component of the NSAP. This module is compatible with Windows functions such as Winsock, Kernel and MPR and also supports network and Internet functions like HTTP, FTP, SMTP, DNS, IRC, and P2P. In other words it's a fully simulated computer, isolated within the NSAP application.

The simulator uses full ROM BIOS capacities, simulated hardware, simulated hard drives, etc. The simulator emulates the entire bootstrap of a regular system at boot-time, starting by loading the operating system files and the command shell from the simulated drive. This drive will contain directories and files that are necessary parts of the system, conforming to system files on physical hard drives.

The file to be analyzed is placed on the simulated hard disk and is started in the simulated environment. Inside the simulated environment the file may do whatever it wants. It can infect files. It can delete files. It can copy itself over networks. It can connect to an IRC server. It can send emails. It can set up listening ports. Every action it takes is being registered by the antivirus program, because it is effectively the emulator that does the actions based on the code in the file. No code is executed on the real CPU except for the antivirus emulator engine; even the hardware in the simulated PC is emulated.

The issue is to figure out what the program would have done if it had been allowed to run wild on an unprotected machine, in an unprotected network.

To ease the analyzing a SandBox summary and an API log from the analyzed file can be created.



Emulator running. Setting up an external connection to download malware for further analysis.

