# Malcode Analysis Software Tools

By Russ McRee


**Figure 1 – SysAnalyzer Wizard**

## Prerequisite

Windows 2000 or XP

For bug hunters there are a great many tools available, from simple command line essentials such as *strings* or *netstat*, to rootkit detectors like Helios, or Joe Stewart's Truman. From David Zimmer of iDefense Labs (now a VeriSign division) you will find an excellent set of tools for malcode analysis on Windows PCs that provide detailed discovery. I asked Ken Dunham of iDefense (see Risk Radar in this publication) about future plans for these tools. According to Ken, they are working on additional tools and techniques. Of those that will be public, they will likely release tools for vulnerability discovery and testing as well as possible future updates to MultiPot and exploit/shellcode handlers.

For further reading on malware analysis give Joe Stewart's work a read on the Secureworks blog[1] and take a close look at Lenny Zeltser's paper on reverse-engineering malware.[2] There are four malcode analysis offerings on the iDefense site but for this effort we'll cover three, specifically SysAnalyzer, Malcode Analysis Pack, and MultiPot.

## SysAnalyzer

SysAnalyzer is described on the iDefense Labs site as "an automated malcode run time analysis application that monitors various aspects of system and process states." From the SysAnalyzer overview comes one **critical note**: SysAnalyzer is not a sandboxing utility. Target executables are run in a fully live test on the system. Thus, if you are testing malicious code, your test system will be infected.[3] The simplest method to test malware under these circumstances is using one of the free VMWare solutions like VMWare Server or VMWare Fusion for Mac (beta), where you can take a Snapshot, then Revert when your research is complete.

In just such an environment I fired up a Windows XP victim and fed SysAnalyzer a Rinbot/VanBot variant packed neatly into `sans.exe`. Yes, this is the same botnet malware that contained truly unkind words for the Internet Storm Center's Johannes Ullrich hidden in the code.[4] SysAnalyzer's initial UI is an efficient little wizard that offers additional options to use *Sniff Hit, API Logger*, and *Directory Watcher*. I selected Sniff Hit and Directory Watcher and clicked *Start*.

Sniff Hit quickly popped up and indicated a Unique IP in the RIPE NCC region (often typical of bots). Sniff Hit is described in the overview as a specialized HTTP and IRC sniffer designed to grab target communication data that includes basic methods to pick up on target traffic that is not on a known or predefined port.[5]

The SysAnalyzer view, after execution, includes *Running Processes, Open Ports, Process Dlls, Loaded Drivers, Reg Monitor,* and *Directory Watch Data*. Our nasty little visitor immediately showed results as seen in Figure 2.

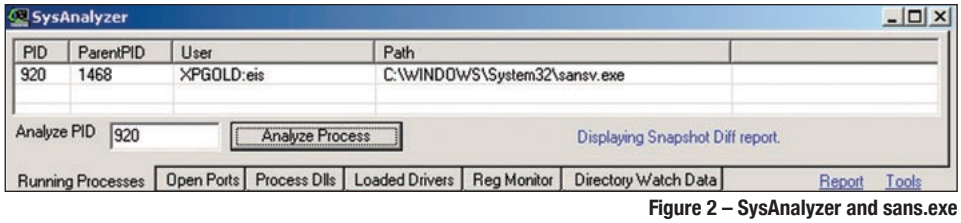


**Figure 2 – SysAnalyzer and sans.exe**

When viewing the Running Processes tab you are afforded the additional opportunity to *Analyze Process* which will spawn *Process Analyzer*, where you can right-click a PID of your choice, logically 920 as seen in Figure 2, representing `sansv.exe`. This will, in turn, *List Data* on the specific process including MD5, packer, and file properties as well as run it through some basic exploit signatures in an attempt to identify the malcode.

Open Ports displayed TCP port 12852 for PID 920 and a process path of `C:\WINDOWS\System32\sansv.exe`, which was consistent with an entry in Directory Watch Data that also indicated `Created: C:\WINDOWS\System32\sansv.exe`.

Reg Monitor let me know that an entry had been made at `HKLM\Software\Microsoft\Windows\CurrentVersion\Run` for the SANS Service, again referring to `C:\`

1 http://www.secureworks.com/research/threats/

2 http://www.zeltser.com/reverse-malware-paper/

3 http://labs.idefense.com/files/labs/releases/previews/SysAnalyzer/

4 See http://isc.sans.org/diary.html?storyid=2295

5 http://labs.idefense.com/files/labs/releases/previews/map/

`WINDOWS\System32\sansv.exe`. How considerate of our bot-writing friend to ensure that such a service always starts for us automatically. We appreciate it, really, we do…like lambs to the slaughter.

SysAnalyzer will also present you with yet more *Tools* including *Snapshot* options and the ability to create a *Known File DB*. Snapshot capabilities are useful for comparing snapshots over a time interval of your choosing.

I chose to run *ApiLogger* in standalone mode with an executable named `card232.exe`, identified later using MD5 hash methodology. ApiLogger adds realtime API logging to the analysis output by injecting a `dll` into the target process. Once loaded, the `dll` will insert a series of detour-style hooks into specific API calls. When these API are accessed by any code in the process, they will trigger a notification message, which is sent to the main SysAnalyzer interface, or the API Call Log in standalone mode.[6]

The results at the end the *Inject & Log* process outlined a couple of malware indicative traits. At address 40335e I saw a connection made to a specific IP address (hosting additional malware at the time this bug was active). At 401433 was an entry to `HKLM\Software\Microsoft\Windows\CurrentVersion\Run`, malware's favorite home in your registry. Finally, at 4014fc, I saw the executable for MSN Messenger copied to a renamed `msrr.exe`. Ruh-roh Rastro, your IM just got 0wn3d.

SysAnalyzer is an excellent framework in which to "quickly collect, compare, and report on the actions" taken by malware on a system.

## Malcode Analysis Pack

For our review of the Malcode Analysis Pack, or MAP, I chose a veritable mixed salad of garden variety bugs with which to explore some of the following tools included in MAP:

- *ShellExt* – 4 explorer shell extensions
- *socketTool* – manual TCP Client for probing functionality.
- *MailPot* – mail server capture pot
- *fakeDNS* – spoofs dns responses to controlled IPs
- *sniff_hit* – HTTP, IRC, and DNS sniffer
- *sclog* – Shellcode research and analysis application
- *IDCDumpFix* – for quick RE of packed applications
- *Shellcode2Exe* – embeds multiple shellcode formats in exe husk
- *GdiProcs* – detect hidden processes

```
c:\iDefense\MAP>gdiprocs /f
GDI Process Scanner -

Scanning GDIShared Handle Table for unique process ids...

    40 processes returned by GDI table

Processes listed in GDI:
------------------------------------------------
  18 - ---- Could not OpenProcess ----
1640 - C:\WINDOWS\TaskManager.exe
2272 - C:\WINDOWS\WinTask.exe
2748 - C:\Program Files\MSN Messenger\MsnMsgr.Exe
```

**Figure 3 – MAP: GDIProcs**

`BackDoor.Java.KBD`, best known as a handy little downloader, immediately turned up in the I-don't-think-you-belong-here category according to GDIProcs. Not so fast you malicious maligner of Microsoft magic, I am certain the good version of Task Manager is neither `C:\WINDOWS\TaskManager.exe` nor `C:\WINDOWS\WinTask.exe` as in Figure 3.

Perhaps a mass mailer might interest you. `W32.boxed` gives us ample fodder for MailPot which captures email sent out by trojans and mass mailers. If the malcode uses Outlook automation you can configure your Outlook client to use MailPot or if it connected to an open relay by domain name use MailPot with fakeDNS to redirect it.[7]

The Shell Extensions are context menu gems, including *Strings* and *MD5 Hash*, with right-click convenience. MD5 Hash lists the target file name, size in bytes and MD5 hash. This is an indispensable method of identification. Feed the hash seen in Figure 4 to Google and you will quickly learn that you are the proud owner of a `Win32/MSNMaker.AB` variant as found in the database at `http://honeynet.cz/?mmenu=malware&smenu_int=0&lang=en&vmetr=1` via hash query.

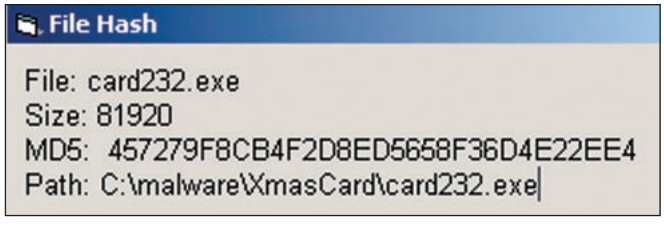


**Figure 4 – MAP: MD5 Hash**

Strings provides invaluable information about certain behavioral attributes of malcode as it extracts all ASCII and Unicode strings from the specified file and displays the results. Sometimes as you are reviewing the output, you may be offered the occasional dead giveaway, convenient particularly for those skilled at seeing only the obvious (like me). Let's review Strings output from `BackDoor.Java.KBD`, specifically `TaskManager.exe`.

```
'(Ljava/lang/String;Ljava/lang/String;)V

addIPListener
```

6 http://labs.idefense.com/files/labs/releases/previews/SysAnalyzer/

7 http://labs.idefense.com/files/labs/releases/previews/map/

```
(LIPChangeListener;)V
removeIPListener
```

Use of an IP listener is a typical malware attribute, yes?

```
Programmed By Kadir BASOL?
```

Remember the dead giveaway? Google the above and the first result refers to `http://www.megasecurity.org/trojans/k/kbd/Kbd1.4.5.html`, which contains a complete write-up from the malware authors, Kadir and Kerim Basol. Nice. Here we learn that the KBD reference in the malcode's common name is an acronym for KADIR BASOL DEVASTATOR. Wow, someone didn't get enough hugs growing up. A final note on this malware. It creates a connection to an IP address in Turkey. Using my superhuman powers of deduction, I see in the Basol brothers' descriptive text that they wrote this code in Turkey in 2003. In my best Forrest Gump voice, "I'm not a smart man, but I know what a backdoor is."

## MultiPot

MultiPot is an emulation-based honeypot designed to capture malicious code as it spreads via various exploits across the net. The captures are such that the host machine requires minimal supervision and is not itself at risk of infection. It was designed specifically to emulate exploitable services in order to safely collect malicious code.

You might find MultiPot useful as an ISP monitoring your network or as corporate security personnel watching for outbreaks. It might also be useful to security and virus researchers to build statistics or collect samples.[8]

MultiPot is very simple to setup, and is offered under the GPL, so you can craft your own handlers or modify those included. In fact, all these tools exist under the GPL, leaving additional opportunities to experiment. Source code is available in the installation or on the iDefense website. MultiPot includes protective measures to avoid disk flooding and the frequency of uploads, and for shellcoders, it includes five shellcode handlers which represent the most commonly seen shellcodes at the time this app was created. Each of these handlers can be tested individually.

MultiPot was last updated in 2005, but as mentioned in the introduction, there will be future work on MultiPot.

While writing this, I did not actually expose MultiPot to the internet, but I did pseudo-fake it out with *nmap*. The results are seen in Figure 5.

MultiPot, along with the right-click context menu MD5 Hash and Strings shell extensions are my favorites, but I'm far from finished with my use and research of all these malcode analysis tools.

## Benefits and drawbacks

Cost to use these tools from iDefense Labs? Zero dollars.

Cost to buy similar commercial offerings? Hundreds or thousands of dollars.

Value of the knowledge gained from using MAP or SysAnalyzer? Priceless.

Enjoy these tools, there are no drawbacks other than the normal caution flags raised by researching malware in production environments. Be careful, but you knew that.

## In conclusion

Studying malware is an endless and evolving process, but tools like MAP and SysAnalyzer offer significant aid in that process, providing an ample framework for experimentation and research. Keep in mind that I really only covered a fraction of the capabilities of these tools, having barely touched on the shellcode tools. If taking a closer look at malware, as a reverse-engineer or system administrator, these tools will serve you well. Cheers…until next month.

## About the Author

*Russ McRee, GCIH, CISSP, is a security analyst working for Expedia. He is a member of numerous security organizations and maintains holisticinfosec.org. Contact him at russ@holisticinfosec.org.*

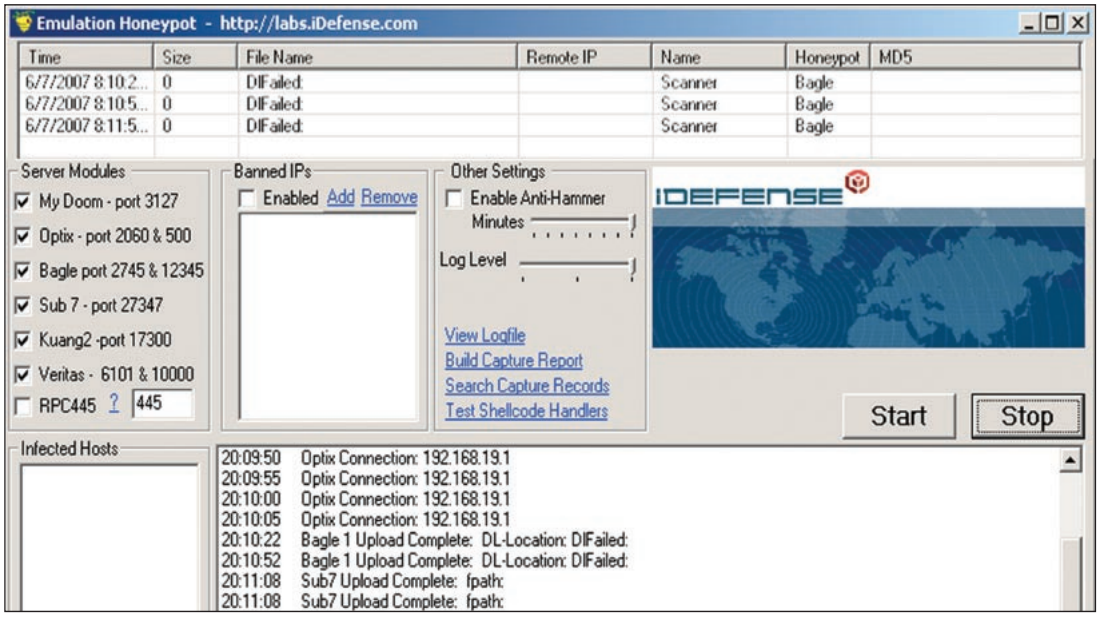


**Figure 5 – MultiPot**

8 http://labs.idefense.com/files/labs/releases/previews/multipot/index.html