Buster Sandbox Analyzer

By Russ McRee – ISSA Senior Member, Puget Sound (Seattle), USA Chapter

Prerequisites

Windows
Sandboxie 3.64 or later[1]

On April 10, 2012, a new version of Sandboxie was released, and on April 16 so too was a new version of the Buster Sandbox Analyzer,[2] which uses Sandboxie at its core. Voila! Instant *toolsmith* fodder.

It's been a few months since we've covered a malware analysis-specific tool, so the timing was excellent.

Buster Sandbox Analyzer (BSA) is intended for use in analysis of process behavior and system changes (file system, registry, ports) during runtime for evaluation as suspicious. You'll find it listed among the Sandbox Tools for Malware Analysis on one of my favorite Internet resources, Grand Stream Dreams.[3]

As always, I pinged the developer and Pedro Lopez (pseudonym) provided me with a number of insightful details.

He releases new versions of Buster Sandbox Analyzer on a fairly regular basis,[4] version 1.59 is current as I write this. There's an update mechanism built right into BSA; just click *Updates* then *Check for Updates*. Pedro has recently improved static analysis, and he's always trying to improve dynamic analysis as he considers it the most important aspect of the tool.

For future releases the TO-DO list is short, given over two years of constant development. The following features are planned:

- A feature to analyze URLs in automatic mode.
- Utilizing the information stored in the SQL database, a feature to generate statistics including used compressors, detected samples, and others.

Pedro continuously looks for new malware behaviors to include and improvements for the features already implemented. Your feedback is welcome here, readership.

Pedro was first motivated to create the tool thanks in large part to Sandboxie. "Before I start coding Buster Sandbox Analyzer back in late 2010, I knew of Sandboxie already. I started using this great software around 2008 and had coded other utilities using Sandboxie as a file container, so I knew already of the potential to write other types of programs for use with Sandboxie. I created Buster Sandbox Analyzer because I didn't like that all publicly available malware analyzers were running under Linux. I like Linux-based operating systems but I'm mainly a Windows user, so I wanted a malware analysis tool running under Windows. I knew Sandboxie was perfect for this task and with the help of Ronen Tzur (Sandboxie's author) it was possible to do it."

Pedro cites several favorite use cases but two are stand outs for him:

1. Use the tool to know what files and registry modifications were created by a program. While this use case is not always directly related to malware analysis, it can be used by any user that wants such information regarding program behavior.
2. Use the tool to learn if a file (executable, PDF document, Word document, etc.) exhibits malware-specific behavior.

Goes without saying, right?

Pedro reports that Buster Sandbox Analyzer suffers from a lack of user feedback (help change that!). He's not really sure how many people have used it to date or how many use it regularly but does recall one success story from a user on the Wilders Security Forums:

"I was shopping on Usenet for some tax software... I found it and ran it in the sandbox. As is my practice, I explored the installed files. Everything worked well. No obvious signs of infection, no writing to Windows, no start/run entries, and no files created in temp folders. But I still wasn't satisfied. I used Buster's program and reran the install...The program logs were literally laced with created events, DNS queries to Russia, and many hidden processes. Needless to say, I kept it in the sandbox."

One message to convey to you, readers: a few versions ago Pedro introduced multi-language support; there are translations for Spanish, Russian and Portuguese (Brazil), while a translation to German may be available soon. He would like to have translations for Italian, French, Japanese, and Chinese and would be grateful if someone can contribute translations for these languages.

Given the likelihood that this article will be read by security professionals, Pedro welcomes anyone who tries out BSA and has suggestions, ideas, feedback, bugs, etc., to send them to his attention at malware dot collector at gmail dot com.

1 http://www.sandboxie.com/.
2 http://bsa.isoftware.nl/.
3 http://grandstreamdreams.blogspot.co.uk/2012/04/malware-analysis-resources.html.
4 http://bsa.isoftware.nl/frame8.htm.

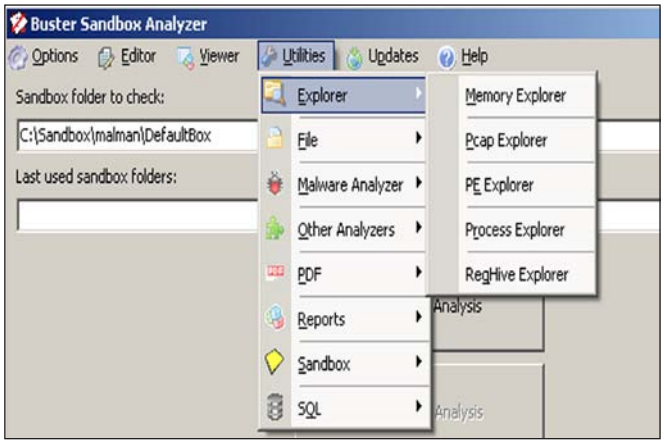

Figure 1 – BSA Explorer features.

Configure BSA

Refer to installation and usage documentation on the BSA site as your primary source, but you may find the BSA guidance at reboot.pro[5] helpful but a bit dated. Consider it documentation reloaded. Actual installation of both Sandboxie and BSA is really straightforward, but there are some configuration tricks worth paying attention to. After reading reboot.pro be sure to add the following to the Sandboxie default configuration file:

```
InjectDLL=C:\BSA\LOG_API.DLL
OpenWinClass=TFormBSA
NotifyDirectDiskAccess=y
```

Even more importantly, this assumes you've installed BSA in C:\bsa. If you choose differently, you must modify the Sandboxie configuration file accordingly. Avoid the Program Files directories on later versions of Windows given the need for administrative permissions to write there.

I'm a big fan of Windows shell integration with any tool that offers it. Under *Options* | *Program Options* | *Windows Shell Integration* select *Add right-click action "Run BSA"* and "Analyze in BSA."

From *Options* set *Common Analysis Options* to include saving packet captures under *Packet Sniffer* via *Save Capture To File*. Be sure to select the correct adapter here as well. Note: BSA utilizes `NetworkMinerConsole.exe` for PCAP analysis. ☺

Also set your *Report Options* from the *Options* menu. I prefer HTML; you may also select PDF and XML. You may also like the SQL options where you can write to a SQL database for analysis and report results.

Be sure to check out the additional features under the *Utilities* menu, including submittal to online analyzers, file tools including disassembly, hashing, hex editing, renaming, signature check, scanning, and strings. There are also "explorers" for memory, PCAPs, PE files, processes, and registry hives as seen in figure 1.

Experiment and fine tune your settings. To then remember settings and load them automatically when the tool starts, select *Options* | *Program Options* | *Save* settings on exit. You can also save multiple configuration files via *Options* | *Program Settings* | *Save Settings As* so as to make use of different analysis patterns.

Lastly, and I imagine you knew I was going to say this, I run BSA in a Windows XP virtual machine and on a bare metal install of Windows 7 running SteadierState. Some malware not only knows when it's running in a VM but it knows when it's running in Sandboxie. If you suspect that's the case, you can hide Sandboxie during a BSA run via *Program Options* | *Hide Sandboxie*.

Using BSA

I wanted to test BSA in two different capacities, one with a browser-borne exploit and one with a "normal" PE.

I am privileged to receive a daily report inclusive of a number of drive-by exploit vehicles so I am always rich in options for exploration, and

```
hxxp:// www.ugpag.cd/index.php?option=com_content&view=
article&id=49&Itemid=75
```

was no exception.

To examine, I started BSA via `bsa.exe` in C:\BSA, tuned my BSA configuration to include some additional reporting options, clicked *Start Analysis*, right-clicked Internet Explorer and chose *Run Sandboxed* (given that Sandboxie is also integrated right into the Windows shell), and finally browsed to the `ugpag.cd` site. Once I willingly stepped through a few browser blocks (yes, I'm sure I want to do that), the "infection" process completed and I chose *Terminate All Programs* by right-clicking on the system tray Sandboxie icon followed by *Finish Analysis in BSA*.

A few key elements jumped right out during BSA analysis and findings.

First, the site spawned an instance of Windows Media Player in order to "play" `hcp_asx` as seen in figure 2.

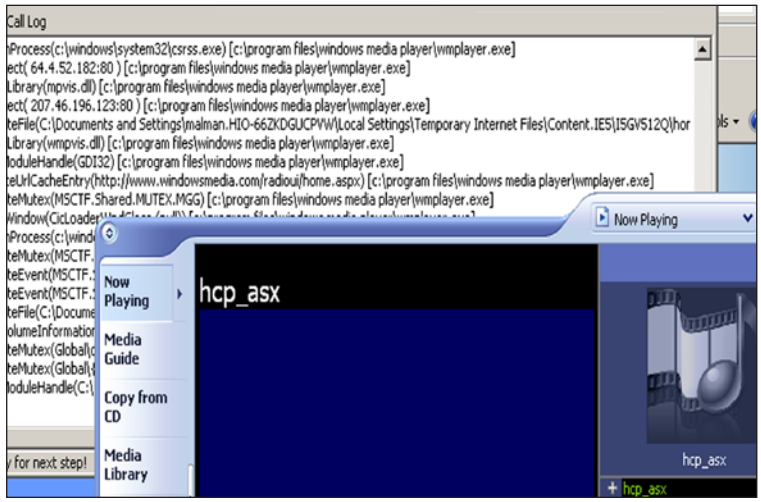

Figure 2 – Pwned site spawns Media Player for hcp_asx.

5 http://reboot.pro/14602/.

Second, when reviewing `Report.html`, I quickly spotted two evil URLs (`lukastroy.in` & `zdravyou.in`) under *Network Services*. Also note the process/window information as seen in figure 3.

A quick URLquery.net search for the URLs called gave me everything I needed to know.

Yep, BlackHole exploit kit. That was easy.

I used a Banload sample (MD5: D03B-F6AE5654550A8A0863F-3A265A412) to validate BSA PE analysis capabilities. As expected, they were robust. The File Disassembler utility immediately discerned that the sample was UPX-packed. Figure 4 points out a number of revealing elements.

Network services
Looks for an Internet connection.
Connects to "www.ugpag.cd" on port 80.
Connects to "127.0.0.1" on port 1077.
Connects to "80.80.228.86" on port 80.
Connects to "lukastroy.in" on port 80.
Connects to "zdravyou.in" on port 80.
Connects to "91.205.74.23" on port 80.
Connects to "" on port :80.
Connects to "www.youtube.com" on port 80.
Connects to "173.194.33.41" on port 80.
Connects to "s.ytimg.com" on port 80.
Connects to "fpdownload2.macromedia.com" on port 80.
Connects to "204.245.34.129" on port 80.
Connects to "173.194.33.38" on port 80.
Connects to "i2.ytimg.com" on port 80.
Connects to "173.194.33.36" on port 80.
Connects to "127.0.0.1" on port 1111.
Connects to "windowsmedia.com" on port 80.
Connects to "64.4.52.182" on port 80.

Figure 3 – BSA reporting reveals BlackHole URLs.

Changes to filesystem	
Creates file:	C:\HIOMALVM02\mplayer2.exe o File entropy: 7.12185 (89.0231%) o Adobe Malware Classifier: Unknown o MD5 hash: d03bf6ae5654550a8a0863f3a265a412 o VirusTotal detections: o CAT-QuickHeal: Trojan.VB.awov o McAfee: PWS-Banker.dldr!h o TheHacker: Trojan/Downloader.Banload.qlw o K7AntiVirus: Trojan o VirusBuster: Trojan.DL.Banload!0BDxx4lm/QM o NOD32: a variant of Win32/VB.ODV o F-Prot: W32/Trojan-Gypikon-based.DM2!Maximus o Norman: W32/Obfuscated.O!genr o TrendMicro-HouseCall: TROJ_GEN.R45C9LQ o Avast: Win32:Rootkit-gen [Rtk] o eSafe: Win32.TRCrypt.Cfi o ClamAV: Suspect.Trojan.Generic.FD-1 o Kaspersky: Trojan.Win32.VB.awov o BitDefender: Gen:Trojan.Heur.dmKfr5J3jUbii o Emsisoft: Trojan.Win32.VB!IK o Comodo: UnclassifiedMalware o F-Secure: Gen:Trojan.Heur.dmKfr5J3jUbii o DrWeb: Trojan.Siggen3.30928 o VIPRE: Trojan.Win32.Generic!BT

Figure 5 – BSA reporting provides Virustotal results with created file.

Of interest is the fact that a connection is made to

`hxxp://alessandrodertolazzi.hospedagemdesites.ws` (187.45.240.69)

in Brazil with attempts to download `mac.rar`. Banload/Banker commonly originates from Brazil, so this comes as no surprise. This sample is a bit dated so the evilware hosted on Alessandro's site is long gone, but you get idea. If you optimize your BSA reporting options to include *Virustotal* results, the changes to file system section will include all the detections for created files as seen in figure 5.

The opportunities for exploration are many with Buster Sandbox Analyzer, and the fact that it's free and regularly developed is of huge benefit to our community. Among the features you may find noteworthy and useful are BSA's ability to automatically analyze a folder in a batch process as well as dump analyzed processes. BSA has moved to the top of my list for sandbox analysis, plain and simple.

In conclusion

The combined strengths of Sandboxie and Buster Sandbox Analyzer make for a truly powerful combination and invaluable malware analysis platform. There's no reason to not start exploring right away. As always, do be careful playing with live samples and remember to provide feedback to the BSA project; your support is welcome.

Ping me via email if you have questions (russ at holisticinfosec dot org).

Cheers…until next month.

```
OpenService(sens) [c:\hiomalvm02\mplayer2.exe]
LoadLibrary(sensapi.dll) [c:\hiomalvm02\mplayer2.exe]
LoadLibrary(urlmon) [c:\hiomalvm02\mplayer2.exe]
URLDownloadToFile(http://alessandrodertolazzi.hospedagemdesites.ws/2811/mac.rar) [c:\hiomalvm02\mplayer2.e
CreateMutex(Local\!IETld!Mutex) [c:\hiomalvm02\mplayer2.exe]
FreeLibrary(C:\WINDOWS\system32\urlmon.dll) [c:\hiomalvm02\mplayer2.exe]
CreateMutex(Local\c:!documents and settings!malman.hio-66zkdgucpvw!ietldcache!) [c:\hiomalvm02\mplayer2.ex
CreateFile(C:\Documents and Settings\malman.HIO-66ZKDGUCPVW\IETldCache\index.dat) [c:\hiomalvm02\mplayer2.
InternetOpen() [c:\hiomalvm02\mplayer2.exe]
InternetConnect(alessandrodertolazzi.hospedagemdesites.ws) [c:\hiomalvm02\mplayer2.exe]
LoadLibrary(c:\windows\system32\mswsock.dll) [c:\hiomalvm02\mplayer2.exe]
HttpOpenRequest(/2811/mac.rar) [c:\hiomalvm02\mplayer2.exe]
LoadLibrary(mswsock.dll) [c:\hiomalvm02\mplayer2.exe]
LoadLibrary(hnetcfg.dll) [c:\hiomalvm02\mplayer2.exe]
LoadLibrary(c:\windows\system32\wshtcpip.dll) [c:\hiomalvm02\mplayer2.exe]
LoadLibrary(wshtcpip.dll) [c:\hiomalvm02\mplayer2.exe]
GetModuleHandle(ws2_32.dll) [c:\hiomalvm02\mplayer2.exe]
bind(port=0) [c:\hiomalvm02\mplayer2.exe]
connect( 127.0.0.1:1096 ) [c:\hiomalvm02\mplayer2.exe]
CreateMutex(Local\ZonesCounterMutex) [c:\hiomalvm02\mplayer2.exe]
CreateMutex(Local\ZoneAttributeCacheCounterMutex) [c:\hiomalvm02\mplayer2.exe]
CreateMutex(Local\ZonesCacheCounterMutex) [c:\hiomalvm02\mplayer2.exe]
CreateMutex(Local\ZonesLockedCacheCounterMutex) [c:\hiomalvm02\mplayer2.exe]
LoadLibrary(rasadhlp.dll) [c:\hiomalvm02\mplayer2.exe]
LoadLibrary(dnsapi.dll) [c:\hiomalvm02\mplayer2.exe]
connect( 187.45.240.69:80 ) [c:\hiomalvm02\mplayer2.exe]
```

Figure 4 – BSA API logging reveals Banload behavior.

Acknowledgements

—Pedro Lopez, lead developer, Buster Sandbox Analyzer

About the Author

Russ McRee leads the incident management and penetration testing functions for Microsoft's Online Services Security team. He advocates a holistic approach to information security via holisticinfosec.org and volunteers as a handler for the SANS Internet Storm Center. Reach him at russ at holisticinfosec dot org or @holisticinfosec.