User Guide

No-Script Automation Tool (NAT)

Name	**No-Script Automation Tool (NAT) User Guide**
Department	Development
Status	Approved
Classification	Public
Version	v1.4.0
Date	**06 August 2018**

Version Information

Version	Updates	Date	Initials
1.0	Initial version	6 Aug 2018	JM

Document Classification: *Public*

Notice

This document is intended only for the use of the individual or entity to which it is provided and contains information that is privileged, confidential, and may be exempt from disclosure under applicable law. This notice serves as warning that any dissemination, distribution, or copying of this document is strictly prohibited without the express written consent of DFLabs, S.p.A.

Contents

Version Information i

About the No-Scrip Automation Tool 1

System Requirements 1

Usage, Support and Warranty 1

Configuring NAT 2

- Specifying OS and Architecture 3
- Specifying Tool Command Line Arguments 3
- Integrity File 4

Running NAT 7

Appendix A – Suggested Tools 9

Appendix B – Windows Versions 12

Contact DFLabs 13

About the No-Scrip Automation Tool

The No-Script Automation Tool (NAT) was designed to solve the complexity and management issues surrounding scripting multiple tools via batch files or other scripting languages for Windows systems. NAT allows users to run sets of pre-defined and pre-verified tools based on user specified input, pre-defined commands and system properties such as architecture and Windows version.

This user guide provides additional detail on configuring and using NAT.

System Requirements

NAT requires that .NET 4.0 or greater be installed on the host system to function properly. While this dependency should be met on most modern systems, it is possible that a legacy system without .NET 4.0 preinstalled may be encountered.

The.NET 4.0 offline installer can be downloaded from Microsoft at https://www.microsoft.com/en-us/download/details.aspx?id=17718. It is recommended that the offline installer be downloaded and stored on the same media which will be used to run NAT in case a legacy system without .NET 4.0 is encountered.

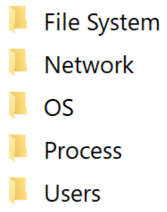
Usage, Support and Warranty

NAT may be used freely for non-commercial purposes, without warrant or support, as outline in the DFLABS GITHUB SOFTWARE PROGRAMS LICENSE AGREEMENT.

Configuring NAT

When NAT is executed, it will look for tools to run in a folder named ‘Tools’ in the same directory as NAT. The Tools folder should contain an additional level of subdirectories which will be used to organize the tools in to functional groups. This will allow the creation of configuration files which will allow NAT to execute subsets of the functional groups, depending on the requirements of the individual execution.

For example, tools may be divided in to functional groups ‘File System’, ‘Network’, ‘OS’, ‘Process’ and ‘Users’ as shown in the following example:



File System
Network
OS
Process
Users

Figure 1 – Tool directory

Once the desired directory structure is created, place each tool in to its respective subdirectory and NAT will execute that tool as directed. Note that while there is no tool naming convention required for NAT, it is *HIGHLY* recommended that each tool be given a name which will distinguish it from any other tools. One common method is to prepend each tool name with ‘NAT_’. Should forensic analysis of the host be required later, this will help distinguish between processes executed as part of the live examination process and processes which were executed through other means. For example, ‘psexec.exe’ may be executed by NAT, however its execution outside of NAT may also be significant. If the executable run by NAT is not renamed, it may be impossible to distinguish between authorized and unauthorized use of this tool or critical forensic artifacts may be altered.

A single configuration file named ‘default.ini’ is required for NAT to execute successfully. This file should be located in the same directory as NAT. This file should contain a list of the subdirectories, one per line, that should be executed by default by NAT. For example, a default configuration file to execute tools in each of the subdirectories shown in Figure 1 would be as follows:

```
File System
Network
OS
Process
Users
```

Figure 2 – Sample ‘default.ini’ file

Additional configuration files may also be created to run additional subsets of tools. These files should be named as they will be called from the command line and given a ‘.ini’ extension. For example, to collect only OS and process information, a second configuration file may be created specifying only the OS and Process subdirectories, and might be named ‘os_process.ini’. NAT would then be executed with the option ‘-i os_process.ini’. This command line option will be discussed in additional detail in subsequent sections.

Specifying OS and Architecture

Some tools may only be applicable to a given architecture or a subset of Windows operating systems. When NAT executes, it will determine both the Windows OS version and the CPU architecture of the host. To restrict a tool to a given architecture or subsets of Windows operating systems additional subfolders may be created within each functional group folder.

Architecture is specified with subdirectories named ‘x86’ or ‘x64’.

Windows OS version is specified with subdirectories named by either the specific version, or an inclusive range of Windows OS versions; for example, ‘6.1’ or ‘6-10.0’. Windows OS version numbers should be specified using the Microsoft version number; for example, Windows 7 is version 6.1. A list of Windows OS common names and version numbers, current as of the creation of this user guide, is available in Appendix B. Appendix B also includes a link to this information on Microsoft’s website.

Architecture and Windows OS version subdirectories may be nested, however there cannot be multiple instances of either in the path to a single tool. For example, ‘Network\6-10.0\x64’ would be valid, however ‘Network\6-10.0\x64\8.2-10.0’ would not be valid.

Specifying Tool Command Line Arguments

Each tool may be executed with one or more sets of command line options. These command line arguments are defined in a text file named ‘*<tool name>*.cmd’ which is stored in the same path as the tool itself. For example, to pass command line arguments to the tool “Network\ping.exe”, save them in a text file named “Network\ping.exe.cmd”.

This text file should include the command line arguments only, not the name of the tool as well. For example, to run the command “ping.exe -t -f 127.0.0.1” the ping.exe.cmd file should contain only “-t -f 127.0.0.1”.

Command line arguments should be entered one per line and end with a blank line. The tool will be executed once for each set of command line arguments that are provided and the output from each execution will be stored in a separate file.

There are three variables which may be used in the .cmd file; %NOOUT%, %OUTDIR% and %SYSROOT%. These variables will be replaced with the appropriate values at runtime. The following table details the use of each of the three variables.

Command Line Argument File Variables

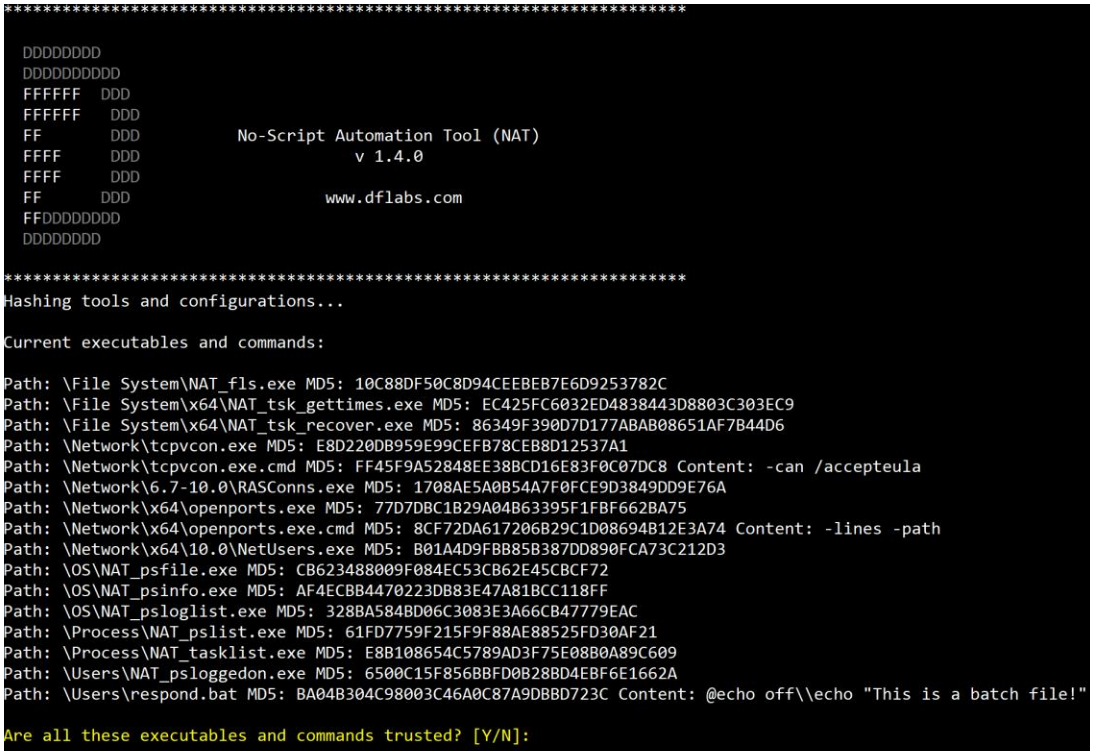
Variable	Use
%NOOUT%	By default, output from each tool will be written to a text file in the output directory specified at runtime. To prevent this for a specific tool, use the variable %NOOUT% as the sole argument in the .cmd file, or at the end of each line of command line arguments if other arguments are specified. This can be used when the output directory is specified as part of the command line arguments for the tool.
%OUTDIR%	To specify the output directory as part of the command line arguments, use the variable %OUTDIR% in place of the output directory. This variable will be dynamically replaced with the correct output directory each time the tool is executed. For example, "-o %OUTDIR%\output.txt" for mytool.exe will result in the command “mytool.exe -o <selected output directory>\output.txt being executed at runtime.
%SYSROOT%	To specify the Windows system root, which may vary between hosts, use the %SYSROOT% variable.

Table 1 – Command line argument file variables

Integrity File

Because NAT is running multiple tools with administrative credentials, there is a risk of executing unintended or malicious processes. To reduce this risk, NAT allows the creation of an integrity file, which will be used to verify that none of the tools or commands have been altered. If no integrity file is created, NAT will warn the user of the potential security implications during each execution and the user must choose to continue.

Once the tools have been added to the correct subdirectories, the appropriate command line argument files have been created and NAT is ready to be used in production, an integrity file can be created by executing NAT with the ‘-c’ option.   This will cause NAT to perform an inventory of executables (.exe, .com and .bat) as well as command line argument files and config files. The path, MD5 hash value and content (for non-compiled files) of each file will be displayed for review. Once it has been confirmed that all files are correct and authorized, the integrity file is created by acknowledging that all executables and commands are trusted, then entering and reentering a password.

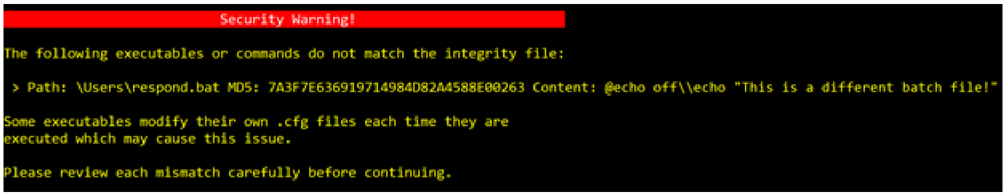


```
**********************************************************************

  DDDDDDDD
  DDDDDDDDDD
  FFFFFF  DDD
  FFFFFF   DDD
  FF       DDD          No-Script Automation Tool (NAT)
  FFFF     DDD                      v 1.4.0
  FFFF     DDD
  FF      DDD                    www.dflabs.com
  FFDDDDDDDD
  DDDDDDDD

**********************************************************************
Hashing tools and configurations...

Current executables and commands:

Path: \File System\NAT_fls.exe MD5: 10C88DF50C8D94CEEBEB7E6D9253782C
Path: \File System\x64\NAT_tsk_gettimes.exe MD5: EC425FC6032ED4838443D8803C303EC9
Path: \File System\x64\NAT_tsk_recover.exe MD5: 86349F390D7D177ABAB08651AF7B44D6
Path: \Network\tcpvcon.exe MD5: E8D220DB959E99CEFB78CEB8D12537A1
Path: \Network\tcpvcon.exe.cmd MD5: FF45F9A52848EE38BCD16E83F0C07DC8 Content: -can /accepteula
Path: \Network\6.7-10.0\RASConns.exe MD5: 1708AE5A0B54A7F0FCE9D3849DD9E76A
Path: \Network\x64\openports.exe MD5: 77D7DBC1B29A04B63395F1FBF662BA75
Path: \Network\x64\openports.exe.cmd MD5: 8CF72DA617206B29C1D08694B12E3A74 Content: -lines -path
Path: \Network\x64\10.0\NetUsers.exe MD5: B01A4D9FBB85B387DD890FCA73C212D3
Path: \OS\NAT_psfile.exe MD5: CB623488009F084EC53CB62E45CBCF72
Path: \OS\NAT_psinfo.exe MD5: AF4ECBB4470223DB83E47A81BCC118FF
Path: \OS\NAT_psloglist.exe MD5: 328BA584BD06C3083E3A66CB47779EAC
Path: \Process\NAT_pslist.exe MD5: 61FD7759F215F9F88AE88525FD30AF21
Path: \Process\NAT_tasklist.exe MD5: E8B108654C5789AD3F75E08B0A89C609
Path: \Users\NAT_psloggedon.exe MD5: 6500C15F856BBFD0B28BD4EBF6E1662A
Path: \Users\respond.bat MD5: BA04B304C98003C46A0C87A9DBBD723C Content: @echo off\\echo "This is a batch file!"

Are all these executables and commands trusted? [Y/N]:
```

Figure 3 – Integrity check creation

This information is encrypted with the specified password using 256-bit AES encryption and stored in a file named 'integrity.ck'.

Once an integrity file is created, NAT will prompt for a password on each execution. If the password is incorrect or the file has been altered in some way, an error message will be displayed. Once decrypted with the correct password, NAT will perform the same inventory which was done when the file was created and compare it to the integrity file. NAT will display any variations from the integrity file to be reviewed by the user.



```
                    Security Warning!

The following executables or commands do not match the integrity file:

 > Path: \Users\respond.bat MD5: 7A3F7E636919714984D82A4588E00263 Content: @echo off\\echo "This is a different batch file!"

Some executables modify their own .cfg files each time they are
executed which may cause this issue.

Please review each mismatch carefully before continuing.
```

Figure 4 – Integrity check mismatch

Mismatches in the integrity file are not always malicious, but should be reviewed carefully before continuing with the execution. In some cases, certain tools may alter their configuration files when they execute. This will cause NAT to detect a mismatch.

If integrity file is deleted, a security warning will be displayed indicating that an integrity file is not present.

It is possible to bypass verifying the integrity of the tools and commands once an integrity file has been created by executing NAT with the ‘-x’ option. However, this is not recommended and a security warning will be displayed.

Running NAT

NAT required administrative privileges and should be executed from an administrative command prompt.

NAT may be executed with the following command line arguments:

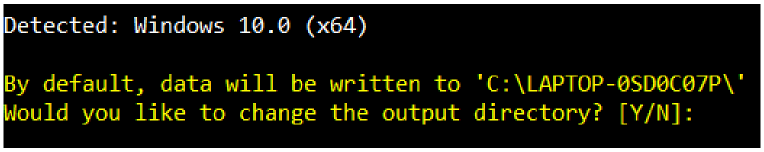
Command Line Argument

Argument	Use
-h	Display help menu and exit
-x	Bypass integrity check
-c	Create integrity check file
-I <file>	Use the specified .ini file (default is default.ini)

Table 2 – Command line argument file variables

For more information regarding integrity checks and ini file configuration, please see the previous sections titled “Integrity File” and “Configuring NAT”, respectively.

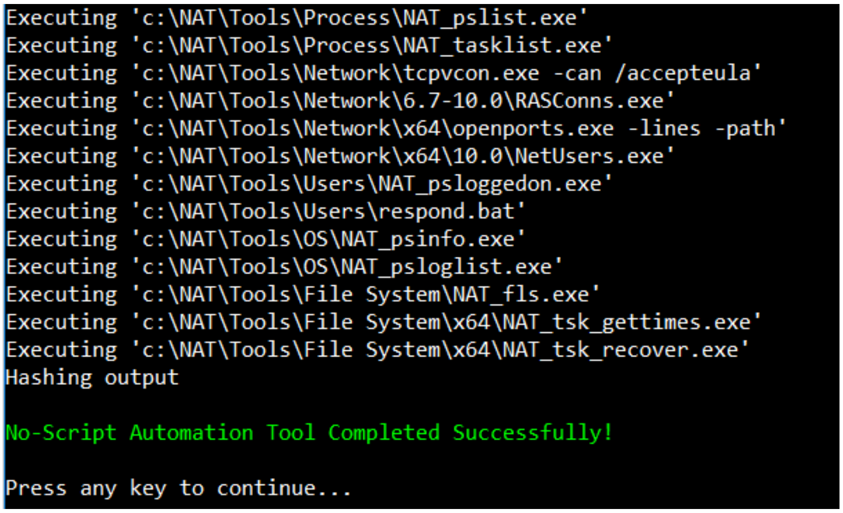
Once any warnings are accepted, NAT will display the detected operating system and ask the user to confirm the directory to which all output will be written. By default, output will be written to a new folder named according to the machine name in the root of the drive on which NAT is executed from. Users may specify an alternate output location at this point.



```
Detected: Windows 10.0 (x64)

By default, data will be written to 'C:\LAPTOP-0SD0C07P\'
Would you like to change the output directory? [Y/N]:
```

Figure 5 – Output Directory

Once the output directory is confirmed, NAT will begin running the provided tools and saving the tool output to the specified output directory. The name of each tool, along with any command line options, is shown as each tool is run.



```
Executing 'c:\NAT\Tools\Process\NAT_pslist.exe'
Executing 'c:\NAT\Tools\Process\NAT_tasklist.exe'
Executing 'c:\NAT\Tools\Network\tcpvcon.exe -can /accepteula'
Executing 'c:\NAT\Tools\Network\6.7-10.0\RASConns.exe'
Executing 'c:\NAT\Tools\Network\x64\openports.exe -lines -path'
Executing 'c:\NAT\Tools\Network\x64\10.0\NetUsers.exe'
Executing 'c:\NAT\Tools\Users\NAT_psloggedon.exe'
Executing 'c:\NAT\Tools\Users\respond.bat'
Executing 'c:\NAT\Tools\OS\NAT_psinfo.exe'
Executing 'c:\NAT\Tools\OS\NAT_psloglist.exe'
Executing 'c:\NAT\Tools\File System\NAT_fls.exe'
Executing 'c:\NAT\Tools\File System\x64\NAT_tsk_gettimes.exe'
Executing 'c:\NAT\Tools\File System\x64\NAT_tsk_recover.exe'
Hashing output

No-Script Automation Tool Completed Successfully!

Press any key to continue...
```

Figure 6 – NAT Execution

A log file named "_NAT.log" will be created in the root of the output directory. This will contain a detailed log of all actions taken by NAT, including each tool run, the tool's MD5 hash value, any command line arguments, and other information.

Any errors encountered while running an individual tool will result in either an error message shown in the console window, a popup box, or both, depending on the tool being run. Any failures of an individual tool should not affect the execution of the other tools.

Once the last tool has been run, NAT will hash each output file in the output directory and save this information in a file named "_MD5.txt" in the root of the output directory.

Appendix A – Suggested Tools

AV Tools

Name	URL
Emsisoft Commandline Scanner	emsisoft.com/en/software/cmd/

Disk Imaging Tools

Name	URL
FTK Imager Lite	http://accessdata.com/product-download/digital-forensics/ftk-imager-lite-version-3.1.1

File System Tools

Name	URL
fls	http://www.sleuthkit.org/sleuthkit/download.php
lads	http://heysoft.de
volume_dump	http://www.gmgsystemsinc.com/fau/

Memory Tools

Name	URL
Dumpit	https://www.comae.io/

Network Tools

Name	URL
DumpWin	https://www.niiconsulting.com/innovation/security-tools.html
fport	http://www.mcafee.com/us/downloads/free-tools/fport.aspx
promiscdetect	http://ntsecurity.nu/toolbox/promiscdetect/
psfile	https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx

RASConns	http://www.westmesatech.com/wast.html
tcpvcon	https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx
DumpWin	https://www.niiconsulting.com/innovation/security-tools.html

OS Tools

Name	URL
autorunsc	https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx
DumpWin	https://www.niiconsulting.com/innovation/security-tools.html
gplist	http://ntsecurity.nu/toolbox/gplist/
ls	https://u-tools.com/msls
ntlast	http://www.mcafee.com/us/downloads/free-tools/ntlast.aspx
OpenedFilesView	http://www.nirsoft.net/utils/opened_files_view.html
psinfo	https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx
psloggedon	https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx
psloglist	https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx
psservice	https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx
reg	https://technet.microsoft.com/en-us/library/cc732643(v=ws.11).aspx
systeminfo	https://technet.microsoft.com/en-us/library/bb491007.aspx
USBDeview	http://www.nirsoft.net/utils/usb_devices_view.html

PROCESS TOOLS

Name	URL
cmdline	http://www.easexp.com/cmdline/
CurrProcess	http://www.nirsoft.net/utils/cprocess.html
DumpWin	https://www.niiconsulting.com/innovation/security-tools.html
handle	https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx
pslist	https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx

Appendix B – Windows Versions

Common Name	Version Number
Windows 10	10.0*
Windows Server 2016	10.0*
Windows 8.1	6.3*
Windows Server 2012 R2	6.3*
Windows 8	6.2
Windows Server 2012	6.2
Windows 7	6.1
Windows Server 2008 R2	6.1
Windows Server 2008	6
Windows Vista	6
Windows Server 2003	5.2
Windows XP 64-Bit Edition	5.2
Windows XP	5.1

** For applications that have been manifested for Windows 8.1 or Windows 10. Applications not manifested for Windows 8.1 or Windows 10 will return the Windows 8 OS version value (6.2).*

Source: https://msdn.microsoft.com/en-us/library/windows/desktop/ms724832(v=vs.85).aspx

Contact DFLabs

Technical Support

DFLabs Customer Support Team
Tel: +39 0373-82416
Email: tech-support@dflabs.com

Sales

VP of Sales
Tel: +1 201-579-0893
Email: sales@dflabs.com

DFLabs Headquarters

DFLabs S.p.A
Address: Via Pietro Donati, 16
26013 Crema (CR)
Tel: +39 0373-82416
Web: www.dflabs.com
VAT: 04547850968

DFLabs Lab

DFLabs S.p.A
Address: Via Bergognone, 31
20144 Milano
Tel: +39 0373-82416