Engineering Development Group

Emotional Simian v2.1
User Manual

Rev. 1.0
April 15, 2013

Change Log

Doc Rev	Doc Date	Rev By	Change Description	Reference	Authority/ Approval Date
DRAFT					

Table of Contents

1. Scope

This document establishes the user manual for Emotional Simian v2.1.

1.1 System Overview and Description

Emotional Simian is a suite of tools which provide the ability to propagate from a primary host to secondary downstream targets via USB thumb drives, or configure a local thumbdrive to execute a configured dll to do all the Payload functionality.

1.2 Assumptions and Constraints

Access to the primary host must already exist, or access to the thumbdrive you would like to whack Furthermore, it is up to the operator to set up persistence for *ES Server(64).exe*.

Any payloads carried downstream must be dropped to run

1.3 Conventions

None.

2. Applicable Documents

The following documents, of the exact issue shown, form a part of this document to the extent specified herein. In the event of a conflict between the documents referenced herein and the contents of this document, the contents of this document will be considered binding. The following documents may be found in the EDG/AED share:

- Emotional Simian V2.1 UserGuide.doc
- Emotional Simian V2.1 TDR.ppt

3. System Description

3.1 Technical References

MD5 Values:

ES Setup.exe
Extract WM Files.exe
Get SN.exe
KeyGen.exe
Post Processor.exe
Whack_Thumbdrive.exe
ES Server.exe
ES Server64.exe
Dll_Payload.dll
Dll_Payload64.dll
Emotional_Simian_Config.exe

3.2 System Concepts and Capabilities

The following is a quick overview of the above mentioned pieces.

- **Emotional_Simian_Config.exe** – This is setup GUI used by the user to create the .cfg file to be laid down on the primary host.
- **./Internal/ES Setup.exe** – This tool is called by *Emotional_Simian_Config.exe* and is used package the .cfg file.
- **./Internal/KeyGen.exe –** This tool creates a public private key. This program is used by *Emotional_Simian_Config.exe*.
- **./Internal/Post Processor.exe** – This is used to decompress, decrypt, and piece together any collected files.
- **./Internal/ES Server.exe** – To be laid down on a 32bit primary host. This tool runs in the background and watches for the insertion of a white list drive. Upon introduction ES Server will infect the drive with the required files.
- **./Internal/ES Server64.exe** – To be laid down on a 64bit primary host. This tool runs in the background and watches for the insertion of a white list drive. Upon introduction ES Server will infect the drive with the required files.
- **./Internal/Extract WM Files.exe** – This tool can extract files stored on the covert storage of the thumb drive to a folder of your choosing.
- **./Internal/Get SN.exe –** This tool can be put on a target to find the serial number for the thumb drive you are targeting. (This can also be done by looking at the registry files.
- **./Internal/Dlls/DllPayload64.dll** - The 64bit version of the Emotional Simian dll payload.
- **./Internal/Dlls/DllPayload.dll** – The 32bit version of Emotional Simian dll payload.

- **./Internal/WhackDrive.exe** – Used by the GUI to weaponize a local thumbdrive.

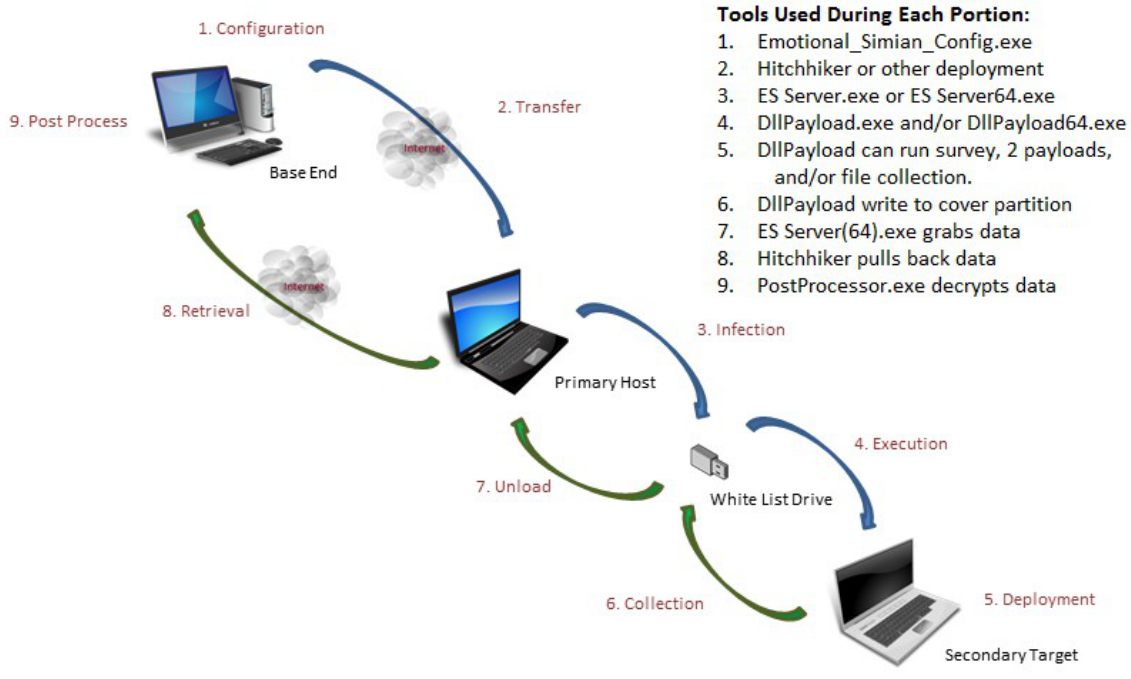
3.3 Prerequisites

- The configuration must be done on a Windows 7 machine.
- Access to the primary host must be gained through other means.
- If not doing a remote operation, you must be able to access thumbdrive in question.
- The primary host must be running Windows XP or later.
- Server.exe or Server64.exe must be run with administrative privileges.
- Persistence is to be set up by the operator.

4. Operation

4.1 Quick Overview

The following is a quick overview of the end to end process for the Emotional Simian suite of tools and what tools are used during each portion of the operation. A more detailed explanation of each step will come later.



1. **Configuration**: Done by the operator on a Windows 7 machine. During this stage *Emotional_Simian_Config.exe* is run and the desired settings are filled in. The tool will then generate the required .cfg file, this file contains all the payloads and configuration information required for *ES Server(64).exe*.
2. **Transfer**: *ES Server(64).exe* (depending what bitness your target is) is transferred downstream with the configuration (.cfg) file via some deployment method. These two files can be put wherever the operator desires and named whatever the operator desires, but they must be named the same e.g. clevername.exe and clevername.cfg. *ES Server(64).exe* should then be executed and persistence must be setup by the operator.
3. **Infection**: Upon introduction of a white list drive *ES Server(64).exe* will place the *DllPayload(64).dll* with the appropriate lnk files on the thumb drive. The lnk files will cause *DllPayload(64).exe* to run when the user sees the lnk file in explorer.
4. **Execution**: If the white list drive is introduced to the proper OS for the generated lnk files and viewed in an explorer window; *DllPayload(64).dll* will gain execution. Immediately following execution, *DllPayload(64).dll* will launch itself as rundll32.exe, attempt to escalate privileges and begin to check the configured kill date, master black list, and if the computer has been seen before. Note execution will always happen but if the blacklist, kill date, or if the computer has been whacked before; the program will immediately quit.
5. **Deployment:** If the initial requirements are met, ES will attempt to deploy the payloads based upon the conditions configured earlier. A unique black list for each payload will be checked and a decision will be made based upon if the computer connects to the internet or not as well as if the process has administrative privileges. The payload also can conduct a survey, or collect files based on the configurations set by *Emotional_Simian_Config.exe.*
6. **Collection:** If *DllPayload(64).dll* does collects files and/or a system survey. These files will be chunked up and written back to the covert partition that exists on the thumb drive.
7. **Unload:** Upon return to the primary host, *ES Server(64).exe* will pull any collected files off the covert partition and store them as hidden system files in the folder designated during configuration. Default is right beside *ES Server(64).exe.*
8. **Retrieval:** The operator will then pull the desired files from the Primary Host and place them on the Base End for post processing.
9. **Post Process:** To decrypt, decompress, and stitch the files back together *PostProcess.exe* is ran. The recreated files will be dumped into the desired location.

4.2 Configuring Emotional Simian

To configure Emotional Simian the operator must run *Emotional_Simian_Config.exe* which must be run on XP SP3 or later, preferably on Windows 7. *Emotional_Simian_Config.exe* will generate a ES Server.exe and ES Server.cfg file which will be laid down on the Primary Host, and a XML file of all the configurations from the configuration tool along with the public private keys.

NOTE: DO NOT LOSE THE PRIVATE KEY! IF THIS IS LOST YOU WILL NOT BE ABLE TO DECRYPT ANY FILES COLLECTED.

4.3 Left behind data

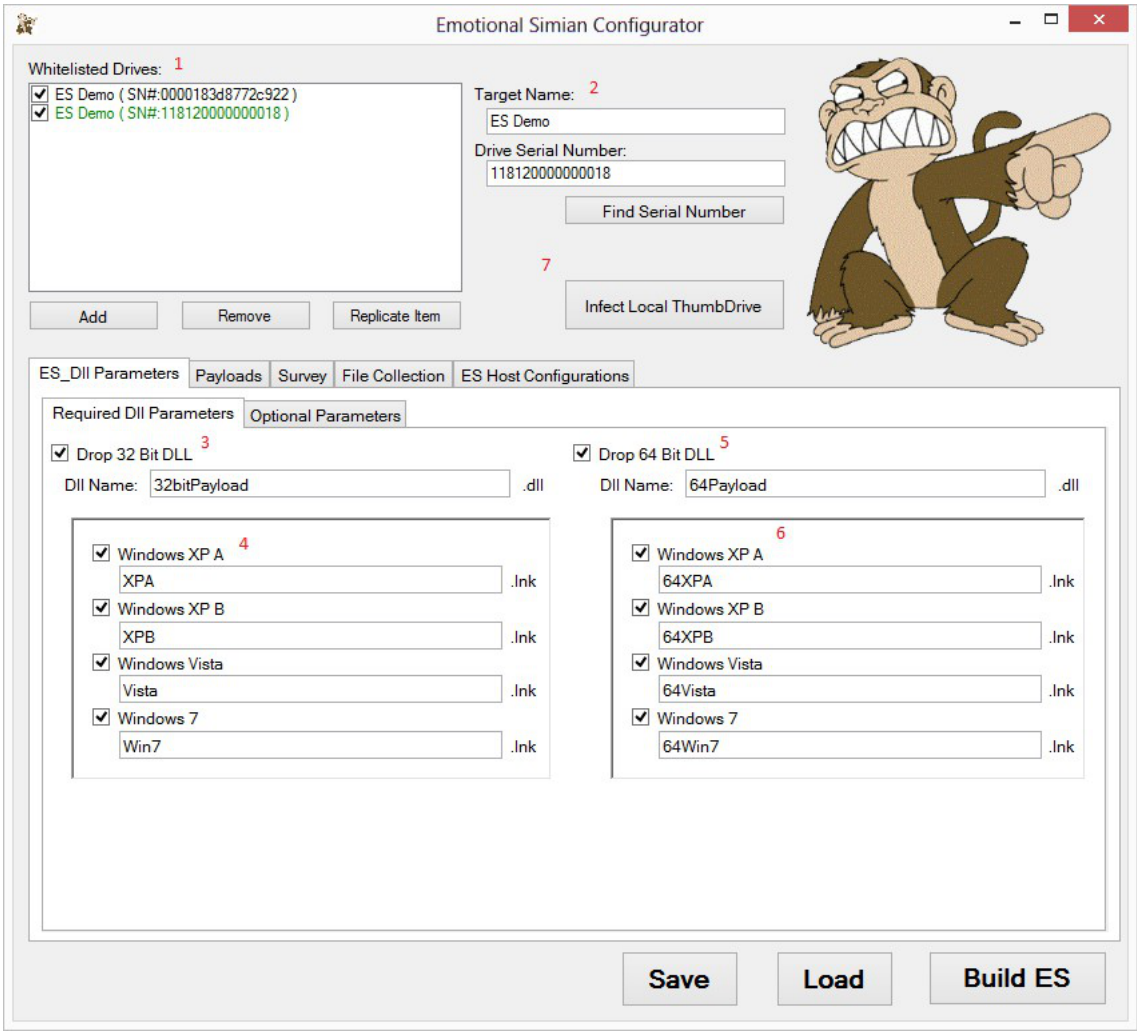
The following things are left behind or altered by ES Server:

1. ES_Server.exe -> wherever you placed it
2. ES_Server.cfg -> wherever you placed it
3. Collection Folder -> Created after seeing the first thumbdrive placed wherever you configured it to drop in field **Collection Directory on Primary Host Target.**

The following files and Reg keys are created by ES Dll Payload:

1. Reg Key -> HKCU\Software\Microsoft\Active Setup
    a. Value: Parameters
2. Reg Key If persist Completed Reg key is checked: HKLM\Software\Microsoft\Active Setup
    a. Value: Some random GUID
3. Reg Key if not persistent: HKLM\Software\Microsoft\MNU
    a. Value: Some random GUID
4. Hash File: Located wherever you configured in field **Hash Collection Directory Location on Secondary Target**
5. Payloads: Wherever you drop them.

Main Form:

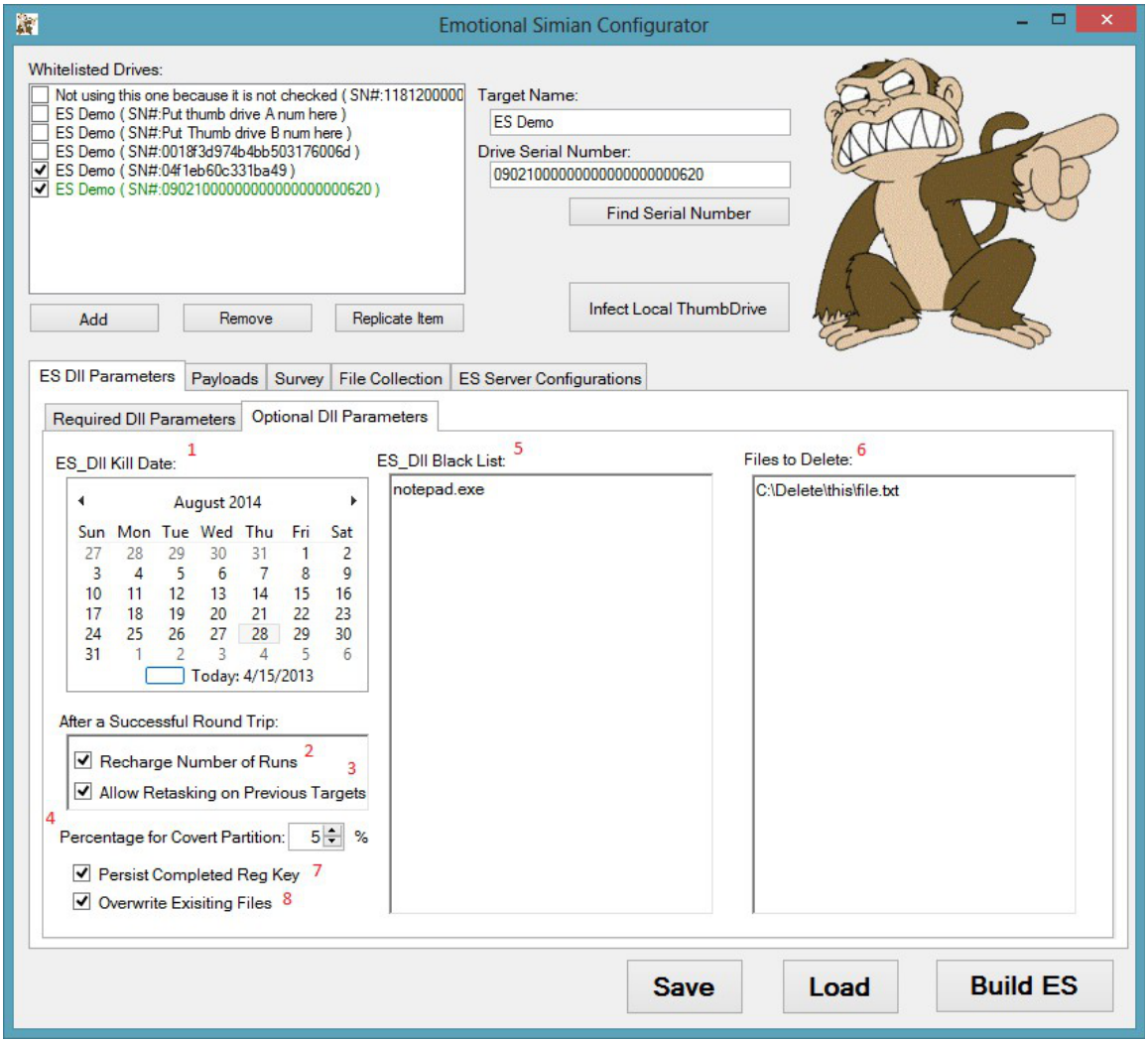


1. **Whitelisted Drives:** This is a list of drives that have been configured for use. If a drive is not checked it will be disabled and will be saved in the XML file but will not be placed in the configuration file. Checked thumb drives may not share the same serial number. The same name with multiple serial numbers is allowed. The drive highlighted in green is the one selected. Checked and selected are two different things. Checked means it will be included in the configurations, highlighted means that the changes that you want to make will happened to the selected member. This theme repeats throughout the entire configuration program.
2. **Target Name:** This is for the operators purposes. This name will only be in the configuration and not the configuration file (.cfg). It helps the operator keep track

of who owns a particular thumb drive without looking up the serial number in your data. I.e. JQJBADGUY/1 (SN# 889238923484)

3. **Drop 32 Dll:** You may fill out the Dll Name: field but not check this box. Checking the box will allow the *ES Server(64).exe* to put down a 32 bit Dll on the listed thumb drive. Each Whitelisted drive may have different names and configurations for the dlls and lnk files or all may have the same names. The dll must end with an extension. It does not need to .dll, but it has to have .something.
4. **32 bit lnk files:** These link files are required to get the dll to run. Fields may be filled out but not checked. The information will be saved in the XML file but *ES Setup.exe* will not put the names of the lnk files in the configuration file.
5. **Drop 64 Dll:** You may fill out the Dll Name: field but not check this box. Checking the box will allow the *ES Server(64).exe* to put down a 64 bit Dll on the listed thumb drive. Each Whitelisted drive may have different names and configurations for the dlls and lnk files or all may have the same names. The dll must end with an extension. It does not need to .dll, but it has to have .something.
6. **64 bit lnk files:** Fields may be filled out but not checked. The information will be saved in the XML file but *ES Setup.exe* will not put the names of the lnk files in the configuration file.
7. **Infect Local Thumbdrive:** This button allows you to instantly infect a thumbdrive that is plugged into your system instead of running ES Server.exe. First click the Whitelisted drive configuration you want and then click this button. *The Pre and Post Build executable/batch scripts will run when you click this button.*

4.3.1 Optional Dll Parameters:

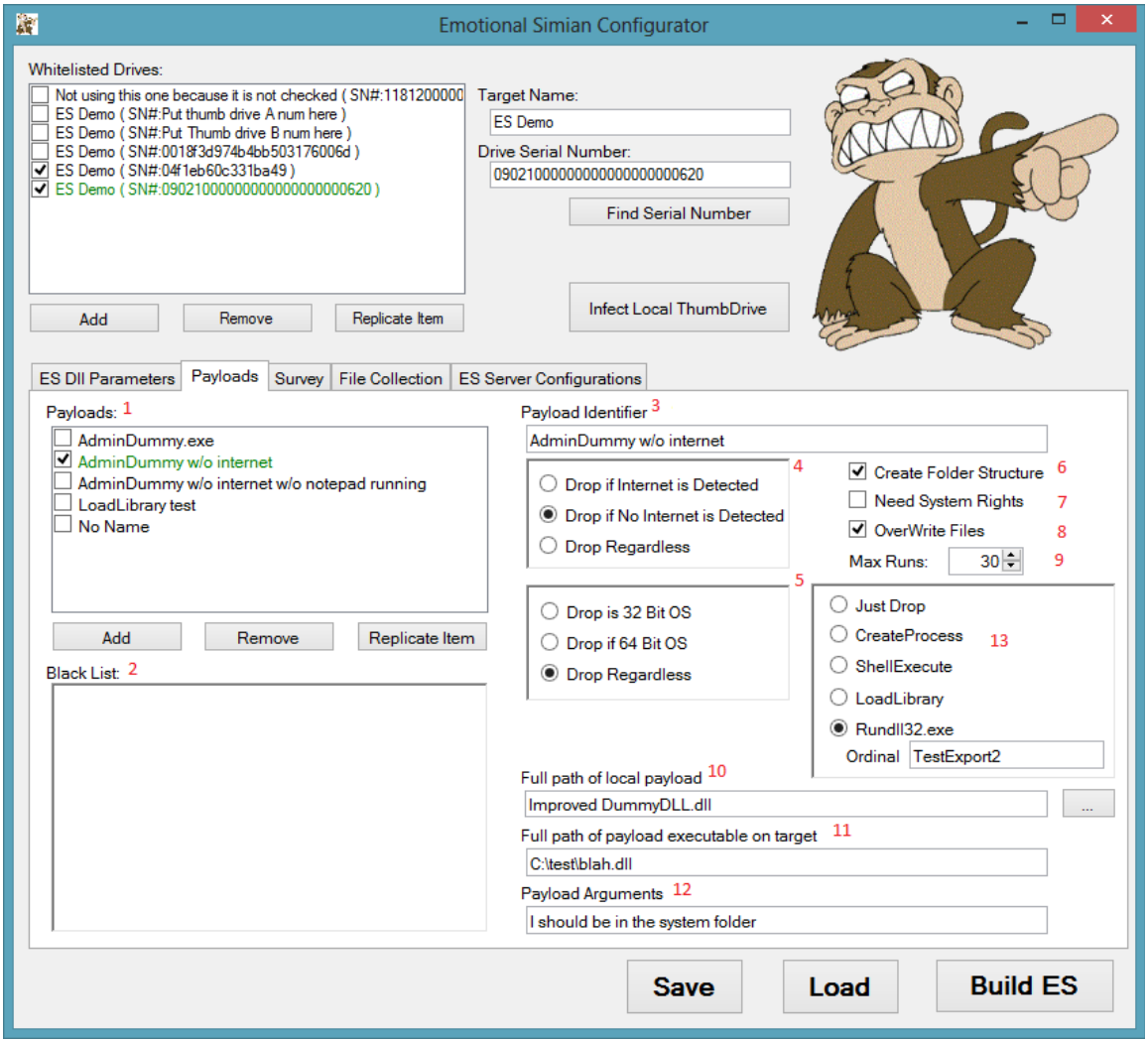


1. **ES_Dll Kill Date:** *DllPayload(64).dll* will exit if this date is surpassed. Default is two years into the future.
2. **Recharge Number of Runs:** After an infected thumb drive comes back to the Primary Host computer, if checked, *ES Server(64).exe* will put the number of surveys and payload runs back to their original amounts.
3. **Allow Retasking on Previous Targets:** Once DllPayload(64).dll whacked a computer, it places a GUID in Registry under HKLM\Software\ Microsoft\Active Setup or HKLM\Software\ Microsoft\MNU depending whether you checked the **Persist Completed Reg Key** box. If the thumb drive has not made it back to the Primary Host running *ES Server(64).exe*, the thumb drive will not run on that

computer. Once the thumb drive comes back to the Primary Host, if checked, *ES Server(64).exe* will change the GUID located in *DllPayload(64).dll*. This will allow *DllPayload(64).dll* to rewhacked downstream targets.

4. **Percentage for Covert Partition:** This is how much covert storage you want to allocate on this thumb drive. Warnings: You are not guaranteed the amount you request. *ES Server(64).exe* will attempt to give you as much as you asked for, but if it is not possible *ES Server(64).exe* will give you the maximum it can without going over the percentage you specify. *Taking more than 10% of the drive could be noticeable by the user. (Default is 5%). 0% will enable ES server to not put a covert storage on the drive, however with 0% you will not be able to collect a survey or files, but it will allow you run payloads.*
5. **Black List:** This is a list of executables that signal a no go for DllPayload(64).exe. *Note Black List is an 'or' condition. Meaning if any of the conditions set forth are met it will not deploy the payload.*
6. **Files to Delete:** After everything has been accomplished (survey, file collection, and\or dropping of the payloads) *DllPayload(64).dll* will attempt to delete these files. *These files have to be absolute paths. If the file is in use, then the file will not be deleted.*
7. **Persist Completed Reg Key:** If this box is not checked, the reg key that indicates the dll has fired will be deleted on reboot. Checked, this reg key will persist a reboot. *You would use this if you needed to run a payload once per reboot.*
8. **Overwrite Existing Files:** If this box is checked, and the Dll or lnk files exist, they will be overwritten. However, once the thumbdrive has been whacked, the files will not be replaced regardless. So, if Target/Owner of the thumbdrive deletes the files, the files will not show up again unless you lay down a new configuration file.

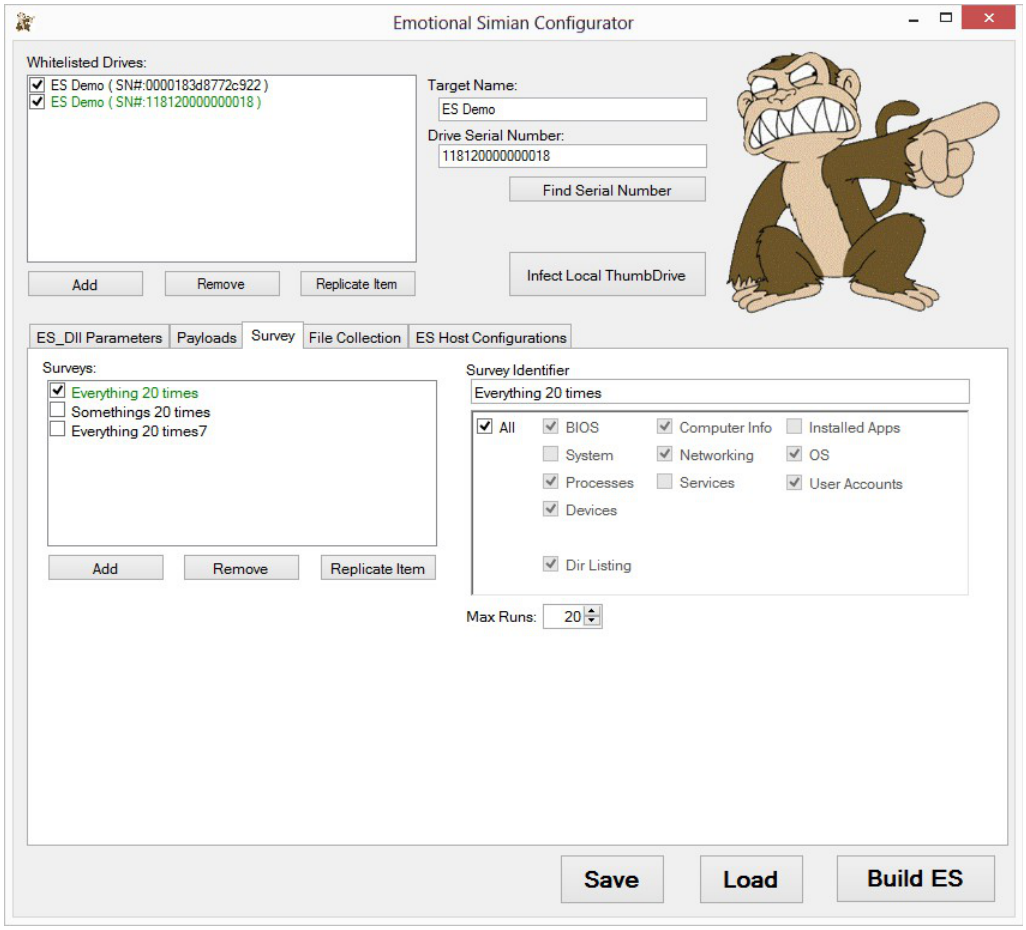
4.3.2 Payload Tab:



1. **Payloads:** You may up to ten payloads in this box, but only three payloads may be checked for each thumb drive. Payload highlighted in green is the current selected payload.
2. **Black List:** This is a list of executables that signal a no go for payload. *Note Black List is an 'or' condition. Meaning if any of the conditions set forth are met it will not deploy the payload.*
3. **Payload Identifier:** This is for the operators purposes. This name will only be in the XML configuration file and not the .cfg file. It gives the operator a quick description of the payload associated with each thumb drive. Only checked members will be associated with each Whitelisted drive.

4. **Drop if No Internet/ Internet:** Self-explanatory.
5. **Drop if 32/64 bit:** Self-explanatory.
6. **Create Folder Structure:** If the path defined in **Full path of payload executable on target** does not exist, if checked; *DllPayload(64).dll* will create the folder structure.
7. **Need System Rights:** Only checked this if your payload absolutely needs System (not Admin) rights.
8. **OverWrite Files:** If this is checked, your file will replace a file of the same name.
9. **Max Runs:** The max number of runs the payload can drop. The number of runs is stored as the creation time. If *DllPayload(64)*.dll cannot modify these times it will not drop and run the payload. This is a simple way of keeping *DllPayload(64).dll* from working from a CD or write blocker. *Note: If you are dropping two dlls 64 bit and 32 bit the total maximum number of runs has now been double since each dll can run X(max_runs) number of times.*
10. **Full path of payload executable:** Path to where your payload you want embedded in the .cfg is located.
11. **Full path of payload executable on target:** Path where the payload will be created on target. Payloads will not be overwritten. If the executable already exist the payload will not run.
12. **Payload Arguments:** Arguments that will be fed into the payload at run time
13. **Run Payload as:** You may choose how your executable is drop and ran.

4.3.3 Survey Tab:



1. **Surveys:** This is the list of surveys they you have created. Only one survey per thumb drive is allowed.
2. **Survey Identifier:** This is for the operators purposes. This name will only be in the XML configuration file and not the .cfg file. It gives the operator a quick description of the survey associated with each thumb drive. Only checked members will be associated with each Whitelisted drive.
3. **Survey checkboxes:** The names are intuitive. Some of the surveys incorporate multiple surveys as seen below in the break down between the configuration and the actual progam
    a. **Hex equivalent for surveys in ES Config.exe**
        i. ALL = 0xFFFFFFF
        ii. OS = 0x0000000F
        iii. BIOS = 0x00000010
        iv. Computer Info = 0x00000020

v.	System Survey =	0x000007FF
vi.	Devices =	0x00003B80
vii.	Networking =	0x001FC000
viii.	Processes =	0x00200000
ix.	Services =	0x00400000
x.	Installed Apps =	0x03800000
xi.	User Accounts =	0x04000000

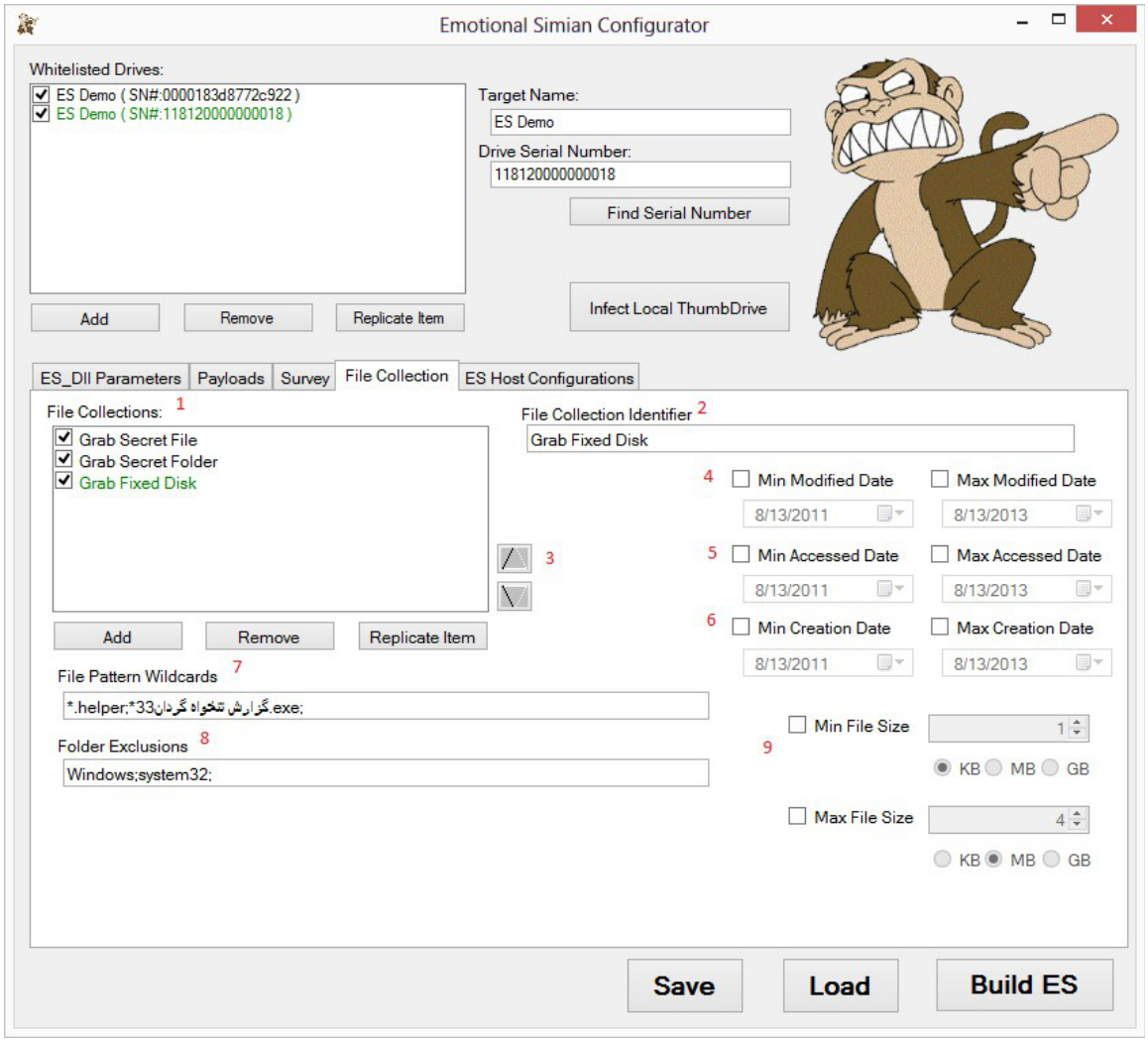
b. Bitwise Flags for Surveys

i.	SYSTEM_DATETIME	1 << 0
ii.	SYSTEM_LOCAL_INFO	1 << 1
iii.	SYSTEM_OS_INFO	1 << 2
iv.	SYSTEM_COMPNAME	1 << 3
v.	SYSTEM_BIOS	1 << 4
vi.	SYSTEM_COMPINFO	1 << 5
vii.	SYSTEM_PROCESSOR	1 << 6
viii.	SYSTEM_SERIAL_PORTS	1 << 7
ix.	SYSTEM_PARRALLEL_PORTS	1 << 8
x.	SYSTEM_USB_INFO	1 << 9
xi.	SYSTEM_REGISTRY_INFO	1 << 10
xii.	DEVICE_CD_INFO	1 << 11
xiii.	DEVICE_PRINTER_INFO	1 << 12
xiv.	DEVICE_DISK_INFO	1 << 13
xv.	NETWORK_IE_CONN_PREFS	1 << 14
xvi.	NETWORK_ADAPTER_INFO	1 << 15
xvii.	NETWORK_ROUTE_INFO	1 << 16
xviii.	NETWORK_ARP_CACHE	1 << 17
xix.	NETWORK_NET_CONNS	1 << 18
xx.	NETWORK_NET_BIOS_INFO	1 << 19
xxi.	NETWORK_PROTOCOLS	1 << 20

xxii.	PROCESS_INFO	1 << 21
xxiii.	PROCESS_SERVICES	1 << 22
xxiv.	APPLICATION_INFO	1 << 23
xxv.	APPLICATION_STARTUP_INFO	1 << 24
xxvi.	APPLICATION_ENVIROMENT_INFO	1 << 25
xxvii.	USER_INFO	1 << 26

4. **Directory Listing:** These boxes allow you to choose to gather more information about each file in your directory listing.
5. **Max Runs:** This is the maximum time you may run a survey for a *DllPayload(64).dll*.

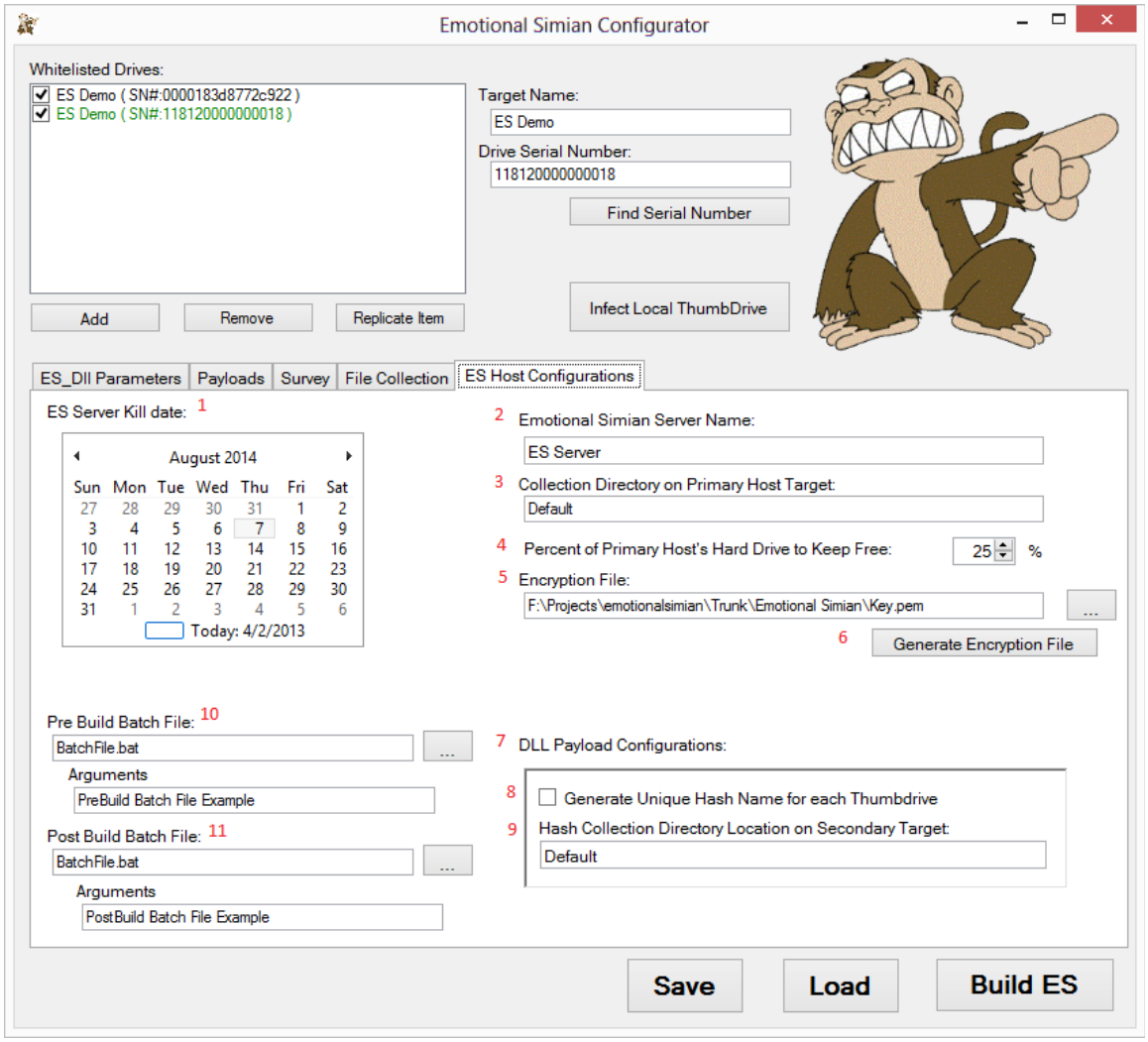
4.3.4 File Collection Tab:



1. **File Collections:** This is the list of file collection they you have created. Unlimited amount of file collections are allowed. Only checked members will be associated with each Whitelisted drive. *Beware that comparing wildcards to every file on the computer can take up a lot of time; thus keep your file collections to a minimum if you are worried about time.*
2. **File Collection Identifier:** This is for the operators purposes. This name will only be in the XML configuration file and not the .cfg file. It gives the operator a quick description of the survey associated with each thumb drive.

3. **Up / Down Arrows:** The file collection will collect based on a priority scheme. For example: if secret.doc gets a hit on the third file collection, but then the next file checked gets a hit on the first file collection it will be put at the top of the file collection queue. This way the most important stuff will be placed on the covert partition.
4. **Min / Max Modified Date:** If the file exceeds the max, file will not be collected; if the file falls below min, file will not be collected.
5. **Min / Max Accessed Date:** If the file exceeds the max, file will not be collected; if the file falls below min, file will not be collected.
6. **Min / Max Creation Date:** If the file exceeds the max, file will not be collected; if the file falls below min, file will not be collected.
7. **File Pattern Wild Cards:** This is a bit tricky. File wildcards are compared against the entire file path. Meaning if you want to collect any file ending with Secret.doc, you must type "*Secret.doc;" because
   Secret.doc; != c:\user\desktop\Secret.doc, but
   *Secret.doc; does equal c:\user\desktop\Top Secret.doc
   Each file wild card must be separated by a semi-colon and end with a semi-colon; if you do not separate them with a semi-colon the program will think the two items are one. e.g. "*\secret.doc blah.txt;" will be compared to every file on the box.
   If you only want files from the C:\ drive then typing c*\secret.doc; will work. Putting the slash before the * means you only files matching that exact name. c*Secret.doc; will collect c:\junkfolder\Not-A-Secret.doc
8. **Folder Exclusion:** These are folders you do not want files from. They must be separated by semi-colons, and end with a semi-colon. Wildcards do not work on these folders, you must type the name perfectly although capitalization does not matter. Program Files and Program Files (x86) are two different folders. If you want exclude all program files you must write both; Program Files;Program Files (x86);.
9. **Min / Max File Size:** If the file exceeds the max, file will not be collected; if the file falls below min, file will not be collected.

4.3.5 ES Host Configurations Tab:



1. **ES Server Kill Date:** *ES Server(64).exe* will check this date on initial startup and whenever a thumb drive is inserted. *Warning: Some targets set their computer date to 50 years in the future so tools will kill themselves.*
2. **ES Server Name:** This is the name of the folder that will be created. Also, the *ES Server.exe* and *ES Server64.exe* will be renamed to what you have chosen. E.g. if you set this box to msconfig the folder will be name msconfig, and inside the folder will contain msconfig.exe, msconfig.cfg, msconfig64.exe, and msconfig64.cfg. msconfig64.cfg and msconfig.cfg are identical; they are renamed and place in the folder for your convenience.

3. **Collection Directory on Primary Host Target:** This is the folder where you want all of your collected data to go to. The default location is right beside *ES Server(64).exe* in a system hidden folder named 0000. The system hidden folder 0000 will always be created in a folder that you choose e.g. if you choose %appdata% your data will be stored in %appdata %/0000/.
4. **Percent of Primary Host's Hard Drive to Keep Free:** This is to protect you from filling up your target's hard drive and being detected. The formula for this logic is [(Size_of_File_to_be_added + Amount_of_space_taken_on_harddrive) / Total_amount_of_storage]*100. So, if ES Server(64).exe is trying to collect a very large file but adding that file will put you over the limit the file will stay on the thumb drive; however, if the next file is a small one and it doesn't put you over then it will be collected. This is the reason the files are collected in 10MB chunks. You might get files back that all of the pieces are not there. The post-processor will be able to sort this out for you.
5. **Encryption File:** This is the location of the encryption file on your computer. This file contains the public and private necessary for encryption. If this file does not exist, but there is a pem file located in your XML file, *Emotional_Simian_Config.exe* will ask you where you would like to store the pem file in the XML document.
6. **Generate Encryption File:** This is an easy button to create a new pem file. *Warning: Losing this key and the XML config file will cause all data collected to be useless. You will never be able to reproduce your old pem files. It would be wise to reuse the same pem file for the same ongoing op.*
7. **Dll Payload Configurations:** These configurations pertain to the *DllPayload(64).dll.* These configurations were placed here to make the same settings apply to all DllPayloads so that you don't have to search all of the target boxes to clean up after *DllPayload(64).dll.* The hash files of all collected files will be saved, but it is up to you were and how they are interpreted.
8. **Generate Unique Hash Name for each Thumb drive:** If this box is checked and you are trying to collect a file named Secret.doc, every infected thumb drive will collect that file. If this box is not checked, then the first thumb drive to collect the file will hold on to it and future infected thumb drives with this box not checked will not collect the file. However, if the file changes in any way the hash will change and any of the infected thumb drives will be able to collect the file again.
9. **Hash Collection Directory Location on Secondary Target:** This is the location of the hash file that records the hash all the files collected from that machine. The default file location is %appdata%/Microsoft/Internet Explorer/hret.cfg. The hash file will always be named hret.cfg; it is the location that you pick.

10. **Pre Build Batch File:** These fields will allow you to put a bat script or executable in to run before you build your ES Payload.
11. **Post Build Batch File:** These fields will allow you to put a bat script to run to clean up your pre build process.

4.4 Deployment to Primary Host

Once the configuration file (*.cfg file), you will grab the appropriate *ES Server(64).exe* program. If you are putting this tool on a 64 bit machine, you will want to grab *ES Server64.exe*, else grab *ES Server.exe*. *ES Server(64).exe* has privilege escalation code in it to allow it to run as System. By default you always try to run the program as Admin because Emotional Simian needs to be at least admin or greater to write data back to the covert partition. However, if you are unable to have admin privileges you will need to put the appropriate version of *ES Sever(64).exe* on the Primary Target's computer. Both *.cfg files are identical, *Emotional_Simian_Config.exe* has just conveniently renamed both to match the executable. *ES Sever(64).cfg* has to be the same name as *ES Server(64).exe*. If you choose Blah.exe, then the .cfg file must be name Blah.cfg.

Once the program is started, *ES Server(64).exe* will load and rename the *.cfg file to *.ini file. This is how you will know it is running correctly. If you ever need to put a new *.cfg file down on target, *ES Server(64).exe* does not need to be shut down; just drop the new *.cfg file down and wait at most 3 seconds. The old *.ini file will be deleted, and the new *.cfg file will be loaded into *ES Server(64).exe* and renamed to *.ini. This will be how you know the file was loaded correctly. The hash list of all whacked thumb drives is stored in the *.ini file, so deleting this file will allow *ES Server(64).exe* to rewhacked thumb drives it has already seen.

ES Server(64).exe will not overwrite any files. If you want to over write the Dlls or lnk files you will need a program to clear those off. *ES Server(64).exe* was designed to be not noticeable. If the target just happened to have a *.lnk file or *.dll of the same name you choose it will be destroyed and overwritten.

4.5 Retrieval of Collected Files

If a white list drive that has a covert partition makes it back to the Primary Host, all data files (Surveys, Directory listings, and/or File collections) will be placed in the folder specified by the **Collection Directory on Primary Host Target** parameter in the configuration program. After that, it is up to you as the operator to retrieve your files. Once *ES Server(64).exe* has collected a file off the thumb drive, it is deleted off the covert storage on the thumb drive to make room for more data.

Next, if you have selected the check box **Recharge Number of Runs** then all payloads and survey will have their max number of runs reset back to their original amount.

Next, if you have selected the check box **Allow Retasking on Previous Targets** then a new GUID will be supplied to the *DllPayload(64).dll* allowing that dll to rewhack computers it had already tagged.

4.6 Post Process of Collected Files

Post processing of the files is very easy. *Post processor.exe* must be ran as Admin. The arguments for the Post Processor are as such:

```
PostProcessor -d <IN:PEM File> <IN:Folder to decrypt> <OUT:Name of output Folder>
```

4.7 Additional Software

4.7.1 Keygen.exe:

Keygen.exe produces a Public / Private key pair. The arguments are below:

```
KenGen.exe <file_to_store_pem.pem>
```

4.7.2 Extract WM Files.exe

This tool allows you to extract files off of a thumb drive with a covert storage space on the thumb drive. This will allow the operator to extract files manually from a thumb drive on target or back at station. The arguments are below:

```
ExtractWMFile.exe <Drive Letter> Optional:<Directory to store files>
```

If the Directory to store files is not filled out, then the files will be stored in a folder named 1111 right next to the Extract WM Files.exe.

4.7.3 Get SN.exe

This tool will allow the operator to find the serial number of a thumb drive, either on the target or back at station. The arguments are below:

```
GetSN.exe <Drive Letter>
```

4.7.4 Whack_Thumbdrive.exe

This tool is used by the GUI to whack a local thumb drive plugged into your computer.

```
Whack_Thumbdrive.exe <Config.xml> <Drive letter>
```