Volume 3, Issue 11, November 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Ultra Registry Cleaner Using Heuristic Analysis and Sync Butler Techniques

Mohammed Aaftab[1], Priya.M[2], Raju.J [3]
[1,2]*School of Information Technology and Engineering*
[3]*School of Electrical and Electronics Engineering*
VIT University
Vellore – 632014, Tamil Nadu, India

Abstract— The system registry is a database that stores the stuffs like configuration settings and options on the Microsoft Windows operating system and other windows.exe installation validation key files with the appropriate class Id.It contains information about low-level operating system components as well as the application level programs which are running on the platform. In this paper we have designed a new registry cleaner; the purpose is to remove the trash files or unused file which eats up the memory and also to improve the performance and speed of the process from the Windows registry.The registry cleaner is developed based on heuristic analysis and Sync buttler techniques.

Keywords— MRU lists; Computer forensic; HKU_Classes; Active-X; Shell Open.

I. INTRODUCTION

The system registry is a large and complex database that contains operating system data, user information, and application software settings for your computer. The registry is a hierarchical database, storing varied information on the Operating System and installed components. In the beginning, it was used for storing the configuration information's of a COM components, but later became a source for peripheral application storage in an attempt to reduce reliance on *.ini* files, and now. It seems we are moving yet in another direction, back towards modular design principles, with application data stored within the application assemblies. The registry scan searches for invalid settings, errors and software paths that no longer exist. Once the scan has completed, user can choose to remove the scanned items, fixing problems and speeding up your computer. In practice our goal includes cleaning up computer registries i.e. do a scan and clean up broken shortcuts. A restore point will be generated before items are removed from the registry. In this paper we used LRU algorithm and heuristic analysis to check for risk and data wastage i.e. detection and guessing the damaged files. Sync Butler keeps a dictionary of checksums for all the files. Sync Butler automates many of the syncing decisions on behalf of the user. There are two major concepts in syncing – Auto Sync and Suggest Action.

Our system aims to reduce the cost of buying a domain from other Developers. Our system also features about MRU list. The system aids in deleting all the unwanted temporary files from the computer and provides a good GUI access our open source application. It also features system font validation. The tool which we designed is named as 'URCleaner'. It provides a flexible streamlined interface, which provides the control and access over the program that allows users to interact with user programs. The user can also get the generic information from the glossy icons embedded in it.

II. EVALUATED TECHNIQUES

A. Heuristic Analysis for Detection

In computer science and Artificial Intelligence heuristic is a technique used to solve a problem more quickly when classic methods are too slow. Indeed in this paper we actually decided to use the Elevator algorithm for Disk scheduling and scanning system images from the sector but the elevator algorithm is a classic and old and simple algorithm which may fail to find the exact solution. This is achieved by completeness, accuracy or precision of a speed. Since it is "greedy", it always finds the nearest local optima of low quality. The goal of modern heuristics is to overcome this disadvantage.

Heuristic analysis is a method employed by many computer scanning programs designed to detect previously unknown computer installed files, as well as new variants of malware or junked files already in the wild. Heuristic analysis is an expert analysis that determines the susceptibility of a system towards particular threat/risk using various decision rules or weighing methods. Multi Criteria analysis (MCA) is one of the means of weighing. This method differs with statistical analysis, which bases itself on the available data/statistics.

Most of the scanning programs utilizes analysis performing these functions by executing the commands of a specific program within a virtual machine, thereby allowing the URC (Ultra Registry Cleaner) program to internally simulate what would happen if the unwanted registry files were to be executed while keeping suspicious code isolated from the real world machine and then check or monitor for several issues such as file overwrite and system start up slowdown and boot sector slow down. Fig. 1 shows how malicious files can be appended with normal files.

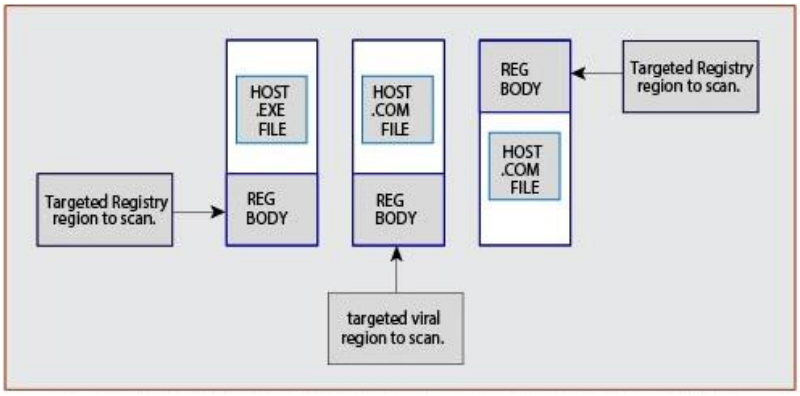

Fig. 1. Appending of Malicious files with normal files.

Another method of Heuristic analysis is to decompile the suspicious program contained in the windows registry and analyses the source code and decimal value contained in it. Then file is flagged and the user is alerted by syncing all the error files to history list. There are few heuristic techniques such as File emulation or dynamic scanning, File analysis and Generic Signature Detection. Heuristic analysis is used to detect the unused registry files and duplication of the files. It significantly limits those regions of the files to be scrutinized. Heuristic analyzer is a module that operates based on heuristic analysis. There are two methods: Static analysis and dynamic analysis. The static analysis scans the suspicious commands typical of malware. Example: modification of executable files modified by typical malware. The analyzer has a counter named "suspect counter" that increases each time when it detects for a suspicious code block in a program. If this "suspect counter" of a program exceeds a certain limit, it is assigned the suspicious status. The dynamic analysis executes the program in a special virtual environment. If the heuristic analyzer detects malicious activity, the program is identified as malware and blocked.

B. Sync Butler for synchronizing partial results

Sync Butler searches and determines automatically the files which are significant to the user and transfers those files while syncing. There are two key areas:

1. Determining the location of files.
2. Operational scenario to select these files.

1. Location Determination – locations can be determined by accessing the ComDlg32 key in the registry which is used to get the list of recent files opened by the user. Then scan the windows folder. Example windows documents.
2. Selection of Files – provides a better interface to select the scanned files to remove after syncing.

Suggest Action, this concept involved when no automation was made. This will suggest a default action and displays the user conflict resolution. In this paper we developed an application with the help of Windows Presentation Foundation (WPF) based Graphical User Interface, which acts as an entry point of the application while project contains application logic in the sense. The GUI is based on the following matrix (Table 1) where rows represent the time of modification and the columns represents the size of the application.

Matrix Reference :(Column: Size | Row: Time Modified)

TABLE .1

	Smaller	**Same**	**Larger**
Later	2	2	1
Same	X	4	3
Earlier	X	X	X

III. SYSTEM ARCHITECTURE

In general the model-view-controller architecture can be used for the program, in order to allow for ease of development and reduce excessive coupling between the logic and the user interface. In this paper we are using application logic and Regscan harness which contains the graphical user interface. The graphical user interface also acts as the entry point of the application and is responsible for calling on the controller to do various checking before starting the program. For example, fig.2 shows the invocation of the Test Single Instance method which determines whether the program will continue execution or bring the current instance to the foreground. The Controller class interacts with the GUI and acts as a main communications layer to the rest of the classes in the program. In order to reduce coupling and dependencies, the GUI should have access only to this class and the data container classes such as conflicts. It contains methods which the controller can call upon to carry out many of the program's functions.

The design of the user interface follows a design goal of having 4 visual elements and 3 sub elements at any one time, in order to achieve an uncluttered look and feel. The aim of having this minimalist design is to ensure that users are not flooded with too many information and allow for clarity of communication with the application program.

With the GUI, the users are provided by the good information about what are all the features available before scanning progress. Once the user selects the option from the menu, it will test for the single instance then if these instance are first then it will receive the action as given in figure 2 Isfirst(), the controller class will return only if it is true

else it will return to the false statement. Then the controller class acts as a communication layers which will handle the rest of the classes in the program.

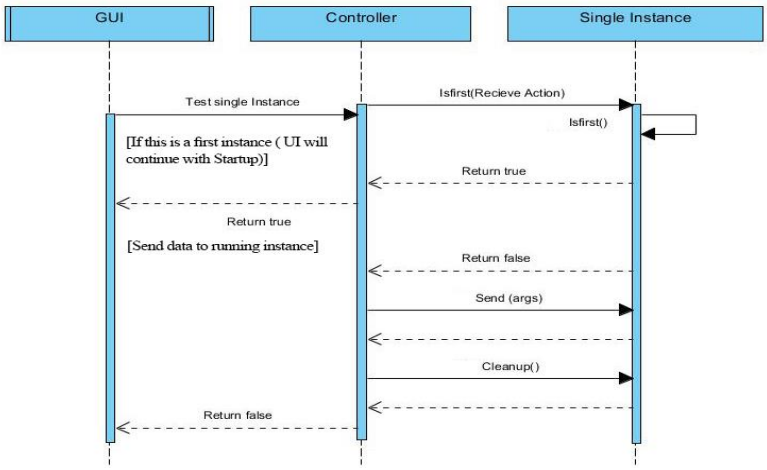

Fig.2 The working Progress of Controller Class.

A. Flexible Interface UI

URCleaner provides a flexible streamlined interface, which provides the control and access over the program that allows users to interact with our program .The user can also get the generic information from the glossy icons embedded in it. In the interface, the left pane gives the information about the current window. The tab structure is located at the left top window of URCleaner that offers three option button to the user: the first option is the main window button, the second option is the custom button and the third option is the Help desk and the about menu which gives the information of the product and the author acknowledgement. Right pane is the main configuration area which is discussed briefly in the system architectures modules. There are list of options like wiping all the damaged content of the registry in the system, to create the restore point in the separate tab where the user can decide whether to wipe the data or not and selection option to select the particular registry. These selection options can be used only by the advanced user who knows about the registry files and this interface .During scanning the progress bar provides the current status of the registry scan, name and the path of the particular registry errors which automatically added in the restore point list.

B. System Scan

System scan is useful to check and identify malware infections like Trojans, worms, adware and rootkits. These tools are forced to work together and their logs are saved in one single report. Using system scan we can also to remove persistent files and registry keys.

C. Libraries

Libraries are same as the "DLL Files". It is a collection of small programs, which can be inherited or invoked while executing .exe files in the windows platform. The registry scans for the unidentified and crashed DLL files and then specifically creates the restore points to the user and then ask the user whether to delete the file after the completion or not. Use of DLL files can reduce the size of the memory during execution. If DLL files get corrupted or missed, URCleaner will fix or replace missing dll files. The DLL files are invoked only if the executable files are declared by the DLL function and the DLL files will be called with the specified parameters. For example consider the Autodesk's AutoCAD whenever the menu is open in the low configured computer the DLL files may get crashed. To fix this here the user mostly uses the Symantec tools.

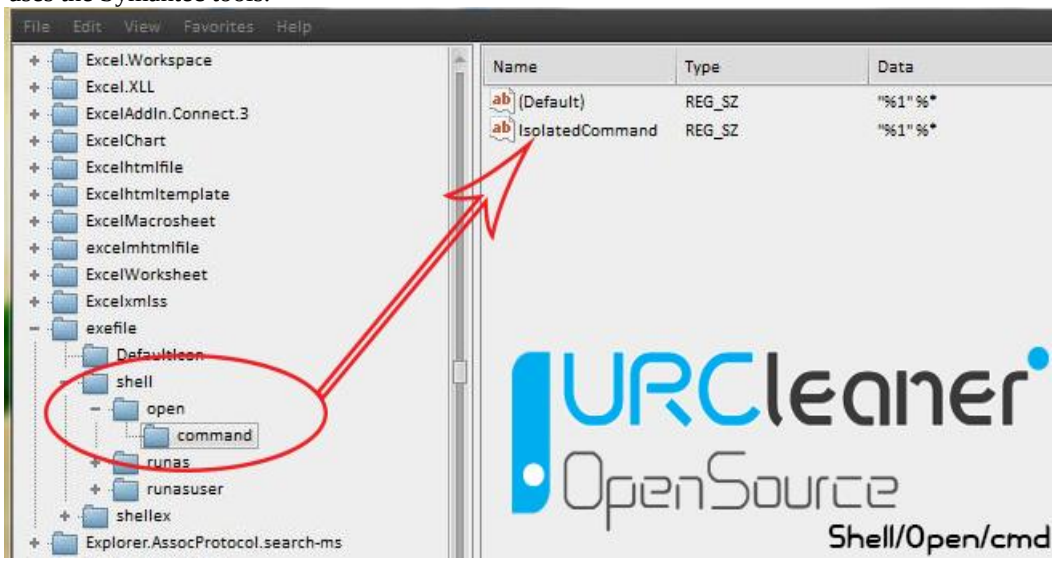

Fig.3 Shell Open Command

Malware can be loaded from a variety of different places in your PC. In addition to the more common modifications to Windows auto start entry points, malware may leverage the shell open command (Fig. 3). This allows it to register itself as the handler for certain file types and the virus, worm or Trojan loads when any of these file types are called.

D. Registry Scan

Windows Registry is a database which stores configuration settings and options on Microsoft Windows systems. This module helps in optimizing and repairing the windows registry by attempting the deep scan. The damages or a crash may occur due to some mismatch or some malfunction occurs in Windows installation which affects the registry. URCleaner scans the overtime cluttered internet junk files, needless archives, unused icons, logs and duplicate files. This utility will also help us in cleaning all our privacy data like history, cache, cookies and records from the internet browser. This utility also provides access control to the user like which application programs and services should have to be start in the system start up to improve the windows boot up time and overall system performance. After scanning, it wipes all the selected files from the system permanently which provides protection against the potential leak. It also features disk cleaning, privacy cleaning and other option like wiping and deletion.

E. Deep Scan

Deep scan module performs the search to identify the data loss, spam, viruses, key words or other content level criteria. It removes the unnecessary entries and corrupted files that cause the system slowdown and other errors. Comprehensive backup offers the user to undo any changes. There are four steps for scanning the system using on demand scan. The first step is configuring the scan location. In this process user can configure the scan locations like clicking checkboxes to perform obsolete scans and unwanted entries. By providing much facilities it also offers the definition of registry areas like start up entries, application path, shared DLLs, font location, uninstall location, and file extension, COM objects, help locations, shared folders, system services, menu orders and ignore list .The second step is scanning the windows registry. The third option is selecting the infected files for deleting or removing. After the scan is completed, the user can select the files to delete based on the results. The registry cleaner automatically selects some dll files for deletion. The confirmation message will pop up by asking whether to delete the selected files or not; by hitting yes the user can delete the scanned and unwanted entries from the system registry. The last step is cleaning the registry. This process creates the restore points and takes backups of registry files and start cleaning up the registry which the user had selected in step three.

F. System Fonts

System fonts should be registered under the windows registry in the operating system. URCleaner detects the reference of the fonts which no longer exists in the registry or the system and removes them. For Widows Arabic edition, the Default Desktop Theme is classic, Classic Desktop Font is Microsoft Sans Serif where the Standard Desktop Font is Tahoma. URCleaner scans the fonts and identifies the fonts which are missing in the System Registry.

G. MRU Lists

This module scans all the MRU lists from the registry. MRU i.e. Most Recently Used lists or the recent places. Mostly the user doesn't have any knowledge about it and these can also slowdown the system. To avoid this, the URCleaner offers an option called MRU lists in which it scans the entire MRU list and deletes the MRU lists and recent visited pages or places in the Operating System. URCleaner generates a backup of the folders/files deleted during the privacy cleaning operations if the Backup files option is enabled. The backups are stored in the date and time stamped folders. The location can be specified by the user. User can store the backup files in their computer locally, on a computer in your network or even in an FTP server. User can restore the folders/ files to their original locations at any time, if they have accidentally chosen vital contents while setting up a privacy cleaning operation. The Backup option in the Privacy Cleaner allows you to view the list of files backed up and to restore the folders/files Improper use of Registry Editor may cause serious problems which lead the user to reinstall operating system. Microsoft cannot provide help for such problems.

H. Settings

The setting option allows the user to configure various option related to the operation of the registry cleaner. Setting tab can be accessed by clicking the setting icon in the structured tab in the top left pane in the main panel window. URCleaner provides the enhanced option to customize the interface of the application where the user can change the themes and enable logging for the registry scan. Enable Logging mode- if the user needs to do deep or a full system scan the user might have to enable this option.

I. System Help files

All of the help files during the installation of the applications have to register under the windows registry. These help files cannot be deleted even after the user uninstalled all the application. Thus it won't take much space in the system but it is in the system registry which can eat ups the memory and it makes the other application helpdesk menu slow. URCleaner works to remove entries by identifying that reference to help files that no longer exist.

J. Progress Bar Indicator

The progress bar indicates the status of the current process. The process may be scanning, deleting or taking backups.

IV. FEATURE SELECTION

Initially, there are several potential features which we had hope to implement, and may still consider to in the future. Out of this large list of features, we went through various brainstorming sessions and decided to focus on what we called the "VTreg scan". This "VTreg scan" revolves around the different environments in which users use computers in, to

allow for an effortless and seamless transition between them. It is the cornerstone of our application and any future addition. The following are the list of potential features we have used:

- General Information about an assembly is controlled through the set of attributes. Change these attribute values to modify the information like assembly description, culture and trademark etc.,
- GUID [assembly: Guid("0e5f22ae-6072-4cef-9aab-60766a046746")] id for the id of TypeLib if this is exposed to COM.
- Region directives using system class and security permission setting out the paths for TypeLib and CLSID.
- cLightning.cs importing dynamic link libraries file to adjust the token privilege and lookup privilege values and process id present in it.
- Dword reading all Dword integer type by using read method and Write Dword will write the Dword value.
- Test for the key that defines multiple restore per cycle.

For the better flexible interface in this study we used circular progress bar template which has a utility control to change the appearance of the program. It includes Utils.cs, circular progress bar view model, grid Utils, and linq to visual trees and Pie piece.

1. Utils.cs - Converts a coordinate from the polar coordinate system to the Cartesian coordinate system.
2. Circular progress bar view model - This method is an alternative to WPF's, which also supports content elements. UI Helper finds a parent of a given item on the visual tree. Check if the parent matches the type we are looking for then use recursion to proceed with next level.
3. Grid Utils - Handles property changed event for the Row Definitions property, constructing the required Row Definitions elements on the grid which this property is attached to.

4. Linq to Visual tree - Defines an interface that must be implemented to generate the Linq to Tree methods generate a Linq to Tree API.

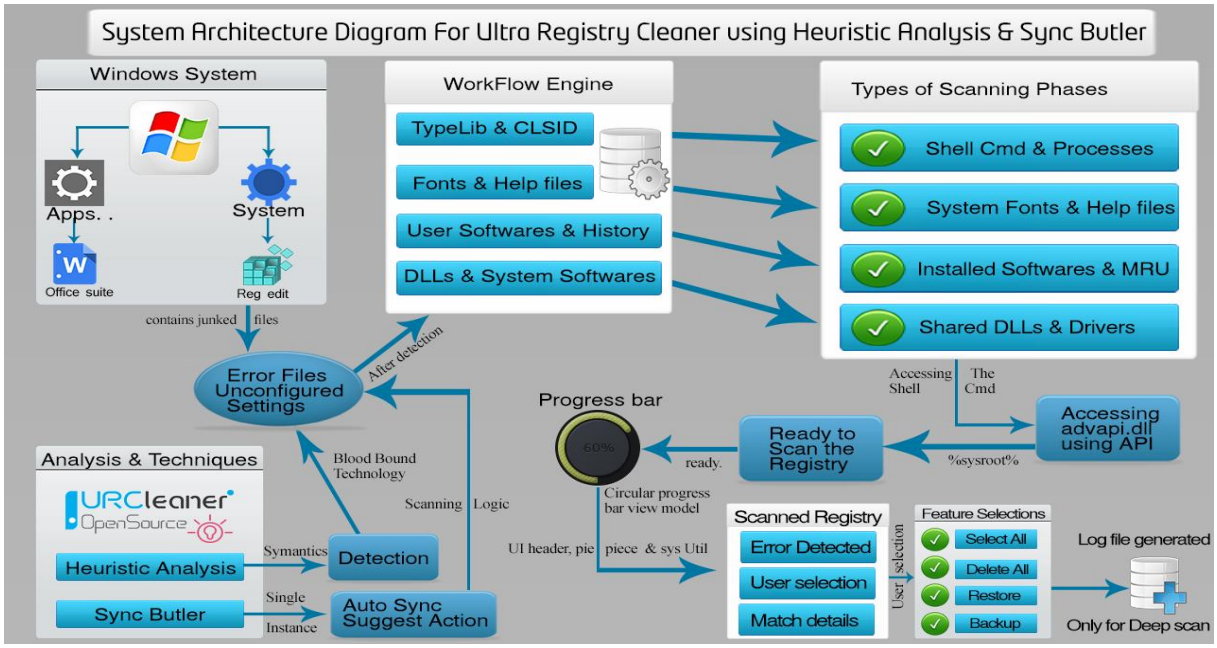

Fig 4. The Architecture diagram of the Registry Cleaner

Figure 4. Describes how the registries are scanned within the modules which are defined in the program. The system damaged registries are detected with heuristics analysis dynamically and the sync butler which offers the default action and displays the user conflicts resolution. Then the workflow engine contains all the core registry directories which contains TypeLib, CLSID, Help files, dll, history and system software's and user software's. By accessing advapi.dll using API it performs the Scanning phases includes shell command open entries, system fonts, dynamic linked libraries, help files, drivers, Most Recently Used (MRU), history and installed files for shortcuts and junked files. After scanning it shows the result providing selection option to user via which the user can manually select the file to delete, by enabling the deep scan method the restore point can be created as a log files.

V. CONCLUSIONS

Whenever the user installs the application, the device driver that are newly connected to system, the initial configuration and the settings will be stored as a keys and values in the system like binary data, hexadecimal values and data values. And whenever the changes made in file associations and control panels during the use of the computer it will be updated in the registry. The older PC contains more junked files and values and keys in the Registry which make it to run slow. Thus our Product and our implementation will work in cleaning the junked system. We are planning to add compression techniques for memory saving purpose. We also thought of adding new module called encryption using

techniques such as AES, Blowfish, and Two fish. Though we have tested this on several different machines (Vista/W7) works fine and hope this implementation will get The Code Project Open License (CPOL).

REFERENCES

[1] Author YoungsooKim and Do-Won Hong: Information Security and Assurance, 2008.International Conference on Windows Registry and Hiding Suspects' Secret in Registry.
[2] M.Russinovich, "Inside the Registry", Windows NT Magazine, 1997.
[3] J.Honeycutt, Microsoft Windows XP Registry Guide, Microsoft Press, 2003.
[4] C.Steel, Windows Forensics, Wiley Publishing, Inc., 2006.
[5] K.J Jones, R. Bejtlich, and C.W. Rose, Real Digital Forensics, Addison-Wesley,2006.
[6] Harvey, Windows Forensics and Incident Recovery, Addison-Wesley, 2004.
[7] L.W.Wong, "Forensic Analysis of the Windows Registry," Forensic Focus, 2006
[8] Timothy D.Morgan, "Recovering Deleted Data From the Windows Registry". Digital Investigation, 5, pp.33-41, 2008.
[9] Dolan-Gavitt.B, "Forensic Analysis of the Windows Registry in Memory". Digital Investigation. 5(Supplement 1), pp.26-32, 2008.
[10] Zhenhua.T, Hong.D, Ming.X, Jian.X, "Carving the windows registry files based on the internal structure", In: The 1st International Conference on Information Science and Engineering,
[11] Russinovich, M. E, Solomon, D. A.: Windows Internals (4th Edition): Microsoft Windows Server 2003, Windows XP, and Windows 2000. Microsoft Press, 2004.
[12] Ruichao.Z, Lianhai. W, Shuhui. Z, "Windows Memory Analysis Based on KPCR". In: Fifth International Conference on Information Assurance and Security, vol. 2, pp.677-680, 2009.
[13] Shuhui.Z, Lianhai.W, Ruichao.Z, Qiuxiang.G, "Exploratory Study on Memory Analysis of Windows 7 Operating System". In: 3rd International Conference on Advanced Computer Theory and Engineering.
[14] Schuster. A, "The impact of Microsoft Windows Pool Allocation Strategies on Memory Forensics". Digital Investigation, 5(Supplement 1), pp. S58-S64, 2008.
[15] Hejazi. S.M, Talhi. C, Debbabi. M, "Extraction of Forensically Sensitive Information from Windows Physical Memory", Digital Investigation. 6, pp. 121-131, 2009
[16] Russinovich, M. E, Solomon, D. A.: Windows Internals (5th Edition): Including Windows Server 2008 and Windows Vista. Microsoft Press, 2009.a

