|  |
| --- |
|  |
| Capstone Experience  IST 894 |
| Lab 11 – Windows Registry Forensics  Scott Finlon |

Table of Contents

[1. General Context 3](#_Toc88079996)

[2. Technical Context 4](#_Toc88079997)

[3. Solution 5](#_Toc88079998)

[References 13](#_Toc88079999)

# General Context

This lab’s focus is on the Windows registry, and how perform some forensic analysis on a sample capture of a registry from a computer. The Windows registry is a collection of databases of configuration settings for Microsoft Windows operating systems (Fisher, n.d.). The registry stores information and settings for software programs, hardware devices, user preferences, and operating system configurations. This can be extremely powerful because looking at the registry can tell you information about the settings of a program or hardware device, information about the computer itself like it’s name and the last time it was shutdown, and also information about individual users like any URLs they typed into a browser or files they’ve opened.

The registry is a hierarchical database that houses several keys. It’s structured into what they call hive files, and it can get extremely complex (Bajpai, 2014). Due to it’s complexity, locating relevant information that is stored in the registry can be difficult and be very time consuming. In this lab we use an application that is written in Perl called ‘RegRipper’ which has over 300 plugins that can be called to analyze registry hives and return specific datasets to you.

One of the best ways to combat the complexity, and prevent from becoming overwhelmed is to start the analysis with specific goals in mind (Bajpai, 2014). Forensic analysts need to have a good understanding of the registry hierarchy at a minimum, this way when they are looking for specific types of information that is relevant to the investigation they know right where to look. It’s one thing to have copious amounts of data, but if you can’t or don’t know how to make use of it, it’s virtually worthless.

# Technical Context

Different registry values that exist can be comprised of keys and subkeys, as a forensic analyst you are expected to know what you’re looking for, where it is located both in the hive and hierarchy, and what to make of it and how it is of use to you once you find it. RegRipper is a tool that was written by Harlan Carvey that is modular and scalable, but it produces reports based on pre-configured saved searched (SANS Digital Forensics and Incident Response Blog | RegRipper: Ripping Registries With Ease | SANS Institute, 2009). The registry is easily exported from the host machine so that it can be investigated on another machine, there are many forensic applications that make parsing registry hives easy but RegRipper is open source and is semi-actively developed.

RegRipper version 3.0 was released 2 years ago, but realistically constant development isn’t completely necessary as it’s only important when new versions of Windows come to market. Using RegRipper is straight forward. You just have to call the application, point it at the registry file in question, and choose what plugin you want. Each plugin, which there are over 300 of, is set to search a specific file type for specific data. This means if you want to know the physical addresses that were installed in a computer, you can run RepRipper and point it at the SYSTEM regisitry hive and type ‘-p macaddr’ and it will list all of the physical addresses that were on the network interfaces in that machine.

User specific data has a few caveats itself. User data is stored in a ‘NTUSER’ registry hive, and there is a hive for each and every user, and these files are deleted if the user account is removed. So if a malicious actor gets on a machine and does a bunch of things, but manages to delete the account when they are done it largely wipes the registry settings for that user which includes any URLs typed into a browser, paths typed into the file system, as well as recently opened applications and files. The registry can really give you a good look at what a specific user has recently done, but this data can be manipulated or erased fairly easily.

# Solution

To begin this lab we log into the US Cyber Range and start up the SANS SIFT virtual machine. We change directories(CD) to ‘~/Desktop/cases/01\_Filesystems’ and use ‘tar’ to extract a gzipped compressed tarball file with `tar -xvf Win7\_Laptop.tar.gz’. This performs a verbose extraction of the file and outputs it to the current working directory. We can CD into the freshly extracted ‘Win7\_Laptop’ directory and run ‘ls’ to list the contents of the directory. Next we call RegRipper with `rip.pl -l` to list the available plugins, of which there are more than 300.

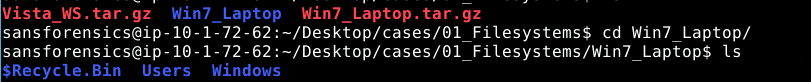


Figure 1 - entering the freshly extracted directly and listing the contents

Text

Description automatically generated

Figure 2 - listing the available plugins in RegRipper

We then start calling RepRipper to run various plugins against different registry hive files. Against the SOFTWARE hive we call the following plugins: winver, profilelist, apppaths, and installer. These plugins tell us information about the operating system and software installed. We can determine that this is a registry from a Windows 7 workstation running Service Pack 1, there are two users John and Dug, and we can see what applications are installed and the path to each of these applications executables.

Text

Description automatically generated

Figure 3 - Determining the operating system version and service pack

Text

Description automatically generated

Figure 4 - Determining the users on the machine along with their SIDs

Text

Description automatically generated

Figure 5 - listing application install information

Text

Description automatically generated

Figure 6 - listing the paths to application executables

Next, we can run compname, timezone, producttype, usbstor, rdpport, and shutdown against the SYSTEM hive to find the hostname, the configured time zone, if this was a server or a workstation, a listing of USB devices that have been connected along with their serial numbers, the port that Remote Desktop is listening on, and the last time that the machine was shut down.

Text

Description automatically generated

Figure 7 - Showing the hostname is JOHN-PC

Text

Description automatically generated

Figure 8 - The timezone is either -500 or -400 depending on daylight saving time

Text

Description automatically generated

Figure 9 - WinNT indicates this is a workstation not a server

Text

Description automatically generated

Figure 10 - a list of all USB devices with serial numbers that have been plugged in and when

Text

Description automatically generated

Figure 11 - RDP is listening on port 3389

Text

Description automatically generated

Figure 12 - System was last shutdown on March 30, 2016 at 23:08:51 UTC

Now, we’ll focus on the user Dug and his NTUSER.dat hive. By running the following plugins ‘recentdocs’, and ‘typedurls’we can see all of the files that were recently opened, and what URLs where typed into a browser.

Text

Description automatically generated

Figure 13 - Dug searched for pizza and went to reddit

Text

Description automatically generated

Figure 14 - Dug opened best.txt and note.txt recently

Finally, we want to run a few more plugins to try to identify additional information. We run the following: ‘auditpol’ against the SECURITY hive to retrieve the audit policy where it appears that auditing is not currently enabled, ‘winlogon’ to help determine autostart information for the machine, ‘comdlg32’ which is the common dialog and shows applications and files recently used, ‘user\_run’ which shows the programs configured to autostart for the Dug user.

A picture containing graphical user interface

Description automatically generated

Figure 15 - Audit policy results

Text

Description automatically generated

Figure 16 - winlogon settings

Text

Description automatically generated

Figure 17 - common dialog output

Text

Description automatically generated

Figure 18 - autostart contents

# ****References****

Bajpai, P. (2014a, August 16). *Windows registry forensics using ‘RegRipper’ command-line on Linux*. Infosec Resources. <https://resources.infosecinstitute.com/topic/registry-forensics-regripper-command-line-linux/>

Bajpai, P. (2014b, August 25). *Windows registry analysis with RegRipper – A ‘hands-on’ case study*. Infosec Resources. <https://resources.infosecinstitute.com/topic/windows-registry-analysis-regripper-hands-case-study-2/>

*Digging Up the Past: Windows Registry Forensics Revisited | Mandiant*. (n.d.). Retrieved November 16, 2021, from <https://www.mandiant.com/resources/digging-up-the-past-windows-registry-forensics-revisited>

Fisher, T. (n.d.). *What Is the Windows Registry & What’s It Used For?* Lifewire. Retrieved November 16, 2021, from <https://www.lifewire.com/windows-registry-2625992>

*SANS Digital Forensics and Incident Response Blog | RegRipper: Ripping Registries With Ease | SANS Institute*. (2009, February 6). <https://www.sans.org/blog/regripper-ripping-registries-with-ease/>