# Fred Cohen & Associates - Analyst Report and Newsletter

***Welcome to our Analyst Report and Newsletter***

## COFEE and the state of digital forensics

Computer Online Forensic Evidence Extractor (COFEE) is a software program developed by Microsoft for use by law enforcement. It was held closely by law enforcement for a period of time until it was revealed in the last year, and subsequently, several individuals released software intended to defeat the utility of COFEE. While a big deal has been made of the secrecy of this tool and other related matters, reasoned examination has been somewhat lacking in the open community, even though there have been validation studies undertaken of the tool. Thus this limited review of the situation is suited to this special end-of-year edition.

## Basics of COFEE

COFEE is, according to its documentation, a collection of programs residing on a mountable media (typically a USB disk drive emulation) designed so that when the USB device is placed in a computer, the COFEE program executable can be run by the investigator. The program is intended to use minimal resources so as to alter as little as possible in the operating environment while allowing the collection of data such as the process, file, and network status, and so forth. It does this by presenting a simple user interface and running copies of other software programs contained on the USB device to collect data.

In this sense, COFEE is really no different from programs like ForensiX or older menu-based systems for running programs, except that it is wrapped in a particular methodology and implemented on a USB drive to be useful for working on "live systems". There are many "live" forensics tools that do similar, or in many cases, what appear to be more forensically sound and larger collections of, jobs of extracting data from systems as they operate.

## Programs run by COFEE

The programs that are, apparently, standard with COFEE, are the programs listed below, as documented within the distribution I retrieved from an Internet archive for the purposes of writing this report. These and similar programs have long existed in various operating environments, such as Linux, Unix, and Windows. Their operation is well known, source code for some versions of some of them may be available, and they can be examined individually for their properties. This also helps in the issues of authenticating their operation for legal purposes, as they are widely published and well known tools that are in widespread use on a day-to-day basis all over the world, and are generally relied upon for normal business purposes for the uses they are normally applied to. That is not to say that they are without flaws, but it is consistent with the normal legal processes associated with the use of tools and writings they produce for admissibility in legal proceedings.

| Program | Description (and command line switches applied) |
|---|---|
| arp.exe | Displays Address Resolution Protocol (ARP) entries from the cache stored |

 

| Program | Description (and command line switches applied) |
|---|---|
| | on the local computer. (-a) |
| at.exe | Lists programs scheduled for future and periodic execution. |
| autorunsc.exe | Shows programs scheduled to be "autorun" at bootstrap. |
| getmac.exe | Shows the MAC address of the network interface(s). |
| handle.exe | Similar to the Unix lsof command, shows information about file, port, registry key, synchronization, thread, and process handles. (-a) |
| hostname.exe | Shows the name of the host. |
| ipconfig.exe | Shows configuration information for network interfaces. (/all) |
| msinfo32.exe | MSINFO32 displays a comprehensive view of your hardware, system components, and software environment. (/report %OUTFILE%) |
| nbtstat.exe | Shows local NETBIOS name (-n) status information of an IP address (-A 127.0.0.1) sessions and their IP addresses (-S) and remote machine names (c) |
| net.exe | Lists network information (share) network shares (use) resource usage (file) open shared files (user) users (Accounts) account settings such as password age, minimum length, etc (view) lists computers in a workgroup and shared resources available per computer (start) can start local services, (Session) list and selectively delete connected sessions, (localgroup administrators /domain) lists members of groups, administrators, guests, etc., (group) and can add, delete, view, or manage network groups. |
| netdom.exe | On a Domain Controller can get information on the domain (query DC). |
| netstat.exe | Shows protocol statistics and current network connections including IP addresses, ports, and process IDs (-ao) (-no_. |
| openfiles.exe | Lists files and folders that have been remotely opened on the system. Must have admin privileges (/query/v) |
| psfile.exe | Local and Remote Network File Lister |
| pslist.exe | Shows status and details of processes (-t) tree format. |
| psloggedon.exe | Shows who is logged in |
| psservice.exe | Lists services on a local or remote system |
| pstat.exe | Shows the status of processes and drivers currently running on the computer. |
| psuptime.exe | Displays the systems current "up time" |
| quser.exe | Lists information about users logged onto the system |
| route.exe | Displays routing information (print) |

| Program | Description (and command line switches applied) |
|---|---|
| sc.exe | Queries the status for a service, or enumerates the status for types of service (query) and extended version (queryex) |
| sclist.exe | Lists services on local machine |
| showgrps.exe | Shows groups that users are members of. |
| srvcheck | Check Server Information on localhost (\\127.0.0.1) |
| tasklist.exe | Displays services hosted on each process (/svc). |
| whoami.exe | Displays the user currently logged in. |

**Validation studies of COFEE**

Unlike most software seen on the market, and unlike many software packages used in digital forensics today, independent validation studies have apparently been undertaken of some elements of COFEE. In particular, three such studies are included in the distribution:

1. "COFEE v1.1.2 GUI CONSOLE - Validation Study" 9/29/2009 by Mark Bowser, CFCE, and Justin Wykes, CFCE, both Computer Crime Specialists at the National White Collar Crime Center.
2. "COFEE version 1.1 Runner and NW3C Profiles - Validation Study ", 9/02/2009 by Charles Matt Weir, CISSP and Sri Harsha Angara, Graduate Research Students Florida State University.
3. "COFEE v1.1.2 – Runner & NW3C Profiles - Validation Study", 9/29/2009 by Justin Wykes, CFCE and Mark Bowser, CFCE, both Computer Crime Specialists at the National White Collar Crime Center.

Study 1 was undertaken under a US Department of Justice Federal grant. "This validation study was conducted to verify COFEE properly formats, wipes, and generates profile(s) to a thumb drive, including its ability to generate a report from collected data. This validation study was conducted to ensure that COFEE consistently completed all of its required actions." The study concludes that COFEE passed all of the tests it was given, and more specifically, that it "successfully generated a listed profile, a user created profile, formatted an attached device as FAT 32 and overwrote or wiped data existing in unallocated space on the device. COFEE successfully generated a detailed report of the results of the collected data from a specified profile. There were no unexpected anomalies found during testing." A list of 18 assertions were identified for testing, and one test environment was configured for the validation. These assertions largely portray the specifics summarized above, and tests included verification that formatting of the drive fails when inadequate space is present and gives a proper error notice. This study did not provide any useful details about performance of the other functions, and makes no particular assertions about tool use, alterations to the target system, or the correctness of results, other than the performance of specific actions by the interface program as identified therein.

Study 2 defines itself well; "COFEE's primary purpose is to create a thumb drive which contains a pre-determined set of applications which are set to run on a suspect's live

 

machine. Upon connecting a COFEE generated thumb drive to a suspect's machine, the investigator executes runner.exe (a program located on the thumb drive) which, in turn, executes all of the programs specified by COFEE, and stores the data collected on the investigator's thumb drive. The programs placed on the generated thumb drives are identified by a "profile" loaded into COFEE. While any user can create their own profile, this validation study will focus only on the profiles created by NW3C: "NW3C – Volatile Data" and "NW3C – Incident Response." This validation study was conducted to ensure that when runner.exe is executed: all of the programs identified by the profile are executed, that the collected data is stored on the investigator's thumb drive, that no applications were run from the suspect's machine, and that no unacceptable writes were made to the suspect's machine. COFEE is currently only supported on the Microsoft Windows XP operating system. No other operating system was tested during this validation study."

The conclusions from this study were a bit over the top, as we will soon see, but for two graduate students, this represents a reasonably strong effort. They conclude "Testing conducted on Runner and the NW3C profiles verified that both the runner.exe application, as well as the selected programs, functioned as expected and are well within acceptable practices for data collection on a live system." ... "NW3C – Volatile Data Profile - There were no writes to the suspect drive's file system using this profile. There were updates made to the Windows Registry on the suspect's machine, however none of the registry updates were of obvious forensic value." ... "NW3C – Incident Response Profile - This profile attempted to make five writes to the target computer's file system. Three of the writes were caused by the program handle.exe and were made to the file "PROCEXP100.sys." The reference to the file PROCEXP100.sys is hard-coded into handle.exe, a product of Sysinternals, and as such it is not possible to restrain handle.exe from writing to this file. However, this file is specifically written as part of the Sysinternals' toolset and is unlikely to be of any evidentiary interest. The other two attempted writes were made to network shares on the target computer, and were also unlikely to be of any evidentiary interest. There were also updates made to the Windows Registry on the suspect's machine, however none of the registry updates were of obvious forensic value."

This study used 3 different configurations of computers, and tested the following conditions:

> "1. All programs identified in the profile were executed.
>
> 2. Results of the tools were properly stored on the investigator's thumb drive.
>
> 3. Executing runner.exe did not cause any direct writes to the suspect drive (file system).
>
> 4. Executing runner.exe did not cause any direct writes to the suspect drive (registry).
>
> 5. The tools executed were run from the thumb drive, not from the suspect's machine."

As reported in the summary, various anomalies were detected, and of course the testing was limited to the specific things identified. But the conclusions drawn were not in fact consistent with the results produced, as will be described in more detail below.

Study 3 appears to be a repetition of Study 2 by the same parties undertaking Study 1. It applies the same methodologies to 6 other computers and applies other configurations. Examples of detailed results include: "An examination of the Process Monitor logs indicates

 

that all of the programs associated with the NW3C–Volatile Data profile were successfully run during the testing period." This implies that the methodology used the appearances presented by the system under test to verify the results of the test. While this is a common methodology, it is somewhat problematic in that it only demonstrates that the system appears to do what it claims to do.

All three of these studies were very limited, very recently performed, and funded by government, and both produced results that favor the law enforcement point of view.

**Analyst comments**

It is normally a beneficial activity to perform such validation tests, even though these particular tests suffer from some particular problems that I will briefly identify below. As such, the activity is to be commended, as is that effort of those involved in doing the validation tests.

My comments and thoughts on COFEE at this time generally include:

- These validation studies are very limited in their coverage. In particular, they only cover the preparation of a "thumb drive" for use, and detection of some sorts of direct modification of the target system by the tools in operation. To the extent that this helps to assure forensic soundness of the process, or at least limits the extent of corruption to the systems under test, that is helpful indeed. In particular, for "live forensics" (in this case attempts to gather data from computer systems while they are operating) using software only (as opposed to placing hardware probes on computer devices), there are some fundamental limitations to the ability to observe without alteration of the digital forensic evidence. These studies help to show some of the limits of such alteration, and thus may have utility for countering various claims of spoliation and alteration during legal proceedings.
- The people who did them were not fully independent of law enforcement and the government. This is not a very serious complaint in this case, because the results of these efforts are now publicly available, and they can therefore be independently tested by others to assess their validity. While their scientific methodologies and validity of the results are not as clearly stated as they might be, the release of this information and the availability of the programs it tested, assuming that the versions are unaltered or that tests on the actual programs as provided can be done under legal mandate, is adequate to perform independent verification of these results. The specific information provided is also adequate to allow it to be reasonably tested, even if it is not as complete and precise as would be desired in the ideal case.
- Some of their conclusions are apparently skewed by their point of view. For example, and without limit, in [2], the conclusion identified in "tester notes" that "While there were slight changes to the registry, the writes were unavoidable in attempting to retrieve the desired information, and as such, the overall rating for this test will be listed “As Expected.” " is clearly problematic and represents a conclusion without adequate basis and that is skewed toward the assertion that the result is positive. This is, in fact, a failure of the tool with regard to the criterion identified, and should be identified as such in the result. In particular, and without limit with regard to this specific example; the interpretation of the changes as "slight" is without basis and interpretive rather than

 

factual in nature; the assertion that these changes were "unavoidable" is not factually based, and in fact, I suspect that it is not true; and the conclusion is the opposite of what it should be - to wit "Anomaly Detected". If this skewing is take into account, every test should be indicated as "Anomaly Detected", while in fact all of them are listed as "As Expected". Results from Study 3 are very similar to those from Study 2, right down to the "As Expected" summary results to summarize "Anomaly Detected" detailed results. At a minimum, the definition of "As Expected" should be changed so as to mean "Anomaly Detected", since it appears that anomalies are expected.

- What they show is very narrow and of limited value. For example, and without limit, any number of other changes may take place that are not listed, and indirect changes may be quite substantial, and include changes such as those indicated as not taking place directly. In addition, these tests don't verify in any way that the results produced reflect the situation at hand. Every single test could be passed perfectly, and the resulting data produced could be completely inaccurate as to the actual target system. They indicate this in their results, so that no misinterpretation is likely by an adequately knowledgeable and skilled examiner. Furthermore, the selection of tests may indicate the desire of those paying for the testing to limit what was studied to things they could be reasonably certain of, or limited by budget or other similar issues. Clearly, these are not tests based on some underlying scientific methodology, or at a minimum, no such methodology was identified as particularly applicable. However, some insight may be gained by viewing these tests in the context of the NIST forensic tool testing program.
- The methodologies implied are not comprehensive in terms of their coverage of possible sources of anomalies. A methodology example from [3] is: "An examination of the Process Monitor logs indicates that there were no direct writes made to the suspect drive by Runner or any of its processes (to include all of the programs within the selected profile). This test was done by filtering the Process Monitor log results to show only Filesystem information, and searching for any “WriteFile” operation." But it may be that these tools perform output that writes to files through other processes. For example, if one of them performs an execution of an external program, sending that program data that gets written, then this will not be detected by this methodology, and if the write operation does not use the file write operations in the operating system, again the write will not be detected by this approach. Hardware write-blockers and detectors are far more effective at detecting such attempts, and could easily be applied to improve the testing methodology and give results that are independent of the system under test. A forensic file difference before and after testing would also bring clarity.
- Because the systems under test are, presumably, not designed to defeat the attempts to do forensics of this sort, their operation in the test environment does not imply proper operation in a live environment in the field. It is easy to devise simple methods to defeat such tools, and indeed, even commonly used tools to defeat forensics might be able to defeat many of the methods used by these common tools. As a simple example, and without limit, suppose the target system runs virtualization and contains multiple environments simultaneously operating. When the facility is entered, the suspect simply presses a key combination, and the screen is filled with a version of Windows running in a virtual machine, and that has nothing of import in terms of

 

> activities the user wishes to conceal. The law enforcement officer uses COFEE and gains information that represents only the subset of activities associated with the virtual machine, perhaps never even becoming aware of the other virtual machines that are operating within the target environment.

Having identified the issues above, some compliments should be given where due. In particular, the testers and their reports identify what they are doing and the basis for their conclusions, even when the conclusions are not supported by the bases offered. Thus the reports and results are reasonably testable, which is a fundamental for meeting the rigors of a scientific process. In addition, and perhaps more importantly, they clearly identify what they are testing for, and thus whatever conclusions they may draw, are properly limited as to scope. While there may be many things to complain about with regard to these tests, the fact that the investigators were clear in what they were seeking to do is a real plus.

Finally, with respect to COFEE as an overall concept, and as apparently implemented, the secrecy associated with the effort is the only real problem I find with it. The notion that somehow suspects would not be aware of the sorts of information gathered and the potential for use of these publicly available and widely distributed programs is hardly worthy of any secrecy at all.

> **Perhaps the big secret surrounding COFEE is that there was no substantial investment in developing better tools or tools that are customized, more reliable, particularly well suited to the task, or otherwise represent a substantial effort.**

As the "anti-forensics" community prepares the public relations campaign against COFEE and puts out its rehashes of old tools to defeat the no-longer secret tool provided to law enforcement by Microsoft for free, the forensics community might take time to reflect on the extent to which this has any significant impact on forensic science. My opinion is that it is little more than a distraction, and my hope is that, by reading this independent review, those who care about the science will be able to return to their work, which is much needed.

The release of the details of COFEE is not only not a game changer, it is somewhat of an embarrassment to the law enforcement forensics community. The fact that this is the sort of "help" they get from Microsoft and that they end up using it because that help is better than the other help they get, shows just how much they are hurting for scientific assistance, competent tool-building, and a lively research community.

**Government funding is not getting it done, and corporate support isn't either.**

**Those who wish to demonstrate the weakness of these approaches do so easily.**

 