Part 1 of 2: The Integrity Project

All's FAIR: Forensics, Analysis, Integrity, and Response with FTimes

By Russ McRee – ISSA member, Puget Sound (Seattle), USA chapter

Prerequisites

libpcre3-dev, libssl-dev

Similar Projects

FTK and Autopsy (free)
F-Response (commercial)

It likely comes as no surprise that incident response is at the core of my professional existence. My daily job duties as an incident handler help drive my search for useful and relevant tools, in turn facilitating discovery of useful topics for *toolsmith*. I often respond with disdain when reading one of the most overused analogies for a great tool: Swiss Army knife. Yet, said analogy is applicable when we explore FTimes.

While I took horrendous liberties with its capabilities in order to craft a witty title for your reading pleasure, one tool that provides forensic functionality, intrusion analysis, integrity monitoring, and incident response is deserving of such flair.

The Integrity Project exists to "build high quality tools that meet the needs of both incident handlers and system administrators." I'm pleased enough with my experimentation with these tools to offer the first two-part toolsmith: FTimes this month, and WebJob in January.

As always, I reached out to the project developers and was received kindly by Bob Austin who provided project information on behalf of Klayton Monroe, who developed FTimes (File Topography and Integrity Monitoring on an Enterprise Scale).

FTimes is an open-source system-baseline and forensic evidence collection and is a lightweight tool in the sense that it doesn't need to be "installed" on a given system to work on that system; it has a small footprint and provides only a command line interface. FTimes is written in C and ported to many popular operating systems, both *nix and Windows. Since its inception, the FTimes Project has evolved from a single tool to an integrated tool suite.[1]

The FTimes tool suite has been used in many scenarios, clearly establishing its usefulness across a broad spectrum:

1. **eDiscovery**: FTimes is being used to support enterprise-scale eDiscovery searches (i.e., find sensitive data such as PII, PHI, clear text passwords, etc., in production environments)
2. **Compliance**: FTimes was used to help a government agency scan 10,000 Windows systems in a day to verify that known malicious files were not present
3. **Security testing**: Used to evaluate a vendor's network gateway appliance by opening the appliance, imaging the hard drive, and using FTimes to analyze the image for clues (credentials, keys, code, DB accounts, etc.) that could be used to attack the appliance
4. **Web application testing**: FTimes was used to search all files for encryption keys in a thick client
5. **Research**: FTimes was the key tool suite used to win the 2006 DFRWS Digital File Carving Challenge[2]

The FTimes roadmap includes:

- Additional validator utilities (e.g., SSNs, CCNs, ZIP codes, etc.)
- Additional carving techniques and utilities
- Enhance basic integrity monitoring via WebJob/SSH (BIMVW/BIMVS) support
- Enhance XMagic and file typing support
- Add GUI support
- Add new hashing algorithms and HashDig utilities
- Improve algorithms and techniques for mapping/digging extremely large and diverse environments
- Add support for digging inside known file types (e.g., compressed files and archives)

With such auspicious capabilities, we have but to begin the discovery process, so let's not delay.

Using FTimes

Installing Ftimes is quite simple. Download *ftimes-3.8.0.tgz*, execute `tar zxvf ftimes-3.8.0.tgz`, and if you don't already have them installed, invoke `sudo apt-get install libpcre-dev libssl-dev` to acquire the prerequisite librar-

1 Bob Austin, KoreLogic Security on behalf of Klayton Monroe, http://ftimes.sourceforge.net/FTimes/index.shtml.

2 http://www.dfrws.org/2006/challenge.

Figure 1 – Registry path found in malware

```
name|type|tag|offset|string
"/home/BadGuy/malware/SweetImSetup.exe"|normal||251408|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/SweetImSetup.exe"|normal||251696|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/SweetImSetup.exe"|normal||253352|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/Generador-Saldo-Telcel2008.exe"|normal||16280|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/iZuosa0.5.0.44.exe"|normal||372324|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/iZuosa0.5.0.44.exe"|normal||399536|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/iZuosa0.5.0.44.exe"|normal||399612|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/IMNames_4.2.0.exe"|normal||77227|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/Kamerali.exe"|normal||64416|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/video.exe"|normal||81893|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/Picture1254.JPG.exe"|normal||16280|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/Generador-Saldo-Telcel2008-101708.exe"|normal||16280|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/kuzen.exe"|normal||77227|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/MessyBetaSetup.exe"|normal||251408|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/MessyBetaSetup.exe"|normal||251696|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/MessyBetaSetup.exe"|normal||253352|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/vidyo.exe"|normal||156380|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/vidyo.exe"|normal||161576|Software\Microsoft\Windows\CurrentVersion
"/home/BadGuy/malware/spaces.scr"|normal||81381|Software\Microsoft\Windows\CurrentVersion
```

ies. Change directories to the uncompressed ftimes install package, follow with the typical `./configure`, `make`, `sudo make install` and you're finished.

FTimes offers two general modes: map mode (file topography) and dig mode (string search). File topography is the process of mapping key attributes (size, permissions, checksum, etc.) of directories and files on a given file system. String search is the process of digging through directories and files (including Alternate Data Streams (ADS)) on a given file system while looking for a specific sequence of keywords or bytes. Dig mode string searches can be expressed as ASCII, HEX, HEX/ASCII, XMagic, and regular expressions. You'll note `DigStringNormal`, `DigStringNoCase`, `DigStringXMagic`, and `DigStringRegExp` as definable configuration parameters below.

As I've spent some of your time in recent months discussing file integrity monitoring, I'd rather focus our review of FTimes in dig mode in a workbench environment. In the workbench environment, FTimes is used to examine evidence (e.g., a disk image or files from a compromised system), analyze snapshots for change, search for files that have specific attributes, verify file integrity, etc."[3] Further, FTimes is a capable tool for live incident response, but in an ideal situation you've successfully imaged the system to be investigated and have returned to your analysis workbench (in keeping with FTimes terminology). Keep in mind it can also be used in a client-server model via the `nph-ftimes.cgi` script. "In the client-server environment, the focus shifts from what the operator can do locally to how the operator can efficiently monitor, manage, and aggregate snapshot data for many hosts. In the client-server environment, the primary goal is to move collected data from the host to a centralized system, known as an Integrity Server, in a secure and authenticated fashion." [4]

You'll benefit from reading Klayton's *System Baselining – A Forensic Perspective*, as it covers compromised system analysis, the importance of identifying the nature of the compromise, as well as recovering quickly, all via system baselining as a technique to support these goals.[5]

For the variety of hoops we'll make FTimes jump through, let's assume an image has been taken and mounted appropriately for analysis. On my workstation, I've added a forensics directory to mount images to, `mkdir /mnt/forensics`, then mounted my image file:

```
sudo mount -o ro,noexec,loop hacker.hda5.dd /mnt/
  forensics
```

Assume further that our image has been taken from a system belong to an alleged malicious insider, allegedly guilty of populating malware on corporate systems (both bots and keystroke loggers) as well as harvesting customer credit card data and SSNs.

FTimes makes use of configuration files; you'll find samples in `/usr/local/ftimes/etc`.

For a simple string search one might utilize a configuration file as follows (strings.cfg). FTimes would then dig through files for matches on "Software\Microsoft\Windows\CurrentVersion," a common reference in malware to ensure its required registry entry.

```
BaseName=SuspectDig
OutDir=.
MatchLimit=3
DigStringNormal=Software\Microsoft\Windows\
  CurrentVersion
Include=/mnt/forensics
```

The command to invoke the appropriate use of FTimes would then be `./ftimes --diglean strings.cfg`.

Setting MatchLimit to 3 ensures that no more than three matches per file of any single string are recorded.[6]

Results are written to a dig file named per your BaseName, in this case SuspectDig.dig.

The results of our search were many, indicating that our alleged malicious insider was in possession of numerous malware samples (Figure 1).

Given your suspicion that the allegedly malicious insider might be harvesting credit cards, a search with a configuration file as follows might be in order.

```
BaseName=SuspectCCDig
```

3 Notes from Bob Austin, KoreLogic Security.

4 Ibid.

5 http://ftimes.sourceforge.net/Files/Papers/baselining.pdf.

6 http://ftimes.sourceforge.net/FTimes/Man+Pages/ftimes.shtml.

Figure 2 – FTimes rooting out stolen CC data (data redacted for obvious reasons)

```
1  name|type|tag|offset|string
2  "/home/BadGuy/certs/FreeMoney.txt"|regexp||0|5491-                    -7085
3  "/home/BadGuy/certs/FreeMoney.txt~"|regexp||0|5491                    -7085
```

```
OutDir=.
MatchLimit=3
#Search for CC
DigStringRegExp=^(?:(?:4\d{3})|(?:5[1-5]\
  d{2})|(?:6011))-?\d{4}-?\d{4}-?\d{4}|3[4,7]\d{13}$
Include=/mnt/forensics
```

The results would look something like Figure 2 were the dig successful.

Finally, perhaps you'd like to dig all .png and .jpg files out of the forensic image in order to really help close the case on your malicious insider. Turns out he has a penchant for inappropriate material and you'd like to add substance to your evidence list. According to the FTimes Cookbook (see next section for more detail on this helpful resource) a PNG image always begins with the following sequence of bytes:

```
0x89 0x50 0x4e 0x47 0x0d
  0x0a 0x1a 0x0a
```

Translating said sequence into an FTimes dig string results first in the following URL-encoded string, `%89PNG%0d%0a%1a%0a`, followed in turn by the following:

```
DigStringNormal=%89PNG%0d%0a%1a%0a
```[7]

A JPG image always begins with the following sequence of bytes:

```
0xff 0xd8 0x?? 0x?? 0x?? 0x?? 0x4a 0x46 0x49 0x46
```

Tranlating that sequence into an FTimes dig string results in the regex string below:

```
DigStringRegExp=(?s)(\xff\xd8....JFIF)
```[8]

Add these together in your FindThePr0n.cfg file and you're ready to go.

```
BaseName=FindThePr0n
OutDir=.
MatchLimit=3
#Search for PNG
DigStringNormal=%89PNG%0d%0a%1a%0a
#Search for JPG
DigStringRegExp=(?s)(\xff\xd8....JFIF)
Include=/mnt/forensics
```

Execute `./ftimes --diglean FindThePr0n.cfg` and the results may look like Figure 3.

You can quickly see where FTimes might be of great use during incident response or investigation; keep in mind it is capable of far more than described here.

Cookbook

For additional FTimes uses you'll definitely want to refer to the Cookbook,[9] which contains a series of "recipes" describing how to solve various problems, and how its FTimes data may be processed and analyzed. Each recipe attempts to solve a particular task or objective and is designed, where possible, to be scripted. This is an electronic cookbook that allows the practitioner to benefit from the experiences of others; for example, there are recipes on how to bulk-load FTimes data from any number of systems into an SQL database, and sample queries to search for files with suspicious names, file permissions, group, etc. Another set of useful references are the `man` pages for Ftimes,[10] which provide online documentation for most of the utilities in the project.

```
1  name|type|tag|offset|string
2  "/home/BadGuy/Pictures/hotty1.png"|normal||0|%89PNG%0d%0a%1a%0a
3  "/home/BadGuy/Pictures/hotty2.png"|normal||0|%89PNG%0d%0a%1a%0a
4  "/home/BadGuy/Pictures/hotty3.png"|normal||0|%89PNG%0d%0a%1a%0a
5  "/home/BadGuy/Pictures/hotty4.png"|normal||0|%89PNG%0d%0a%1a%0a
6  "/home/BadGuy/Pictures/couple01.png"|normal||0|%89PNG%0d%0a%1a%0a
7  "/home/BadGuy/Pictures/cheerleader.jpg"|regexp||0|%ff%d8%ff%e0%00%10JFIF
8  "/home/BadGuy/Pictures/horny.png"|normal||0|%89PNG%0d%0a%1a%0a
9  "/home/BadGuy/Pictures/sheila6.jpg"|regexp||0|%ff%d8%ff%e0%00%10JFIF
10 "/home/BadGuy/Pictures/wendy01.jpg"|regexp||0|%ff%d8%ff%e0%00%10JFIF
11 "/home/BadGuy/Pictures/1024x768.jpg"|regexp||0|%ff%d8%ff%e0%00%10JFIF
12 "/home/BadGuy/Pictures/2muchluv-1280x1024.jpg"|regexp||0|%ff%d8%ff%e0%00%10JFIF
13 "/home/BadGuy/Pictures/2muchluv-1280x1024.jpg"|regexp||332|%ff%d8%ff%e0%00%10JFIF
14 "/home/BadGuy/Pictures/2muchluv-1280x1024.jpg"|regexp||5359|%ff%d8%ff%e0%00%10JFIF
```

Figure 3 – FTimes uncovers inappropriate material

In conclusion

The FTime usage scenarios I've detailed are extraordinary oversimplifications in light of the tool suite's numerous capabilities. Once you dive into the Cookbook, you'll realize that the possibilities are endless.

Remember to try FTimes for mapping and file integrity checking as well. There's not much it cannot lend itself to when it comes to eDiscovery and meeting compliance mandates.

Enjoy FTimes for all its power and note that I'll bring you another Integrity Project tool next month, WebJob, for part two of this series.

Cheers…until next month.

Acknowledgments

Bob Austin, KoreLogic Security, on behalf of Klayton Monroe, who's notes enhanced this article greatly.

About the Author

Russ McRee, GCIH, GCFA, CISSP, is a security analyst working in the Seattle area. As an advocate of a holistic approach to information security, Russ' website is holisticinfosec.org. Contact him at russ@holisticinfosec.org.

7 http://ftimes.sourceforge.net/Files/Recipes/ftimes-dig-extract-png-from-tar.txt.

8 Ibid.

9 http://ftimes.sourceforge.net/FTimes/Cookbook.shtml.

10 http://ftimes.sourceforge.net/FTimes/Man+Pages/index.shtml.