

# ANTI-FORENSICS

## Breaking the Forensic Process

By Mark Whitteker – ISSA member, Raleigh, USA chapter

**Tainted digital evidence can render examination or analysis difficult if not impossible to perform. When data manipulation is performed to specifically compromise forensic analysis, this is known as anti-forensics.**

The process of digital forensics treats digital components as individual crime scenes, each capable of possessing thousands of pieces of interrelated evidence and clues.<sup>1</sup> The digital investigator uses this information to perform a structured examination and analysis to determine what happened, when and where, and by whom. Unlike physical evidence, however, digital evidence is much easier to manipulate, hide, or even remove from detection.<sup>2</sup> Because it exists only as electrical charges representing ones and zeros, tools, techniques, and even normal system behavior can compromise the availability or usefulness of digital evidence. Once compromised, this tainted evidence can render further examination or analysis difficult if not impossible to perform. When such data manipulation is performed to specifically countermeasure forensic analysis, this is known as anti-forensics.

### Investigation process challenges and weaknesses

The purpose of digital forensics is to determine if an information storage or processing device was used to perform a specific act. In doing so, the results of an investigation should yield reproducible results of high quality and adequate quantity to corroborate an investigator's conclusions. Unfortunately this premise is fraught with challenges and inherent weaknesses.

One of the greatest challenges faced by a forensic examiner is time – both the imminent loss of data due to changes over time and the limited amount of time an investigator has to complete the investigation process. As time progresses, many system logs and file remnants can be erased or overwritten due to the transient nature of file systems and limited capacity of storage devices. As changes occur to a system's state, network connections are terminated and processes concluded. Once modified or overwritten, extracting or reconstructing this information can be prohibitively difficult if not impossible. In addition to its effect on the state of information,

1 Brian Carrier and Eugene H. Spafford, "Getting Physical with the Digital Investigation Process," Center for Education and Research in Information Assurance and Security, Purdue University, 2003, 2.

2 Michael A. Caloyannides, "Digital 'Evidence' is Often Evidence of Nothing," in *Digital Crime and Forensic Science*, ed. Panagiotis Kanellis, Evangelos Kiountouzis, Nicholas Kolokotronis, and Drakoulis Martakos (Hershey, PA: Idea Group Publishing, 2006), 334-335.

# Anti-Forensics

- Attack the **Data**
- Attack the **Tools**
- Attack the **Investigator**

time can also place constraints on the ability of an investigator to acquire evidentiary data.

Because most investigators are assigned to multiple cases and have a finite amount of time to allocate to each investigation, they are often unable to devote adequate time to each examination. This is especially true as the storage capacity and bandwidth available to end-users continues to grow rapidly and prices decrease. As a result of this trend, the time required by traditional forensic tools to image and complete an analysis, combined with the increased number of investigative targets, has created scalability issues.<sup>3</sup> As the size of drives increases and the number of targets expands, investigators may have issue with the amount of time required to process a single hard drive, let alone multiple drives from numerous targeted systems. And once a drive has been analyzed, sifting through the enormous amount of data can be like searching for a needle in a mile-high haystack.

## Anti-forensic techniques

Anti-forensic techniques attempt to exploit the inherent weaknesses in the digital investigation processes. Primarily focused on the collection, analysis, and presentation phases, anti-forensics serves to locate and exploit issues found in all phases of the process where technically possible. This is especially true when an investigation is based on three fundamental assumptions:

- The data can provide evidence
- The investigator can trust the results of selected tools
- The investigator is able to find the evidence present on the targeted system

Anti-forensics attacks these three assumptions by targeting the data, the tools, and the investigator.<sup>4</sup>

## Attacking the data

One of the primary goals of a digital forensics investigation is to acquire evidentiary data from a target system. This requires the investigator to use specialized forensic tools to evaluate the content of a system's memory and storage devices, perform a manual review of all system data for relevance in a case, or a combination of the two. The key element of either method is the acquisition of relevant data. By attacking the data, anti-forensics seeks to make information unreadable,

hidden, or completely erased from the target system. Several methods can be used to accomplish these goals.

## Encryption

The impact of encryption on a digital forensic investigation is largely determined by the type of data being encrypted and how. The extent of what is encrypted combined with the strength of encryption methodology will have the greatest impact on the level of difficulty imposed on the investigator.

At the least granular level, the entire hard drive of a system can be encrypted, guarding its contents from unauthorized access and disclosure. While this does not prevent an investigator from imaging the drive, once imaged, the contents of the drive appear as virtually indecipherable randomized bits. Without the proper decryption key(s), the investigator must rely on performing brute force attacks against the key(s) in an attempt to decrypt the data. Given that a drive encrypted using the Advanced Encryption Standard (AES) with a 128-bit key has  $2^{128}$  possible keys, the likelihood of a successful brute force attack is miniscule at best. Adding to this difficulty, even if the investigator were to select the correct key, the disk contents must be examined with each possible key to determine if the data is properly decrypted. Given this level of difficulty, the investigator is likely to have more success by attempting to acquire the encryption key using other methods such as a previously installed keystroke logger or attempting to retrieve the decryption key from system memory.

While whole disk encryption is becoming more commonplace in government and select corporate environments, implementation of such solutions is currently most prevalent on Microsoft Windows based systems. For other operating systems such as Linux, Unix, and Apple's Mac OS X, the investigator is more likely to encounter partition-level encryption. With partition-level encryption, only a segmented portion of the hard drive is encrypted. Although the investigator may easily examine unencrypted partitions of the hard drive, as with whole disk encryption, without the appropriate decryption key(s), the contents of the encrypted partition appear as randomized bits.

Similar to partition encryption, the investigator may also encounter encrypted "containers" on the drive. While not a separate division of the hard drive, an encrypted container acts as a virtual drive on the system, encrypting any file or directory placed within the container. Unless mounted as an active drive on the target host, an encrypted container may even be hidden on the hard drive, appearing as random bits in unallocated drive space. Unless specifically looking for its existence, an investigator may not even be aware of such a container existing. And, as with other encryption mechanisms, without the decryption key, the data will remain inaccessible.

Finally, at the most granular level, encryption may be applied to an individual file or part of a file's contents. While this does not hide the evidence that the file exists, without the decryption key, its contents remain indecipherable by the

3 Golden G. Richard III and Vassil Roussev, "Digital Forensic Tools: The Next Generation," in *Digital Crime and Forensic Science*, ed. Panagiotis Kanellis, Evangelos Kiountouzis, Nicholas Kolokotronis, and Drakoulis Martakos (Hershey, PA: Idea Group Publishing, 2006), 75-77.

4 Vincent Liu and Francis Brown, "Bleeding-Edge Anti-Forensics," Stach & Liu, LLC, April 3, 2006, 12.

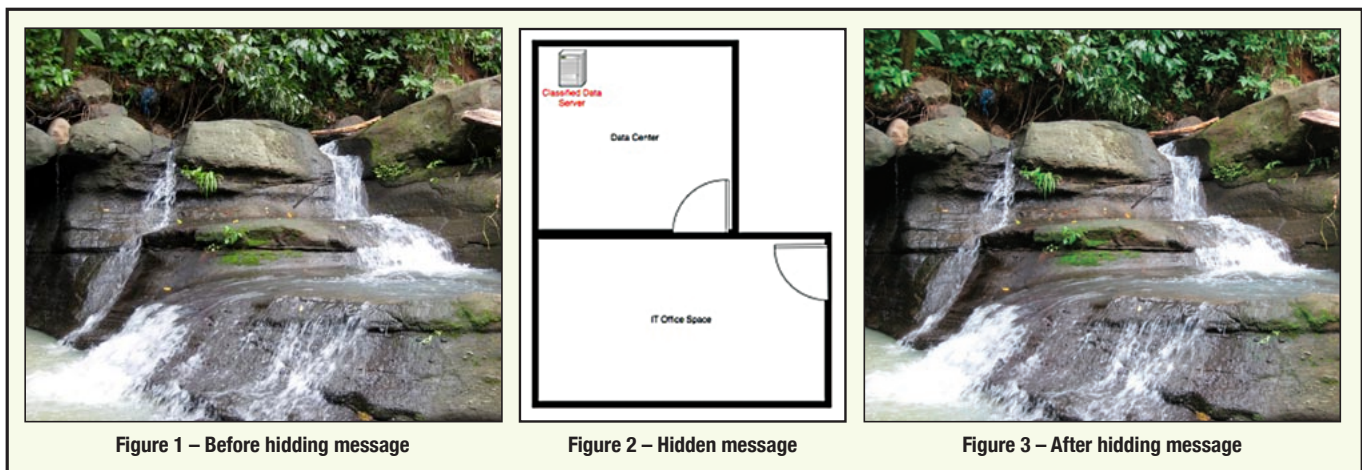


Figure 1 – Before hiding message

Figure 2 – Hidden message

Figure 3 – After hiding message

investigator. The only valuable information the investigator may ascertain would be the file's name and timestamps.

## Steganography

While the use of encryption is aimed at obscuring data so that only the intended recipient is able to read or decipher its contents, steganography is used to hide the fact that such information even exists. Steganography, a term of Greek origin, means, "covered, or hidden writing."<sup>5</sup> By applying steganographic techniques, data may be concealed, such as a message or even an entire file inside of another file without changing the containing file's outward appearance.

As an example, suppose that an individual wanted to hide a confidential image containing the blueprint of a classified facility. Using an application such as the Digital Invisible Ink Toolkit,<sup>6</sup> the individual inserts the blueprint image into an otherwise nondescript image file, such as a vacation picture from the beach. Using the least significant bits in the destination image, the blueprint image is broken into multiple segments and inserted into the destination file. The size of the destination file remains unchanged after this process, and the degradation in image quality is negligible. Unless the forensic investigator were looking for such hidden data, it is unlikely to be found.

To further increase the success of hiding data in a source file, steganography techniques can be combined with data encryption. By encrypting the target data prior to inserting it into the innocuous file, the ability for the investigator to find such data is minute.

To counter steganography, one technique employed by investigators is to perform a bitwise or hash comparison of target files against known "clean" source files to detect if a file has been modified. This, unfortunately, requires that the investigator have a known clean version of the source files. While

this may be possible for certain application- and OS-specific files, it is highly unlikely for most user-generated files such as digital images.

The first image (Figure 1) is the original image before a hidden message (Figure 2) was encrypted and hidden inside, and the second image (Figure 3) is the result. Note that there is no distinguishable difference in quality between the two files. Without performing a bitwise or hash comparison, the hidden message would go undetected.

## Unallocated space

Unallocated space consists of the storage area on digital media that has not been assigned to an active file. This may be clean, unallocated space on a new drive where no data has ever been written, or it may be areas on the drive where files existed prior to being deleted. Investigators have known for years that when a file is deleted on a hard drive, the file contents are not actually removed; only the pointer to the file recorded in the File Allocation Table (FAT) or Master File Table (MFT) is deleted. Once this occurs, the original contents of the file remain on the hard drive until overwritten as new files are created. Until overwritten, forensic tools can be used to extract and reconstruct the original, previously deleted files. Given the current trend of enormous drive space, it can often take months or years before a deleted file is completely overwritten.<sup>7</sup>

From an anti-forensics perspective, this creates several opportunities or areas of interest. The first is in storing encrypted data in unallocated space. Because encrypted data looks like random bits of unrelated information, similar to most areas of previously occupied unallocated space, this provides an excellent area for an individual to store hidden encrypted data. The user is now able to both obfuscate the information so it cannot be read by unintended parties, and be provided with plausible deniability that encrypted data even exists. After all, it looks just like remnants of old files in unallocated space. Unless the investigator is aware of such data, it is likely to be overlooked.

5 Sos S. Aghaian and Benjamin M. Rodriguez, "Basic Steganalysis Techniques for the Digital Media Forensics Examiner," University of Texas and Air Force Institute of Technology, 2006, 177.

6 K. Hempstalk, Digital Invisible Ink Toolkit, <http://diit.sourceforge.net/index.html>. A Java-based application, so interested readers would be able to try it out on any platform.

7 New Technologies, Inc., "Unallocated File Space Defined," <http://www.forensics-intl.com/def8.html>.

**Knowing that simply deleting a file does not remove its contents from the system's unallocated space, an attacker can simply perform a secure overwrite of the file, replacing its content with randomized data, prior to removing it from the FAT or MFT.**

The second area of interest is from a data deletion standpoint. Knowing that simply deleting a file does not remove its contents from the system's unallocated space, an attacker can simply perform a secure overwrite of the file, replacing its content with randomized data, prior to removing it from the FAT or MFT. Once overwritten, the prior contents of the file are unrecoverable by forensic tools.

### Slack space

When an operating system such as Windows 2000 or XP creates a file, space is allocated for the file in blocks of space called clusters. Before saving a file to disk, enough clusters are allocated to the file so that all of its contents can be stored. If a file only requires slightly more than five clusters, it will still be allocated six clusters of space. The space that exists between the end of the file's data and the end of the last cluster is known as slack space. The following diagram illustrates this concept:

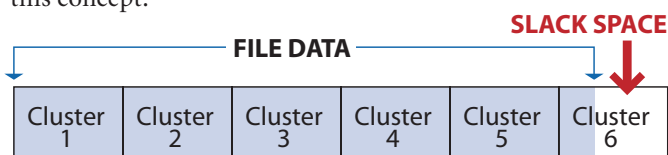


Figure 4 – Slack space

When this occurs, the slack space located at the end of the last allocated cluster can be utilized to store hidden data using specialized tools such as Metasploit's Slacker utility.<sup>8</sup> Upon normal examination of the file, the contents residing within the slack space would not be disclosed.

Although forensic investigators are aware of the possibility of storing data within file slack space, this data would go undetected unless a detailed forensic examination were performed on the file system to discover such data. To increase the effectiveness of slack space use, an individual could also encrypt the data prior to insertion into the slack space. Once this is done, the data appears as random bits, typical of innocuous data remnants from prior file allocations.

### Secure Deletion

One of the keys to anti-forensic techniques is to go unnoticed; to remove potential evidence from a system before an investigator has the opportunity to retrieve it. For files, slack

space, and unallocated disk space this can be accomplished by performing a secure deletion of data.

When data is deleted from most operating systems using the standard file deletion command or Graphical User Interface (GUI), the file itself isn't actually removed from the system. As already mentioned, only the pointer to the file's location on the disk is removed, allowing the file's previous storage location to be reallocated if needed by other files. While this is efficient from a system and operating system performance standpoint, it results in an undesirable side effect. Using standard forensic analysis tools, an investigator is able to retrieve that previously deleted file unless it has been overwritten by another file's data.

To mitigate this capability, specially designed applications can be used to perform a secure deletion of files instead of using the operating system's built-in functionality. Rather than simply removing the file's pointer from the FAT or MFT, leaving the data undisturbed, secure deletion applications will perform one or more overwrites of the file prior to removal. This replaces the file's sensitive data with random bits, rendering slack and unallocated space analysis by the forensic investigator a fruitless effort.

While anti-forensic techniques are most effective when attacking the data directly, other anti-forensic attacks can be performed against the tools used by forensic investigators.

### Attacking the Tools

Whereas anti-forensic data attacks seek to make information unreadable, hidden, or completely erased from the target system, tool attacks attempt to manipulate target systems in such a way that interferes with or misleads forensic examination. Data, files, and processes can all be manipulated so that when examined, they may appear harmless to the investigator.

### Alternate data streams

Beginning with Windows NT and the introduction of the NTFS file system, alternate data streams were introduced as a method of attaching additional data to a file without affecting its functionality or appearance when displayed using traditional file browsing tools like Windows Explorer.<sup>9</sup> By using this functionality, a user is able to hide data or even executable applications within another file's alternate data stream, undetectable by a system administrator or investigator who is browsing through a system's file structure. While most modern forensic analysis tools such as EnCase, FTK and The Sleuth Kit are able to detect and extract the contents of alternate data streams, this method is highly effective at subverting routine examination.<sup>10</sup>

<sup>8</sup> Metasploit, "Metasploit Anti-Forensic Project," <http://www.metasploit.com/research/projects/antiforensics>.

<sup>9</sup> WindowSecurity.com, "Hidden Threat: Alternate Data Streams," [http://www.windowsecurity.com/articles/alternate\\_data\\_streams.html](http://www.windowsecurity.com/articles/alternate_data_streams.html).

<sup>10</sup> Chris Davis, Aaron Philipp and David Cowen, *Hacking Exposed Computer Forensics: Secrets & Solutions*, (McGraw-Hill Professional, 2004), 174.

## MD5 hash collisions

One automated method of validating file integrity used by many forensic investigators is the MD5 hash. MD5 hashes are used to compute a unique identifier for a file based on its contents. Changing a single bit in a file will cause its MD5 hash value to change.

Relying on this methodology, many forensic analysis tools can be used to compute the MD5 hash of all files on a system, comparing the computed hash value to a previously computed hash value from a known “good” state, such as from the original media. If a change is detected, the file is identified as suspect. This process, however, relies on the assertion that a file cannot be modified in such a way that the MD5 hash doesn’t change. Unfortunately for the forensic investigator, tools and techniques now exist which enable a user to generate a hash collision in under an hour.<sup>11</sup> This provides a method for creating a file with modified or completely replaced contents that still matches the originally computed hash of the “good” file. Without performing a bit-by-bit file comparison, such modifications will go undetected.

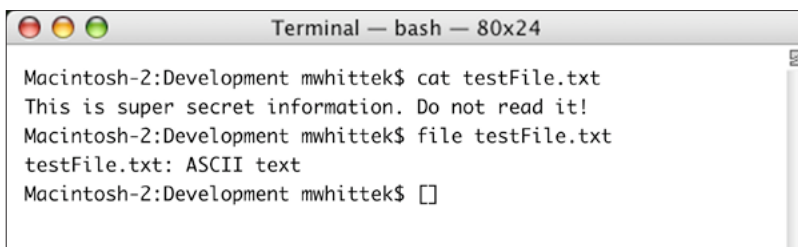
## Timestamp modification

One of the cornerstones of a forensic investigation involves the creation of a time line, a sequential visual representation of related events over a period of time. This enables the investigator to establish a logical chain of events when presenting evidence.<sup>12</sup> A primary source of this evidence is the timestamp information associated with files on the target system. Forensic tools are used to extract the creation, modification, and access times of files in order for the investigator to correlate various system activities to a user’s actions.

Anti-forensic tools exist that enable the modification of timestamps. This enables the user to create misleading or inaccurate changes to file times or log file entries. By doing so, the forensic investigator’s time line can be corrupted, incomplete, or worse yet, lead an investigator to the wrong conclusion.

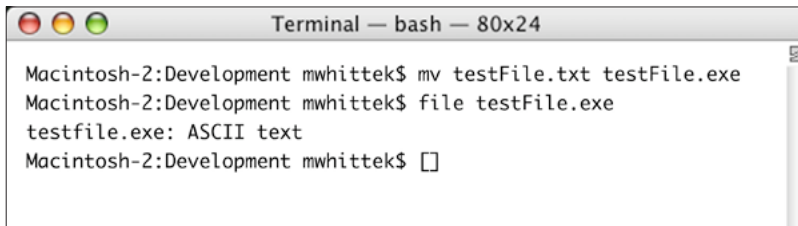
## File extension and magic number manipulation

During the course of an investigation, forensic tools are often used to analyze target systems for the presence of certain file types. Rather than performing a file-by-file manual review by the investigator, forensic tools are used to search through the file system identifying files by their extension and file signature. The simplest, and easiest to detect, modification involves changing the extension of a file. For example, if a user knows that a forensic tool is going to be used to identify



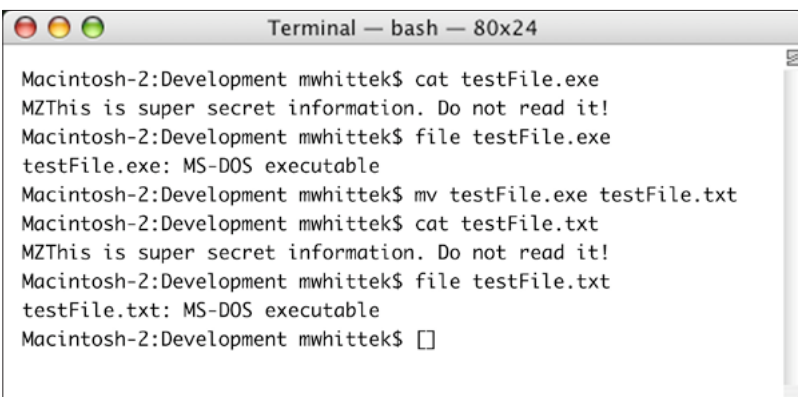
```
Terminal — bash — 80x24
Macintosh-2:Development mwhittek$ cat testFile.txt
This is super secret information. Do not read it!
Macintosh-2:Development mwhittek$ file testFile.txt
testFile.txt: ASCII text
Macintosh-2:Development mwhittek$
```

Figure 5 – Original testFile.txt



```
Terminal — bash — 80x24
Macintosh-2:Development mwhittek$ mv testFile.txt testFile.exe
Macintosh-2:Development mwhittek$ file testFile.exe
testfile.exe: ASCII text
Macintosh-2:Development mwhittek$
```

Figure 6 – Modified testFile.exe



```
Terminal — bash — 80x24
Macintosh-2:Development mwhittek$ cat testFile.exe
MZThis is super secret information. Do not read it!
Macintosh-2:Development mwhittek$ file testFile.exe
testFile.exe: MS-DOS executable
Macintosh-2:Development mwhittek$ mv testFile.exe testFile.txt
Macintosh-2:Development mwhittek$ cat testFile.txt
MZThis is super secret information. Do not read it!
Macintosh-2:Development mwhittek$ file testFile.txt
testFile.txt: MS-DOS executable
Macintosh-2:Development mwhittek$
```

Figure 7 – testFile.txt with modified magic number

executable files, he can change the extension of a suspect executable file (samplefile.exe) to a text file (samplefile.txt).

Another method used by forensic tools to detect certain file types is by analyzing a file’s signature, or “magic number.”<sup>13</sup> Many files contain constant values at the beginning of the file that identify the contained data. If this signature, or magic number, is modified, many forensic tools can be misled to believe a file is of a different type. This can cause suspect files on the target system to be overlooked or ignored during a forensic examination. The following illustrates how changing the magic number of a text file can cause the operating system to incorrectly identify it as an executable.

The original testFile.txt containing ASCII text is displayed and accurately identified by the “file” command as an ASCII text file. (Figure 5)

The same file is modified to change its extension from a .txt text file to an .exe executable file. Notice that the “file” command still identifies the file as an ASCII text file, regardless of the extension modification. (Figure 6)

<sup>11</sup> Xiaoyun Wang and Hongbo Yu, “How to Break MD5 and Other Hash Functions,” Shandong University, China, May, 2005, 8.

<sup>12</sup> Peter Stephenson, “Using Evidence Effectively,” Elsevier Advanced Technology, 2002/2006, 4.

<sup>13</sup> Optima SC, Inc., “File Formats,” <http://magicdb.org>.

Notice now that once the magic number identifier “MZ” has been added to the text file the “file” command identifies the file as an MS-DOS executable, even though the file still contains ASCII text. Even after changing the file’s extension back to .txt, it is identified as an MS-DOS executable. This could cause the investigator to overlook the file if searching for specific keywords within text files. (Figure 7)

## Attacking the Investigator

Two of the biggest limiting factors affecting a forensic investigator are time and money. Knowing this, anti-forensic techniques can be employed that attack the efforts of an investigator by making an investigation prohibitively expensive in both time and resources. Most organizations must, at some point, make the decision whether or not to terminate an investigation. Once the invested costs have exceeded the perceived benefits, many investigators can not afford to continue with a case.

There are a few simple techniques that can be used to increase the time and resources required to conduct an investigation. Initial system design decisions such as selecting the largest hard drive available or implementing a RAID array can greatly increase the amount of time needed to scan a drive and extract potential evidence.<sup>14</sup> Additionally, if performing an attack, it behooves an attacker to utilize as many

systems as possible. This greatly increases the complexity of an investigation, and can affect the investigator’s ability to obtain system images and log files. This is especially true if the infrastructure involves resources residing in other countries.

## Conclusion

The proliferation of anti-forensic tools and techniques has continued to rise over the past few years. Their ease of use and widespread availability to the mainstream user population is now allowing common criminals and inexperienced hackers to thwart the efforts of even the most seasoned forensic investigators. By manipulating, hiding, and removing digital evidence, they render many investigations difficult if not impossible to perform. By understanding anti-forensic tools and techniques, investigators can become better educated and more aware of the challenges and opportunities facing them, improving their success and continuing the evolution of the digital forensics industry.

## About the Author

*Mark Whittaker, CISSP, GSNA, GCFA, is a Security Architect and Information Systems Security Officer at Cisco Systems, Inc. He can be reached at [mwhittek@cisco.com](mailto:mwhittek@cisco.com).*



<sup>14</sup> Vincent Liu and Francis Brown, “Bleeding-Edge Anti-Forensics,” Stach & Liu, LLC, April 3, 2006, 21.