



**McAfee® Endpoint Encryption (Safeboot®) Evaluation**  
Beth Israel Deaconess Medical Center  
IS Security Department

2008-09-12, Rev 1.0

Michael K. Yamamoto, CISSP <[myamamot@bidmc.harvard.edu](mailto:myamamot@bidmc.harvard.edu)>

***FOR INTERNAL DISTRIBUTION ONLY***

**Abstract**

This document describes the evaluation of McAfee, Inc's full disk endpoint encryption product "Safeboot". It discusses issues surrounding adoption of this technology as part of BIDMC's "Defense in Depth" security strategy.

**Scope**

The evaluation will address the technical and security-focused issues surrounding the use of Safeboot at BIDMC, and its relevance to "safe harbor" and Massachusetts General Law.

**Safe Harbor**

Massachusetts recently became the 39<sup>th</sup> state with a data security breach notification law. Should the loss of personal information occur, notification must be sent to affected parties, along with the Attorney General and Director of Consumer Affairs and Business Regulations<sup>1</sup>.

If a lost or stolen device employs an encryption mechanism (and the breach does not also include the means to access the encrypted data), the loss is said to fall under safe harbor, where no notification need occur. According to M.G.L. ch.93H §1, Encryption is defined as "transformation of data through the use of a 128-bit or higher algorithmic process into a form in which there is a low probability of assigning meaning without use of a confidential process or key."

**Encryption and Authentication**

By default Safeboot uses the Advanced Encryption Standard (AES/Rijndael) block cipher. It uses a 256-bit secret key to encrypt the hard disk of the target endpoint (a "symmetric key" cryptographic process). AES has been adopted as a Federal Information Processing Standard (FIPS-197)<sup>2</sup>, and is a NSA approved cipher for the encryption of top secret information.

The process used by Safeboot to encrypt the drive has received FIPS 140-2<sup>3</sup> and Common Criteria<sup>4</sup> EAL4 certification<sup>5</sup>.

After the Safeboot client has been installed on the target endpoint (manually, through ePO

- 1 <http://www.mass.gov/legis/laws/seslaw07/sl070082.htm>
- 2 <http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- 3 [http://en.wikipedia.org/wiki/FIPS\\_140-2](http://en.wikipedia.org/wiki/FIPS_140-2)
- 4 [http://en.wikipedia.org/wiki/Common\\_Criteria](http://en.wikipedia.org/wiki/Common_Criteria)
- 5 [http://www.mcafee.com/us/local\\_content/datasheets/ds\\_mcafee\\_endpoint\\_encryption.pdf](http://www.mcafee.com/us/local_content/datasheets/ds_mcafee_endpoint_encryption.pdf)

or as part of the imaging process), the encryption process will begin. It may take several hours to complete, depending on a number of variable factors (disk size, processor speed, RAM, cache, etc).

When the initial encryption process has completed, the endpoint will contain a new MBR, referred to as the "Safeboot file system". This is an area of disk reserved for pre-boot authentication (granting the ability to start the endpoint from disk) and authentication to the secret key (granting the ability to decrypt the data stored on disk). The target partitions will be fully encrypted and unreadable to any person or automated program that does not have access to the secret key.

Authentication to the pre-boot environment and to the secret key is handled through the Management Console, where an administrator may grant or revoke privileges to members of BIDMC's Active Directory tree. This access control list must be carefully maintained to preserve the confidentiality, integrity and availability of the target endpoint.

## **Decryption**

In order to access the encrypted data stored on disk, a user must first authenticate to the Safeboot filesystem. From there, the user can "unlock" the secret key, which provides access to the rest of the disk. This allows the operating system to function, files to be opened, etc. Although accessing the Safeboot filesystem and unlocking the key usually occurs in a single step, a legitimate user will still need proper login credentials to the operating system (an Active Directory login). A user needs one password to boot the machine, and another to log into the operating system.

Access to the disk is handled by Safeboot's filesystem driver. Generally speaking, as the operating system needs access to files (which at this point are encrypted), the Safeboot driver dynamically provides an unencrypted stream containing the requested files back to the calling process. Likewise as the OS writes files, the Safeboot driver dynamically takes an unencrypted stream from a process, encrypts it, then writes it to disk. The disk remains encrypted, but any arbitrary file may be retrieved in an unencrypted state once the system has been booted by any process with the proper OS-controlled credentials.

As an example, suppose we took an encrypted system, booted/logged-in, then shared the "C:" drive over the network. Any user who was able to log-in over the network would be able to read the contents of the entire drive, even though that drive is encrypted. This is because all of their file I/O operations are being handled by the Safeboot driver, which is unlocked and able to access encrypted data on our behalf.

## **Recovery**

In the event that the Safeboot file system becomes damaged, McAfee provides several methods of recovery. Through the use of a the Management Console, a recovery disk, a daily rotating code (obtained through McAfee's support line) or an auto-generated keypair an administrator/technician can access the Safeboot filesystem and possibly the encrypted data in the event of a system failure.

It is important to note that an unauthenticated user will be unable to access the encrypted section of disk. A technician would be unable to repair the NTFS file system on a target endpoint if they did not have authorization to the secret key, which is not granted through simple authentication to the Safeboot file system.

## Disk Performance

During normal operation of a Safeboot-encrypted endpoint, data is dynamically encrypted/decrypted as needed. Therefore disk read/write operations may be negatively impacted by the Safeboot driver.

For this evaluation, 2 systems were tested with the IOzone Filesystem Benchmark tool<sup>6</sup>: A private-build laptop and a private-build desktop (as a comparison of system performance pre/post encryption, specific systems stats are irrelevant but available upon request).

The full range of file system benchmarks were run, including basic read/write, random read/write, stride read/write, etc. File sizes were tested up to twice the addressable memory space (RAM) so that cached and non-cached values would be measured. Excel spreadsheets containing the results and specific command-line options are included with this report.

The largest average speed reduction occurred during basic write operations (see appendix, "writer report"). A reduction of up to 10,000 Kbytes/sec (1K = 2<sup>10</sup> bytes), about a 9% loss in performance. Other operations generally show a much smaller or even negligible difference.

## Miscellaneous Tests

Several tests were run to validate basic operational functions of Safeboot. These include:

1. **Installation and Removal** - The target endpoint can be fully encrypted, then fully decrypted without a loss of data.
2. **Authorization and Authority** - A username/password can be configured with the Management Console and applied to the endpoint. Those credentials can then be used to boot the system, and access encrypted data.
3. **Back-channel Access** - The target endpoint (in this case a WinXP system) can be booted with removable media (Linux cdrom), the disk fully accessed with ntfsmount<sup>7</sup>, and plain-text data may be read through a raw block device. After encrypting with Safeboot, this process no longer succeeds.
4. **Recovery** - Authentication to the Safeboot filesystem is obtained through both the "magic key" (daily rotating value from McAfee support) and the built-in recovery method (key pairs generated by the target endpoint and the Management Console).

## Conclusion

Through the use of Safeboot's default encryption method, proper documentation and maintenance of the system access control list the data contained on a lost or stolen endpoint would be useless to an unauthorized user. Given the strength of default encryption algorithm, such a loss would fall under safe harbor and BIDMC would have no legal obligation to report the breach.

The performance penalties are not likely to impact an average desktop user. Tasks such as web browsing, email, office applications (Word, Excel, etc) were not noticeably slower during testing. Should we want to deploy this technology to a high performance environment, additional system tuning would be recommended before deployment.

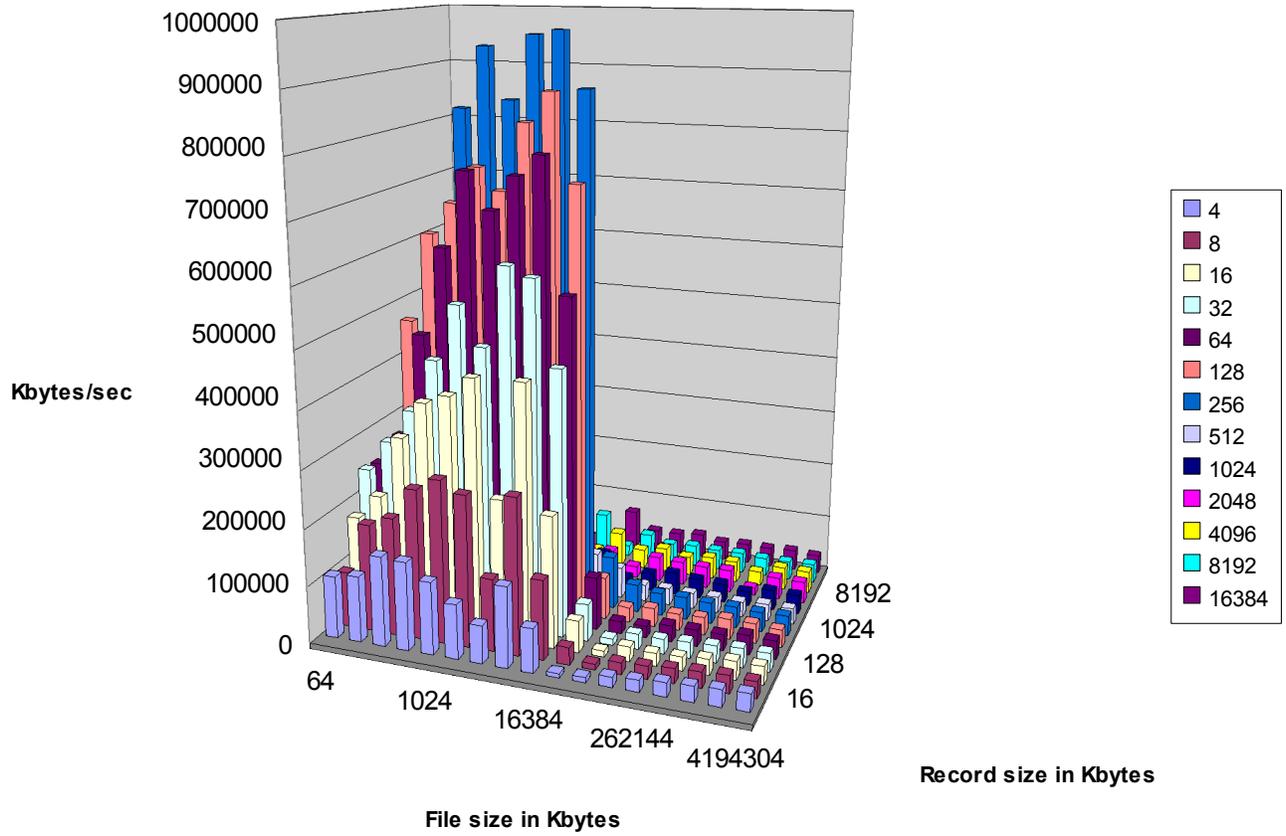
While Safeboot requires virtually no training for end users, administrators and technicians must fully understand its operation, process and caveats in order to maintain proper security and to assist in a useful way should a problem occur.

---

6 <http://www.iozone.org/>

7 <http://www.linux-ntfs.org/doku.php?id=ntfsmount>

**Writer Report (unencrypted)**



**Writer Report (encrypted)**

