A Security Analysis of My.MP3.com and the Beam-it Protocol

Adam B. Stubblefield and Dan S. Wallach
Department of Computer Science
Rice University

Abstract

My.MP3.com is a service that streams audio in the MP3 format to its users. In order to resolve copyright concerns, the service first requires that a user prove he or she owns the right to listen to a particular CD. The mechanism used for the verification is a program called Beam-it which reads a random subset of an audio CD and interacts with the My.MP3.com servers using a proprietary protocol. This paper presents a reverse-engineering of the protocol and the client-side code which implements it. An analysis of Beam-it's security implications and speculations as to the Beam-it server architecture are also presented. We found the protocol to provide strong protection against a user pretending to have a music CD without actually possessing it, however we found the protocol to be unnecessarily verbose and includes information that some users may prefer to keep private.

1 Introduction

MP3 [1, 5] is a compressed audio format that offers audio quality comparable to that of a traditional compact disc at roughly 1/10 of the storage space (128 kbits/sec vs. 1.5 Mbits/sec). MP3 compression has proven quite popular, appearing in a number of portable consumer products, in addition to software to play MP3 files on a computer. Normally, a user inserts a traditional audio CD into a computer CD-ROM drive and uses a program, sometimes called a "ripper," to read the raw data from the audio CD and compress it to MP3 files, typically one file per song. Song titles and artist information are obtained through a free Internet service, CDDB[1], which maintains a database indexed by the lengths of each audio track.

MP3 files, by virtue of their small size and ease of sharing, have proven to be quite a thorn in the side of the music recording industry. The Recording Industry Association of America (RIAA) has brought legal pressure on sites that indiscriminantly host MP3 files and has even sued MP3 search engines [3] and services like Napster [7], which help users locate MP3 files, whether or not they are authorized to listen to them.

1.1 My.MP3.com

Into this fray of copyright enforcement, MP3.com, a popular Web site that hosts MP3 files for all kind of music bands, has recently created a new service called My.MP3.com. Users begin to use the service by registering with their e-mail addresses and passwords. Then, users can add music to their personal libraries either from existing CDs or by purchasing a CD at an MP3.com-affiliated reseller. My.MP3.com allows the user to stream the music over the Internet using a standard web-based interface.

As a technical means to avoid copyright infringement, MP3.com created a system called Beam-it[2]. A user inserts a CD into a CD-ROM drive, which is them "beamed" to their account on My.MP3.com's server, where it can then be normally downloaded. Very little music data is actually transmitted across the network. Instead, Beam-it implements a challenge-response protocol that allows a user to prove they have physical possession of a CD. In essence, the server picks several random blocks of data from the music disc and challenges the client to respond with those blocks. If the client happens

[1] http://www.cddb.com/about.html

[2] http://help.mp3.com/help/faqs/beaming.html

to read corrupt data from the CD (perhaps it was scratched) or if the client attempts to forge data (perhaps by decompressing an illegally obtained MP3 file), the server considers the challenge to be a failure and issues a new challenge. Once the challenge succeeds, MP3 files representing the CD in question are added to an account for the user.

MP3.com apparently purchases the audio CDs itself and stores the MP3 files on its own server. The server need not store the original audio CD, even for the challenge-response system. Instead, the server most likely computes hashes of the responses to its challenges and compares that to pre-computed hashes. Even though users could conceivably precompute these hashes as well, the original data must be transmitted in the challenge-response protocol, so user-computed hashes would have no benefit.

The authors decided to investigate My.MP3.com after reading an article[3] posted to Slashdot on February 4, 2000 announcing the release of the Beam-it Linux client.

1.2 Account sharing security

Once a user has proven to My.MP3.com's satisfaction that she possessed an actual CD, she may then download any track from that CD at any time, and from any location (and with either low or high-fidelity sound). This account is accessed using standard Web password-based access controls, and thus suffering from all the usual weaknesses of password-based authentication. Of likely concern to the recording companies, users can trivially share passwords and thus share accounts. A cartel of geographically distributed users could easily pool their money, purchase individual copies of music CDs, add them to their My.MP3.com account, and then the whole cartel can access the music. To limit account sharing, My.MP3.com could certainly apply hardware token-based authentication methods, such as SecurID[4], which are very difficult for a user to duplicate. Even then, a user wishing to access the cartel's account could telephone the user holding the hardware token and perform the authentication as before.

1.3 Privacy issues

Any user who uses My.MP3.com is inherently giving up a remarkable amount of privacy. My.MP3.com knows every CD in a user's collection that they "beamed" to the server along with the user's e-mail address, network IP address and and Ethernet MAC address. An unscrupulous marketer could correlate musical preferences with other lifestyle choices and use this for targeted advertisement. MP3.com's privacy policy [5] does not offer strong guarantees against this kind of behavior, and the ability to opt-out is at the bottom of the user-preferences page – something that most users will never do.

A similar privacy issue with Real Network's RealJukebox [6] caused Real Networks to redesign their system. Unless MP3.com redesigns their privacy policy, users who wish to preserve their privacy should consider more traditional means to acquire MP3 files.

1.4 Legal issues

Legal issues are largely beyond the scope of this paper. The RIAA is currently suing MP3.com precisely over the legality of their service[6]. MP3.com has also filed a counter-suit[7].

1.5 This paper

The remainder of this paper is structured as follows. Section 2 presents the details of how the Beam-it system works. Section 3 presents the methodology we used to reverse engineer Beam-it. Section 4 presents our conclusions.

2 System Structure

2.1 Program Architecture

The Beam-it system is available for a number of platforms. Under Windows and MacOS, it is distributed as a binary application. Under Linux, however, it is distributed in two parts – an open-source application and a closed-course library. The application is responsible for collecting user input and

[3] http://slashdot.org/article.pl?sid=00/02/04/1027208

[4] http://www.securid.com/products/securid/

[5] http://www.mp3.com/privacy.html

[6] http://www.wired.com/news/politics/0,1283,33634,00.html

[7] http://www.wired.com/news/politics/0,1283,34191,00.html

No.	Direction	Purpose	Data
User Authentication			
1	Server→Client	Hello	version information
2	Client→Server	User ID	username, machine-id, version information
3	Server→Client	Challenge	challenge, user-id
4	Client→Server	Response	MD5(challenge, password, challenge)
5	Server→Client	ACK	user-id, machine-id
CD Identification			
6	Client→Server	CD Info	length of CD, track offsets
7	Server→Client	CD Identifiers	possible disk identifiers
CD Verification			
8	Client→Server	Specific CD Identifier	disk identifier to check against
9	Server→Client	CD Data Challenge	track-id, offset, length
10	Client→Server	ACK	track-id, offset, length
11	Server→Client	ACK	amount of data expected
12	Client→Server	CD Data Response	CD data
13	Server→Client	ACK	success on disk identifier
Clean-up			
14	Client→Server	Quit	quit
15	Server→Client	ACK	bye

Table 1: Beam-it protocol summary.

1	210 vers=1.0 prog=msp_server-1.0 plat=linux-i386 srvr=cdver02.mp3.com:8094
2	HELO mail=*username* vers=1.00 cver=LINUX010 sern=*machineid*
3	320 strg=*challenge* usid=*userid* meth=md5
4	AUTH meth=md5 pass=*md5sum*
5	220 sern=*machineid* usid=*userid*
6	MDID time=*secs* tkof=*list*
7	232 mdid=*diskids*
8	VFCD mdid=*diskid*
9	331 bits=16 trck=*t* chnl=stereo nsec=*l* encd=pcm size=*s* rate=22050 sect=*offs*
10	RVDT trck=*t* sect=*offs* nsec=*len* rate=22050 chnl=stereo bits=16 size=*s*
11	350 size=*s*
12	*data*
13	231 mdid=*diskid*
14	QUIT
15	212 help=Bye time=*slen*

Table 2: Beam-it protocol syntax.

username	the user’s e-mail address	`astubble@rice.edu`
machineid	’m’ appended with the MAC of the NIC in hex	`m006097060a92`
challenge	a random string	`cClCloXFstp3CKq2OyCXxW`
userid	a number associated with your e-mail address	`118826483`
md5sum	MD5(*challenge*, *password*, *challenge*)	`6d08c230659872b27657c97e4a8eab49`
secs	the total number of seconds on the disk	`4287`
list	a comma delimited list of the track offsets	`150,18932,43907,66491,98797`
diskids	a comma delimited list of possible disk ids	`517897,724912`
diskid	the disk id that we would like to verify against	`517897`
t	track number that should be read from the CD	`3`
l	the length in sectors that should be read from the CD	`7`
s	the size in bytes that the server expects	`8232`
offs	the sector offset from the beginning of the track	`8173`
data	every other 32-bit word from the data read from the CD	*raw data*
slen	the length of the session	`32`

Table 3: Meaning of variables in tables 1 and 2.

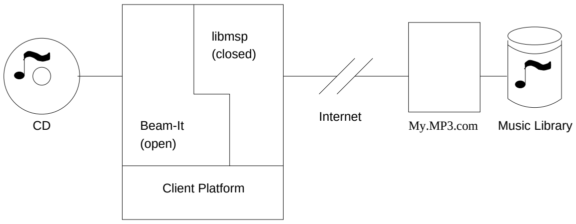

Figure 1: Beam-it Architecture

providing a standard interface to the CD-ROM. The closed source “Music Service Provider Protocol” library (libmsp) is responsible for the networking and implementation of the Beam-it protocol.

2.2 Security Architecture

The security functionality of the Beam-it protocol is broken into two distinct parts. First the user must be authenticated, then the client must prove to the server’s satisfaction that it possesses a specific audio CD. Both sections are performed using a challenge-response protocol.

2.2.1 The Protocol

The protocol used by My.MP3.com to verify CDs is, for the most part, a text based challenge-response protocol. Commands are of the form “*command arguments*” where *command* is a number for messages intended for the client and mnemonics for messages intended for the server. The specific messages of the protocol are listed in tables 1 and 2.

2.2.2 User Authentication

User authentication begins in step 2 where the client sends the e-mail address the user entered when registering for the My.MP3.com service. Also, in step 2 the client sends the machine’s Ethernet MAC address.

The user authentication utilizes a random nonce sent by the server appended to the users password and hashed with MD5 [4], a cryptographic one-way hash function. As with any challenge / hashed-response authentication of passwords (CHRAP) protocol, this protocol is vulnerable to off-line dictionary attacks against the password. Furthermore, many Web users reuse a single password for all sites they visit, so an eavesdropper may well observe the plaintext password in an earlier session. Stronger protocols are available for password-based authentication [2], but could be considered overkill for this application.

If an attacker successfully acquired a password, the risks to the compromised user would be minimal. The attacker could beam new CDs to the user’s account, and the attacker could download music from the user’s account. While the user might suffer a loss of privacy by having their music collection revealed, the real issue is that a attacker might be able to acquire music for which she did not pay.

2.2.3 Disk Identification

The process of disk identification is completed in steps 6 and 7. The Beam-it client provides information about the CD and the server responds with possible CD identifiers associated with the information.

The client begins the process of determining the identity of a particular CD by sending the server the total number of seconds of music on the CD along with the offset from the beginning of the disk for each track in step 6. Because of possible collisions (i.e. multiple disks with the same length and track offsets) the server returns a list of possible disk identifiers in step 7.

2.2.4 Disk Verification

For the CD verification phase, the server asks the client for random blocks of data from the CD. These are transmitted to the server which compares the client data to the correct values. To accomplish this, the server does not need to store the original CD in its uncompressed form. Instead, the server most likely maintains hashes computed from the original data. The CD verification responses can be hashed and compared in the server to its stored hashes. Regardless of what happens on the server, the user must still have the original CD (or a bit-for-bit accurate copy of it) to respond to the server's challenges.

In the event of bit-errors on the user's CD, perhaps due to scratches which cause data loss, the challenge/response will fail, causing the server to issue a new challenge, repeating the process until either a challenge has a successful response or some server-determined maximum number of challenges are failed. Upon a successful response to a challenge, the server indicates success and the user may either terminate the protocol or "beam" additional CDs to the server.

An interesting property of the Beam-it protocol is the way the client encodes the data for transmission to the server. Before transmission, the client discards every other 32-bit word. We believe this data is dropped such that a network eavesdropper cannot recover full-fidelity music solely by observing the Beam-it protocol.

3 Methodology

The "Music Service Provider Protocol" library is provided to users in binary form only. In order to better understand the protocol we decided to reverse engineer the library.

Because the library is responsible for all net access, the authors first ran the supplied client over a sniffed network to determine the basic protocol used. A "fake" server was then constructed so that values of our choice could be fed to the client. For some steps, such as determining the authentication algorithm, a debugger was used to examine the activities of the library. The library itself provided some hints, even though most symbols were stripped. There were dynamic link symbols named msppEncryptMD5, MD5Init, MD5Update, and MD5Final that were not referenced in the API header file. We placed breakpoints at these functions during the authentication stage so that the values passed to the MD5 functions could be examined. After downloading and compiling the MD5 reference code from RFC 1321 [4], the authors noted that many sections of the compiled source were identical to code in the msp library.

To test that we correctly understood the Beam-it protocol, we developed a replacement library that was compatible with the original msp library, and was able to successfully "beam" disks to My.MP3.com.

4 Analysis

Though the protocol for Beam-it was supplied in a closed library, it would appear to be relatively secure. In particular, the only way a user may successfully "beam" a CD to a My.MP3.com account is to have possession of the CD itself or access to the original CD image. We see no security reason for portions of the Beam-it system to be distributed as a closed binary except perhaps to compromise the users' privacy (see section 1.3 for details).

By studying the Beam-it protocol, we observe several of the messages are redundant or unnecessary. In particular, messages 10 and 11 within the CD verification phase, seem to serve no useful purpose. We also observed the password challenge / response system can be subjected to off-line password attack.

While we do not have access to the My.MP3.com server's source code, we attempted to study its behavior. In particular, we wanted to know whether the server could conceivably request every block of music with equal probability or whether the server had any biases. If the server only used a small number of challenges, then users could share only those responses and fool the My.MP3.com server. We attempted to beam the same CD a large number of times from two different machines. Of the approximately 2500 sectors requested by the server, only 100 or so were requested 2 or more times and no track was requested more than 3 times. The server appears to be sampling the disk purely at random.

If the server detects bad data, it simply asks for a another block, probably to get around scratches on legitimate disks. If the client repeatedly sends bad data the server reports that the disk can not be beamed and the server terminates the protocol.

5 Conclusions

Our analysis has revealed no glaring security flaws in the My.MP3.com Beam-it protocol. A user must have possession of the original music CD (or a bit-for-bit perfect copy) in order to "beam" a disk to the My.MP3.com server. Our analysis did reveal some privacy issues, but the My.MP3.com architecture fundamentally compromises the privacy of its users in order to provide a centralized service. Users desiring privacy can continue to use traditional MP3 "ripping" software.

While we had to reverse engineer a closed-source module within the Beam-it client, we found no compelling reason for this client to have its source code hidden. The security of the system is not dependent on the module's secrecy.

Future work might consider more comprehensive analysis of the challenges presented by the server, looking for patterns in the requests. Also, our replacement for the msp module was not particularly robust in the face of error-handling within the Beam-it protocol. Additional effort could yield a fully compatible open-source replacement.

References

[1] FRAUNHOFER IIS. *MPEG Audio Layer-3*. `http://www.iis.fhg.de/amm/techinf/layer3/index.html`.

[2] INTEGRITY SCIENCES, INC. *Publications on Strong Password Authentication*, Jan. 2000. `http://www.integritysciences.com/links.html`.

[3] REUTERS. RIAA may sue Lycos over MP3. Wired News, Mar. 1999. `http://www.wired.com/news/news/culture/mpthree/story/18723.html`.

[4] RIVEST, R. The MD5 message-digest algorithm. Tech. Rep. RFC-1321, Internet Engineering Task Force, 1992. `http://info.internet.isi.edu/in-notes/rfc/files/rfc1321.txt`.

[5] ROBERTSON, M. Top 10 things everyone should know about MP3, July 1998. `http://www.mp3.com/news/070.html`.

[6] SMITH, R. M. The RealJukebox monitoring system, Oct. 1999. `http://www.tiac.net/users/smiths/privacy/realjb.htm`.

[7] SULLIVAN, J. RIAA suing upstart startup. Wired News, Nov. 1999. `http://www.wired.com/news/business/0,1367,32559,00.html`.