

## CS166 Project 3: SecurePlayer

Due on Thursday, April 9, 2009 at 11:59 pm EST

Project Out: March 18, 2009

### Silly Premise

You are an everyday student of Brown University. As a typical college student, there are few things you enjoy as much as listening to some music, free from the fascist constraints of everyday life full of professors, bosses, and the RIAA. With this in mind, you started using Brown's new music service, SecurePlayer, to download and listen to music on demand. There are two hitches to the SecurePlayer service, however. First, they offer mostly music you've never heard of. The second hitch is the "Secure" part of SecurePlayer. While the music player portion of SecurePlayer offers the ability to play MP3 music formats, you can only download the MPS music format (which the player also plays). Unlike MP3s, the MPS format disallows you from sharing music with your friends. Well, this is a serious problem, because you'll be damned if you're going to rock out with the man and not illegally share your music! Thus, your quest to get your friends your new copies of Johnny Cash begins...

### Instructions

SecurePlayer is a music player equipped with a digital rights management system. SecurePlayer plays both MP3 and MPS files, where MPS is an encrypted music format. SecurePlayer can connect to a music server and download an unlimited number of songs from the server. However, songs downloaded from the music server are stored as MPS music files. To play MPS files on your computer, you need to enter your password, which enables the decryption of the file. Since you don't want to give your password out to other people, or else they can mess with your server account, only you can play the MPS files.

To get set up for the project, copy over the following directory into your CS 166 project space:

```
/course/cs166/asgn/secureplayer/files
```

This will give you `SecurePlayer.jar`, the executable file of SecurePlayer. The interface of SecurePlayer is rather simple. There are three windows to use. One is the *track control*, where you can play, pause, fast forward, and generally control the track you're listening to. The second is the *playlist*. It is here that you add tracks to play. Just go to the top menu, select "add tracks," browse for any MP3 or MPS files you might have, and add them to the playlist. From there, double click to play them in the play window. Finally, the *music service* window allows you to connect to the music server and download music. When you connect, a list of available tracks appears. When you click "download" on a track, it adds the music file to the `./music` directory in your current directory.

Also of note is that you can use SecurePlayer either in a virtual machine (VM) or outside of one. If you want to use a VM, use shared folders to bring it into your Linux VM from SynCity. We will have a music server available both in the VM network and outside of it. You are free to use either one. The only catch is that you can only use tools such as Ethereal and nmap (which require root permissions) from within the VM, but we also recognize that it's annoying to always use a VM. Thus, it is your choice, and you can mix and match as you please.

If you are working inside of a VM and are thus using the virtual net, connect to 192.168.100.113. If you are working from within the department, but not within a virtual machine, connect to engelbart. However, do not SSH into engelbart as that will drastically slow down everyone involved. Your password will be mailed to you shortly after this assignment goes out.

The goal of the assignment is to devise and implement as many methods as possible for getting the MPS music downloaded by SecurePlayer into MP3 format. We will allow almost any methods as there probably are several we haven't come up with ourselves. We will NOT, however, accept re-recording the music (by playing it and recording it) as a solution. If you're ever in doubt about a solution, please feel free to go to TA hours and ask the TA about it.

You need to find and implement a minimum of four significantly different methods of "freeing" the music. In addition, you are competing against the other students for how many methods of attack you can find. We will be keeping track publicly of how many attacks people have found, so get creative!

Beyond that, it's up to you how to go about getting the music to your friends. Enjoy, and good luck!

## Handing in

Handing in will be done identically to the other projects.

You will be required have a README that documents everything you did and if there are any bugs in the project. You should also comment your code so that we can read it easily.

## Competition

We will be running a friendly competition, recording how many exploits you and your peers have found. This will provide all involved with an idea of the general standard for success, and should be fun. Whenever you find an exploit, email the cs166tas (@cs.brown.edu) with "[exploit]" as the subject of your message and a description of your exploit so we can update the scoreboard. The scoreboard will have columns for total exploits found, exploits you've found which you were the first to find, and exploits which only you have found so far. Your e-mail should contain a DETAILED description of the exploit, as well as evidence that you have implemented and tested this exploit to prove to us that it works. This explanation should be rigorous. If you would like to remain relatively anonymous on the scoreboard, we will allow codenames. Every exploit e-mail should include the alias/codename you would like to be posted as.

## Grading

We will be having interactive grading for this assignment. We will be grading you based not just on functionality but also on the design, efficiency, commenting, and documentation.

## Special Notes for this Project

On this project, and this project only, you may assume that it is OK to use any third-party programs, tools, etc. that you can find. We want you to be as creative as possible in how you go about ripping this program apart.

The collaboration policy will be enforced especially strictly on this assignment. You may not, under any circumstances, discuss techniques, tools, ideas, etc. for finding or implementing exploits with anyone not on the CS166 staff. We will be having a special wrapup lecture to discuss the exploits found for SecurePlayer. To be safe, you should not discuss any aspect of the project with other students until after this discussion.

## Internet Lab Rules

For this project, some of your coding may be carried out in virtual machines running in the Internet Lab. You may use VMware shared folders to back up your code into your normal home directory or into your CS 166 project directory, as a safeguard against VMware crashes.

The Internet Lab has around 16 individual machines, and it should be obvious that this class has significantly more students than machines. Keeping this in mind, you might want to plan your schedule so that none or very little of your work is compressed into the final moments before this project is due. If it becomes necessary we will institute a wait list for machines in the Internet lab. Our use of the Internet lab is contingent upon everyone respecting the computers in there, not carrying drinks in, and not abusing access to this lab. Doing any of those things would lead to 166 being thrown out of the Internet Lab, and a total existence failure for most of the course. This would not bode well for you, and you should hence not engage in such actions.