Surreptitious Software — Exercise

Attacks
5

Christian Collberg
Department of Computer Science, University of Arizona
July 11, 2010

Crack — Differential Attacks

p. 81

In this exercise you'll see how it's possible to discover the location of a watermark by using a *differential attack*, i.e. to compare two differently marked copies of a program.

The Program

In this version of the player program we've added a one-word watermark, an identifier that uniquely identifies the person who bought the program. Here's the version Bob bought:

```
uint play(uint user_key, uint encrypted_media[], int media_len) {
   int code;
   *key = user_key ^ player_key;

   int i;
   for(i=0;i<media_len;i++) {
      uint decrypted = *key ^ encrypted_media[i];
      asm volatile (
         "jmp L1              \n\t"
         ".align 4            \n\t"
         ".long     0xb0b5b0b5\n\t"
         "L1:                 \n\t"
      );
      float decoded = (float)decrypted;
      fprintf(audio,"%f\n",decoded); fflush(audio);
   }
}
```

And here's the version Alice bought:

```
uint play(uint user_key, uint encrypted_media[], int media_len) {
   int code;
   *key = user_key ^ player_key;

   int i;
   for(i=0;i<media_len;i++) {
      uint decrypted = *key ^ encrypted_media[i];
      asm volatile (
         "jmp L1              \n\t"
         ".align 4            \n\t"
         ".long     0xada5ada5\n\t"
         "L1:                 \n\t"
      );
      float decoded = (float)decrypted;
      fprintf(audio,"%f\n",decoded); fflush(audio);
```

```
    }
}
```

The Differential Attack

1. Compile the two programs:

```
> gcc -O0 -falign-functions -o bob bob.c
> gcc -O0 -falign-functions -o alice alice.c
```

2. Compare the binary code using `vbindiff`:

```
> vbindiff bob alice
```

   Use the space-bar to go through the binary. Did you find any interesting differences?

Preventing a Differential Attack

How can we make it harder for an attacker to perform a differential attack? Well, we can try to make `bob.c` and `alice.c` as different as possible!

1. Make some random changes to `bob.c` and/or `alice.c`. Any changes are OK, as long as they don't alter the behavior of the program; For example:

   (a) Add reduntant (non-functional) code;

   (b) Reorder statements and/or functions;

   (c) Replace code with code that looks different but has the same behavior (for example, replace `x = y * 2` with `x = y + y`).

   These code transformations are called *code obfuscations* and we will learn more about them in Chapter 4 of the book.

p. 202

2. Compile `bob.c` and `alice.c` as before.

3. Use `vbindiff` to compare `bob` and `alice`. Is it now harder to find the watermarks?