Computer Science E-1
Spring 2010
Scribe Notes

Lecture 7: April 5, 2010
Andrew Sellergren

Contents

Computer Science E-1
Spring 2010
Scribe Notes

Lecture 7: April 5, 2010
Andrew Sellergren

1 Introduction (0:00–5:00)

- Why did David and Dan buy an iPad? In short, because it exists, they had to have it. But taking a moment to rationalize his decision, David argues that the user interface for computers has been overly complicated and largely unchanged for a number of years. That so many people have so many seemingly stupid computer questions is a testament to how (probably unnecessarily) complicated these machines really are. Do Dan and David really need an iPad? Probably not. Do they want one to play with? Yes.
- Should you wait for the second generation of the iPad? Perhaps. David initially was disappointed by the iPhone 3G and returned it for his Blackberry. However, when the 3GS came out, he gave it a try and ended up loving it. So, yes, there's a chance that the second generation of the iPad will be much more satisfying to you than the first generation, but given that Apple has now had time to work out the kinks of the iPhone, hopefully they'll get the iPad right the first time. The best solution is to play with it in the store before you buy it.

2 Security (5:00–112:00)

2.1 Hackers (5:00–39:00)

- The term *hacker* is a loaded one. Although it didn't originally, the term has taken on a pejorative connotation in some circles. Truthfully, a hacker is best defined as someone who is savvy with computers. You might say that a computer science major at MIT is a hacker just as readily as you would say that someone who gained access to the Department of Defense's network is a hacker.
- A hacker might write a script that would circumvent a program's prompt for a serial number. A *program* is simply a set of instructions which a computer can understand and execute. In the case of a piece of software that requires a serial number, the instructions might be summarize as *a*) load program *b*) ask for serial number *c*) wait for input. Of course these instructions are not written as English sentences, but rather as clusters of zeroes and ones. Just as they can be used to express letters and words in ASCII, patterns of zeroes and ones can be used to express arithmetic operations like add and subtract. A hacker who understands these zeroes and ones—or rather, the programming languages which are eventually converted to those zeroes and ones—could skip from step 1 to 3 and bypass the serial number check. Alternatively, they might figure out how to generate a seemingly valid serial number based on the pattern inherent in other legitimate serial numbers.
- To combat these hackers' efforts, software developers might issue new versions of their software with improved defenses. They might also make

each copies of the software slightly different so that one copy's key won't unlock the others. This, of course, is expensive, both in terms of time and money. And, for the most part, no defense is utterly impenetrable.

- Microsoft's newest initiative involves scanning a computer's hardware configuration and using it as a kind of fingerprint to identify that computer and associate it with a software license. A problem arises, however, when the owner of the computer switches out a piece of hardware and suddenly his fingerprint has changed. As the owner, then, you must call up customer service and explain the situation if you ever need to reinstall that software. Unfortunately, there's not much to stop a determined adversary from doing the exact same in order to illegitimately gain a software license.

- Question: could you make the case that streaming any sequence of zeroes and ones, even if it's the exact sequence that corresponds to Adobe Photoshop, is an act of free speech? Sounds interesting, but probably isn't viable as a legal argument. The precedent would almost certainly indicate that this is copyright infringement.

- What is the purpose of hacking? Some might say that it's to prove it can be done. Some feel it's not as clearly wrong as, say, breaking into a store (not that we advocate either). Another motivation, of course, is money. If you can hijack a shared computer on a university campus and use it to distribute pornography, you don't have to pay for disk space or bandwidth, but you might make money off the advertisements.

- A few buzzwords: *warez* is illegally distributed software, a *zombie* is a computer which has been seized by a hacker for malicious purposes, and a *botnet* is a collection of zombies. In general, the idea is that if you can carry out malicious tasks through a computer which you don't personally own, it will be harder to trace them back to you. Also, the more computers you can amass to do your bidding—like attacking a server with phony requests—the more of a threat you pose.

- Most compromises in security arise because the human who was programming the software failed to account for a scenario and thereby introduced a *bug*, or a glitch in the program. After all, it's virtually impossible to imagine every possible input a user might provide. Most user input, even that which hasn't been imagined by a programmer, is innocuous. Consider if we were to enter the URL for the course website but to add a long string of random characters to the end. If we do this, we get a HTTP 403 Forbidden error, which is uninteresting. But what if instead we ended up crashing the server? Although this isn't directly beneficial to us, it does reveal a weakness of the server which we might be able to exploit in other ways. Next we might try passing a phrase like "DELETE FROM DATABASE" via the URL. If the programmer was stupid enough to pass this input directly to his program, it might end up doing some damage.

2.2 Buffer Overflow Exploits (39:00–44:00)

- Most usernames and passwords are short in length. When a program stores this user input, it seems reasonable that it might only allocate space for usernames and passwords of reasonable length, perhaps 50 characters. But what if a user types in more than 50 characters? When this input is stored, it will overrun the bounds of the memory allocated for it. In other words, if we planned on user's input being less than 50 characters, then we will only have allocated 50 bytes for it. If we try to write more than 50 characters to that memory, we'll end up writing characters to memory that we didn't explicitly ask the operating system to use. A problem arises if that memory we don't own that we've written to actually contains instructions for the program. Now we've overwritten them and probably prevented certain instructions from being carried out.

- What's more, if the user input is not only long, but also well-crafted, the program might not just crash, but rather do something completely different than it was intended to. Within his input, a malicious user could include instructions in binary so that when the program's original instructions are overwritten, they will be overwritten with the malicious user's instructions. Now, the program has been hijacked.

- What's the defense? Surprisingly, it's very simple and effective although frequently overlooked. All the program needs to do is check the length of the user's input and make sure it fits in the memory that's been allocated for it.

2.3 Viruses and Worms (45:00–55:00)

- Computer viruses are named after their biological counterparts because both of them require a host in order to live and propagate. In other words, if your computer is infected with a virus, it's probably because you executed a program that was infected which, when run, latched itself onto your hard drive. This is why you should never open attachments from sources you don't trust!

- Worms are more insidious because they don't require user interaction in order to propagate. Once a worm is unleashed, it will automatically attempt to connect to and infect other computers on the network.

- Popular anti-virus programs include McAfee, Norton, and AVG. These programs scan your hard drive for patterns of zeroes and ones that represent recognized viruses. That's why your anti-virus software frequently updates its database with virus signatures that it will scan for. What's the problem with this approach? Obviously, the anti-virus software is always one step behind. If a virus is relatively new or even obscure, your anti-virus software may not recognize it. Zero-day attacks are those launched

by malicious users to exploit security vulnerabilities in software or networks before the developers become aware of it and are able to fix it.

- Question: is it possible to send viruses via e-mail but not as an attachment? Not really, no. A while ago, there was a virus which somehow embedded executable code in a JPEG image file. This security hole has since been plugged.

- These days, even hardware that you buy can be infected with a virus. Consider an employee of the company that makes your USB flash drive who was disgruntled or perhaps just unaware that his computer had been compromised. Now every flash drive he loads with software will be infected and the moment you plug that flash drive into your computer at home, your computer might also be infected.

- The more complicated a computer is, the most vulnerabilities it has. For this reason, devices like the iPad are actually compelling in their simplicity because they are more "locked down" and thus less likely to be infected with a virus or worm. Interestingly, through a process called jailbreaking, even these devices can be unlocked. As a result, they become more powerful but also more insecure.

- One approach you might consider is defenses in depth, which essentially means installing multiple anti-virus programs in the hopes that whatever one misses, the other will pick up. However, anti-virus programs tend not to play nicely with one another, so honestly your best bet is just to practice safe computing and not open suspicious attachments and programs!

2.4 SSH (55:00–67:00)

- We've already discussed HTTP which is a protocol for sending and receiving data from websites. SSH is a protocol which enables control of a remote computer without all the extra data of graphics and the like. SSH allows you to interact with a remote computer via a simple black terminal window with a command line that only takes specific textual input. It might sound arcane, but it's tremendously useful in administering servers.

- Because Mac OS is Unix-based, there is a built-in terminal with which you can interact using these simple text commands. If we open up the Terminal application and type `ls`, we get a list of the contents of the directory we're currently in. We can also run the `ssh` command to connect to and control a remote computer. If you jailbreak an iPhone, you can install an SSH server so that you can actually connect to it remotely from any other computer.

- The security holes which make jailbreaking possible are constantly being patched by Apple because the devices were never intended to be used in this way. If you don't know what you're doing, jailbreaking an iPhone and

installing an SSH server on it can be a seriously bad idea. When installing an SSH server, there is a default username and password that are enabled. If you don't change these credentials, then anyone who knows this default username and password will be able to login to your iPhone.

2.5 Phishing (67:00–73:00)

- Phishing is an attack in which malicious users send out e-mails impersonating major companies like Bank of America or Facebook. These e-mails may look very real and the websites they link to might look identical to the real thing, which is why some people actually fall for it when they are asked to provide their login credentials. It may seem ridiculous, but consider that it costs the adversary virtually nothing to send out thousands of e-mails and if only a hundred people fall for the scam, he has still succeeded.

- Question: is phishing the only way to steal passwords? No, but it's probably the easiest. Other attacks include hacking into a website's database or a man-in-the-middle attack in which an adversary places himself between you and the server, sending and receiving all data, including your login credentials, to both ends.

- One way of defending against phishing attacks is to make sure you type in the URL yourself whenever you access an important website. That way, you won't be fooled into entering your username and password on an illegitimate website. If you're not sure, try calling customer service! Chances are that legitimate websites will never ask you for your username and password for any reason other than to login to the site.

2.6 SSL and Cryptography (73:00–93:00)

- We've discussed how SSL encryption can be used to defend against a man-in-the-middle attack: with SSL, if an adversary intercepts the packets you were trying to send to the server, they'll be of no use to him since he can't decrypt them.

- SSL is a practical implementation of cryptography. Let's consider a simpler example. Say that you wanted to send the message "HELLO" to a friend without anyone in between being able to understand it. You and your friend might agree to shift each letter in the message up 1 place in the alphabet. The message would then become "IFMMP." Perhaps more devious would be to shift each letter by 13 places so that the message would be "URYYB." This second scheme is called ROT-13, meaning "rotate by 13."

- Of course, neither of these encryption schemes, which are known more generally as Caesar ciphers, is very hard to crack. If your adversary knows

that each letter is shifted by the same amount, he only has to try a maximum of 26 different combinations to find the correct message.

- A slightly more complex encryption scheme is the Vigenère cipher. Instead of rotating each letter by the same amount, we'll rotate them by different amounts. We can use a keyword, each of whose letters represents an amount by which we'll rotate a character of the message. So if our message is still "HELLO" and our keyword is "KEY," we'll rotate H by 10 (because K is the 10th letter of the alphabet, assuming A is 0), we'll rotate E by 4 (because E is the 4th letter of the alphabet, assuming A is 0), and so on. In this way, there are many more possible combinations that an adversary would need to consider before he stumbled on the correct message.

- Modern implementations of cryptography are much more complicated (and thus much harder to crack) than Caesar or Vigenère ciphers. Public-key cryptography is analogous to sending a letter and stamping the envelope with a wax seal. If that letter gets to its destination but the seal is missing or broken, the recipient will know not to trust the information in the letter.

- Public-key cryptography, which is employed by SSL, relies on two very large prime numbers called a private key and a public key. Both the sender and the recipient have their own private keys which are mathematically related to the shared public key. The public key, as its name suggests, can be widely distributed. It is used to encrypt the message that's being sent. Because of the mathematics behind this scheme, only the recipient's private key can decrypt the message. Even if an adversary gets a hold of the message and the public key, he can't decrypt the message without also stealing the private key.[1]

- Private-key cryptography is simpler than public-key cryptography. Every message that is sent is both encrypted and decrypted by the private key which both sender and recipient know. The problem, of course, is how to send that private key to the recipient without it being intercepted. One clever solution is to use public-key cryptography to send the private key. Once both parties have the private key, then private-key cryptography can be used.

- Imagine a box whose contents you want to secure. You can, of course, put a padlock on it and then send the key to the recipient. Having a single lock on the box that can be opened by a single key (copies of which are possessed by both sender and recipient) is analogous to private-key cryptography. As before, we have the problem of how to send the key to the recipient without it being intercepted. To do this, we might use a method analogous to public-key cryptography.

[1] Dan actually misspoke in lecture and said that the private key is used to encrypt and the public key is used to decrypt. Apologies for any confusion this might have caused!

2.7 Firewalls and Ports (93:00–97:00)

- In order to watch TV on a plane, David must configure his Slingbox to accept external connections. In doing so, he is making his network somewhat more vulnerable to attack. However, because his router implements a firewall, he can configure it to deny external connections to all ports except the single one which the Slingbox is listening for. That way, an adversary will need to know not only the login credentials but also which single port out of tens of thousands he can connect to on David's network. The risk is thus much smaller than if all the ports are open.

- Similarly, if you wanted to configure your home computer to serve up web content so that you could display your work to others, you might want to change the HTTP port from 80 to something that's hard to guess. This is another example of security through obscurity in that an adversary, who might otherwise easily guess that port 80 is open on your home machine, will have to work a little harder. This is a common theme in the world of computer security. Although we can't eliminate threats entirely, we can at least raise the bar a little higher for adversaries.

2.8 Digital Rights Management (97:00–111:00)

- Digital Rights Management (DRM) is a way for companies to protect copyrighted material. Music you download on iTunes used to be protected with DRM so that only you could listen to it and only on a specific device.

- Although the iTunes Store is now DRM-free, you still cannot download a song and share it with your friends. Your Apple username is actually embedded in the file when you download it. So, if you did manage to find a way to share a song with a lot of friends and Apple found out about it, they'd know right where to find you!

- Of course, it's easy enough to circumvent DRM. If you simply burn a CD with iTunes songs on it, you can give it to your friend who can then rip the songs off it.

- Many DVDs come with encryption nowadays, but almost every form of it has been cracked. The first example was CSS encryption which was cracked by Jon Lech Johansen of Norway.

- Sony briefly experimented with DRM on CDs. During this period of time, the CDs they released had a bit of code on them that would install software on your computer when you first inserted the disc. This software would then prevent you from doing things like copying the songs onto your hard drive. Because this software installed itself without the user's permission, many people considered it malware and filed suit against Sony. Sony, in response, released programs to remove this software from computers but took a few tries to get it right.

- In the world of television, there is also a flag that is included in the digital broadcasts of shows that often prevents that show from being copied between devices. Interestingly, only some DVRs pay attention to the flag. Not to mention that the data isn't encrypted in any way, so there's nothing to stop you from copying the raw bits from one device to another despite this flag.

2.9 The Take-away Message (111:00–112:00)

- If you take nothing else away from this lecture, take away this advice: back up your data! You should always have a back up of your data on a drive or discs that aren't connected to your computer. That way, if your computer is ever infected with a virus or worm, you can restore it back to normal. If your back up is connected to your computer, however, then it too might be compromised by the same virus or worm.