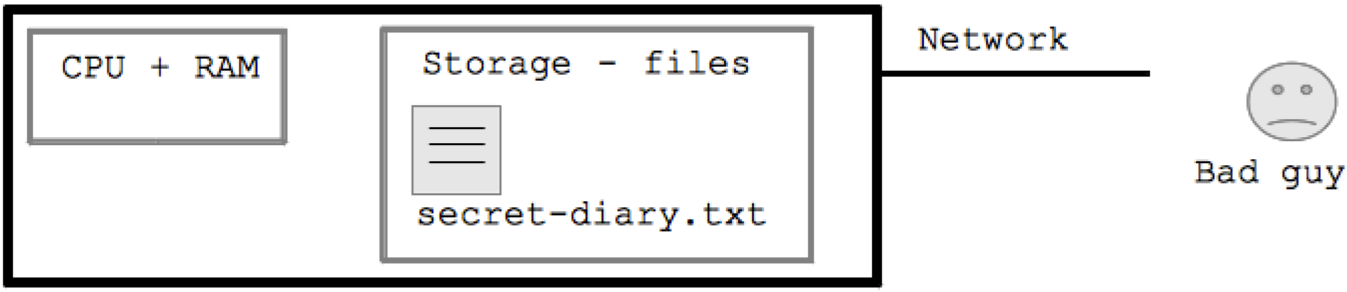
Security 1 - Passwords

Computer security is a big and kind of dramatic area which lends itself to movie plots and fear. There are real threats our there, but staying safe is not that hard.

Computer -- The Castle

- The computer is like a castle with walls
- Inside and outside are very different
- Bad guy cannot just access the bytes inside inside the computer at will
- Bad guy will need to work at it
- A couple bad guy strategies:
  - obtain a password allowing access
  - trick the computer into running bad guy code
- This lecture is scary, but we're going to be ok



Computer - "castle" model

Computer Attacks

In the following sections we'll look at the three most common types of attack, lumping into broad categories: 1. Password attacks, 2. Phishing attacks, 3. Malware attacks.

Typical Bad Guy Attacks - Bulk

- Typically the bad guys are not crafting some attack just for you
- They send out millions of generic attacks, just snaring who falls for it
- If you avoid the most common errors, you will probably be fine
- We'll concentrate on this typical case

- I don't have any anti-virus software on my computer, and I have not had any problems (I'm not running Windows which probably helps me)

Although I'll talk about problems most of the time, don't get all scared. I use the internet all day long, I don't have any anti-virus software installed, and I have not had any problems (that I know of!). It probably helps that I don't use any Microsoft software, which is a popular target.

Aside: Atypical Spear Phishing Case

- "Spear phishing" - rare
- A specifically crafted and sophisticated attack against a specific person
- Likely to succeed if the attacker has money and motivation
- e.g. if the CIA really wants your files, they will get them
- Except "encryption" which nobody can break (a later topic)
- We won't concentrate on this case

Password Dictionary Attacks

A favorite CS101 question: list all the ways a bad guy can get your password? We'll go through them.

- The bad guy could try to guess your password to a site
- This is the "outside" case - bad guy is outside the site, guessing
- Known as "dictionary attack"
  - as if they are trying all the words in a dictionary
- Bad guys tries to log in again and again
- Bad guys will try common passwords as guesses
- Works if the password is common, e.g. "password" or "password1"
- The attack fails mostly, but works some percentage of the time with an account with a weak password
- There are 86400 seconds in a day
- 1 guess/second = 31 million guesses per year
- There is not time to make 100 billion guesses
- Just avoid the weakest 10 million passwords, probably ok

The bad guy could try to just guess your password, attempting to log in again and again, hoping to get lucky. They might know the username and just guess the password, or more likely they are guessing both. There are 86400 seconds in a day, and suppose your bank permits 1 login attempt per second. The bad guy could just go through the list of 100000 common passwords ("password", "password123", "janexyz", ...) trying to get lucky. This is good enough for the bad guys. Since they launch the attack in bulk, just getting a fraction of a percent is worthwhile.

Clearly, the bank or whatever should detect thousands of bad logins and slow down or freeze the account. This can cause problems for the legitimate user however, so it's a balance. One simple policy is that the Bank can process login attempts at a slowish rate, such one every second to prevent the bad guys from trying 100 billion different passwords.

Dictionary Attack Example

Here's a real "log file" from my codingbat.com server where it routinely records what happens each day. What you see here is the attacker is trying guess both the username and password on the account. It happens that the username for each attempt is printed in the log file but the password is not. No doubt they are trying common passwords, such as "secret" "password12" etc. It's funny to me that you can see that their list of usernames to try is sort of alphabetical order, and they are just running through it in the most obvious way. So what you need to understand is .. this sort of attack is clicking along, every second of every day aimed at basically all the servers on the internet. They just need to succeed with a few accounts here and there, even though they fail 99.99% of the time. This is why you should not have a password which is close to a dictionary word or someone's name, or is a password people often choose. The good news is .. with just 4 random letters added to your password .. suddenly this dictionary attack is not going to work -- there's not enough seconds in the day. Note that 49.212.7.205 is the IP address of the machine attacking codingbat.com. It appears to be in Japan -- it's probably some person's Windows machine that has been compromised and is now used as a "zombie" under the control of the bad guy to launch more attacks. The zombie is probably running attacks at many servers all at the same time, but here we just see the ones directed at codingbat, about one login attempt every 3 seconds.

```
...

Mar  6 06:26:20 codingbat sshd[30924]: Failed password

for invalid user alex from 49.212.7.205 port 36268 ssh2
```

```
Mar  6 06:26:22 codingbat sshd[30926]: Failed password
for invalid user alex from 49.212.7.205 port 36605 ssh2
Mar  6 06:26:26 codingbat sshd[30928]: Failed password
for invalid user alex from 49.212.7.205 port 36937 ssh2
Mar  6 06:26:29 codingbat sshd[30930]: Failed password
for invalid user adam from 49.212.7.205 port 37212 ssh2
Mar  6 06:26:32 codingbat sshd[30932]: Failed password
for invalid user fax from 49.212.7.205 port 37546 ssh2
Mar  6 06:26:34 codingbat sshd[30934]: Failed password
for invalid user fax from 49.212.7.205 port 37864 ssh2
Mar  6 06:26:38 codingbat sshd[30936]: Failed password
for invalid user demo from 49.212.7.205 port 38201 ssh2
Mar  6 06:26:41 codingbat sshd[30938]: Failed password
for invalid user demo from 49.212.7.205 port 38561 ssh2
Mar  6 06:26:44 codingbat sshd[30940]: Failed password
for invalid user amanda from 49.212.7.205 port 38911
ssh2
Mar  6 06:26:47 codingbat sshd[30942]: Failed password
for invalid user angie from 49.212.7.205 port 39244
ssh2
```

```
Mar  6 06:26:51 codingbat sshd[30944]: Failed password
for invalid user angie from 49.212.7.205 port 39552
ssh2
...
```

Weak Passwords

- The bad guys have lists of the top few millions of common passwords
  -words and puns and tricks are on this list
- Patterns of weak passwords to avoid
- 1. Passwords should not be a plain word
  - kittens
- 2. Passwords should not be too short - 6 characters is marginal, longer is better
- 3. Passwords with only lowercase letters are weaker
  - upper case, digits, punctuation are all stronger
- 4. Passwords should not be a pun or pattern that someone else would think of (this one is the killer!)
  - opensesame
  - qwerty123
  - catfish
  - remaincalmandcarryon
  - these sorts of passwords are on the common password list
  - When asked to make a random, memorable password, the pun instinct is strong!
- 5. When required to add a digit to a password, many people just add 1 at the end
- Here is a list of commonly used passwords, most popular at the top, basically demonstrating all the patterns of bad passwords:
- `password`
- `password1`
- `123456789`
- `12345678`
- `1234567890`
- `abc123`
- `computer`
- `tigger`
- `1234`
- `qwerty`

Weak Passwords - The Bad Guy Perspective

- How do bad guys guess passwords?
- 1. Dictionary of words
- 2. List of commonly used passwords from other sites
  - this includes whatever joke or pun you are thinking of!
  - they are more likely to use (2) than (1), better hit rate
- 3. Heuristic changes (scary)
  - say bad guys have "catfishr" from their list
  - bad guy code tries variations automatically:
  - catfishr1
  - catfishr2
  - cat.fishr
  - iheartcatfishr - (add common stuff on the ends)
- Therefore: our strategy must avoid anything from the common list

Strong Passwords

- Passwords do not need to be super elaborate to be secure (some sites go crazy with this)
- What makes a password stronger:
  - longer
  - more characters: lower case, upper case, digits, punctuation
  - not a word or pun
- Here is what I do for secure passwords, e.g a bank site
- Start with a word, say "kittens"
- Change it with a random misspelling, then add some random stuff
  - kottens4x -- simple but fine password
  - not a word, not a pun, not digit-at-end
- Here are stronger versions
  - kottens,erx -- better
  - Kottens,9erx -- better
  - KottensX,97erx -- probably more complex than necessary
- Key: the random misspelling cannot be a joke or pun

Password Outside Guessing vs. "Cracking"

- Thus far the "outside" case
  - The bad guys is outside the site, guessing passwords at the rate of 1/sec or so
- vs. the "cracking" case:
- The bad guys has stolen all the encrypted passwords from the site itself
- "cracking" is trying to decrypt the stolen passwords
- Cracking can be done at rates of a billions of guesses per second
- Therefore: If a site is compromised, assume the passwords will be exposed
- ArsTechnica Cracking Article - yikes!
- Scary example:
  - cracked password was: momof3g8kids
  - bad guy list of 111 million passwords from around the web
  - "momof3g" was on the list of 111 million
  - bad guy had shorter list with "8kids"
  - bad guy just tried all the combinations!
  - With a billion guesses per second, you can just do that
- Just remember 2 things:
- 1. If a site is compromised, your password for that site will be cracked
  - we are going to lose to the billions / second guesses
- 2. This is why you cannot re-use passwords across sites!

What To Do

- Avoid weak passwords
- Don't have to go crazy with it
- Bad guys are probably guessing thousands, not billions on you
  - e.g. kottens4x is not terrible
- Don't re-use passwords across sites
- Do consider writing down important passwords
- Not all passwords need to be super secure
- Email password is extra important, due to password re-sets
- 1 scheme: memorize suffix "x23" for passwords
- Write down passwords but not the suffix

For an important site like a bank, you should use a password different from your other passwords. It should not be the case that by stealing your facebook or twitter password, they now have access to your bank. I write the passwords down on a piece of paper at my house in case I forget. The bad guy in Russia or whatever does not have some team of ninjas that's going to break into my house and get passwords off my slip of paper. The attacks are bulk, mindless affairs that work on the low-hanging fruit. One technique for writing down passwords is to pick a little suffix you memorize, like "x936" or whatever, and that always goes on the end of your passwords. Write the passwords down, but never the suffix. That way, even with the piece of paper, a bad guy still does not have the passwords. Or maybe its better to just write the passwords out clearly, so your family can access your email etc. if you are in the hospital.

Email is tricky -- once they have your email password, then they may be able to do a password reset and get into your account. In that sense, your email password is the most important.

Two-Factor Authentication

- A second piece of information to log in, a "second factor"
- aka "Multi-Factor Authentication"
- Password is 1 pice of info
- Require 2nd info to log in
- example 1: The site texts a little number to the user's registered cell phone
- example 2: The user has a free One Time Password (OTP) app on their phone
- example 3: U2F (below)
- Two-factor makes it much more difficult for the bad guy
- Although not impossible
- Great side effect: with two-factor, perhaps the password can be simple "kitten2"
- Ideally, 2nd-factor not required every time
  - maybe once a month or from a new computer
- Two-factor may still fail for phishing (not U2F though)

Next-Generation Two-Factor: U2F

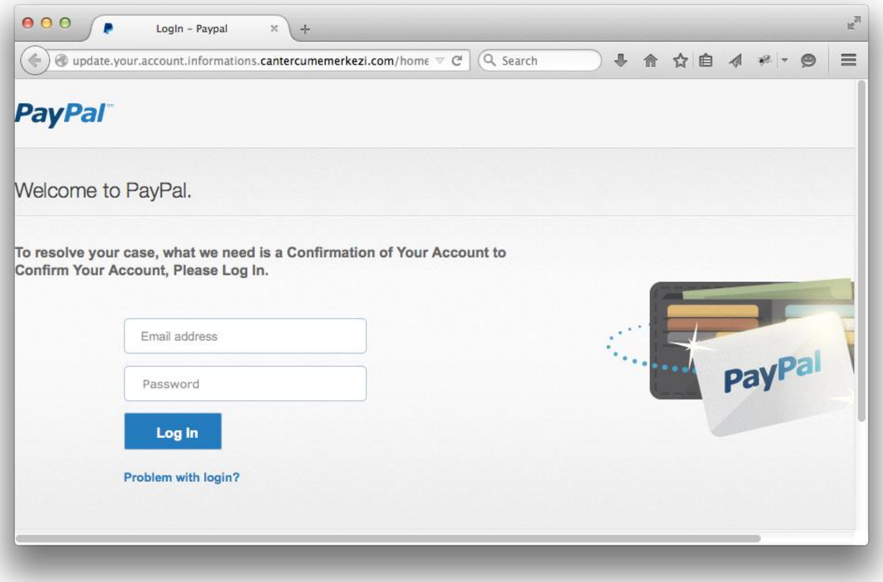
- **Device** is better than passwords in our heads
- The world will not continue to use passwords as we know them today
- The new free and open Fido U2F "universal two-factor" shows a next-gen solution
- Right now works in Chrome
- It's an inexpensive little device you can carry around, it has one button on it
- You click a button on the device when asked to prove you are you, and the rest is automatic
- Secure and convenient - you don't have to type anything.
- Ultimately your phone will work as a U2F token too
- It is also phishing proof - click the button on a "bad guy" site and there's no harm
- U2F is so secure, the password can be trivial, like 4 digit PIN or perhaps nothing

Security 2 - Phishing

2. - Phishing Attacks

- "Phishing" is a type of attack where the bad guy tricks you into typing your password into a bad guy site, thus the bad guy gets your password
- Phishing and dictionary attacks (previous section) are two major ways that people's accounts are broken into
- You have probably received many phishing emails. The phishing email most often includes a link to a page such as the one shown below

Here is an example phish site from one of the many paypal phishing emails in my inbox:



Welcome to PayPal.

To resolve your case, what we need is a Confirmation of Your Account to Confirm Your Account, Please Log In.

- The above is phishing site, not the real paypal site. If you type your username and password into the phishing site, they are sent to the bad guy who can use them to break into your account.
- The graphics and coloring are correct. Those are trivial for the bad guys to copy and mean nothing.
- The title of the tab - "Paypal Login" - is also meaningless.
- How do you detect that this is a phishing site? Look at the **url**, near the top of the browser window. Does it say "www.paypal.com"? Checking that is your one defense.
- The word "phishing" is a little joke. The bad guy is "fishing" for**you**.
- Bad guys are skilled at writing provocative emails, prompting you to click, e.g. "fraud alert, click immediately"

- Alternately, the bad guy might sprinkle links to the phishing site across web in forum comments and anywhere else they can. Anything to get people to visit the page and get phished.

Probably the most common form of attack. The email is forged to appear from someone you might trust -- including logos etc -- I have gotten ones which I personally found quite convincing. I had to slow down and really pay attention to realize that it was a phishing attempt. It does not help that ATT, Schwab, Citigroup, etc. do in fact send you email all the time about your accounts.

Fake ATM Machine -- Real World Analogy

- Funny "phishing" crime in real life
- Fake ATM in front of bank .. prints error message, but records card details and PIN for bad guy

Criminals put up a fake ATM machine made of plywood in front of a real ATM, with a "under construction" sign. The victim would put their card into the fake ATM and type in their PIN. Then the machine would print an "out of order" message and give the card back. The bad guys in this way collected all the card numbers and PINs and drained the accounts over the weekend. This is a nice real-world analog of fake-site phishing.

Avoiding Phishing

- Don't trust urls in emails or sites when they lead to a login page
- The bad guy is hoping that when the username/password fields appear in front of you, you will just type it in out of a habit. When asked for your password, slow down for a moment and look.
- Technique 1: Scrutinize the url near the top of the browser window
- The bad guy url will try to look legitimate but it is not the correct url, e.g. **www.ebay.bad-guy.ru** is not the official ebay site, which should end in "ebay.com"

Avoiding Phishing - URL / typing

- The screenshot above shows this bad guy technique. The bad guy added a series of plausible words to the left of the domain name, hoping you would stop reading: "update.your.account.information.cantercumemerkezi.com". Does "cantercumemerkezi.com" seem like a site where you should be typing your paypal password?

- The 2 rightmost words of the url define the organization controlling the url, so that's what you are checking.
- Safe: accounts.google.com, login.gmail.com
- Not safe: accounts.google.bad-guy.ru, gmail.login.quicksite.hk
- Technique 2: (more secure) Type the url into your browser yourself -- if the email claims you need to log into ebay, go to your browser manually and type in "www.ebay.com". This is what I tell my parents to do, as it's a simple rule. In this way you sidestep the www.ebay.bad-guy.ru url the bad guy placed in the email.
- Chrome and the other major browsers have an ongoing effort to detect and warn users about phishing sites in realtime. Sometimes this will save you with a warning about a phishing page. If you see a phishing site, you can use the "report web forgery" menu item in our browser to report the site, contributing against the bad guys.
- Look for **https** in the url (below)

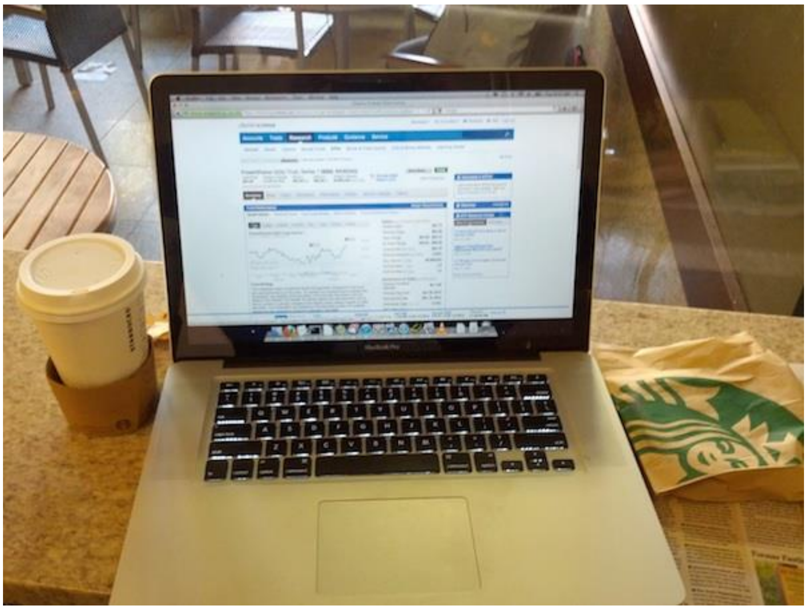
HTTPS

- HTTPS - "secure" variant of HTTP to transfer the bytes of a web page over the internet
- HTTPS does two things:
- a. HTTPS requires some paperwork to set up, so that the domain www.schwab.com domain name really is coming from the Schwab organization.
  -Helps prevent phishing, but the user still needs to look at the url
  -e.g. Checking that it's www.schwab.com not www.schwarb.bad-guy.ru
- b. HTTPS encrypts all the traffic, so interception of the bytes does not yield anything intelligible

HTTPS is the "secure" http variant, https://www.ebay.com/. In particular, the server must hold a certificate which is verified beforehand by an authority that the server really represents www.ebay.com or whatever. You have to pay money and file some paperwork to get an https certificate -- hard for a bad guy to do, although not impossible. The idea with https is that the user can see the identity of who they are talking to. In the browser interface, https is typically accompanied by a little lock icon, and some banks etc. mention to their users to look for the lock. Having users pay attention 100% of the time is not a perfect security solution, but it helps.

Encryption vs. Bad Guy Packet Eavesdropping

- "Encryption" is a way of scrambling data before it is sent out in packets, so that even if intercepted, they are meaningless
- HTTPS provides encryption, in addition to its url-verification feature
- Example: this is a picture where I went to Starbucks and logged in to their free Wi-Fi. Then I went to the www.schwab.com home page and type typed in my password to log in
- My password was sent out in Wi-Fi radio packets sent from my laptop to the Wi-Fi router and on to www.schwab.com. These packets could be intercepted by all the other laptops at starbucks at that moment.
- Probably several laptops there were infected with malware, and malware often specifically targets financial info (e.g. Schwab)
- Could someone there have gotten my password out of a packet?
- No -- HTTPS encrypted all the packets before they were sent from my laptop
- Note if my machine had malware on it, it could steal my password as a typed it in. HTTPS only takes care of the networking.



When you go on to wifi and visit a web page and type something in ... the packets for all that are just being broadcast in the room, so anyone nearby can observe the packets, listening in (recall the ethernet-packet-broadcast material from the networking section). For the most part, this is harmless. In some cases, say when you are typing in a credit card number, you want the communication to be **encrypted** (encoded), so that someone listening in cannot read it. The**https** scheme above also does encryption, so you will notice that when you go the page to type in a credit card number, the url begins with "https://". On such page, all the packets are encrypted, so someone can listen and see the packets, but they will appear to be random garbage. The eavesdropper cannot unscramble the packets to see what's inside, or forge a packet. So Https blends two security provisions -- (a) verifying that it really is the www.mybank.com or whatever server on the other end and (b) encrypting all the packets of the communication.

Security 3 - Malware

Pre-Malware Category: Social Engineering Attacks

- Social engineering - talk to people, ask for help
- Famously, social engineering techniques are very effective
- e.g. call on the phone, claim to be an employee, ask for password info
- e.g. walk in campaign office, be the same age and look as the volunteers, ask what the password is to log in
- e.g. "Hi this is Larry, I'm on the client site to give a presentation, but it's a disaster. What's the password for the share drive again?"
- Walk in wearing overalls and with a clipboard, put a little black box on the printer. Really the black box is listening to all the local wi-fi packets trying to get a password
- Social engineering works because people are generally helpful

"Social engineering" means using human to human contact, say on the phone, to get into a system. Some people can be quite persuasive on the phone, and most people are polite and helpful by default (see, we're not such a bad species!). A bad guy might pose as technician showing up, trying to fix the printer. People will often be polite to a well dressed person on site who appears to be doing something proper. An example from a few years ago was leaving USB keys in the parking lot containing malware, counting on the curiosity of those picking them up and taking them inside and plug them in to their machines. Windows has (had?) an extremely insecure "autorun" feature where it will automatically run certain code on an inserted drive. On a properly designed operating system, plugging in a flash drive lets you look at its contents, not start trusting and running the bytes found there.

Malware Attacks

This is a big category, where the bad guy tricks the victim into running bad software ("malware") on the victim's computer. I'm lumping viruses, worms, and trojans all into this category.

How Do I Feel About This File?

- Suppose a bad guy emails you the following sort of file:
- A plain .TXT file, which you open and read on your computer
- A .JPG file, which you then open and look at on your computer
- A program .EXE file -- a program or "app"
  - copy on to your computer and run it
- A .DOC document file which you then open and read on your computer

Passive Content = Safe, Program = Unsafe

- If the bad guy gets you to run bad guy authored code on your computer, the computer is compromised, the bad guy has won.
- The code can **take actions** and it's inside the computer
- "Malware" - generic term for program that does something bad
- Recall the "castle" analogy - the bad guy program is running inside the castle
- **Key:** if bad guy authored code is downloaded to the computer and runs .. the bad guys has won
- Variations below will all center on this downloaded ".EXE" case
- So we trust passive content (.TXT .JPG) but not active programs (.DOC .EXE).
- Unfortunately, many seemingly passive formats, such as .DOC, can have "program" type qualities in them as an advanced feature
- e.g. .DOC can be unsafe because of Microsoft Visual Basic macros embedded ... this used to be a huge source of problems (search for "macro virus")

The last, .exe case is the main risk here, although all of the above scenarios could result in problems. If code written by the bad guy, a .exe, runs on your computer, the bad guy now in essence has control and access to the data on your computer.

Suppose the bad guy is sitting at the keyboard of your unlocked computer, obviously they can do whatever they want with your data. The .exe case is very similar -- the code in the .exe could do what the bad guy wants -- look for a particular file, email it off the machine, etc. Most of the "attacks" listed below in essence try to re-create the .exe case, and the defenses center on preventing the .exe case.

"Malware" is the general term for a program written by the bad guy to do bad things to your machine - break in to the machine, steal passwords, send spam, etc.

Bad Guy Malware .EXE Techniques

Malware 1 - Trojan

- A "trojan" is malware disguised as something else
- So the user downloads it or accesses it, not aware that it will do something bad
- e.g. JustinBieberJPEG.exe
- e.g. SuperAntiVirus.exe -- this is actually a common Trojan ruse!
- The trojan tries to look like harmless content, not a program
- The trojan claims to be a program that does something many people want, but really it's malware
- Operating systems may have helpful warning "this is a program you downloaded, do you want to run it?"
- Therefore:
- Don't run programs from random sources
- What I do: google the name/domain-name of it first, see what people say
- If something is from a well known domain and has lots of downloads, I figure someone would have flagged it if it was malware

A "trojan" is a malware disguised as something else, like "awesome-cursors.exe" or "fun-game.exe" or "JustinBeiber.JPEG.exe" (Windows is vulnerable to extensions other than .exe, it's just used for the examples here). The term refers to the Trojan Horse story from antiquity. If the user can be tricked into double clicking the trojan, running it, then the bad guys have won. The https is no defense. On Mac OS X, heuristics have been added where it puts up a dialog "This was just downloaded and it is a program; are you sure you want to run it?"

Malware Example - Keylogger

- A good example piece of malware - its running on your computer
- Keylogger - watches the screen, what urls you visit, records all the keystrokes
- HTTPS is no help - it's **inside** the castle
- Sends this info back to the bad guy
- Easy to imagine how this could allow the bad guy to abuse your accounts
- Important: do not log into your email account from some random internet-cafe machine!

Malware 2 - Vulnerability

- Suppose there is a **bug** in the Flash web-animation program

- When fed an "exploit" of the vulnerability (a special pathological pattern of bytes) the program breaks and gives access to the machine
- So the bad guy writes this exploit file, and then sends links to it on sites or in spam
- Just visiting the page containing the exploit is enough to compromise the visiting machine if it is vulnerable
- This is probably the most scary case, as the user does very little
- Solution:
- Keep web-facing software up to date
- All browsers now have strong auto-update channels, so by default the right thing tends to happen
- "Zero-day" vulnerability = no fix is available yet
- "Exploited in the wild" = bad guys are actively using this nowWeb news search: zero-day vulnerabilityy
- What does this headline mean "Adobe flash vulnerability found, exploits seen in the wild, update now."
- Translation: a new vulnerability bug has been discovered in Adobe's flash player. Exploits of this bug are seen out on the internet. The latest version of Flash fixes the bug, install it now. Increasingly, Flash and the browsers and whatnot detect new versions and install them automatically precisely because of this "vulnerability" scenario.
- "Zero-day" means a vulnerability in the wild before Adobe has a fix out
- Aside: this is also why having a proprietary format like Flash be a key part of the internet is not a good architecture. People are very dependent on Adobe to fix the software very quickly, and Adobe's record has been very uneven.

Suppose there is an engineering flaw in Firefox or the Flash player or some other software on your machine, such that if it sees a particular sequence of bytes as input, there is bug that allows a takeover of the machine. This is called a "vulnerability", and it is one of the scary cases. If the user browses over to a web site this is hosting the "attack" content and their browser is vulnerable, then the bad guy can get it just from that. The bad guy can make the web site appear attractive, post links on reddit or whatever to try to drive traffic to the site. This attack is scary because it does not require the user to do anything especially foolish.

Such vulnerabilities in Flash and IE used to be quite common. However, the engineering culture seems to be catching up, and this case is becoming more rare. The most important step is being sure to run the most up to date, current version of your browser and any plugins such as Flash. Firefox et al have switched to make programs auto-check for new versions, so the user does not need to do much to have the most recent version. Often a vulnerability is fixed, and months later attackers start using it on sites, but they can still succeed with users using old versions.

Malware Example: Zombie Botnets

- A group of machines with malware on them allowing "bot herder" to control them
- How to obtain the zombies?
- -Bot herder sends out million emails pointing to a site that attacks a Flash vulnerability, installing the malware onto vulnerable machines
- The bot herder sends out commands for all the zombies to do something
- Botnets can be rented, there's an active botnet market in the bad-guy community (suggested interesting B-School research direction)

In parallel with other harm, the malware may set up the compromised machine as a "zombie" or "bot". A zombie is a machine, one of thousands, which all together form a "botnet". The owner of the botnet can distribute tasks to be done by all the zombies, like this: "ok everyone, here is a list of 10 million email addresses, start sending spam email to them." Because the number of zombies is large, the botnet can accomplish things that require a lot of machines. Sending spam is a great example. Another great example is doing dictionary-password attacks on random websites, as shown previously.

Malware Example - Encryption "Ransomware"

- Malware encrypts with a random password all the files on the victim machine, deletes the originals
- The victim must send the bad guys money (bitcoin) to get the "unencrypt" password
- The bad guys typically do send the password when given the money
- The scheme depends on the bad guys having a reputation of reliability
- Have backups of your important files!

Malware Example - DDOS Attack

- "Distributed Denial of Service" DDOS attack
- Attackers coordinates a large number of machines to send many requests to a site all at once

- Overwhelm the site's connection to the internet with so many packets, it becomes effectively unreachable
- When bad guys "take down" a web site in the news, typically this means a DDOS attack done with zombie machines
- **Technical Fix** many DDOS techniques depend on sending packets with a forged ("spoofed") From: IP address field. The router upstream of the attacking zombie could block such forged packets from leaving their network to help blunt DDOS attacks. Nobody is very motivated to do this currently. If I were dictator of the internet, I would require it to reduce this silly pollution. (This is a tragedy of the commons.)
- Demo: search for "ddos attack" news

The zombies can also be used to "attack" a web site, by all trying to access it at the same time. With some tends of thousands of machines all hitting a site at the same time, it is possible to in effect make the site unavailable to the internet. This is called a "denial of service" (DOS) attack. It's not breaking into the site or stealing passwords or money; instead it's making the proper function of something unavailable.

Obviously the botnet is not **paying** the owner of the machine. The botnet is stealing the use of the machine from its proper owner. If a machine seems sluggish in regular use, and the networking lights are blinking like mad all the time... the machine may be a zombie. Like a parasite in the real world, the zombie software wants the machine to still mostly work for its owner, otherwise they would be motivated to clean it.

One problem with zombies is that the owners may not be all that motivated to fix it. The millions of compromised Windows machines out there are putting out this pollution that causes problems for us all. If you think a machine is a zombie, you should erase it and fix it. The zombie may be doing who knows what with your passwords, your data, there's too many risks.

In what would make a most interesting Business School case study, there are active markets in botnets. The botnet owners basically rent out their botnets for spamming or whatever use a bad guys wants to pay for that day.

Malware Example - CEO Payment Email Scam

- Get email password of finance manager at company
- (guessing, phishing, keylogger)
- Study past emails about making wire transfers
- Study when the CEO is traveling
- Send an email like this:
- Hi Bob, Im' traveling, to grab this deal, I need you to wire $25,000 to 11243-4732626 ASAP
- May break into CEO's account to send, it or just make it look like from CEO to attack swimming.com bad guys register swimning.com, send mail from it
- Not a bulk targeted attack, bad guy effort required
- Bad guys at times have placed voice calls, pretending to be people
- Works because email about wiring money is the common practice
- Solution: have some other channel to verify, e.g. call the CEO

Security 4 - Safe Practices

How To Stay Safe? A pretty short list!

1. Password Safety

- Avoiding the bad guy guessing or obtaining your password
- Don't use a weak password for an important site (e.g. bank, email)
- -email is especially important because of password-reset
- Don't re-use passwords across important sites
- -Bad guys have software to re-try username/passwords across a zillion sites
- Don't type in your password on some random machine in a cafe (it could have a keylogger)
- -Cheap wi-fi phones are great for this case (e.g. Nick Starbucks example)
- Do write your passwords down, consider 2-factor authentication for important sites
- -Not all sites are important! Hey, your time is valuable too.

2. Phishing Safety

- Avoiding the bad guy tricking you into disclosing your password
- Something is asking for your password? Look up at the browser url area
- Something is asking for your password? Look up at the browser url area
- Something is asking for your password? Look up at the browser url area (demo)
- Watch out for clever bad guy urls: weblogin.stanford.edu-xnr-xyzldlwerou.ru
- Proceed carefully with content from email or random pages with provocative "click this" content
- Or just type in "www.schwab.com" yourself in the browser instead of clicking in the email - super simple and secure practice
- **U2F Two Factor** - solves both password and phishing problems, a technical fix
- Let's hope something like this catches on

3. Malware Safety

- Avoiding the bad guy installing software on your machine
- Trojan - be wary of downloading and running an application
- Trojan - be wary of .zip file in email
- -like phishing, what domain is hosting this thing I'm downloading?
- -google the name of the site ... lots of complaints?
- Trojans are commonly sent in email, often in a .zip file

- (phone) - best to install apps from official apple/google stores only
- Vulnerability case - keep internet-facing software on auto-update to stay at the latest

Let's stay safe out there!

Nick's Favorite Question

List all the ways you can think of that a bad guy could obtain your password.

Security 5 - Bad Guys

What do bad guys do?

Bad guys most often want money. How can they convert either (a) your password or (b) access to your computer into money?

To read a series of often entertaining stories about various bad guy activities, a fantastic source is Krebs On Security blog

Bad Guy 101 - Generic

- Send Spam using your account to your address book (exploit higher trust) -spam email could be just ads, or malware
- Use login on site to post spammy forum comments with links to bad stuff
- Sell fake goods on your ebay account (exploit account rating, clever!)
- (phone) Call expensive 900 numbers, money goes back to bad guy
- Turn your computer/phone into a "zombie"
- Try the same password on other accounts you might have
- Dig through your computer or steal your name, SSN etc. for fraud posing as you
- Dig through your computer for financial accounts, CC numbers
- Dig through your computer for bitcoins .. super easy to steal

Bad Guy 101 - At Stanford

Here are some specific things that have happened at Stanford with stolen passwords. Dictionary attacks and phishing are two of the top ways that Stanford accounts get broken into.

- Stanford accounts are desirable in the bad-guy world, so we get a lot of bad-guy attention
- Host #6 google ranked beastiality porn on Stanford wiki using stolen login (this actually happened)
- Use Stanford "fat pipes" internet connection: DDOS
- Steal research information (China)
- Put links to bad spam/malware sites on Stanford pages .. use Stanford's high page-rank credibility to make the bad sites show up better in search results
- Steal paychecks by changing direct-deposit account info in axess (this actually happened)

Security 6 - Cryptography

Cryptography - History

- "Encryption" deals with scrambling information so it is readable only someone with the secret key
- Plaintext -- the original bytes (text, image, ...)
- Ciphertext -- the encrypted, unreadable form
- Key -- the short secret needed to encrypt/decrypt (a password basically)
- Aside: cryptography shows up throughout history, especially World War 2. See "The Code Book" by Singh for some great summer reading on the history.
- Notably, the German Enigma machine was cracked by Polish and British cryptographers, probably shortening the war by a couple years and saving many millions of lives.
- The Enigma effort helped spark the beginning of Computer Science

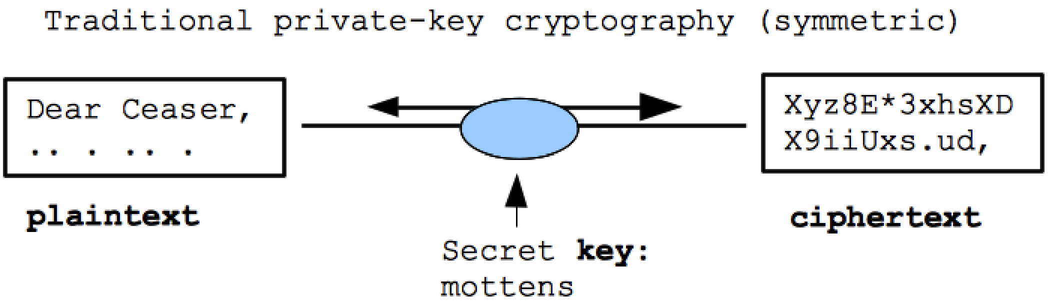
Open Machinery + Secret Key

- The best practice is that the cryptography scheme in use is open, not secret
- The security derives from keeping the per-message **key** secret
- The machinery is wide open ... you need to really trust your per-key encryption!
- World War 2: The German Enigma machine internal wiring was secret, but the Poles and British figured that out eventually (amazingly building workalike machines just from intercepting encrypted messages but never seeing a machine). The Enigma still had a key for each message which provided excellent, although ultimately breakable encryption.

Classical: Private Key Cryptography ("symmetric")

- Traditional -- encrypt and decrypt with a single key which is kept secret
- aka "symmetric" encryption, since the key is used both for encryption and decryption
- Current tech: the "AES" standard, mature, reliable, widely used
- It is thought that no government has "broken" AES - it works
- Two results one obvious, one subtle:
- 1. Provides: **secrecy** attacker intercepts the ciphertext, but cannot recover the plaintext from it

- 2. Provides: **authenticity** attacker cannot "spoof" data, sent to be decrypted. If the ciphertext decrypts cleanly, it must have come from a party with the secret key
- The only "attack" known is "brute force" .. try to guess through the space of all possible keys (if the original key is long, it's safe)
- Note: any encryption can be attacked brute force, so if that's the best attack known, the encryption is regarded as un-broken
- Problem: key distribution. How do you share the key securely between parties?



Demo - Code Wheel Machine

- Plaintext wheel
- Ciphertext wheel
- Key = Starting position of both wheels
- Select a key, see encryption and decryption

Private Key Application: File / Disk Encryption

- One day, your laptop is going to get stolen
- Suppose you have a spreadsheet on their with people's SSNs?
- Suppose your browser is already logged in to important sites
- Most likely, the thief just wants to sell your laptop, but naturally you're a little worried about your information
- Approach 1: Encrypt a sensitive file, deleting the original
- Approach 2: set up whole-disk encryption on the laptop (or whole-disk encryption)
- Disk encryption:
  - a big random key is created automatically
  - the whole file system is encrypted with it
  - your typed password encrypts/decrypts that key
- Machine wakes: your password unlocks the key to decrypt all the files

- No password, no files!
- The files in your home directory are all stored in encrypted form .. docs, browser prefs, everything
- Good: thief gets your laptop but no files. They don't care, they just sell it on craigslist.
- Problem: annoying to type in your password
- Problem: if you forget the password, your data is truly inaccessible - AES is not broken!
- Still, this is a pretty good solution (Stanford staff are supposed to enable disk-encryption)
- The "ransomware" malware uses symmetric encryption - the malware encrypts your files, you send the bad guys money, they send you the key to decrypt your files

In The News - Phone Encryption

- San Bernardino Terrorists
- FBI had their phone
- Phone used whole-disk encryption
- It was in a state where 10 wrong PIN entries would delete all the files
- The FBI took apart the phone and with great effort recovered the PIN anyway
- There was a good chance that they would not have been successful (50%?)
- Point 1: encryption pretty much works
- Point 2: With huge budget/effort, a govt may find a way around

Modern: Public Key Cryptography ("asymmetric")

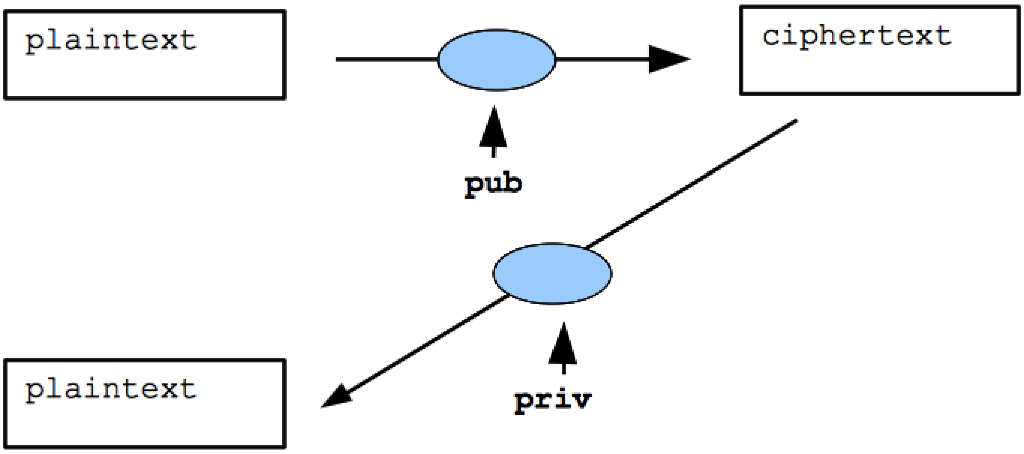
- Public-key cryptography ("asymmetric")
- Amazing technology, surprising that it is even possible
- Used on the internet all the time
- Some of the early work was out of Stanford: Diffie/Hellman

1. pub/priv key pair

- Instead of a simple key, the key is created with two parts: public and private
- Your computer can just make up a pub/priv key pair out of thin air based on random numbers
- Key features:

- 1. The pub/priv keys work as encryption opposites of each other: encrypt with pub, decrypt with priv, **Or** encrypt with priv and decrypt with pub.
- 2. If someone knows pub, they cannot easily compute priv from it. This is an impressive feature, given how closely pub/priv must be related to work as opposites.

Public-key cryptography: Encrypt with pub, decrypt with priv (or encrypt with priv, decrypt with pub)



Public key Application -- Packet Secrecy

- 1. Suppose www.example.com wants to accept packets from anyone, but so they are secret from eavesdropping
- 2. www.example.com makes up a pub/priv pair
- 3. www.example.com publishes the pub part of the pair so everyone can see it. Priv is kept inside www.example.com.
- 4. Someone who wants to send a packet to www.example.com first encrypts it with pub
- 5. At www.example.com the encrypted packet is decrypted with priv
- Provides: no key exchange problem!
- HTTPS is built on something like structure - that's how your browser does the encryption with HTTPS sites

Public key Application -- Digital Signatures

- Another amazing feature enabled by public key cryptography
- A party can make up a pub/priv pair as usual
- Use their private priv key to "sign" a file, distribute that sig

- The sig is not forgeable without the priv key
- Another party can look at the sig and verify that it is valid

Computer Security 7 - Governments vs. The Internet

Free speech and freedom:

Beware of he who would deny you access to information, for
in his heart he dreams himself your master.
  -- Sid Meier

"Read him his rights" .. we all know what that means

The Starsky And Hutch theory of human rights:

We seem much more comfortable with propagating...values to future
generations nonverbally, through a process of being steeped in
media. Apparently this actually works to some degree, for police in
many lands are now complaining that local arrestees are insisting on
having their Miranda rights read to them, just like perps in American
TV cop shows. When it's explained to them that they are in a different

country, where those rights do not exist, they become outraged. Starsky and Hutch reruns, dubbed into diverse languages, may
turn out, in the long run, to be a greater force for human rights than
the Declaration of Independence.

-- Neal Stephenson

- I'll talk about 2 major trends of Governments vs. The Internet
- 1. Government Surveillance - In the news!
- 2. Governments vs. Free Speech
-

Category 1: Government Surveillance

- Snowden disclosures
- NSA spying on everybody
- How much?
- 2 scenarios are clear, 1 unknown

1. Motivated Spying

- Say a government/agency is highly motivated to spy on a particular person
- e.g. Government trying to prosecute a Mafia type conspiracy
- In this case, the attacker can likely get all sorts of info (mostly legally):
- 1. Warrant -- they could get warrant for email records, phone records, wiretaps, ...
- 2. They could physically break in and install a keylogger on a laptop or gadget on a car ...
- 3. They could exploit an unknown vulnerability to install malware on a particular person's machine (China and the NSA are accused of this)
- "Spear Phishing" .. like phishing, but crafted for just that person
- e.g. send a specially crafted email trojan to a particular person
- Summary: for a motivated, well financed attacker, the attack is probably successful

- With a warrant, it's legal too

2. Motivated Encryption

- Now from the other side
- Encryption is reliable, e.g. AES encryption
- If someone encrypts a hard drive or file with a good password, it's unbreakable
- So long as the password is nice and long
- e.g. Child Porn encrypted hard drive
- For months the government could not crack/guess the password, child porn
- If the password was, say, twice as long, it would never have been cracked
- Doubling password length squares the space of possible passwords
- Interesting edge case in the law: can a suspect be required to divulge a password?
- Can a suspect be required to give a PIN or fingerprint?
- Those are not yet settled 5th-amendment issues in the US:
  - Can a suspect be forced to reveal a password/PIN under probable cause

3. Government Surveillance

- The US government does not have the resources to monitor everything going on on the internet
- But they could be monitoring some things in coarse detail (Snowden revelations)
- e.g. who calls who (which numbers call which numbers, not the audio)
- e.g. who visits certain web sites
- Is that legal without a specific warrant?
- How much are they doing?
- Nobody is sure, but it's some -- the Snowden disclosure
- It looks like theUS Government was collecting information without a warrant
- The level of surveillance was too high - illegal

Category 2: Governments vs. Free Speech

- Another category of Governments vs. The Internet
- Autocracies are against sharing ideas: free speech, opinions, blogs, newspapers, twitter
- Reasons for hope:
- Fall of communism: VHS video tapes showing western life overpowered the 24/7 regime propaganda (Francis Fukuyama, Stanford)

- Fukuyama in a nutshell: people can tell when they're being lied to all the time, they deeply dislike it, and this dislike cannot be erased, no matter how voluminous the propaganda
- Neal Stephenson story about Miranda rights (above)
- China, dictatorships etc. strongly against free speech
- Being critical of your own institutions is an important value
- The US Government has many flaws
- The US Government is extremely good on freedom of speech

How Governments Censor Free Speech

- e.g China, North Korea: traditional autocratic,
  what-the-people-are-allowed-to-say / read
- e.g. Pakistan, Saudi Arabia .. mixing in limitations on un-islamic thought, combined with autocracy
- Control TCP/IP routers the connect country to whole internet
- Block certain IP addresses, domain names
- China has an army of people who monitor blogs etc., delete ideas that are not officially permitted
- China also has an army that floods the forums etc. with government-view posts
- I wonder how effective this is, probably not zero
- What "Tiananmen" search looks like with censorship inside China
- Washington Post Censorship Works
- e.g. my free CS video materials are blocked in China
- This all seems like a tragic error that's going to hold China back
- I strongly believe in the value of free speech, China's censorship looks insane
- The censorship has the appearance of a kleptocracy
  - just trying to keep out-of-power people in the dark
- Hope: email, twitter, video ... increasing ease of information sharing making it harder to suppress the truth, per Francis Fukuyama

Security 8 - New EMV Cards

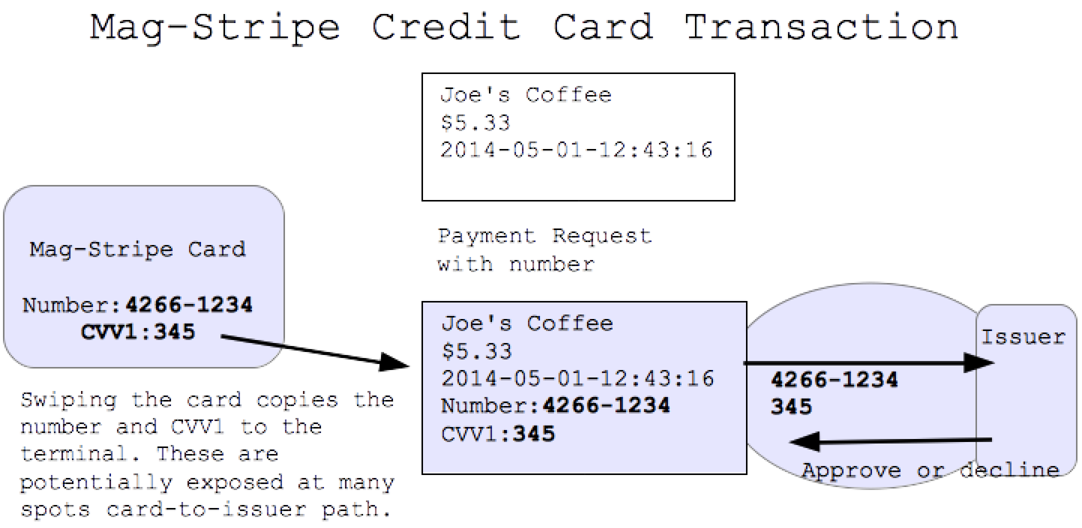
EMV cards are about to be rolled out massively in the US, so you heard it here first!

EMV Card - aka Chip Card

- EMV (Europay Mastercard Visa) have been in use in Europe for over 20 years (EMV on wikipedia)
- Known as "Chip and Pin" or "Chip and Signature" cards
- There are various attacks to nibble at EMV
- However EMV is **much** more secure than old mag-stripe
- The details are complicated, but the basics are simple

How Do Mag-Stripe Credit Cards Work?

- The old way - really quite insecure
- One "secret" is the Credit Card number, printed on the front of the card
- There's also a "CVV1" number on the mag stripe, not printed on the back
- Anyone who has these "secrets" can try making charges
- There is a different CVV2 number printed on the back
- Card-swipe: "secret" is the CC number + CVV1



Swiping the card copies the number and CVV1 to the terminal. These are potentially exposed at many spots card-to-issuer path.

Mag-Stripe Credit Cards Weaknesses

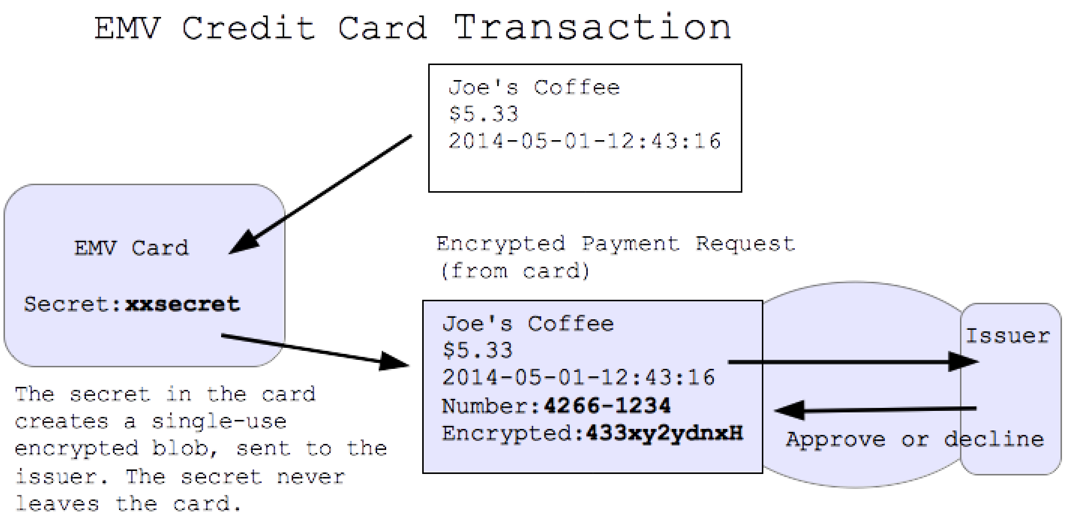
- Weakness: the "secret" can be stolen along the whole path
- Call this "skimming" .. many forms
- e.g. a. in the magstripe reader ("skimmer")
- e.g. b. in the POS (Point of Sale) device
- - Target POS breach, 40 million cards
- e.g. c. the waiter covertly swipes it to get the secret
- e.g. d. along the path to the issuing bank
- A "skimmer" bad guy device attached to ATM, steals "secret"
- e.g. Krebs Skimmer Example
- Conclusion: sending the "secret" unencrypted is a terrible system
- There are many places the bad guys can get to it

EMV Chip on Card

- There's a chip on the card
- The chip has a secret "key" on it, coded at the bank
- EMV uses keys and encryption (below)
- Important strategy improvement: the secret key never leaves the chip
- If the retailer's equipment is compromised .. we're still fine!

How EMV Works

- Insert the card chip-side up into the terminal
- Leave the card inserted, so it can communicate
- Merchant creates a payment request, sends it into the card
- The chip in the card creates with its secret an encrypted "blob" request
- -e.g. imagine the card encrypts the request with the secret key
- Merchant sends the encrypted blob to the issuer to request payment
- The issuer has a copy of the secret to decrypt the request
- **Key improvement:** Encrypted blob is useless to the bad guys if intercepted
- The secret never leaves the chip
- The secret is what one needs to make encrypted blobs
- Avoid needing 100% security along the card-to-issuer path
- I estimate EMV is about 100x more secure than mag-stripe for this "swipe" case
- The request includes the CC number but not the needed CVV1


EMV Credit Card Transaction
Joe's Coffee
$5.33
2014-05-01-12:43:16
EMV Card
Secret:**xxsecret**
The secret in the card creates a single-use encrypted blob, sent to the issuer. The secret never leaves the card.
Encrypted Payment Request (from card)
Joe's Coffee
$5.33
2014-05-01-12:43:16
Number:**4266-1234**
Encrypted:**433xy2ydnxH**
Issuer
Approve or decline

EMV - Contactless

- The little wireless-pay terminals
- There are based on EMV too
- Apple Pay, Google Pay, etc.
- Retain the key feature: data interception is harmless

EMV Card Shift

- Mag stripe: anyone holding the card can get the info off it
- -Bad guy makes a duplicate card, goes to Best Buy etc.
- EMV: only the chip in the card can make a valid request
- -Having the card briefly does not allow one to make a duplicate
- Therefore: the bad guys must steal the physical card

EMV Weakness 1 - Card Stolen

- Now the physical card itself is vital
- Bad guy could steal it from your wallet, go use it
- Some EMV cards require a PIN, guarding against the card-stolen case
- In the US, issuers are choosing to not do PIN
- My guess: skimming was a big problem, card-stolen is relatively rare
- US issuers may add PIN back, once people are used to EMV

EMV Weakness 2 - On The Internet

- Typing your CC number on a web site is CNP (Card Not Present)
- CNP is not getting the benefit of the EMV chip
- When Europe introduced EMV, CNP fraud went up!
- Like there's a pool of bad guys, and the need something to do
- There's possible solutions, but no great solution yet
- Maybe cell phones will supplant cards eventually
- Basic agency problem: the merchants eat the loss for CNP fraud
- -But the issuers control the tech .. lack motivation
- The issuers eat the fraud for "skimming" fraud

EMV Summary

- This is going to be a big improvement
- Makes the skimming (Target) case disappear
- There are weaknesses, but not fatal ones
- We might see PIN get rolled out in the USA eventually
- Next we need to solve the CNP "internet" case