Paxson
Spring 2011

CS 161
Computer Security

Discussion 11

April 20, 2011

Question 1 ***Ransomware*** **(7 min)**

(a) In lecture we discussed 'Ransomware,' in which the malware holds a computer system or data on it hostage by demanding a ransom for its restoration. List some of the ways malware can do this. Be original and try to think of methods not discussed in lecture.

> **Solution:** The question is open-ended and has lots of possible solutions. Examples include locking out the user from his computer by changing his password, slowing down the computer by changing clockspeed, or distorting the display.

(b) Describe how either symmetric or asymmetric encryption could be used for this purpose.

> **Solution:** Using symmetric encryption, the virus would have to first encrypt the files, then send the key elsewhere (and delete it locally). The remote party would return the key if payment is made. This works so long as the victim's system doesn't record the key before it is deleted.
>
> If asymmetric encryption is used, it would also be possible to generate a key pair ahead of time and embed the public key in the virus while retaining the private key remotely. In this case the user would not be able to decrypt the files even if they record all of the keys present on their system (at any point).

(c) Last year, an intriguing virus called Kenzero was out in the wild. Kenzero searches a victim's web browsing history / cache for pornographic websites, takes a screenshot of what it finds, and posts it on a public website with the victim's name. The virus then directs the user to a site where they must make a credit card payment of $16-400 in order to remove the post before Google crawls it.

What other types of threats (apart from deleting or encrypting data) can you imagine a virus making to extract payment?

> **Solution:**
>
> Perhaps the most interesting possibilities are those that leverage social connections in some way. For example, the virus could find the victim's email contacts

or facebook friends, then threaten to send them something embarrassing. Things that could be sent include porn the victim has viewed, or emails/IMs to and from other users (especially those that include the name of the contact to which the virus will send the messages). Apart from forwarding existing information, the virus might attempt to make it appear that the victim has taken actions that in fact they haven't (e.g., forge messages or change Facebook relationship statuses). Other possible threats include publicly revealing passwords, secret keys, or credit card information, or consuming resources (e.g., telling other virus instances to constantly send SMS messages to the user's phone).

In order to ensure the user cannot prevent these threats by simply shutting down their machine, the virus could first set up the process so it will be automatically executed by a third-party instance of the virus unless the user makes a payment within some time limit. Once this is in place, the virus would demand payment.

(d) Instead of making threats, can you imagine any ways viruses or worms might offer incentives or rewards in exchange for payments? Can you think of anything a virus or worm might ask the user to do other than make a payment?

Solution: Again, you can think of interesting incentives based on a user's social connections. The virus / worm could offer to allow the user to read emails or messages from the account of a friend that has been compromised (perhaps showing promising excerpts as a "teaser"), or allowing the user to access their friend's webcam / microphone. Similarly, the virus may could the user the ability to control other infected machines.

More speculatively, one might imagine a virus or worm that somehow asks a user for assistance in propagation (perhaps in exchange for a portion of the resulting payments, in the style of an Amway-type scheme; or a reduction in the user's ransom). While we might not expect any users to be willing to sell out their own friends by, say, sending them email attachment laden with a virus, one might be able to get them to do so to strangers. It can sometimes be easy to identify accounts on, say, MySpace that are spreading viruses because they are either (a) a phony account that obviously was created in an automated fashion, or (b) making a series of identical or exceedingly bland wall posts. If the virus is instead spread by a large number of independently motivated, real people, it could be harder to identify.

Question 2 *Worm Spread* (7 min)

(a) In class, Prof. Paxson mentioned that typical network worms propagate using scanning. Can you think of other ways to spread a worm?

> **Solution:** Again, this whole question is open-ended and has multiple solutions. One solution is to post links to friends via email/twitter/facebook. Another is to read logs or other files on the local machine to find other likely victims. A third is to use a search engine ("outsourcing" the scanning), as discussed in lecture on Thursday. A fourth is to contact a server whose job it is to track other servers (for example, some online games have "meta-servers" that you can contact to find a list of servers available for playing the game).

(b) The typical virus exploits a benign application to execute its own (malicious) code. Exploiting real world applications is getting tougher every year because of the mitigations for buffer overflows that we discussed. Can you think of a way that a virus would not require an exploit to achieve code execution?

> **Solution:** You can just fool a user to download 'critical updates' through social engineering.
>
> The Koobface worm spread using this technique. Once a user's computer is infected with Koobface, Koobface would post a link on the user's friends' walls. When clicked, the linked page would ask the user to download and install an 'update' for their Adobe Flash Player. If the 'update' is installed, the computer is now infected with Koobface.

Question 3 *Trusting Trust* (5 min)

The title of the Ken Thompson talk that Prof. Paxson mentioned in class was "Trusting Trust." Think of your daily computer usage, and list down the corporations that you *have* to trust to keep your private data private. We can start the list by the obvious: Apple, Microsoft, Google, Facebook. Place yourself in the DoD's shoes when you go through the exercise and consider whether it's appropriate to trust the companies. Arguably, you are also trusting the computer manufacturers that Apple developers wrote the iPhone software on. You are also trusting the compilers, the OS and the whole toolchain that takes source code from Apple developers and converts it into ARM assembly that runs on the iPhone. You also have to trust the chip designer Qualcomm. Again, Qualcomm only designs chips and so you have to trust the actual manufacturer.

Remember that trust is transitive: if you trust EvilCorp, you also have to trust corporations that EvilCorp relies on.

Solution: This is again an open-ended question. One of the examples we talked about in class was your phone conversation on your iPhone. You have to trust the carriers and the maker of the phone. In addition, you have to trust the corporation that assembled the phone (Apple only designs the iPhone). You have to trust Apple too because it is Apple's software on the phone. In addition, you have to trust the manufacturer of the speaker, the microphone, the camera, the GPS, and so on too.

Interestingly, one of the conclusions that Ken Thompson mentions in his lecture is that you shouldn't trust him or companies that hire him. It's illuminating to go through the list of things you have to abandon if you don't trust Thompson: it gives a clear insight on the massive influence he has had on our daily lives. His Wikipedia page (http://en.wikipedia.org/wiki/Ken_Thompson) is a good place to start.