The Economics of Cybersecurity — Lecture 4 Notes

Adam Hastings

January 23, 2024

Homework Review

Discussion: General Economics of Patching

What are the economic forces that affect the ecosystem of patching?

- Who benefits from patching?
- Who loses out from patching?
- What are the costs of patching? Who pays them?
- What do we think about the "Ship it tomorrow and fix in the next release" line from Anderson's paper?
- What does a typical patch look like? How many lines of code is it?
- What are some deployment considerations when it comes to patching?
  - How easy is it to patch software that you wrote yoursefl on your own system? Even this may not be simple
- Why might someone *not* want to patch a system?
  - May be because they're an IT manager who is busy with other tasks
  - May be because patching a system will cause unacceptable downtime
  - May be because system is remote or unaccessible (e.g. pacemaker).
    - This brings up another question: Should companies be forced to make their systems patchable?
    - What happens if someone discovers an issue with pacemaker code? Could be a security issue or simply a functional one.
    - Healthcare devices are pretty strictly regulated by the FDA. Should other domains be regulated the same amount? Should in-home security cameras be required to have a patching system in place? What about a children's toys?

    - Side note: These questions are being asked and discussed by governments around the world today, and with some regulations (in Europe mostly) already affecting product vendors.
    - There are lots of very difficult questions to answer here that intersect with computer science, economics, policy, and law. We will discuss these topics more later in the semester.

- **Important Question:** Does giving products the ability to be updated increase or decrease the product's security level?
  - Increase: If vulnerabilities are found, they can be fixed
  - Decrease: It means that someone somewhere has the ability to change the code that's running on a device. Is the update mechanism secure?
    - I did a hardware security internship at Bloomberg, who makes their own biometric authentication devices (fingerprint swipe). One of the projects I worked on was writing firmware that decided whether or not to allow a patch. A lot of steps need to happen to make sure this is done securely!
  - It's not immediately clear if patching is always better. I suspect it is (and suspect that the majority of security professionals would agree with me) but this is based on intuition rather than evidence.

- What might be done to make sure that patches can be done securely?
  - Depends on the domain. But generally is going to involve some level of certificates.
  - Security I was listed as a prerequisite for this class. Can someone tell me what a certificate is?
  - Let's back up even more: Can someone tell me what a digital signature is?
    - It's a set of algorithms:
    - Key generation: Create a public-private keypair. Based on special properties of fields usually.
    - Signing: Using the private key, create a signature. In ECDSA (common signature scheme), this is 64-bytes.
    - Verification: Using a hash of the signed data, the public key (available to everyone), and the signature, verify that the signature was created using the public key.
  - Conclusion: If I'm a product vendor, I could sign some piece of data (like a patch) using my private key, and then using my public key, you (or your device) could verify the signature before deciding to accept the patch.
  - What's the problem here?
    - The problem is that someone could impersonate me. A malicious attacker could make their own keypair, give you a patch with a signature, and ask you to verify the signature. And the signature verification would succeed! So we need some trusted way of verifying that the public key belongs to a specific person or company. This is where certificate come into play.
  - What's a certificate?
    - A certificate is a signature by a trusted party that a certain public key belongs to a certain individual or company.

        * Who here has used cerificates before? Trick question—all of you! Whenever you access an HTTPS website, it means that communication beween you and the website is encrypted using public key cryptography. But to ensure that you are actually communicating with the website you think you're communicating with, your browser will check the website's certificate, which says "this website's public key is XYZ" and will be signed by a certificate authority, which are companies that are trusted just to
    - What's the problem here? It's requires you to trust the cerificate authority to do the vetting process for you. This process has been compromised before!
    - If you are the device manufacturer though, you may be able to act as your own Certificate Authority though. But this brings up other questions...
    - What happens if my private key gets stolen?
        * I'd have to create a new keypair
        * What if your device was programmed on only accept one single public key though?
- Discussion conclusion: Patching is not simple. There can be significant costs involved depending on the level of security required.

A Large-Scale Empirical Study of Security Patches

How Much is Performance Worth to Users?

- Why do you think I assigned this paper to discuss on the day we talk about patching? Is this paper even about patching?
- You may find it interesting that the initial reason why we did this work was because we were interested in the effects that patching has on users. What does this paper have to do with patching?
    - My research area is in hardware security, and in hardware

(Give 15-minute conference presetation)

Post-talk discussion:

- What did we think of this work?
- (Answer any other questions students may have)
- One thing that you may find interesting is that the motivation for this

I want to subsantively discuss the *content* of this paper, but I do also want to briefly discuss the

- What do you think I did well in this presentation?