Identifying Threats to Privacy

Cryptoparty in Boston*

November 30, 2016

1 Basic thoughts about threat modeling

1.1 Metaphor

How many possible ways could a stranger get into your house/apartment? Just some examples off the top of my head:

Steal your key when you're not looking
Pick your lock
Use a crowbar
Copy your key at a hardware store; then replace it
Make a wax/clay impression of your key, as in *The Day of the Jackal*
Trick someone with a key into unlocking the door
Get in through an unlocked window
Take the key by force/threat (including legal)
Try the knob and find it unlocked

Can you think of other ways someone could get in without permission? From the would-be burglar's perspective, what are some advantages/disadvantages of any of these methods? Which ones do you think someone is most likely to ever try against your apartment in real life? Which ones are most likely to *succeed* in real life? How would that change if it was widely known that you owned $100,000 worth of diamonds? Or if you were a bank or electronics store?

1.2 My assessment of threats to my privacy

Now take the thought process we just applied to the physical security of your apartment, and try to apply it to privacy. For our purposes, *privacy* means *keeping yourself or information about you secluded, or free from observation or intrusion.* So, what if the intruders were after information, rather than access to your place (think credit card numbers, photos of you, Watergate style snooping, etc.)? Below are some ways that I personally think my privacy could

be compromised. (Again, can you think of other ways? I don't intend this list to be a complete "answer sheet.")

Related to my non-digital actions: being followed, videotaped, photographed.

Tracking by servers including my home internet company, any relaying server, or destination server. If they keep a big pot of data on what I connect to, what times of day, contents of communications, etc., then this could later be accessed inappropriately from within, breached, or used as circumstantial evidence. Mini thought experiment: if you know only that someone is connecting to `doIllegalThings.com`, how much can you guess about what they're reading? What if it were `wikipedia.org`?

Breach of single password because of: loss or confiscation of a written password, someone tricking me into disclosing a password, or data breach directly at a server. This can compromise the privacy of that account and any accounts that used the same password. Note: I also worry about losing a written password, forgetting a memorized password, and having to pull out a written password "too often"—strictly these are threats to convenience, not privacy.

Guessing a password to a valuable/sensitive account. This can be brute-force guessing if it's a weak password (e.g. a password of six lowercase letters, even if random, like `nkpukf, orfpvv`). In the case of some technologies, this can be facilitated by stealing a private key. This could also be intelligent guessing, if it's an extremely common password, or if it relies on significant people's names, dates, etc.[1] Examples of extremely common passwords: `iloveyou1 dragons1 michael1 princess1 1qaz2wsx trustno1 sunshine1`

Random numbers: Someone somehow nabbing the randomness used to generate passwords. Someone inferring pseudorandom numbers (if used).

General threats to digital privacy (not related to passwords) can emerge any time the algorithm or the specific implementation has a flaw.

Why/who? It can also be useful to think about which one(s) of the following scenarios is/are likely. Are the bad guys after *you specifically* (trying to break into your e-mail because of who you are and information that only you have)? Or do they profit from *anyone's* data/resources (trying to steal any financial details, collect data about everyone's movents, or remotely control anyone's computer)? Or are they just malicious?[2] Are they going to try entering passwords in the same web form where you log on, or can they try passwords offline—as I will describe in the story in section 2—or are they going to go for the data directly on the server?

What harm is likely to occur if someone accesses or discloses information about you? Would people stalk/harass you? Would you get fired? End up with

[1] Somewhat nice discussion of common passwords at http://wpengine.com/unmasked/ .

[2] "He said the hack was simply a grab for my three-character Twitter handle. That's all they wanted. They just wanted to take it, and fuck shit up, and watch it burn. It wasn't personal." Mat Honan, How Apple and Amazon Security Flaws Led to My Epic Hacking. *Wired* 2012-08-06.

a drained bank account or ruined credit score? Would you be under suspicion (e.g. for searching "how to steal an identity")?[3]

1.3 My current approach to these threats

Tracking: I use Tor Browser for plenty of things (such as all the research I did for this write-up). I purchased a subscription to a non-Gmail e-mail account. Note that if the bad guys want to know only the names of the people you contact, encrypted e-mail doesn't help. Or if they want to know only *whether* you're using Tor (not the contents of your communications), then you need something additional.

Current password practices: I *never* reuse passwords used to secure highly sensitive data. My passwords are completely random. To encourage myself to use strong passwords, I write them down. I avoid pseudorandom numbers and use `random.org` or literally a box of dice to generate passwords.[4] I used to use short passwords with many types of characters like: `f+xW'a)w`. Now I use longer passwords made of English words, like: `dellairydaycavern`. I can carry some randomly pre-selected words as "entropy in my pocket," so it's easy to change passwords. I'm sure that we will discuss more password practices beyond this short list.

Advantages: Because I use an external source of randomness, I know exactly how much randomness any one of my passwords embodies. In my opinion it's extremely hard to determine this for any passwords I would invent by myself. "Intelligent guessing" of the password because it doesn't have anything to do with names/dates pertinent to my own friends, family, etc. I can type in all lower case. It's relatively easy to change passwords. When I switched from random characters to random words, it became somewhat easier to remember passwords, but not in the way I expected (story on request). If I have the password memorized, I don't need to resort to any external tool at all.

Drawbacks: I have to keep any written passwords very safe. Written passwords also change the secret from "something you know" to "something you have." My current word list contains weird words that are harder than average to remember. It takes a little while to "warm up" to a long password just after it's been changed.

2 A personal experience

2.1 Story about trying to guess a password

Some time ago, I helped someone set up e-mail encryption using GPG software, but they forgot the password, so I decided to try to guess it. I had to obtain

[3] What if there was no identity theft or public embarrassment? If a stranger knew about your messages, receipts, income, movements, or how you probably vote, would that fact alone bother you? I'm not saying it's *supposed* to.

[4] http://world.std.com/~reinhold/diceware.html

the private key, and I also had the cooperation of the person in question, who gave several examples of probable passwords and what theme might be present in the password.

I wrote a script that reads a short list of candidate passwords outputs many variations on the theme. The script could check several passwords per second; a big proportion of this time is probably a delay built in to GPG. By far the longest time spent on this project was writing the script.

This required only modest knowledge. I am not a particularly "fancy" programmer—details on request :). Probably there are millions of people in the U.S. who are at least as knowledgeable. Ultimately I didn't guess the password, but it was easy to check about 1400 passwords, which I think is a lot for certain types of password.

2.2 Lessons learned

Stay usable. Don't lose your password. For me it is always a very hard decision whether to write down a password. I don't think there is a right or wrong answer. This *directly ties in* to our discussion about threat modeling. What are the benefits/risks to keeping the password in your head or on a piece of paper?

Stay tidy. For GPG, generate a revocation certificate and keep it. Reminder: you can't generate one if you forgot your password! Expiration dates of your keypair can also help, and there is probably no "one size fits all" expiration date either.

Stay secure. Don't let your private key loose! Lock the hardware where it lives. That is, if someone is looking at your desktop icons, don't imagine that it takes a long time to find and copy your key. It took me probably 1 minute. I doubt there were any logs to give me away. A password on the device itself would've stopped me if I hadn't had the owner's permission.

Use a strong password. If the private key is compromised, password guessing is "embarrassingly parallel." Given more time, I could have scaled up to many compute instances and/or used a more optimized implementation without delays. It's probably within the means of one individual to exhaustively search some password classes in a matter of days.

3 Conclusion

To decide on my approach to privacy, I first think *how* someone may try to compromise my privacy, *why* they may do so, *who* they may be, and *what harm* may occur if they succeed. Keeping a written vs. a memorized password is a prime example of a choice that can trade off the likelihood of different threats.

There is obvious overlap between this threat modeling discussion and many other topics in cryptography. I've tried to limit this discussion to privacy and not bleed over into security, and I've tried not to focus on passwords exclusively. Probably I didn't completely achieve either of those goals—I encourage comments!