CS107, Lecture 21
Reverse Engineering, Privacy and Trust

Reading: None

Ed Discussion: https://edstem.org/us/courses/28214/discussion/2135675

Privacy and Trust

How does a computer interpret and execute C programs?

Why is answering this question important?

- Learning how our code is really translated and executed helps us write better code
- **We can learn how to reverse engineer and exploit programs at the assembly level**

assign5: find and exploit vulnerabilities in an ATM program, reverse engineer a program without seeing its code, and de-anonymize users given a data leak.

Privacy and Trust

- Learning about machine code and program execution helps us better understand computer security.
- Computer security (the protection of data, devices, and networks from disruption, harm, theft, unauthorized access or modification) is important in part because it enables privacy.
- In understanding computer security, it's essential to understand the context in which it comes up (privacy and trust).

Have you been affected by a data breach/hack or other improper access of your data?

How did that make you feel?

Privacy

What is privacy? Here are 4 ways of framing privacy:

- Privacy as **control of information**: controlling how our private information is shared with others.
- Privacy as **autonomy**: agency to decide for ourselves what is valuable.
- Privacy as **social good**: social life would be severely compromised without privacy.
- Privacy and protection as based in **trust**: privacy enables trusting relationships.

First two are *individualist* –the value of privacy as an individual right.

Second two are *social* – the value of privacy for a group.

Privacy

Privacy as **control of information**: controlling how our private information is shared with others.

- Consent requires *free* choice with available alternatives and *informed* understanding of what is being offered.
- How many of you just skip past the terms of service for new online services?
- Control over personal data being collected (e.g., data exports from services you use, privacy dashboards, device privacy protections)

Privacy

Privacy as **autonomy**: agency to decide for ourselves what is valuable.

- Links to autonomy over our own lives and our ability to lead them as we choose.
- Do you feel your autonomy is consistently respected when using products and services? Why or why not?

"[P]rivacy is valuable because it acknowledges our respect for persons as autonomous beings with the capacity to love, care and like—in other words, persons with the potential to freely develop close relationships" (Innes 1992)

Individualist Models of Privacy

Privacy as **autonomy** and privacy as **control over information** focus on the value of privacy at an individual level.

- Individual privacy can conflict with interests of society or the state.
- Many debates over "privacy vs. security" – whether one should be sacrificed for the other
  - Apple v. FBI case re: unlocking iPhones (link)
  - Debates around encryption (link)

Privacy

Privacy as **social good**: social life would be severely compromised without privacy.

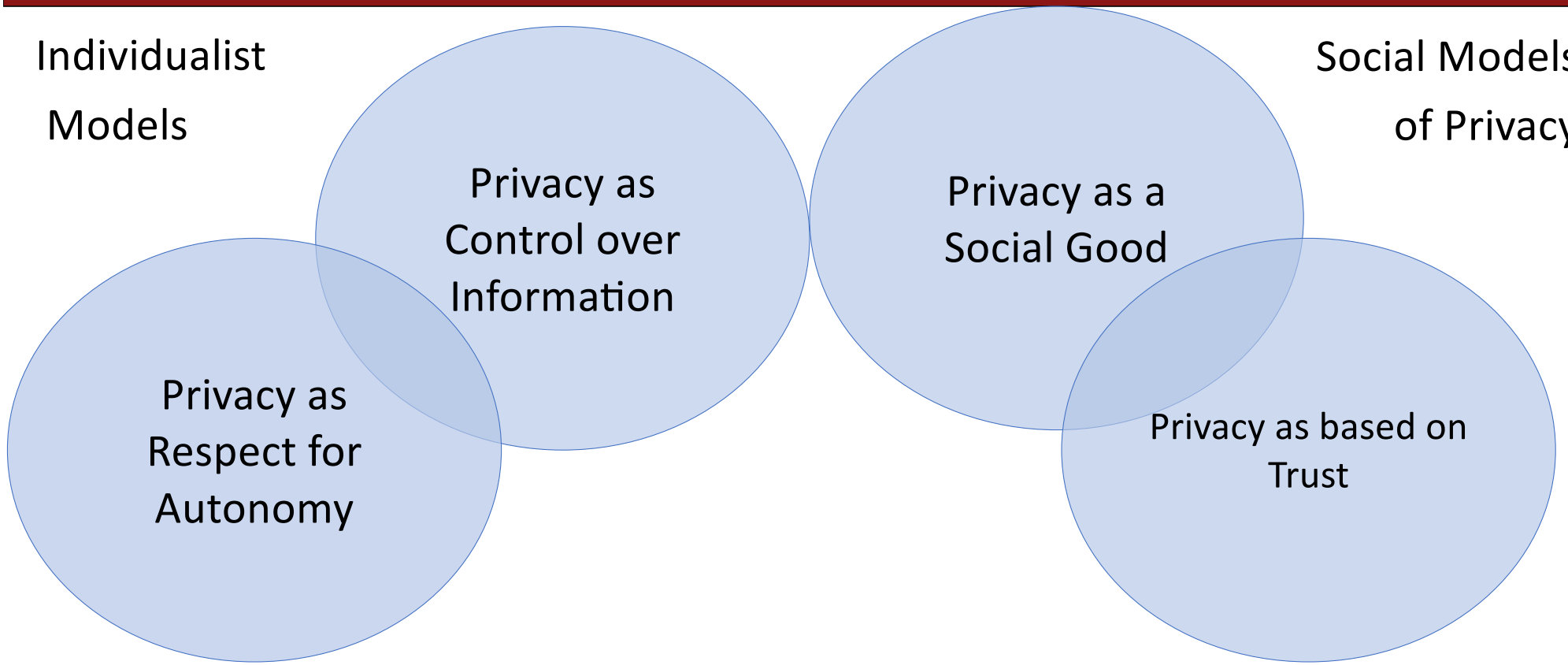
- Privacy has a social value in bringing about the kind of society we live in.
- What would society look like without privacy?

Privacy

Privacy and protection as based in **trust**: privacy enables trusting relationships.

- Privacy may help enable trusting relationships essential for cooperation. For instance, a *fiduciary*: someone who stands in a legal or ethical relationship of trust with another person (or group). The fiduciary must act for the benefit of and in the best interest of the other person.
  - e.g., tax filer with access to your bank account
  - Should anyone who has access to personal info have a *fiduciary* responsibility? (Richards & Hartzog 2020).
- This model of privacy stresses the essential relationship between trust placed in any holder of personal data and responsibilities that come with this trust.

Models of Privacy



Loss of Privacy

Loss of privacy can cause us various harms, including:

- ***Aggregation*:** combining personal information from various sources to build a profile of someone
- ***Exclusion:*** not knowing how our information is being used, or being unable to access or modify it (Google removing personal info from search – link)
- ***Secondary Use***: using your information for purposes other than what was intended without permission.

Who Should We Trust?

Both security and privacy rely on trusted people (who administer security, perform penetration tests, submit vulnerabilities to databases, or keep private information secret). The final piece of the security puzzle is understanding trust.

Trust = Reliance + Risk of Betrayal

What makes trust unique to relationships between people is that trust exposes one to being *betrayed or being let down* (Baier 1986).

Penetration Testing & Trust

Penetration testing is the practice of encouraging or hiring security researchers to find vulnerabilities in one's own code or system.

The tester is placed in a position of trust: they are given access to the system itself and encouraged to find easily and not-so-easily exploitd vulnerabilities, with the expectation that the tester will share what they have found with you.

Hiring a penetration tester means *relying on* their skill at finding vulnerabilities but also *trusting* their ethical compass will lead them to tell you and to act as a trustworthy *fiduciary* (guardian of your interests). In assign5, you will have the opportunity to test your own ethical compass!

Example: Differential Privacy

Imagine a large database—perhaps a medical one—with personal information and records of past activity tied to a name.

The records might be useful for research purposes, or to train a machine learning model to predict future health outcomes, but what if giving access to the records exposed the privacy of individual person's health records?

Differential privacy is a formal measure of privacy that attempts to address these concerns. By adding inconsequential noise (changing a birthday from 2001 to 2002, for example) or removing records, differential privacy protects individuals from *aggregation* by making them harder to identify (Dwork 2008).

Differential Privacy's Trust Model

Differential privacy assumes that the only threat to privacy is an *external user accessing the database* who must be prevented from aggregating data that could identify a user.

In other words, the *trust model* of differential privacy is that the database owners and maintainers are to be fully trusted, and no one else.

Differential Privacy: The Other Threats

But is that the only threat? Differential privacy does not protect against improper use by people with full access to data or against data leaks from the database itself, which may be the primary data exposure risks.

Differential privacy also does not question the assumption that amassing & storing large amounts of personal data is worth the risk of inevitable leaks (Rogaway 2015).

In every evaluation of privacy, we can ask: who is trusted? Who is distrusted? Does this model concentrate trust (and therefore power) in a single individual or small group, or does it distribute trust?

Optimizations you'll see

`nop`

- **`nop/nopl`** are "no-op" instructions – they do nothing!
- intent: Make functions align on address boundaries that are nice multiples of 8.

`mov %ebx,%ebx`

- zeros out the top 32 register bits (because a **`mov`** int an e-pseudo-register zeros out the upper 32 bits).

`xor %ebx,%ebx`

- Optimizes for performance as well as code size (read more here):

```
b8 00 00 00 00          mov $0x0,%eax
31 c0                   xor %eax,%eax
```

Funky Assembly you'll see

Some functions like **printf** take a *variable numbers of arguments.*

- It turns out that in assembly when we call these functions, we must indicate the presence of any float/double arguments by setting **%rax** to the count of vector registers used. If none are used (i.e., no parameters of float/double type), it sets **%rax** to zero.