Trust and Operating Systems

Mendel Rosenblum

Trust and Operating Systems

Optional readings: Operating Systems: Principles and Practice: None

What is Trust?

‘the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party’

The Ethics of Advanced AI Assistants https://arxiv.org/pdf/2404.16244

- willingness … to be vulnerable
  - Must be willing, not under duress
  - Open yourself up to the risks that trustee performs in undesirable way
- expectation
  - Belief that the trustee will perform satisfactorily
- irrespective of ability to monitor or control the other party
  - Even if trustor cannot control or monitor the trustee

What is Trust (philosophy version)?

- Trust is an unquestioning attitude
- With trust, we stop questioning its dependability and assume it will work.

Why Trust? Trust extends our agency

Agency definition:

> “Agency is the sense of control that you feel in your life, your capacity to influence your own thoughts and behavior, and have faith in your ability to handle a wide range of tasks and situations. Your sense of agency helps you to be psychologically stable, yet flexible in the face of conflict or change."

Agency in regards to computer systems:

> Agency means a person’s practical sense that they can understand, influence, and direct what happens in their interactions with systems.
>
> > Agency is the feeling that you understand what a system is doing and can make it do what you want.

Why Trust? Improves efficiency

Trust lets us accomplish more than we could through direct personal control alone

Trust is **empowering**, can do more with trust than without

Example: Trusting a computer to add a bunch of numbers

Trust is fundamental to all social systems

Enormous number of things you have to trust to get through a single day

Example: Trust Empowering

- A person with diabetes might need to monitor their blood glucose level
  - Test a small sample of blood for sugar level
- Technology: microprocessor + sensors + radio
  - Blood glucose monitor device that connects over Bluetooth to user's phone
  - Notified user when blood sugar changes
- Allows people to rely on the system instead of constantly monitoring

Why Trust? Sanity

- Eliminates constant worrying
- Without trust you would have to do everything for yourself
  - Food and product safety
  - Stanford education
- True even if you lived totally alone

Risks of Trust

- Trust creates dependence and vulnerability
  - Dependence: Trustor might depend on trustee to accomplish task
  - Vulnerability: Failure of the trustee can harm trustor
- Violation of trust can be deeply upsetting, even dangerous

Example: OS dropped Bluetooth connections silently, which caused glucose monitors to stop sending alerts.

Over-Trust - Trusting more than warranted

Formal definitions:

- **Over-trust:** cognitive error where an individual extends trust beyond reasonable limits, leading to potential exploitation
  - This is a property of the trustor
- **Untrustworthiness**: objective failure of a person (or piece of technology) to display integrity, reliability, or care, making them unworthy of trust
  - This is a property of the trustee

Terminology uncertainty: Over-trust, Unwarranted Trust, Agential gullibility

Discussion

Discussion with neighbor:

- Things you both trust and why
  - How does it extend your agency?
- Examples of over-trust?
  - What was the result of your trust being violated?
- Examples of untrustworthiness?
  - What was the result of your trust being violated?

How Trust is Established

- Assumption
- Inference
- Substitution

Trust by Assumption

- Trust without evidence of trustworthiness
- Sometimes we trust simply because we must act quickly
  - Example: someone yelling 'Watch out! A car is coming.'

Trust by Inference

Indicators suggest trust is warranted

- Past behavior and presumed generalizability
  - Assume past experience predicts future behavior
- Characteristics
  - How it was constructed
  - Brand or institution
  - Some indicators are weaker, others stronger

Inference is the strongest form of trust

Example: evidence suggests the system is reliable

Trust by Substitution

- Trust due to backup plan
  - Structural arrangements that will compensate for misplaced trust
- Examples:
  - Trapeze artists trust each other partly because a safety net exists
  - Take Uber to the airport? If no cars available, can drive
- Substitution moves the trust requirement to the backup system
  - Substitution only works if the backup is trustworthy

Discussion

Did your group have examples of things you trusted through:

- Assumption
- Inference
- Substitution

Trust and Software

Software underpins modern life

Trust is essential in software systems

We now depend on software for virtually every aspect of our lives:

- Business
- Transportation
- Utilities: water, electricity, telecommunications, etc.
- Science
- Education
- News, social interactions
- Social media and recommendation algorithms

Establishing Trust in Software

- Assumption: ineffective, not used
- Inference: the path to trust is through distrust
  - Testing and verification
  - Instrumentation
  - Code reviews
- Substitution: detect errors when they occur, correct if possible
  - Logging
  - Consistency checks
  - Timeouts
  - Redundancy

Aside: Software unlike other products

- Software licenses typically absolve developers of responsibility for failures
- Expectation that software has flaws (i.e., bugs)

Challenge: confirmation bias

We trust systems that appear to work correctly

Confirmation bias causes us to stop questioning systems that appear to work

Failure to scrutinize system when it seems to be behaving properly

- This is a form of over-trust

Over-trust may also be amplified by some design aspects of the system

- Institutional trust (e.g., our governments or Stanford)
- Access to technology without training what technology is good for

Operating System: Root of Trust

- All software runs on the OS

  Applications rely on the operating system for security, protection, and correct execution

  Applications are only as trustworthy as the OS they run on

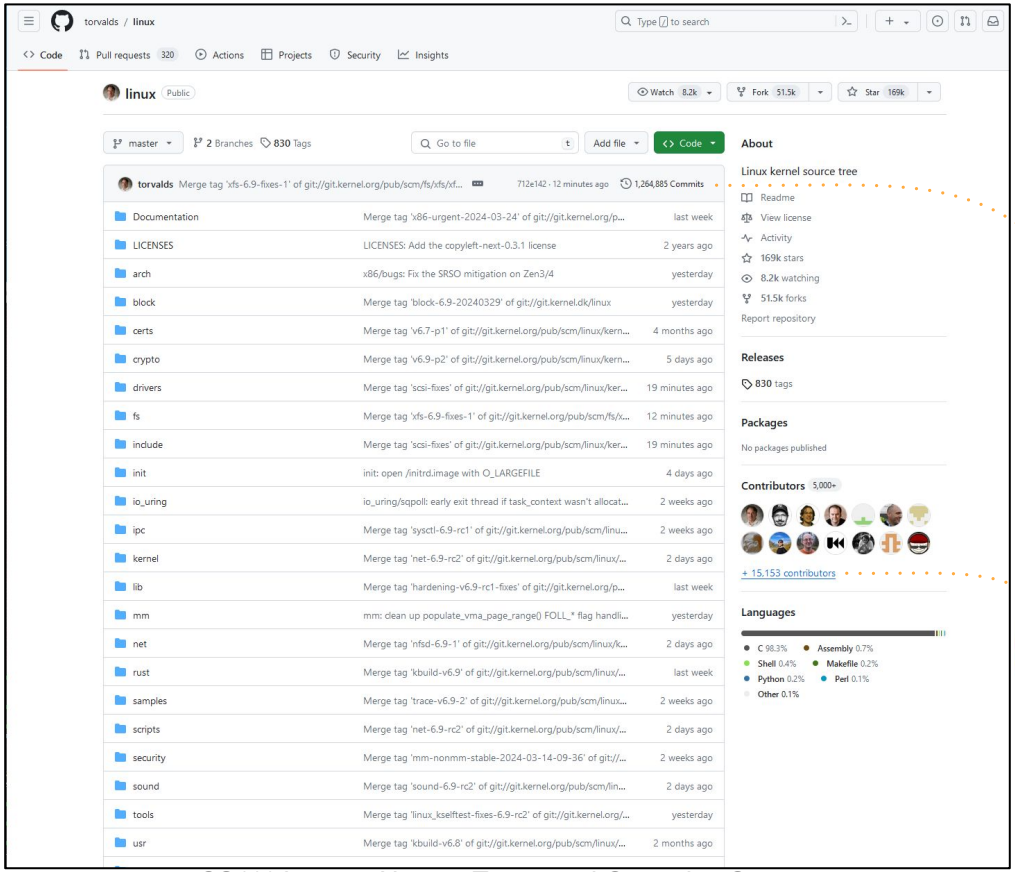
- Operating systems implement mechanisms used by applications to ensure trust
  - Operating systems are part of the **trusted computing base**
    - Computing hardware and firmware (BIOS)
    - Operating system kernel

When you think you are running a particular application on your phone, how do you know you really are?

Linux Kernel Example

> 8M lines of code

Why trust it?



1,264,885 Commits

+ 15,153 contributors

Why Users Trust Linux

- Assumption
  - "Never thought about it"
  - "No other option"
- Inference
  - General trust in open source software
  - Many eyes to detect and fix problems
  - Have used it before without problems
- Substitution
  - 3rd party antivirus software
  - Replicate/encrypt important files

Why Developers Trust Linux

- Assumption: rare
- Inference:
  - Used by other app developers
  - GitHub stars
  - Trust Linus Torvalds
- Substitution:
  - Code is open source
    - Read it
    - Clone the repo to fix bugs

Trust Within Linux Developers

- Assumption: none (risks of bugs)
- Inference: community reputation
  - Known in community: community reputation
  - Previous patches were high quality
- Substitution
  - Changes must be reviewed, accepted in layered process
  - Linus has final authority

Trojan Horse Example: xz / ssh attack

Recent (2024) Trojan Horse discovered in the Linux ssh program

- Would have enabled attackers to gain access to any Linux system
- Discovered before widely deployed, but only by chance
  - Someone curious about a small slow down in sshd

Example of over-trust enabling a supply chain attack

xz / ssh attack

- sshd uses the system log (`systemd/libsystemd`) which uses `xz` for compression
  - ssh developers trusted Linux system log which trusted `xz`
- Someone posed as a legitimate open source developer named Jia Tan for multiple years
- Tan began to express impatience with the xz lead maintainer
- Multiple other developers pressured the lead maintainer to accept help from Tan
- Tan eventually was given permission to merge changes into xz.
- Tan also submitted a pull request for OSS-Fuzz, which scans open source packages for malicious code; the patch disabled a check that would have exposed the Trojan Horse
- Infected systemd was dynamically linked when the `sshd` command was run

Should Linux trust/accept AI generated code?

- Decision released April 2026
- What do you think?

Linux AI policy

- Linux kernel developer beliefs:
  - Going to be hard to detect and stop submissions that use AI
  - Some AI use is genuinely useful
  - Would be a disaster to let AIs submit code

Decision:

AI may help write code, but only humans may contribute code

Humans take full responsibility for every line

Key Lesson

Trust systems can fail

Even strong trust systems can fail due to over-trust.

Recap

What is trust?

- 1) willingness to be vulnerable
- 2) expectation that the trustee will complete the action
- 3) irrespective of ability to monitor or control the other party
- Beneficial because it extends agency

Trust is essential, but it comes with risks

Trust emerges through:

- Assumption
- Inference (most powerful)
- Substitution

Because of software's ubiquity and high impact, it is important for software to be trustworthy

Operating systems sit at the root of software trust (trusted computing base)

CrapTrap from Brex: https://www.brex.com/crabtrap

CrabTrap: an LLM-as-a-judge HTTP proxy to secure agents in production

- Inspect outbound web requests made by AI agents
- Block unsafe or unauthorized actions
- Filter requests for policy violations
- Prevent prompt-injection-driven behavior from websites
- Constrain agents to approved domains / actions
- Log and audit agent activity

How does this software fit into today's lecture?