CSci 4271W
Development of Secure Software Systems
Day 25: AI safety threat modeling, XZ/SSH backdoor

Stephen McCamant
University of Minnesota, Computer Science & Engineering

Outline

AI safety threat modeling

Announcements intermission

The XZ/SSH backdoor

More crypto failures

DNSSEC

Kinds of AI safety concerns

- AI failure and misuse: present-day negative consequences of AI not being smart enough, or being used by adversarial people
- AI alignment: long-term risks of AI behavior being inconsistent with human values

Business and social context

- Recent advances in AI are novel software being driven by big tech companies
- Short-term concern is showcasing the technology as useful and low-risk
  - Worthy of future investment but only light regulation
- The reading is a whitepaper from OpenAI around the time GPT-4 was released
  - Incentives to not leave risks out, but make them seem manageable

Normal security concerns

- Companies deploying LLMs have most of the normal security concerns
  - E.g., running a large public web site
- For commercial providers, keeping the models secret is a critical requirement

Relevance of threat modeling

- For AI-specific concerns, the main intersection with security is thinking about adversarial threats
- Main adversaries are:
  - Malicious users (short term)
  - Rogue AIs (longer term)

Unwanted/harmful content

- "Unwanted" for generative AI covers both:
  - Unwanted by the user: not following directions
  - Unwanted by the provider: fulfilling user requests would harm third parties or damage the provider's reputation

Exemplary harms from a chatbot

- Facilitating disinformation and political influence
  - Avoid things social media platforms have been criticized for
- Facilitating development of weapons
  - E.g., help an individual or low-resource group build a biological weapon
  - Support going beyond web search results

LLMs in computer security

- Lowest-hanging fruit is augmenting social engineering
- What about finding security bugs?
  - Dual use between defenders and attackers
  - Not yet very effective, interesting cases are harder than other code-support tasks
  - But could be a cause of a high-profile harmful incident

Emergent risks

- Scaling LLMs have often shown novel capabilities
  - Which ones are most concerning in amplifying AI risk?
- Planning, pursuing goals (positive applications too)
- Self-replication (e.g., compare computer worm)
- Real world influence and deception
  - Example: TaskRabbit to solve a CAPTCHA

Medium-term concerns

- Economic disruption
  - E.g., widespread job losses and unemployment
- Acceleration: positive feedback increasing the rate of AI development
  - Reckless competition towards AI goals
  - AI facilitating science and technological development

Some reasons alignment is hard

- Humans already can't agree among themselves on universal values
- Human desires have a lot of implicit side conditions and unstated restrictions
- We don't understand many details of how LLMs work internally
- If AIs become smarter than people, why would they want to obey us?

Hypothetical endpoints

- Paperclip maximizer
  - Seemingly simple goal + great capability = deeply undesirable result
- Will super-human AIs treat humans the way humans have treated non-human animals?
  - Extreme loss of agency is possible without destruction
  - Many different example animals and possible perspectives
  - Too close of an analogy may be unrealistic, since AI may be much less like us than animals are

Precaution and p(doom)

- A trending conversation topic is comparing estimates on the probability of a catastrophic outcome from AI
- Surprisingly many people working in AI have a significant p(doom)
  - Progress is inevitable, or it would be worse without me
- Choosing not to pursue technology because of downside risks is rare
  - Compare: nuclear weapons and energy

Outline

AI safety threat modeling

Announcements intermission

The XZ/SSH backdoor

More crypto failures

DNSSEC

Midterm 2 grade statistics

```
<5 | *
 5 | 02334668999
 6 | 124556667778999
 7 | 001122223444567799
 8 | 001123344667899
 9 | 1234
```

- Mean: 69.99, Median: 71
- There is a +10 points difficulty adjustment on Canvas

Outline

AI safety threat modeling

Announcements intermission

The XZ/SSH backdoor

More crypto failures

DNSSEC

When "fun" is also scary

- Security vulnerabilities and attacks are interesting to hear about when they:
  - Had high impact
  - Use clever or unusual techniques
- These can also be worrying bad news about the overall state of security

One-slide overview

- Maliciously-added code was recently discovered in the XZ-Utils compression package use on Linux systems
- When the affected library was loaded by OpenSSH, it opened a "backdoor" to allow login using an embedded key
- The problem was found only after it had started being incorporated into major Linux distributions

Context of the changes

- XZ-Utils provides the `xz` high-ratio compression tool and a matching `liblzma` library
  - Relatively small and un-glamorous, with one long-term primary maintainer
- The backdoored changes were supplied by a developer JiaT75 who started contributing in 2021
- Common to have rancorous email exchanges with no more direct communication

Contents of the changes

- Random-looking "compression test files" actually had hidden x86-64 code
  - Only these were in the regular Git repository
- Backdoor was incorporated only conditionally for the `.tar.gz` release
  - Various checks performed by obfuscated and encrypted Makefiles and shell scripts

Backdoor functionality

- Back door triggered when the affected library was dynamically linked in the OpenSSH server
- Modified RSA signature checking looks for an elliptic curve signature hidden inside the RSA modulus (e.g., of an OpenSSH certificate)
- If matched, the payload is passed to `system`

Integration story

- SSH isn't supposed to use LZMA compression, and the standard OpenSSH version doesn't
- Major Linux distributions had patched SSH to integrate login notifications with `systemd`
- Easiest way was to link with a `systemd` library, which linked with `liblzma` for other functionality
- In hindsight, these dependencies can be removed

Function replacement mechanism

- Runtime function replacement uses a GNU ELF variant feature named IFUNC (indirect functions)
- Benign use is to switch implementations of a function (e.g., using different CPU feature) without an extra function pointer layer
- The GNU C Library is normally the main user

Who was JiaT75?

- In short: we don't really know
- Likely an assumed name
  - No traces found outside open source
  - Some other identities in conversations seem to be sock puppets
- Not impossible to be a single impressive and motivated individual
- But a coordinated group seems more likely

Outline

AI safety threat modeling

Announcements intermission

The XZ/SSH backdoor

More crypto failures

DNSSEC

WEP "privacy"

- First WiFi encryption standard: Wired Equivalent Privacy (WEP)
- F&S: designed by a committee that contained no cryptographers
- Problem 1: note "privacy": what about integrity?
  - Nope: stream cipher + CRC = easy bit flipping

WEP shared key

- Single key known by all parties on network
- Easy to compromise
- Hard to change
- Also often disabled by default
- Example: a previous employer

WEP key size and IV size

- Original sizes: 40-bit shared key (export restrictions) plus 24-bit IV = 64-bit RC4 key
  - Both too small
- 128-bit upgrade kept 24-bit IV
  - Vague about how to choose IVs
  - Least bad: sequential, collision takes hours
  - Worse: random or everyone starts at zero

WEP RC4 related key attacks

- Only true crypto weakness
- RC4 "key schedule" vulnerable when:
  - RC4 keys very similar (e.g., same key, similar IV)
  - First stream bytes used
- Not such a problem for other RC4 users like SSL
  - Key from a hash, skip first output bytes

Newer problem with WPA (CCS'17)

- Session key set up in a 4-message handshake
- Key reinstallation attack: replay #3
  - Causes most implementations to reset nonce and replay counter
  - In turn allowing many other attacks
  - One especially bad case: reset key to 0
- Protocol state machine behavior poorly described in spec
  - Outside the scope of previous security proofs

Trustworthiness of primitives

- Classic worry: DES S-boxes
- Obviously in trouble if cipher chosen by your adversary
- In a public spec, most worrying are unexplained elements
- Best practice: choose constants from well-known math, like digits of π

Dual_EC_DRBG (1)

- Pseudorandom generator in NIST standard, based on elliptic curve
- Looks like provable (slow enough!) but strangely no proof
- Specification includes long unexplained constants
- Academic researchers find:
  - Some EC parts look good
  - But outputs are statistically distinguishable

Dual_EC_DRBG (2)

- Found 2007: special choice of constants allows prediction attacks
  - Big red flag for paranoid academics
- Significant adoption in products sold to US govt. FIPS-140 standards
  - Semi-plausible rationale from RSA (EMC)
- NSA scenario basically confirmed by Snowden leaks
  - NIST and RSA immediately recommend withdrawal

Outline

AI safety threat modeling

Announcements intermission

The XZ/SSH backdoor

More crypto failures

DNSSEC

DNS: trusted but vulnerable

- Almost every higher-level service interacts with DNS
- UDP protocol with no authentication or crypto
  - Lots of attacks possible
- Problems known for a long time, but challenge to fix compatibly

DNSSEC goals and non-goals

+ Authenticity of positive replies
+ Authenticity of negative replies
+ Integrity
− Confidentiality
− Availability

First cut: signatures and certificates

- Each resource record gets an `RRSIG` signature
  - E.g., `A` record for one name→address mapping
  - Observe: signature often larger than data
- Signature validation keys in `DNSKEY` RRs
- Recursive chain up to the root (or other "anchor")

Add more indirection

- DNS needs to scale to very large flat domains like `.com`
- Facilitated by having single `DS` RR in parent indicating delegation
- Chain to root now includes `DS`es as well

Negative answers

- Also don't want attackers to spoof non-existence
  - Gratuitous denial of service, force fallback, etc.
- But don't want to sign "`x` does not exist" for all `x`
- Solution 1, `NSEC`: "there is no name between `acacia` and `baobab`"

Preventing zone enumeration

- Many domains would not like people enumerating all their entries
- DNS is public, but "not that public"
- Unfortunately `NSEC` makes this trivial
- Compromise: `NSEC3` uses password-like salt and repeated hash, allows opt-out

DANE: linking TLS to DNSSEC

- "DNS-based Authentication of Named Entities"
- DNS contains hash of TLS cert, don't need CAs
- How is DNSSEC's tree of certs better than TLS's?

Signing the root

- Political problem: many already distrust US-centered nature of DNS infrastructure
- Practical problem: must be very secure with no single point of failure
- Finally accomplished in 2010
  - Solution involves 'key ceremonies', international committees, smart cards, safe deposit boxes, etc.

Deployment

- Standard deployment problem: all cost and no benefit to being first mover
- Servers working on it, mostly top-down
- Clients: estimated around 30%
- Will probably be common for a while: insecure connection to secure resolver

What about privacy?

- Users increasingly want privacy for their DNS queries as well
- Older DNSCurve and DNSCrypt protocols were not standardized
- More recent "DNS over TLS" and "DNS over HTTPS" are RFCs
- DNS over HTTPS in major browsers might have serious centralization effects