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

Stephen McCamant University of Minnesota, Computer Science & Engineering

Outline

Al safety threat modeling

Announcements intermission

The XZ/SSH backdoor

More crypto failures

DNSSEC

Kinds of AI safety concerns

- Al failure and misuse: present-day negative consequences of Al not being smart enough, or being used by adversarial people
- Al alignment: long-term risks of Al 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

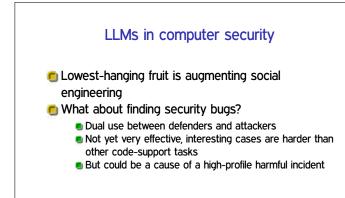
For Al-specific concerns, the main intersection with security is thinking about adversarial threats

- Main adversaries are:
 - Malicious users (short term)
 - Rogue Als (longer term)



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



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
 - Al 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 Als become smarter than people, why would they want to obey us?

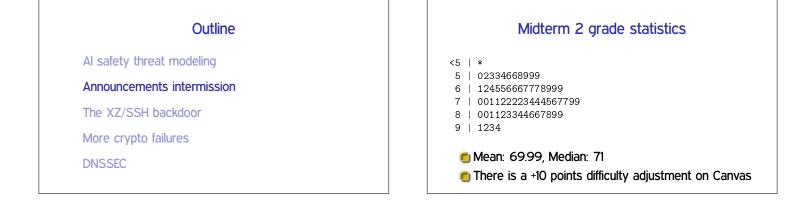
Hypothetical endpoints

🖲 Paperclip maximizer

- Seemingly simple goal + great capability = deeply undesirable result
- Will super-human Als 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 Al 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	
Al safety threat modeling	
Announcements intermission	•
The XZ/SSH backdoor	
More crypto failures	0
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

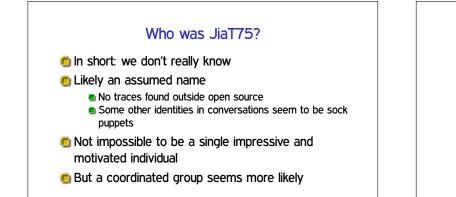
- Back door triggered when the affected library was dynamically linked in the OpenSSH server
- Modified RSA signature checking looks for an elliptic curve signature hidden inside the RSA modulus (e.g., of an OpenSSH certificate)
- If matched, the payload is passed to system

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



Outline

Al 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

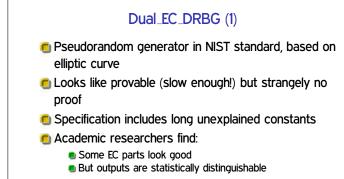
WEP RC4 related key attacks

- 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

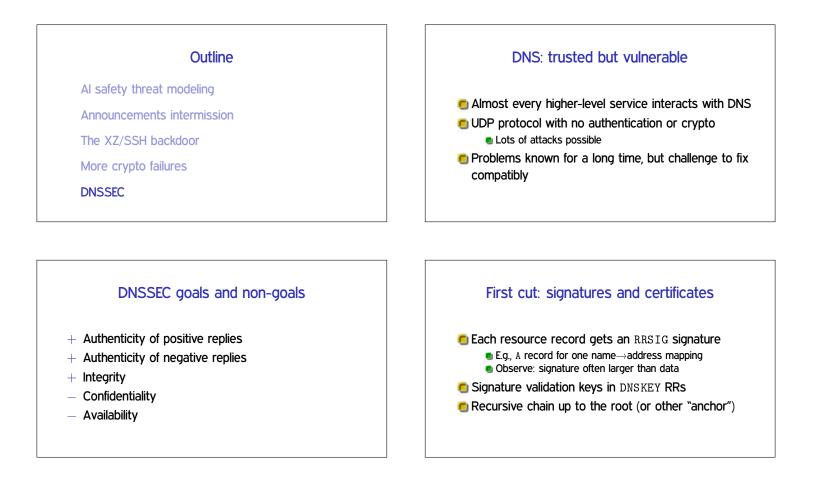
Trustworthiness of primitives

- 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 (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

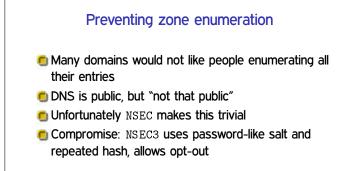


Add more indirection

- 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 DSes 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"



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