CSE 543 - Computer Security

Lecture 2 - Introduction
August 30, 2007
URL: http://www.cse.psu.edu/~tjaeger/cse543-f07/
A historical moment …

• Mary Queen of Scots is being held by Queen Elizabeth …
• … and accused of treason.
• All communication with co-conspirators encrypted.
• Walsingham needs to prove complicity.
Intuition

• Cryptography is the art (and sometimes science) of secret writing
  – Less well known is that it is also used to guarantee other properties, e.g., authenticity of data
  – This is an enormously deep and important field
  – However, much of our trust in cryptographic systems is based on faith (particularly in efficient secret key algorithms)
    – … ask Mary Queen of Scots how that worked out.

• This set of lectures will provide the intuition and some specifics of modern cryptography, seek others for additional details (Menezes et. al.).
Cryptography

- Cryptography (cryptographer)
  - Creating ciphers
- Cryptanalysis (cryptanalyst)
  - Break ciphers

- The history of cryptography is an arms race between cryptographers and cryptanalysts
Encryption algorithm

- Algorithm used to make content unreadable by all but the intended receivers

\[ E(\text{plaintext}, \text{key}) = \text{ciphertext} \]
\[ D(\text{ciphertext}, \text{key}) = \text{plaintext} \]

- *Algorithm is public, key is private*

- Block vs. Stream Ciphers
  - Block: input is fixed blocks of same length
  - Stream: stream of input
Hardness

• Functions
  – Plaintext $P$
  – Ciphertext $C$
  – Encryption key $k_e$
  – Decryption key $k_d$

$$D(E(P, k_e), k_d) = P$$

• Computing $C$ from $P$ is hard, computing $C$ from $P$ with $k_e$ is easy
• Computing $P$ from $C$ is hard, computing $P$ from $C$ with $k_d$ is easy
Example: Caesar Cipher

• Substitution cipher
• Every character is replaced with the character three slots to the right

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |

• Q: What is the key?
Cyptanalyze this ….

“beg nqmf rug”
Cryptanalysis of ROTx Ciphers

• Goal: to find plaintext of encoded message
• Given: ciphertext
• How: simply try all possible keys
  – Known as a brute force attack

1 T F D V S J U Z B M E Q S J W B D Z
2 U G E W T K V A C N F R T H X C E A
3 W H F X U L W B D Q G S U L Y D F B
   S E C U R I T Y A N D P R I V A C Y
Shared key cryptography

• Traditional use of cryptography
• Symmetric keys, where a single key \((k)\) is used is used for \(E\) and \(D\)

\[
D( E( p, k ), k ) = p
\]

• All (intended) receivers have access to key
• **Note**: Management of keys determines who has access to encrypted data
  – E.g., password encrypted email
• Also known as symmetric key cryptography
Key size and algorithm strength

- Key size is an oft-cited measure of the strength of an algorithm, but is strength strongly correlated (or perfectly correlated with key length)?
  - Say we have two algorithms, A and B with key sizes of 128 and 160 bits (the common measure)
  - Is A less secure than B?
  - What if A=B (for variable key-length algorithms)?
Is there an unbreakable cipher?

• As it turns out, yes ....
  – (Claude Shannon proved it)
The one-time pad (OTP)

- Assume you have a secret bit string s of length n known only to two parties, Alice and Bob.
  - Alice sends a message m of length of n to Bob.
  - Alice uses the following encryption function to generate ciphertext c:
    \[ \text{forall } i=1 \text{ to } n : c_i = m_i \oplus s_i \]
  - E.g., XOR the data with the secret bit string.
  - An adversary Mallory cannot retrieve any part of the data.

- Simple version of the proof of security:
  - Assume for simplicity that value of each bit in m is equally likely, then you have no information to work with.
Reading papers …

• What is the purpose of reading papers?
• How do you read papers?
Understanding what you read

- Things you should be getting out of a paper
  - What is the central idea proposed/explored in the paper?
    - Abstract
    - Introduction
    - Conclusions
  - How does this work fit into others in the area?
    - Related work - often a separate section, sometimes not, every paper should detail the relevant literature. Papers that do not do this or do a superficial job are almost sure to be bad ones.
    - An informed reader should be able to read the related work and understand the basic approaches in the area, and how they differ from the present work.
• What scientific devices are the authors using to communicate their point?

• Methodology - this is how they evaluate their solution.
  • Theoretical papers typically validate a model using mathematical arguments (e.g., proofs)
  • Experimental papers evaluate results based on test apparatus (e.g., measurements, data mining, synthetic workload simulation, trace-based simulation).
  • Some papers have no evaluation at all, but argue the merits of the solution in prose (e.g., paper design papers)
• What do the authors claim?
  • Results - statement of new scientific discovery.
  • Typically some abbreviated form of the results will be present in the abstract, introduction, and/or conclusions.
  • Note that just because a result was accepted into a conference or journal does necessarily not mean that it is true. Always be circumspect.

• What should you remember about this paper?
  • Take away - what general lesson or fact should you take away from the paper.
  • Note that really good papers will have take-aways that are more general than the paper topic.
Summarize Thompson Article

- Contribution
- Related work
- Methodology
- Results
- Take away
A Sample Summary

- **Contribution:** Ken Thompson shows how hard it is to trust the security of software in this paper. He describes an approach whereby he can embed a Trojan horse in a compiler that can insert malicious code on a trigger (e.g., recognizing a login program).

- **Related Work:** This approach is an example of a Trojan horse program. A Trojan horse is a program that serves a legitimate purpose on the surface, but includes malicious code that will be executed with it. Examples include the Sony/BMG rootkit: the program provided music legitimately, but also installed spyware.

- **Methodology:** The approach works by generating a malicious binary that is used to compile compilers. Since the compiler code looks OK and the malice is in the binary compiler compiler, it is difficult to detect.

- **Results:** The resulting system identifies construction of login programs and miscompiles the command to accept a particular password known to the attacker.

- **Take Away:** Thompson states the “obvious” moral that “you cannot trust code that you did not totally create yourself.” We all depend on code, but constructing a basis for trusting it is very hard, even today.
Reading a paper

• Everyone has a different way of reading a paper.

• Here are some guidelines I use:

  • Always have a copy to mark-up. Your margin notes will serve as invaluable sign-posts when you come back to the paper (e.g., “here is the experimental setup” or “main result described here”)

  • After reading, write a summary of the paper containing answers to the questions in the preceding slides. If you can’t answer (at least at a high level) these questions without referring to the paper, it may be worth scanning again.

  • Over the semester, try different strategies for reading papers (e.g., Honeyman approach) and see which one is the most effective for you.
Reading a systems security paper

- What is the security model?
  - Who are the participants and adversaries
  - What are the assumptions of trust (trust model)
  - What are the relevant risks/threats

- What are the constraints?
  - What are the practical limitations of the environment
  - To what degree are the participants available

- What is the solution?
  - How are the threats reasonably addressed
  - How do they evaluate the solution

- Take away: key idea that drives the design, e.g., generalization (not solely engineering)
- Hint: I will ask these questions when evaluating course project
Armando Fox’s Presentation Hints

• Know Thy Jargon
• Keep the Big Picture in Mind
• Tell a Story
• Pace Yourself
• Tell ‘em What You Told ‘em
• Be Ready for Questions
Presenting a Paper

• Similar to Summary
  • Same basic areas as a summary

• Different Than a Summary
  • Engage the audience
  • Identify an insight
  • Argue a point
  • Make an extension

• Relate to Security Concepts

• What Strikes You
  • Passion is good
Presenting Thompson’s Idea

• Do you trust this program?
  • What is the *trust model* of a typical program?

• How does the Thompson compiler impact the trust model?
  • What are the *threats*?

• Shows that we cannot assume trust blindly
Methodology

• Define **Trojan horse**
  • Program that performs as expected or looks legitimate
  • But, it also contains a malicious function
    • Capture secrets, impact integrity, cause DoS

• Trojan horse in a compiler
  • Compilers convert one language to another
  • How do we know that it works correctly?
    • Check the code of the compiler

• Put the Trojan horse in the compiler compiler binary
  • No source code gives it away
  • Supposedly worked for a while
Establishing Trust

• How do we establish trust in a program?
  
  • Code: binary and/or source
  
  • Source: E.g., MS authenticode, signed Jar files
  
  • Behavior: Track how it runs
  
  • Community: Ask someone else if it’s OK
  
  • Inputs: What about what we put in it?

• What are the risks of these approaches?
  
  • How does it accumulate?

• Even if we get a valid program...
  
  • May be modified (viruses) and reconfigured
The course project requires the student to execute some limited research in security.
- Demonstrate applied knowledge
- Don’t try to learn some new non-security field
- Be realistic about what can be accomplished in a single semester.
- However, the work should reflect real thought and effort.

The grade will be based on the following factors: novelty, depth, correctness, clarity of presentation, and effort.
Deliverables

- The chief product of the project will be a conference style paper. There will be several milestones:
  - Project Choice (9/11/07)
  - Background and Related Work (10/9/07)
  - Experiment Proposal (10/30/07)
  - Project Status Slides (11/15/07)
  - Final Project Write-up (12/20/07)
- Everyone will present to 12/3-4/07, describing the project, progress, expected results and related work
- This is the most important factor in your grade (30%) so you better take it seriously
  - E.g., an exceptionally good project may help your grade
Project Choice

- Due on Sept 11, 5:00pm
- Order list of projects
  - Choose three projects in order of interest
- Choose up to 3 collaborators
  - Optional
  - Get a sense of groupings
- I will choose your project and group
  - Hopefully, I can resolve the constraints implied
  - One group per project
  - A functional group
- Project choices -- see course calendar