CSE433/533 - Computer and Network Security
Security Research Methods

Professor Kevin Butler
Winter 2011
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.

These are the best areas to find an overview of the contribution
Understanding what you read (cont.)

• 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).

  ‣ **Empirical** research evaluates by measurement.

  • Some papers have no evaluation at all, but argue the merits of the solution in prose (e.g., paper design papers)
Understanding what you read (cont.)

• 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**: 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
- Motivation
- 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).

- **Motivation**: People need to recognize the security limitations of programming.

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

- **Take away**: *What is the transcendent truth?????* (see next slide)
Turtles all the way down ...

• **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.

• ... or “**trust in security is an infinite regression** ...”

“A well-known scientist (some say it was Bertrand Russell) once gave a public lecture on astronomy. He described how the earth orbits around the sun and how the sun, in turn, orbits around the center of a vast collection of stars called our galaxy. At the end of the lecture, a little old lady at the back of the room got up and said: "What you have told us is rubbish. The world is really a flat plate supported on the back of a giant tortoise." The scientist gave a superior smile before replying, "What is the tortoise standing on?" "You're very clever, young man, very clever", said the old lady. "But it's turtles all the way down!"

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 term, try different strategies for reading papers 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

• What is the take away?
  ‣ key idea/design, e.g., generalization (not solely engineering)

• **Hint:** I will ask these questions when evaluating course project.
Course Projects

• The course project requires the students execute some limited research in security.
  ‣ Demonstrate applied knowledge
  ‣ Don’t try to learn some new non-security field
  ‣ Be realistic about what is possible in a one quarter.
  ‣ 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 8 page conference style paper (not including refs). There will be several milestones:
  ‣ Project Choice (1/22/13)
  ‣ Background and Related Work (2/12/13)
  ‣ Abstract/Intro (2/28/13)
  ‣ Final Project Presentation (3/14/13)

• This is a very important factor in your grade (25%)
  ‣ An exceptionally good (poor) project may help (kill) grade
Project Choice

• Project choices due before class on Jan 22
• Group meetings Jan 22 and Jan 23 (sign up for time)
• Order list of projects
  ‣ Propose at least three unique projects in order of interest
• Choose groups of 1,2 (up to 4 with approval)
  ‣ Get a sense of groupings
• I will approve/choose your project and group
  ‣ Hopefully, I can resolve the constraints implied
  ‣ One group per project
  ‣ A functional group
Topic Examples

• Web systems and Networking
  ‣ Evaluate the security of a Web 2.0 application
  ‣ Develop a secure anonymous communication scheme

• Mobile Systems
  ‣ Design and build an Android/iPhone security application

• Cloud Systems
  ‣ Design a cloud component for ensuring data security

• User Studies
  ‣ Measure the effectiveness of passwords, card systems

• Embedded Systems
  ‣ Build a security app for an embedded device or eval security

• **Note**: picking a topic is very important, and should almost certainly involve an area that you know well
Bad Ideas

• Encryption library for SMS (or anything else)
• Password wallet
  ‣ these have both been done ad nauseum
• Firewall rule checker
• Steganographic schemes
• Anything that requires massive amounts of data that you don’t have access to
  ‣ e.g., online gaming trends requiring snapshots of all users
• Anything you can’t hit the ground running with
Idea Formulation

• The essential part of successful research is picking good problems and solutions
• Q: how do you do this?
Idea Formulation

• Good approaches to finding ideas:
  ‣ First, read several (good) papers in a particular area
  ‣ If a new topic area, you need to become familiar with the problems, solutions and terminology of the community
  ‣ Ask the following questions and write down answers:
    • What are the problems this area addressss?
    • What are the methodological tools that people bring to bear in addressing problems in this area?
    • How is the field evolving?
    • How does your skill set apply to problems addressed?
    • How are expected changes in the larger CS community going to affect known problems and solutions?
Idea Formulation - LISTING

Do the following exercises:

- (5 min) listing: make a quick list of 1-5 word phrases that would be used by/related to/observance of field and problems & solutions
  - Not an outline, no ordering to list: use your imagination
  - Don’t overthink: some of list will be nonsense, don’t filter thoughts

- Example: if I was looking at a paper about firewalls, I might come up with the following as a start:
  - policy validation, distributed firewalls, bad for detecting viruses... this is general, should contain thoughts more specific to paper content
  - e.g., better algorithm than author -- use graph theory
Brainstream

- storage provenance, network provenance, tracking information as it goes between systems in the cloud, state of systems when creating data, processing data, sending data to the next stage, pipelines of information flow, pipelines in SCADA systems, relation of provenance to real world workflows, real world workflows vs workflows of information between applications, how isolated are applications in their data use?, many phone applications are isolated, but communicate with cloud servers, are smartphone apps producers or consumers of information?, does this relate to provenance anymore? healthcare workers use smartphones rather frequently, can geographic location be used as a provenance source in a phone-cloud system? location and provenance are both sometimes used for access control.
Using the results

• Examine contents closely - they’ll tell a story
• Find singletons or clusters or phrases and see if they provide some new angle on a problem/issue
• E.g., geographic location used as provenance source
  ‣ Leads to the following idea:
    • Q: in what environments can location provenance be used?
    • Q: what real world analogies are there?
  ‣ Only read something written in similar spatial/provenance context
    • Paper: “Situational Memory Recall for Access Control Policy”
Class Expectations

• This class will test you as a student
  ‣ Not a lot of downtime this quarter: be ready

• You’ll need to do more than just regurgitate class
  ‣ If you can’t apply what you’ve learned, defend a position, argue against one another, you’re not going to have fun

• Security and scholarship are mindsets
Why write a paper?

• There are many reasons to write a paper:
  ‣ Articulate a new idea, thought, or observation ...
  ‣ Document your research ...
  ‣ Talk about new (observed) phenomenon ....
  ‣ Advance your career ...
  ‣ Because you have to ...

• **Reality**: publication is the coin of the realm in science, failure to do this successfully will lead to failure. You have to be effective at this to be a good (a) graduate student, (b) faculty member, or [sometimes] (c) researcher in professional research laboratory (IBM/AT&T/MS)
Where to publish?

• Venues for publication:
  ‣ Tech report
  ‣ Workshop
  ‣ Conference
  ‣ Journal

• Often your work will work through these from *preliminary* to *archival* versions of the work, sometimes branching or joining.

• *Book*: less frequent, more work.
Publication Tiers

• Not all publication venues are valued the same. Publication “tiers” tell the story

• 1st tier - IEEE S&P, USENIX Sec, CCS, \textit{TISSEC}, \textit{JCS}
  ‣ 1.5 NDSS

• 2nd tier - ACSAC, ACNS, ESORICS, CSF, RAID, \textit{TOIT}

• 3rd tier - SecureComm, ICISS

• 4th tier - HICSS
  ‣ SClgen (WMSCI 2005)
Journal publication

- The **editor-in-chief** (EIC) receives the papers as they are submitted.
- The papers are assigned to **associate editors** for handling.
- **Anonymous reviewers** rate the paper:
  - Accept without changes
  - Minor revision
  - Major revision
  - Reject
• The **PC Chair** is the person who marshals the reviewing and decisions of a conference. This is different than the **general chair**.

• **PC members** review, rate and discuss, the paper, then vote on which ones are accepted.

• The **acceptance rate** is the ratio of accepted to submitted papers.
Paper evaluation

• A paper is evaluated on
  ‣ Novelty
  ‣ Correctness
  ‣ Impact
  ‣ Presentation
  ‣ Relevance
  ‣ “hotness”
Parts of a paper

- Parts of paper (vast generalization)

1. Abstract
2. Introduction
3. Related Work/Background
4. Solution/Problem
5. Evaluation/Analysis/Experiment
6. Discussion (often, but not always)
7. Conclusions
Abstract

• One sentence each for:
  ‣ Area
    • Topic of work
  ‣ Problem
    • What’s the issue?
  ‣ Solution
    • How do you propose to address the problem?
  ‣ Methodology
    • What’s the experiment?
  ‣ Results
    • What did you find?
  ‣ Take Away: Lesson
Introduction

• One paragraph each on:
  • Area
    ‣ More elaborate
  • Problem
    ‣ Scenario
  • Why is problem not solved
    ‣ Brief of related work or the challenge
  • Proposed insight ("In this paper, ...")
    ‣ What is the experiment?
  • Contributions -- What will the reader learn?
  • Boilerplate outline (?)
Related work/Background

• This is a statement of the work that led to this one.
  ‣ who this work relies on
  ‣ who has done work in the area
  ‣ areas that inspired this work (not just technology)

• There are several reasons for related work section:
  ‣ Motivate the current work
  ‣ Differentiate from past work
  ‣ Establish “bona fides”

• Background
  ‣ Outline the Problem
    • May use an example scenario
  ‣ Material Related to the Solution
    • Why hasn’t it been solved
Background and Experiment

• Experiment
  ‣ Means of showing truth
  ‣ Big Insight -- Hypothesis -- Claim
    • Show why it is interesting
  ‣ Expected Results
    • Informal proof/argument that is true

• Experiment types
  ‣ Empirical - measure some aspect of the solution
  ‣ Analytical - prove something about solution
  ‣ Observational - show something about solution
Implementation and Results

• Implementation: Experimental Platform
  ‣ Exact specification of platform
  ‣ Design may have more than implementation -- what did you implement?
  ‣ How are key design features/mechanisms implemented?

• Results
  ‣ Summarize -- what do the results mean?
  ‣ Specific experiments
    • We did X, saw Y
  ‣ What do the experiments prove
  ‣ What other experiments would you want to do based on these results?
Conclusion

• Like the abstract in past tense
• Problem
  ‣ What was the problem?
• Solution
  ‣ What was the insight and why was it expected to work?
• Method and Results
  ‣ What did you find?
• Take away: Lesson
• Future work
Hint

• Intro: tell them what you are going to tell them
• Body: tell them
• Conclusion: tell them what you told them.
What is Research?

• Which activities are research?
  ‣ Designing a new protocol?
  ‣ Building an implementation of a protocol?
  ‣ Measuring the cost of a protocol?
  ‣ Formally evaluating the correctness of a protocol?
  ‣ Developing methods of simplifying the protocol?
What is not research?

- Arguing quality of a protocol?
- Arguing appropriateness of a protocol?
- Surveying a field?
- Illustrating limitation of common practice or system?
Research vs engineering

• Novelty...

• Importance... (sort of)

• Discovering new fact/idea
Why is there bad research?

• Lots of papers are bad research. Authors:
  ‣ don’t formulate the problem well (or at all)
  ‣ don’t motivate the problem well/at all
  ‣ address unimportant/moot problem
  ‣ aren’t familiar with breadth/depth of area
  ‣ don’t know the related work
  ‣ don’t have solution that is coherent
  ‣ don’t apply methodology well
  ‣ don’t draw correct conclusions from results
  ‣ don’t present work well enough to be understood