CS 161 Fall 2005 — Computer Security — Joseph/Tygar/Vazirani/Wagner — HW 1

Due Friday, September 23 at 11am

Please include the following at the top of the first page of your homework solution:

Your full name
Your login name
The name of the homework assignment (e.g. hw3)
Your section number

Staple all pages together, and drop them off in drop box #2 (labeled CS161/Fall 2005) in 283 Soda by 11am on the due date.

Homework exercises:

1. **(1 pts.) Any questions?**
   What's the one thing you'd most like to see explained better in lecture or discussion sections? A one-line answer would be appreciated.

2. **(4 pts.) Getting started**

   (a) Read the course web page. Write on your homework, immediately after your name, the following sentence: "I understand and will comply with the academic integrity policy."

   (b) What is the course policy regarding working on homework in groups?

   (c) Register with the grading system. These are the instruction for becoming registered for the class:

   i. Login to your instructional named account (cory account)

   ii. Set your environment variable $MASTERDIR to /home/ff/cs161
   - if your are using tcsh or csh: `setenv MASTERDIR /home/ff/cs161`
   - if you are using bash: `export MASTERDIR=/home/ff/cs161`

   iii. Run `register`

   That's it. If you want to, you can put the environment setting for the MASTERDIR variable in your .cshrc or .bashrc, so you don't have to set it every time you login.

   (d) What is David Wagner's favorite security-related book? The answer is found on the course newsgroup, `ucb.class.cs161`. Look for the post from David Wagner titled "The answer to question 1(d)," and write down the answer you find there. Instructions on how to access the newsgroup may be found on the course web page.

   (Why are we having you do this? The class newsgroup is your best source for recent announcements, clarifications on homeworks, and related matters, and we want you to be familiar with how to read the newsgroup.)

3. (45 pts.) Attack Trees

An attack tree is an AND-OR tree. Each node is labeled with an attack goal, i.e., an effect that an attacker might try to achieve by mounting some kind of attack. The root node corresponds to the ultimate attack goal (e.g., violate one of the security goals). The child nodes of a node represent subgoals that help the attacker make progress towards the goal at the parent. If the parent is an OR node, then achieving any one of the subgoals suffices to achieve the goal at the parent. If the parent is an AND node, then the goal at the parent is achieved when you achieve all of the subgoals at the children. You stop refining the goals when you reach an acceptable level of detail: e.g., when each leaf contains a simple elementary goal whose difficulty of achieving it can be easily assessed.

You can find more information about attack trees, and a number of example trees (including the example from the Sept 8th sections) at: `http://www.schneier.com/paper-attacktrees-ddj-ft.html`

(a) Your attack target is to find the contents of a file owned by USER and belonging to GROUP with chmod value 640 on a UNIX file system (owner: read-write access; group: read-access). Here USER is some arbitrary username, and GROUP is some Unix group. Describe as many different ways to read the file as possible (at least four). For example, one way is to discover the root password. Express these goals using an attack tree with one level. (If you don't know how UNIX file system security works, you may refer to the textbooks or do a google search for this topic on the web, for example: `http://tille.xalasys.com/training/unix/x262.html`)

(b) For each of those goals, design successive sub-goals. For example, one way to get the root password is to watch the system administrator log in and remember his password. Incorporate these sub-goals into the attack tree from part (a). The final tree should include a total of 15 to 40 nodes (the more the better).

(c) Once you have an attack tree, you can do interesting things with it. For instance, if you label each leaf with the cost of achieving the corresponding goal, then you can propagate costs up the tree by summing at each AND node and taking the min at each OR node; the result is the cost of achieving the top-level security goal (and the cheapest ways to do so).

Assign rough costs to the tasks in your attack tree, measured in terms of $t/(1-p)$, where $t =$ the time required to achieve the task, and $p =$ the chance of detection. For example, watching the system administrator might require $t = 8$ hours (to catch the right time that he types his password) and the chance of detection is maybe $p = 50\%$. This gives it a cost of $8/(1-0.5) = 16$ units.

(d) Calculate the total attack cost of reading a private file using your model.

(e) Make a constructive suggestion (based on your attack tree) to make private UNIX files more secure.

(f) Construct a new attack tree based upon your suggested change, and calculate the improvement in security that your change will have.

(g) Do you think this is a good way to analyze security? Why or why not?

4. (30 pts.) In-band and Out-of-band Signaling

(a) In-band signaling in a communication architecture shares the same communication infrastructure for both data (e.g., voice) and control information (e.g., connection setup, billing, connection teardown).

Out-of-band signaling in a communication architecture relies on a separate communication infrastructure for data and another one for control information.

Explain one advantage and one disadvantage of a communication architecture that uses in-band signaling versus one that uses out-of-band signaling.

(b) Signaling System #7 (SS#7) provides a separate, out-of-band signaling system. Explain how this signaling system improved the security of the telephone network when it was introduced, and explain why it is inadequate today.

(c) Suggest a way to improve SS#7's security without forcing everyone to buy a new telephone, or changing the way people place calls.

5. (20 pts.) Default Configurations

Give two reasons/examples of why default configurations for software/hardware can be a security problem.