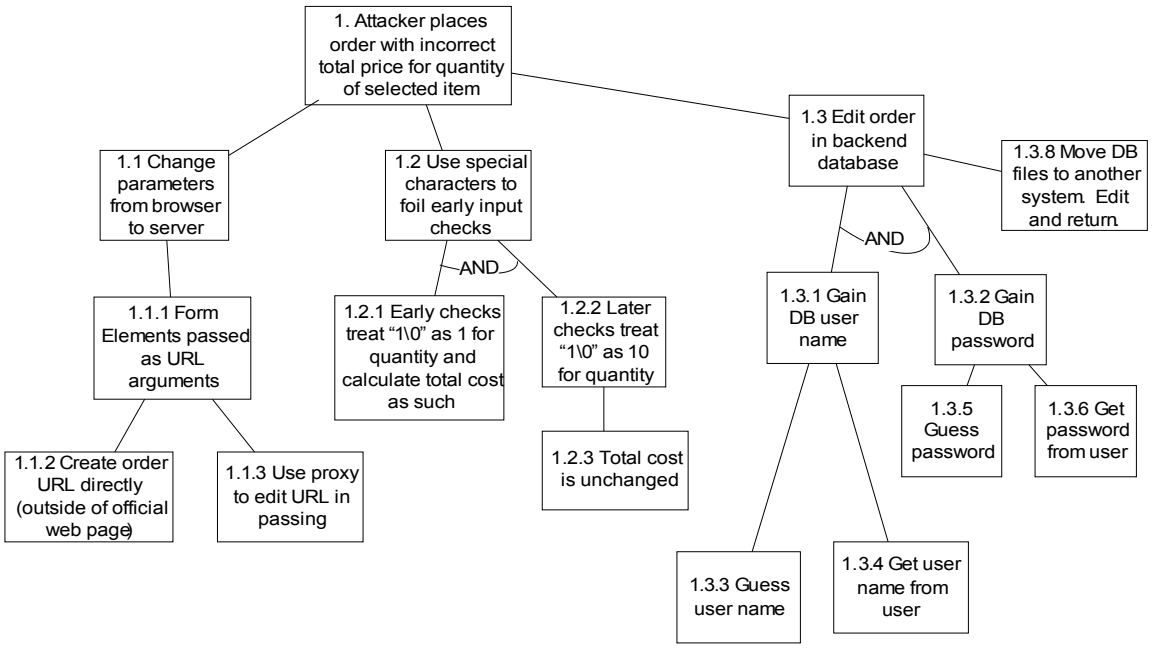
Name:

Information Assurance: Homework 8

Due November 11, 2009. No late handins, so we may post the answer key in time to help students study for exam 2.

1. Consider the threat tree below. This represents a threat from a web based shopping page.



   a. Enumerate all the attack paths in the tree.
   b. Identify a set of controls that would address all attack paths.
   c. Can this threat be completely controlled? Why or why not?

Name:

2. The text describes polymorphic viruses as one technique used by virus writers to avoid detection via scanning. Assume you were given the job of creating a virus detection program that would catch polymorphic viruses. What would your design be? You can assume that the virus only uses polymorphic techniques, no encryption.

3. Consider a program posted at http://www.cs.illinois.edu/class/fa09/cs461/hw8.zip. This simple program has not one but two buffer overflow vulnerabilities in two different functions. The program takes two arguments: -f1 or -f2 to indicate which function to invoke and a count of the number of bytes to allocate for a buffer.
    a. The zip file includes a Makefile which will create two binaries (tested on csil-linux-ts1). The hw8-plain binary is a vanilla gcc compile. Try this program on both version of the function with different buffer sizes. What happens?
    b. The other binary creates hw8-protected which is the same program compiled with the stack-guard canary values (details of stack guard at http://www.usenix.org/publications/library/proceedings/sec98/full_papers/cowan/cowan.pdf). On the csil machines stack guard is not enabled by default, but this is not the case on all distributions. If you compile on some other machine, check your compiler's man page to determine whether you need to enable stack guard in this part or disable it in part a. Run the same experiments again. What happens this time?
    c. Libsafe uses a runtime approach to detect and protect against stack smashing. The man page is posted at http://www.cs.uiuc.edu/class/fa07/cs461/libsafe.8.html. Unfortunately, this library doesn't seem to work any more. Based on the man page, how do you think hw-plain would operate in the presence of libsafe?
    d. What is one unique benefit of each approach (stack guard and libsafe)?

4. Consider ARP cache poisoning.
    a. Is this attack enabled by a deficiency in Confidentiality, Integrity, or Availability? How?
    b. With a successful ARP cache poisoning, can the attacker attack Confidentiality, Integrity, and/or Availability? How?
    c. How can a client be aware that they are the victim of ARP cache poisoning? How can the client limit the impact of a successful ARP cache poisoning?
    d. If you have control of an installation, how could you set up a network to eliminate or reduce the occurrence of an ARP cache poisoning?