CSci 5271: Introduction to Computer Security

Exercise Set 4 **due: Thursday, November 21st, 2013**

Ground Rules. You may choose to complete these exercises in a group of up to three students. Each group should turn in **one** copy with the names of all group members on it. You may use any source you can find to help with this assignment but you **must** explicitly reference any source you use besides the lecture notes or textbook. An electronic (plain text or PDF) copy of your solution should be submitted on the course Moodle by 11:55pm on Thursday, November 21st.

1. Random numbers with limited entropy. (20 pts) Alice, Bob, and Carol are employees of a company (in a small island nation) setting up an online casino website based on card games like blackjack. They realize that if users could predict the sequence of pseudorandom numbers used to deal cards, they could win reliably and hurt the company's bottom line. They've found a good cryptographically-strong pseudorandom number generation algorithm to use in the shuffling process, but they're having trouble deciding what to use as the seed when they initialize the generator at the start of each user's session.

(Following the usual good security design principles, they don't want the security of the games to depend on the choice of the pseudorandom generator or the shuffling algorithm being secret; they might also want to franchise their casino out in the future. But practically speaking, reverse-engineering those algorithms would be a significant effort, so attacks that worked without the attacker needing to do so would be particularly damaging.)

(a) Alice suggests seeding the PRNG with the time: specifically the date and time as returned by the Unix `time` system call, equal to the number of seconds since midnight, January 1st 1970 UTC. Explain why this is a bad idea by describing an easy attack.

(b) Bob suggests seeding the PRNG with the process ID of the login CGI script. Assuming this script runs once for each login, and process ID numbers are assigned sequentially in the range of 2 to 65535, describe an attack against this scheme.

(c) Carol suggests combining Alice and Bob's ideas by taking the time and the PID and XORing them together. But Alice points out a problem with this scheme that involves a user logging in once every second. Explain the details of her attack and why it's a problem.

(d) After the problems with their previous schemes, Alice, Bob, and Carol have called you in as a consultant. Suppose that because of the architecture of the system, the seed is required to be a deterministic function of the time in seconds and the PID. Propose a better combining function that takes these two pieces of information as input and produces a bit string (of any length) than can be used as a seed. Evaluate the security of your approach.

2. Firewall Schmirewall. (20 pts) Sarah is installing a network firewall for her company. Being familiar with the principle of fail-safe defaults, she has configured the firewall to DENY all packets by default. Now she needs to identify the minimal access rules that will allow her organization to use its Internet connection. For example, her organization will need to be able to send and receive email through the firewall, and uses a central mail server at IP address 10.1.100.100. So she has added rules to the firewall that look like this:

SRC ADDR	DEST ADDR	SRC PORT	DST PORT	PROTOCOL	ACTION
10.1.100.100	*	*	sendmail	TCP	ALLOW
*	10.1.100.100	*	sendmail	TCP	ALLOW

The organization has determined that it will also require the following kinds of Internet access:

- Incoming SSH access to a VPN server, at 10.1.100.200.
- Access to the web, through a proxy that whitelists approved sites. The proxy's address is 10.1.200.200.
- Outgoing SSH access to three client sites: 0.1.2.3, 42.42.42.42, and 3.14.15.9.

List the minimal set of firewall rules necessary to allow these connections. List some potential vulnerabilities associated with this ruleset. Can the firewall and proxy servers defend against these vulnerabilities?

3. False Positive Answer. (20 pts) Anderson's chapter 11 details several ways to defeat physical intrusion detection systems (a.k.a. "burglar alarms"). One of the common ones is to artificially create "false" alarms so that the true alarm is ignored. Let's investigate this idea with respect to computer intrusion detection systems.

(a) An old Snort rule says that any HTTP packet that includes "`/..%c0%af../`" should trigger an alarm, as an attempted IIS exploit. Explain why in "normal" usage this rule would have a low false positive rate.

(b) Suppose Eve discovers a web server, `vulnerable.org`, that is vulnerable to the IIS Unicode exploit and she wants to exploit the hole without having it noticed. What are a few ways Eve can temporarily increase the false positive rate at `vulnerable.org` for the rule, without getting her IP address noticed?

(c) What can you conclude about "advertised" false positive and false negative rates?

4. Virus Virii. (20 pts) Sam has invented a brand-new virus detector, ViruSniff, and he claims it is 100% effective - if executable F is a virus, then ViruSniff(F) will output "`VIRUS!!!`".

(a) Does ViruSniff's claim conflict with the undecidability of the halting problem? Why or why not? (Hint: is there a simple program that can do exactly what Sam says ViruSniff can do?)

(b) Some hackers reverse engineer ViruSniff and post its algorithm online: it turns out that ViruSniff does processor emulation of the first 10000 instructions of an executable, and then applies a fancy signature matching algorithm (that no one seems to understand) to the sequence of instructions and memory changes to decide if the program is a virus or not. Explain how to change any program that runs for at least 10001 instructions, and does not trigger the `VIRUS!!!` alert, to propagate a virus such that the altered program will also fail to trigger the alert. What does your strategy say about Sam's claim?

(c) Given your knowledge of the attack from (b), how might you enhance ViruSniff to work against the new virus-writing strategy? Evaluate the potential effect of your change on the false-positive rate.

5. Denial of Service Denial. (20 pts) Sly and Carl are really concerned about the possibility of DoS attacks against their web server program.

(a) Sly has developed a new module for his web server that he claims will prevent DoS attacks by slowing them down. In Sly's module, every incoming HTTP request is put into a queue, with a timestamp and a "delayed" bit marked as false. When it is ready to serve a request, the web server takes the first request in the queue. If the "delayed" bit is false and there are no other requests from the same IP address in the queue, it serves the request immediately. If the "delayed" bit is false and there is at least one other request from the same IP address in the queue, the "delayed" bit is set to true and the request is re-inserted at the end of the queue. If the delayed bit is set to "true," then the request is served **if** the current time is at least 1 second greater than the request timestamp, and **otherwise** the request is sent to the end of the queue again. Sly's idea is that this will allow the site to deal with requests from legitimate users in preference to DoS attack requests.

Will Sly's scheme work to prevent a DoS attack from making his web server unusable by normal users? Give a detailed explanation.

(b) Inspired by BitTorrent, Carl has a different suggestion for preventing DoS. In Carl's solution, whenever client downloads a page, he also downloads an ActiveX control that acts as a mini web server for that page and its contents only. Then when the main server starts to be overloaded, it uses HTTP redirects to point new clients to servers running on old clients. The new clients can then download the pages from old clients directly, without using any more of the main server's bandwidth.

There are some implementation challenges with Carl's scheme, such as browsers that are behind firewalls or don't support ActiveX. But let's assume these are adequately solved. How well will Carl's scheme work against attackers who want to make his site unusable?