CS 155: Spring 2009
June 2009

CS 155 Final Exam

This exam is open books and open notes. You may use course notes and documents that you have stored on a laptop, but you may NOT use the network connection on your laptop in any way, especially not to search the web or communicate with a friend. **You have 2 hours.** Print your name legibly and sign and abide by the honor code written below. All of the intended answers may be written well within the space provided. You may use the back of the preceding page for scratch work. If you want to use the back side of a page to write part of your answer, be sure to mark your answer clearly.

The following is a statement of the Stanford University Honor Code:

A. *The Honor Code is an undertaking of the students, individually and collectively:*

   (1) *that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;*

   (2) *that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.*

B. *The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.*

C. *While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.*

I acknowledge and accept the Honor Code.

_____

(Signature)

_____

☐ **GRADUATING?** *(Print your name,* legibly!*)*

Prob	# 1	# 2	# 3	# 4	# 5	# 6	# 7	Total
Score								
Max	18	12	14	12	18	12	14	100

1. (18 points) ..... Short Answer

(a) (3 points) How does salt slow down an offline dictionary attacks? Specifically, how many hash computations are needed to run a dictionary attack on n passwords, with and without salt? Assume d passwords in the dictionary.

(b) (3 points) Assume that you want to slow down an offline dictionary attack even further. What can you do ? *Hint:* The goal is to add a time factor that is not related to the complexity of the password.

(c) (3 points) What is a race condition and why are they a security problem?

(d) (3 points) Suppose you find a memory leak in a third-party library, e.g. the library `cleanup` function forgets to call free on a certain pointer `p`. You notice that some applications fix the problem by freeing the memory themselves. How could you fix the library `cleanup` function to free `p` without introducing a double-free vulnerability in applications that free the memory themselves?
You are only allowed to change the library. If you need to make an assumption about how the `free` function works, make sure to state your assumption explicitly.

(e) (3 points) In the DRM lecture we discussed two offline DRM systems, CSS and AACS. What is the main difference in key management between the two and why is one better than the other?

(f) (3 points) Traceback is a proposed defense against DDoS attacks. Briefly explain the limitations of this defense.

2. (*12 points*) ..... Control hijacking

In class we described various defenses against control hijacking attacks, such as NX bit, stackguard, and ASLR. Recall that NX and stackguard crash an application once an attack is detected.

(a) (*3 points*) Suppose a web server uses all the techniques above. Moreover, suppose it takes a minute to restart the web server after a crash. Now, suppose the web server comes under a repeated stack-based control hijacking attack that would compromise the server unless the defenses above are in place. What is the effect of the attack on the web site? Why?

(b) (*3 points*) Propose a web server architecture that avoids the problem above.

(c) (*3 points*) Write sample server-side code that contains a buffer overflow vulnerability such that the overflow can result in malicious data being written to the database *without hijacking program flow control*. Are any of the techniques above effective at stopping this?

(d) (*3 points*) Propose a defense against the form of attack illustrated in part (c).

3. (*14 points*) ..... HTTPS

(a) (*3 points*) Suppose an HTTPS page links to an HTTP iframe where both are loaded from the same origin. The browser shows a mixed content warning dialog. Explain the risk of clicking OK on this dialog.

(b) (*3 points*) Suppose an HTTPS page links to an HTTP iframe where the two are loaded from different origins. Should the browser display a warning dialog? If so explain why. If not, explain why not.

(c) (*3 points*) In class we discussed the `ssl_strip` attack on HTTP. Suggest a browser extension that can defend against the attack. You may assume the user only visits a small number of sites where SSL is needed and the user can supply this list of sites to the extension.

(d) (*5 points*) Suppose a corporate network wants to monitor all web traffic originating inside the corporate network and headed towards an external web site. The monitoring needs to be done transparently, i.e. without any change to the user experience. The admin can route all web traffic through a transparent web proxy and the web proxy will monitor the traffic. The problem is that when an employee uses SSL from his office computer, the proxy sees encrypted traffic and cannot do its job.

Propose an architecture that will enable the proxy to monitor SSL traffic in cleartext. Again, this should be done without affecting the user experience, i.e. without causing any security dialogs to popup on the employee's browser.

Hint: You many assume that the employee's computer is installed by the admin and that the admin installed a corporate CA certificate in the browser's certificate store. You may also assume the browser is configured to use TLS client hello extensions. Be precise in your explanation of how the proxy works.

4. (*12 points*) ..... Access Control

One way to express or analyze access control policies is using logical "if–then" rules, often written backward in "then–if" form (as in the language Prolog). For example, the rule below says that a Principal has a specific Permission to access a Resource, if the Principal is a member of a Group that is given that Permission for the Resource.

$$\text{canAccess}(\text{Principal}, \text{Permission}, \text{Resource}) \leftarrow \text{member}(\text{Principal}, \text{Group}), \text{canAccess}(\text{Group}, \text{Permission}, \text{Resource})$$

A single rule such as this can apply to all permissions in the system (such as read permission, write permission, and execute permission) and all resources (such as all files). A comma "," on the right-hand-side of a rule means *and*. Names such as Group and Principal that begin in upper-case are variables and can be replaced by any value.

(a) (*3 points*) In order to decide whether a user can access a file, for example, rules are combined with facts. For example, consider the facts

$$\text{member}(\text{alice}, \text{administrators})$$
$$\text{member}(\text{bob}, \text{users})$$
$$\text{canAccess}(\text{administrators}, \text{Permission}, \text{Resource})$$

Explain which user(s) these facts (combined with the rule above) allow to execute file `'C:\\Program Files\\Adobe\\Adobe Help Center\\BIB.dll'`.

(b) (*3 points*) We can associate access lists with files using facts and rules based on formulas of the form

$$\text{file}(\text{FileID}, \text{ReadList}, \text{WriteList}, \text{ExecList}, \text{Owner}),$$

where ReadList, WriteList, and ExecList are the groups allowed to read, write, and execute the file (respectively), and Owner is the owner of the file. For example:

$$\text{file}(\text{f10345}, \text{read10345}, \text{write10345}, \text{exec10345}, \text{alice})$$
$$\text{member}(\text{bob}, \text{read10345})$$
$$\text{member}(\text{carol}, \text{write10345})$$
$$\text{member}(\text{carol}, \text{exec10345})$$

Write a single rule that allows the WriteList group to write that file (when combined with other rules given in this problem). Your rule should imply canAccess(carol, write, f10345), for example, but should not mention carol or f10345.

$$\text{canAccess}(\text{WriteList}, \text{write}, \text{FileID})$$

$$\leftarrow \text{file}(\text{FileID}, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_)$$

(c) (*1 point*) An access control system could give a process created by executing a file has the same permissions as the owner of the file. What feature of Unix achieves this?

(d) (*5 points*) We can write a rule expressing that a process created by executing a file the same permissions as the owner of the file. For simplicity, assume that when a process created by executing a file tries to perform an action, the system determines whether canAccess(FileID, Permission, Resource) is granted by the system. Under this assumption, we can use the following rule:

$$\text{canAccess}(\text{FileID}, \text{Permission}, \text{Resource}) \\ \leftarrow \text{canAccess}(\text{Owner}, \text{Permission}, \text{Resource}), \text{file}(\text{FileID}, \text{R}, \text{W}, \text{X}, \text{Owner})$$

Something is very wrong with this, however, if writing to a file does not change the owner of the file. Explain how the access control system given by the rules and facts stated in this problem (including all parts above) allows carol to execute any sequence of commands that an administrator is allowed to execute. Assume rules as in part (b) so that every member of the readlist of a file can read it, and similarly for write and execute.

Aside: A similar vulnerability in certain Windows files was discovered by researchers from Princeton, using a translation of Windows access control into rules like the ones given in this problem.

5. (*18 points*) ..... S-BGP

The Border Gateway Protocol (BGP) is used to maintain routing tables across different Internet service providers, referred to as Autonomous Systems (AS). This was described in lecture.

Here's a brief review of BGP, using the map of the United States above: the New York AS may announce to its neighbors Chicago and Washington that it will receive traffic for Brooklyn, Bronx, and Manhattan (burroughs of New York). Once this announcement is received in Chicago, for example, Chicago will send a routing announcement to Minneapolis, Denver, and Washington that Chicago has a path of length 1 to New York. Washington will ignore this information, since it does not provide a better path than it already has. However, Denver will announce its path of length 2 to San Francisco, and so on. Each node forwards its shortest path to its neighbors in each round, eventually giving each node its shortest path to New York.

BGP itself is vulnerable to various forms of attack. For example, it is easy for a malicious node to announce a nonexistant shorter path to its neighbors.

The Secure Border Gateway Protocol (S-BGP) is intended to provide security against malicious use of the BGP protocol. To run S-BGP, each node is assigned a private signing key, and each node has the public verification of all the other nodes. Assume that S-BGP uses signatures in two ways:

- Each AS, such as New York, has signed data records stating the addresses it can announce, such as Brooklyn and Bronx.
- Whenever one node makes a routing announcement to another, such as Chicago announcing a path (to New York) to Denver, the sender signs the entire path it is announcing (including all signed data from previous nodes) and the neighbor it is sending the announcement to.

Explain which of the following attacks are still allowed by S-BGP and which are not:

(a) (*3 points*) Add nodes to path: Can a single malicious node (say Chicago, with Chicago's signing key) announce a path that contains a node that is not actually on its path to New York?

(b) (*3 points*) Remove nodes from path: Can a single malicious node (say Chicago, with key) announce a path that is missing a node needed to reach New York?

(c) (*3 points*) Collusion: Can two ASes claim to have an edge that does not exist, and add this to an announced path?

(d) (*3 points*) Data-plane attacks: Can an AS announce one path using BGP, but use another for actually routing packets that are sent to it?

(e) (*3 points*) Policy violation or resource exhaustion: Can an AS announce more than one "shortest path" to its neighbors?

(f) (*3 points*) What goes wrong if the node sending a route announcement only signs the path from that node to the destination, and does not sign data specifying the neighbor it is sending this announcement? Consider (a) and (b) above.

6. (*12 points*) ..... Bot detection

A bot is remotely controlled software, executing on a compromised host. A botnet is a network of bots and a controller that controls their operation. Most bots are highly programmable, allowing the bot controller to send programs that are executed by bots. Bot detection and remediation can be carried out on a network by examining network traffic, or on a host by trying to identify software that is acting as a bot.

(a) (*3 points*) Bots are widely used for relaying email spam. Describe a network defense that detects bots used for spam.

(b) (*3 points*) Bots have been used for launching distributed denial of service (DDoS) attacks. Describe a network defense that detects bots carrying out a DDoS attack. Use some characteristic of the way DDoS attacks are usually done *other* than measuring the amount of network traffic coming from a host on the network.

(c) (*3 points*) Explain why the Confinker binary contains a copy of the botmaster public key.

(d) (*3 points*) One possible way to do host-based bot detection is to compare contents of network packets that might be commands from the controller with system calls on the host. Explain how this idea might help you detect a bot executing a port redirect command (i.e. receive input on one port and send it back out on another).

7. (*14 points*) ..... CSRF defenses

(a) (*3 points*) In class we discussed Cross Site Request Forgery (CSRF) attacks against web sites that rely solely on cookies for session management. Briefly explain what a CSRF attack is.

(b) (*4 points*) A common CSRF defense is to place a token in the DOM of every page (e.g. as a hidden form element) in addition to the cookie. An HTTP request is accepted by the server only if it contains both a valid HTTP cookie header and a valid token in the POST parameters. Why does this prevent the attack from part (a)?

(c) (*4 points*) One approach to choosing a CSRF token is to choose one at random. Suppose a web site chooses the token as a fresh random string for every HTTP response. The server checks that this random string is present in the next HTTP request for that session. Does this prevent CSRF attacks? If so, explain why. If not, describe an attack.

(d) (*3 points*) Another approach is to choose the token as a *fixed* random string chosen by the server. That is, the same random string is used as the CSRF token in all HTTP responses from the server. Does this prevent CSRF attacks? If so, explain why. If not, describe an attack.