CS 155: Spring 2008
June 2008

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 a laptop 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)

☐ **SENIOR?**

_____

(Print your name, legibly!*)*

Prob	# 1	# 2	# 3	# 4	# 5	# 6	# 7	Total
Score								
Max	18	13	13	14	14	12	16	100

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

(a) (*3 points*) Basic buffer overflow attacks use the fact that the return address is at a higher memory address than the local variables (buffer). Explain how to carry out a buffer overflow attack if the stack layout is reversed. In the reversed layout, the stack grows from lower-numbered memory locations to higher ones, and the return address is at a lower memory address than the local variables allocated in the same stack activation record.

(b) (*3 points*) The same origin policy (SOP) for DOM access is based on the triple (protocol, host, port). Suppose SOP did not include protocol (i.e. SOP was defined using only host and port — as was the case in Safari until Safari 3.0). What goes wrong? For example, explain how a network attacker could steal gmail secure cookies (i.e. cookies sent only over HTTPS). Note that reading document.cookie in an HTTP context does not reveal secure cookies.

(c) WPAD is a protocol used by IE to automatically configure the browser's HTTP and HTTPS proxy settings. Before fetching its first page, IE will use DNS to locate a WPAD file, and if one is found, will use its contents to configure IE's proxy settings. If the network name for a computer is pc.cs.stanford.edu the WPAD protocol iteratively looks for wpad files at the following locations:

http : //wpad.cs.stanford.edu/wpad.dat
http : //wpad.stanford.edu/wpad.dat
http : //wpad.edu/wpad.dat (prior to 2005)

i. (*4 points*) Explain what capabilities were inadvertently given to the owner of the domain wpad.edu as a result of this protocol. Explain how personal information can be exposed as a result of this issue.

ii. (*2 points*) Are pages served over SSL protected from the problem you described? If so, explain why; if not, explain why not.

(d) (*6 points*) A stateless packet-filter firewall decides whether to allow a packet to traverse the firewall based on the TCP/IP header of the packet, without regard to past traffic through the firewall. Assume a stateless packet-filter firewall is installed between an enterprise network and the external Internet, for the purpose of protecting users on the enterprise network.

Circle the following attacks that can be detected and mitigated (to a significant degree) by the firewall:

i. Port sweep
ii. Syn flooding
iii. DNS cache poisoning
iv. a Phishing attack in which users are asked to visit a known bad web site
v. viruses in incoming email addressed to enterprise users
vi. DNS rebinding

2. (*13 points*) ..... Detecting executable tampering

Media players enforcing content protection rules need to ensure that their executable image on disk has not been modified by the user (otherwise, one could bypass content protection by disabling the content protection component). These mechanisms are intended to defend against attackers who modify the executable instructions.

(a) (*2 points*) A simple method for a program to detect tampering with its executable is follows: at startup the program hashes (using SHA-1) the executable image loaded in memory and compares the result with a pre-computed hash value (say, stored in the executable header). The program exits in case of mismatch. Explain how an attacker could defeat this mechanism with a single word change to the executable image.

(b) (*7 points*) Many proposals try to improve on the method outlined in part (a) by relying on obfuscation and repeated hashing. Let us examine a generic attack on this approach. Modern processors have an *instruction cache* used to cache memory pages containing code and a *data cache* used to cache memory pages containing data (a page that contains both code and data may be cached twice). When the processor wants to load a page into either cache, it first translates the page's virtual address into a physical address. This translation is done using the TLB. If the TLB does not contain an entry for the required address (a TLB miss) then an exception is triggered requesting the operating system to populate the TLB with an appropriate entry.

On the UltraSparc a different exception is signaled depending on whether the page is to be loaded into the data cache or into the instruction cache. In other words, the OS can tell whether the page is being accessed as data or as code.

Design an OS memory manager for the UltraSparc that defeats any tamper detection mechanism based on hashing segments of the executable image in memory. Explain exactly how your memory manager responds to TLB misses.
Hint: You are allowed to keep two copies of the media player application in memory.

(C) (*4 points*) Suppose you are the media player developer. Can you think of a way to defeat this attack?

3. (*13 points*) ..... Threat models

This question asks you to compare various threat models. A threat model defines a set of actions that an adversary may use to attack a system.

In network security, protocols such as SSL/TLS are designed to be secure against a *network attacker* who can intercept, block, and insert messages into the network between the client and server, but cannot guess randomly chosen cryptographic keys. We assume the network attacker can intercept every network packet because we want network security mechanisms to protect against an attacker who manages to do this.

In web security, many mechanisms are designed to be secure against a *web attacker* who can set up and control any number of web servers. It is assumed that the web attacker can buy certificates for the sites the attacker owns (but not forge certificates to pretend to be someone else), and can operate web browsers and other programs if needed. We assume that the user (potential victim) visits any web site set up by the web attacker as part of an attack, including any designated HTTP and HTTPS site.

(a) (*3 points*) In studying web security mechanisms, such as the *same-origin policy,* why is it part of the threat model to assume that the user (potential victim) visits a web site controlled by the attacker? In other words, why is the threat model above better for studying web security than a weaker model without this assumption?

(b) (*3 points*) Explain briefly why a web attacker (defined above) has sufficient resources to mount a cross-site request forgery (CSRF or XSRF) attack against a victim site that is vulnerable to CSRF.

(c) (*3 points*) What attack action is assumed possible for the web attacker but is not part of the network attacker threat model?

(d) (*4 points*) *iGoogle* allows users to build their own custom web pages and install Google gadgets (content from another source) on them. Suppose we want to study how well a user is protected from iGoogle pages that may contain malicious gadgets. Using the web attacker threat modeal as a starting point, what additional actions would you add and/or remove to define a *gadget attacker* threat model? Explain.

4. (*14 points*) ..... Web authentication

PwdHash is a web browser extension that that transparently produces a different password for each site. When a user types in a plaintext password like rover, destined for a site like wellsfargo.com, the browser sends a series of characters determined by h(rover, wellsfargo.com), where h is a cryptographic hash function. Assume that the hash function is publicly known, since anyone can download PwdHash and run it. However, it is not feasible to compute x from h(x), or find collisions. When a customer uses PwdHash, the bank server sees the hashed password as the user's password.

(a) (*3 points*) Suppose a phisher sets up a site that looks like Citibank, at a domain different from the real Citibank login site. How will PwdHash protect a customer entering a password to the phshing site?

(b) (*2 points*) How does PwdHash protect a bank customer against a web attacker (as defined in the previous problem)? Explain.

(c) (*3 points*) Does PwdHash protect a bank customer against a network attacker (as defined in the previous problem) who acts as a man-in-the-middle? Explain.

(d) (*3 points*) Does PwdHash protect a bank customer against a web attacker (as defined in the previous problem) who can do dictionary attacks? Explain.

(e) (*3 points*) Some people use the same password at high-security sites, such as their bank, and low-security sites, such as a high-school reunion web site. Will PwdHash help protect the bank accounts of such people? Explain.

5. (*14 points*) ..... Privacy and MIX nets

(a) (*6 points*) In class we described a theoretical MIX net where MIX nodes function as stateless routers (2nd attempt in lecture 15): MIX nodes maintain no state information beyond their secret keys. Explain how to support bi-directional communication in this network.
Hint: sender includes a "response onion" in its packet to the server. Explain what goes into the response onion and how it used to send data back from the server to the sender.

(b) Let us design a system providing (minimal) mutual anonymity. Imagine user A wishes to download an mp3 file with title T. Server B has the file. Our goal is to design a system that enables A to download the file from B so that no single network entity knows the identity of both A and B. At your disposal are thousands of stateless MIX nodes as well as a directory service.

Note: you need not hide the fact that someone downloaded a copy of title T. In particular, the directory service will know that a download of T took place. We are only protecting the identity of the end points.

i. (*4 points*) Explain how B can register anonymously at the directory service (letting the service know that it has title T).
Hint: try using a return onion as in part (a) as part of the registration process.

ii. (*4 points*) Explain how A uses the directory service to obtain the file from B. The directory service does not take part in the file transfer.

6. (*12 points*) ..... Mobility

In Mobile IPv6, a mobile node can send a message called a *binding update* to the other endpoint of a network connection, which is called the *corresponding node.* The binding update messages gives the corresponding node a new IP address for the mobile node, so that the two nodes may communicate directly using the new address.

(a) (*4 points*) If no authentication is required for binding updates, how can an attacker operating a computer with no special control over the network eavesdrop on a connection between two nodes running mobile IPv6?

(b) (*4 points*) If no authentication is required for binding updates, how can an attacker operating a computer with no special control over the network launch a denial of service attack on a chosen site, such as cnn.com? (*Hint:* The attacker can use a browser to download video from youtube.com or high quality streaming video sites.)

(c) (*4 points*) Suppose that in order to authenticate binding updates, the mobile node signs a message that has two parts: its current address and its new address. Assume that the corresponding node has the verification key to verify the signature. Could a network attacker who has eavesdropped on past communication use binding updates to disconnect a conversation?

7. (*16 points*) ..... Malware detection and evasion

BotHunter implements a method for botnet detection which entails correlating alarms from different network intrusion detection systems (NIDS). Assume that BotHunter is deployed at the gateway to an enterprise network. Snort is a standard NIDS.

BotHunter uses a model of a bot infection sequence which entails certain combinations of the following events: inbound port scan (E1), inbound exploit (E2), internal-to-external binary download (E3), internal-to-external C&C communications (E4), and outbound port scan (E5). A port scan detection engine identifies E1 and E5, Snort signatures and a payload-anomaly detection engine identify E2, and Snort signatures detect E3 and E4. Different combinations of alarms — which occur within a particular time and which all reference the same internal host (IP) — can satisfy the threshold for declaring a bot infection. In particular, E2 followed by E3, E4, or E5 results in an alarm as does any two of {E3, E4, E5}.

For each of the following ways that a bot writer might try to avoid detection by BotHunter, explain why method can or cannot be effective:

(a) (*4 points*) *Tactic #1: Threshold Manipulation.* Malicious activities can be spread out in time so as to fly beneath the radar.

(b) (*4 points*) *Tactic #2: Perturb Network Flows.* Use compromised nodes inside or outside the enterprise network to modify the sender and receiver of network traffic.

(c) (*4 points*) *Tactic #3: Encrypt Traffic.* Use SSL/TLS to encrypt command and control network communication between the installed bot and the bot master.

(d) (*4 points*) *Tactic #4: Launder Bot Internal Dataflow.* An installed bot may defeat detection on the host by "laundering" command data received over the network. There are several techniques for doing so, such as writing commands to the file system, and then reading them back later.