CS 155: Spring 2011
June 3, 2011

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	Total
Score							
Max	15	16	12	16	12	14	85

1. (*15 points*) ..... Short Answer

(a) (*3 points*) What are covert channels and how can they be used to leak information between isolated VMs running on a single machine.

(b) (*3 points*) Which of the following technologies can help in defending against a heap-based control hijacking attack:

Stackguard, LibSafe, ASLR, DEP, StackShield

Briefly explain how the technologies you chose help.

(c) (*3 points*) Suppose an organization wants to block employees from sending HTTP requests to external web sites whenever the content of the request matches a certain regular expression. This is not difficult to do for HTTP traffic using a web proxy, but how would the organization enforce this policy for HTTPS traffic?

(d) (*3 points*) What capability is enabled by the TPM's sealed storage mechanism? Describe one way in which physical access to the insides of the machine can defeat this mechanism.

(e) (*3 points*) Consider the following C code:

```
if (canAccess( getFilename() )) {
   fp = fopen(getFilename(), "w");
   do-something(fp);
}
```

where `canAccess(file)` returns false if the current context is not allowed to access file `file`. You may assume that this code is running in a single threaded environment so that there are no concurrency issues.

Can this code result in an access control violation? If you answer yes, give an example function `getFilename()` that results in a call to `fopen(filename,"w")` even though `canAccess(filename)` returns false. Function `getFilename()` takes no input and does not do I/O.

2. (*16 points*) ..... Unix access control and Android

In Unix, every process has a real user id (*ruid*), an effective user id (*euid*), and a saved user id (*suid*). Processes with an euid of 0 have special root privileges.

(a) (*1 point*) If a process with user id n forks to create another process, what user id does the new process have? (*Hint:* it's the same answer for euid, ruid, and suid.)

(b) (*4 points*) If a process with euid n makes a setuid system call, what possible euids can the process run with after the call? Consider the following subcases and write your answers in the underlined areas.

i. Before: $euid = n \neq 0$, saved user id $suid = m$. After:_____

ii. Before: $n = 0$ After:_____

(c) (*3 points*) In qmail, most modules run under separate user ids. Similarly, each Android application runs in a separate process using a separate user id. From a security standpoint, what is the advantage of assigning separate uids instead of using the same uid for all?

(d) (*2 points*) Why should the separate uids be *non*-zero?

(e) (*2 points*) The Android zygote process that creates new processes runs as root. After forking to create a new process, setuid is normally called. Explain why it is important to call setuid? What security purpose does this serve?

(f) (*2 points*) One stage of the *droiddream* malware takes advantage of the fact that Android has a limit RLIMIT_NPROC on the maximum number of process uids. The zygote process uses the following code to call setuid:

```
err = setuid(uid);
if (err < 0) {
   LOGW("cannot setuid(%d) errno: %d", uid, errno);
}
```

Assume the call to setuid fails when the call tries to exceed the RLIMIT_NPROC limit. How does this code leave the Android device vulnerable to an attack that the programmers intended to prevent?

(g) (*2 points*) Explain why this patched code addresses the problem.

```
err = setuid(uid);
         if (err < 0) {
            LOGE("cannot setuid(%d): %s", uid, strerror(errno));
            dvmAbort();
        }
```

3. (*12 points*) ..... Phishing scams

Phishing web sites create a copy of a legitimate web site (e.g. a bank) and present to the user an authentic-looking login page. When the user enters a login credential (username/password) the data is recorded and later collected by the Phisher. The Phisher can drive traffic to the phishing page using a number of techniques, including spam email and ads.

(a) (*2 points*) Login pages are typically served over HTTPS using the site's certificate. How can phishers who do not want to pay for a certificate get around this?

(b) (*3 points*) Some phishers copy the login page as is. That is, they copy the login page, but leave the embedded image links pointing to the real banking site. Explain how a banking site can use this fact to detect phishing sites.

(c) (*4 points*) Some phishers may make a complete copy of the phished site, duplicating all images and scripts on the target page and store them on the phishing server. They copy Javascript on the phished page, but without altering the script. Explain how a bank can use this fact to not only detect phishing sites, but also detect which of its customers fell victim to the phishing scam. The bank can then move to block those customers' accounts.

(d) (*3 points*) Suppose the banking login page has an XSS vulnerability. Explain how this can make the phisher's life easier.

4. (*16 points*) ..... Are two browsers better than one?

Some security experts advise users to use more than one browser: one for surfing the wild web and another for visiting "sensitive" web sites such as online banking web sites. For example, you could use Chrome to read blogs and Firefox for banking. The advice raises the question of whether two browsers are better than one, and if so, how.

For the purposes of this question, assume that each browser uses a specific directory to store temporary files and cookies on the local host. Also assume that the user never uses the sensitive browser to visit non-sensitive sites and never uses the "wild-web" browser to visit sensitive sites.

(a) (*2 points*) Briefly define reflected cross-site scripting (XSS). If you need example sites to write your definition, assume a blog is controlled by an attacker and a bank site is an honest victim.

(b) (*2 points*) Briefly define cross-site request forgery (CSRF). If you need example sites to write your definition, assume a blog is controlled by an attacker and a bank site is an honest victim.

(c) (*2 points*) Briefly define click-jacking. If you need example sites to write your definition, assume a blog is controlled by an attacker and a bank site is an honest victim.

(d) (*3 points*) Which one of the three attacks listed above that can be *directly prevented* when two browsers are used as recommended. Describe why in 2 sentences. Assume that sensitive-sites do not launch attacks for the rest of this question.

(e) (*4 points*) A browser vendor wants to make the security advantages of two browsers available in a single browser. They decide to create two storage directories for their browser, called "sensitive" and "non-sensitive". The browser stores a list of sensitive sites. If the location bar of a browser tab names a sensitive site, all temporary files and cookies for that tab are stored in the sensitive directory, where they are only accessible to other tabs whose location bars name a sensitive site. If a user opens a tab, logs into bank.com, and then opens another tab to visit attacker.com that contains an `iframe` for bank.com, the requests issued for the `iframe` will *not* contain the bank.com user credentials. Explain an attack that succeeds against this two-in-one browser implementation but would fail if two actual separate browsers are used. (*Hint:* Malicious JavaScript can open new tabs.)

(f) (*3 points*) Can you think of a *simple* browser mechanism that can be implemented to thwart attacks like the one you describe? Concisely describe the mechanism in less than 4 sentences.

5. (*12 points*) ..... Stealing traffic

The IP protocol supports fragmentation where a packet can be fragmented and re-assembled when it reaches the destination. When a packet is fragmented it is assigned a 16-bit packet ID and then each fragment is identified by its offset within the original packet. The fragments travel to the destination as separate packets. At the destination they are grouped by their packet ID and assembled into a complete packet using the packet offset of each fragment. Every fragment contains a one bit field called "more fragments" which is set to true if this is an intermediate fragment and set to false if this is the last fragment in the packet.

(a) (*4 points*) In class we mentioned that when fragments with overlapping segments are re-assembled at the destination, the results can vary from OS to OS. Give an example where this can cause a problem for a network-based packet filtering engine (an engine that blocks packets containing certain keywords). How should a filtering engine handle overlapping fragments to ensure that its filtering policy is not violated?

(b) (*5 points*) Suppose two machines are behind a NAT and one machine (the attacker) wishes to eavesdrop on traffic intended for the other machine (the victim). Suppose the NAT re-assembles all fragments before forwarding the full packets to the endhosts.

Now, consider a sequence of fragments that arrives at the NAT, all with the same packet ID and intended for the victim. Suppose the attacker knows this packet ID and wishes to have the re-assembled packet sent to him. Explain how the attacker can send two well-timed fragments to the NAT so that he receives all but the first fragment of the re-assembled packet. You may assume that fragments arrive at the NAT in increasing packet offset order.

Hint: Try to force the NAT to produce a packet where the IP header contains the attacker's IP address as the destination IP.

(c) (*3 points*) Suppose the data source in part (2) assigns packet IDs using a counter. That is, the packet ID for an outgoing packet is set to the current value of a counter and then the counter is incremented by one. Explain how the attacker can use this to learn the packet ID assigned to packets sent to the victim. For simplicity you may assume that the attacker knows the precise time at which the source sends packets to the victim.

6. (*14 points*) ..... The .bank Top Level DNS Domain

The banking industry is discussing a new top-level domain. Let's call this .bank and assume that only banks are supposed to be assigned domain names in this top-level domain (TLD). For example, Wells Fargo Bank might be assigned wellsfargo.bank and use this in place of its current wellsfargo.com. Assume that banks must be verified as legitimate in order to obtain certificates and that the organization which runs the .bank TLD verifies that all of its entries belong to legitimate banks.

(a) (*2 points*) Suppose an attacker poisons the cache of a local DNS resolver, inserting the name server (NS) entry for wellsfargo.com. What domain(s) and subdomains does the attacker now control, and for which end-users?

(b) (*2 points*) What improvements in the security of domain-name lookups will the new TLD accomplish if ordinary DNS is used for .bank,?

(c) (*2 points*) Suppose the new .bank TLD uses DNSSEC instead of DNS. Can a network attacker, who can read and write network traffic in and out of a local resolver, insert DNSSEC entries into the resolver's cache which are not part of the legitimate DNS record? You must briefly describe why or why not to get credit. (Ignore issues related to NSEC3 in this part of the problem.)

(d) (*2 points*) Why might the banking industry consider using DNSSEC with *opt-out* for .bank? Explain briefly, using one or more of the reasons that opt-out was designed originally.

(e) (*4 points*) Suppose DNSSEC with NSEC3 opt-out is used for .bank. Describe how attackers can "create" a child domain of .bank visible to Web users that points to an attack site posing as a legitimate bank. Specifically, describe how attackers can insert a name server (NS) entry for a child domain of the .bank domain into the cache of a local DNSSEC resolver. Assume the network attacker can read and write network traffic in and out of the local resolver.

(f) (*2 points*) Assume that the attacker in (part **6**e) succeeds and is able to insert a name server (NS) entry for benignlyNamedEvil.bank into the local resolver. What defenses-in-depth might prevent Web users who are savvy about the usual online banking security measures from suffering harm, even if they try to create an account at benignlyNamedEvil.bank?