CS 155: Spring 2010
June 2010

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	19	8	13	23	22	15	100

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

(a) (*3 points*) Are one time password authentication systems vulnerable to eavesdropping attacks? If so, explain why. If not, explain why not.

(b) (*2 points*) Briefly explain how you would use access control lists or capabilities to enforce the principle of least privilege.

(c) (*2 points*) You run a web vulnerability scanner against your web site and the tool finds no vulnerabilities in your site. Can you conclude that your site is secure?

(d) (*3 points*) How are client puzzles used to defend against Denial of Service attacks?

(e) (*3 points*) In a TPM-based file encryption system such as BitLocker explain what happens if the master boot record (MBR) which loads the operating system is infected with a root kit. Explain in detail why the Operating System will not boot.

(f) (*3 points*) Suppose a movie player only plays movies that contain a valid MAC of the movie and ignores all other movies (the player contains the MAC verification key needed to verify the MAC). Can you fuzz this movie player? If so explain how. If not explain why not.

(g) (*3 points*) Suppose a movie player only plays movies that contain a valid signature of the movie and ignores all other movies (the player contains the signature verification key). Can you fuzz this movie player? If so explain how. If not explain why not.

2. (*8 points*) .... Threat models

Answer each of these questions in 1–2 sentences.

(a) (*2 points*) What is the difference between an active network attacker and an eavesdropper?

(b) (*2 points*) What is a dictionary attack and why is it often effective?

(c) (*2 points*) If you have an authentication system that is vulnerable to online dictionary attacks (and not vulnerable to offline dictionary attacks) what easy mechanism can you use to reduce this vulnerability?

(d) (*2 points*) In web security, why do we assume that the user visits a site set up by the web attacker?

3. (*13 points*) .... HTTPS vs. Signatures

Suppose a software company *xyz.com* sells a product P and wants to distribute a software update D. The company wants to ensure that its clients only install software updates published by the company. They decide to use the following approach:

> The company places D on its web server and designs the software P to periodically check this server for updates over HTTPS.

(a) (*1 point*) Explain what can go wrong if P downloads the software update over HTTP.

(b) (*2 points*) The company decides to buy a public key certificate for its web server from a reputable CA. Explain what checks P should apply to the server's certificate to defeat a network attacker. Is your design vulnerable to `ssl_strip`?

(c) (*3 points*) How would you design the program P and the web server so that the update is secure against a network attacker, but there is no need to buy a certificate from a CA? Your design should use an HTTPS web server as before.

(d) (*3 points*) The company worries that an attacker will break in and steal the web server's secret key. How would you design the software update system so that it can recover from such an event?

Later on engineers at the company proposed the following very different design:

> Sign D using an *xyz* private key to obtain a signature s and then distribute (s, D) in the clear to all customers. The corresponding public key is embedded in the P.

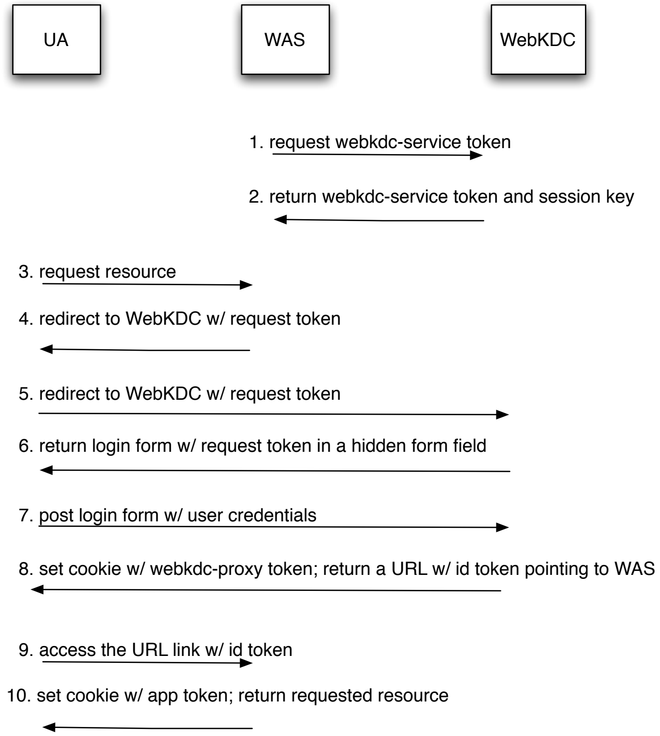
Let's compare the signature vs. HTTPS designs:

- (*2 points*) If we want to distribute the patch D using a content distribution network like BitTorent, which of the two designs should we use? Explain why.

- (*2 points*) How much computing time does *xyz.com* spend doing crypto calculations in each of the designs? Which is better?

4. (*23 points*) .... Web single-sign-on protocols

In his guest lecture, Dirk Balfanz from Google talked about the OAuth protocol, and the use of protocols that allow users of one site to authenticate using a password from another. A related protocol is WebAuth, used at Stanford. To check your grades on Axess, for example, you can navigate your browser to `axess.stanford.edu`. However, to log on, Axess redirects your browser to another site, `weblogin.stanford.edu`, that asks you for your password. After you enter your Stanford user name and password, you are sent back to Axess. Here's a diagram of how the basic WebAuth protocol works, described below. Your goal is to find a problem with the protocol as described here.


UA
WAS
WebKDC
1. request webkdc-service token
2. return webkdc-service token and session key
3. request resource
4. redirect to WebKDC w/ request token
5. redirect to WebKDC w/ request token
6. return login form w/ request token in a hidden form field
7. post login form w/ user credentials
8. set cookie w/ webkdc-proxy token; return a URL w/ id token pointing to WAS
9. access the URL link w/ id token
10. set cookie w/ app token; return requested resource

The WebAuth protocol involves three entities:

(a) User-Agent (UA), the user's browser,

(b) WebAuth-enabled Application Server (WAS), a web server that is integrated with WebAuth, such as Axess or CourseWare, and

(c) WebKDC, the web login server (such as `weblogin.stanford.edu`).

WebAuth uses so-called "secure" cookies to store its state and HTTPS to transmit its messages, protecting the protocol from network attackers. In general, the WAS and WebKDC will be hosted on different domains.

The figure shows two parts of the protocol:

- *WAS Initialization (Steps 1–2).* At startup, the WAS (Axess) authenticates itself to the WebKDC (weblogin) using a private key, and receives something called a *webkdc-service token* and a session key. This occurs before anyone tries to log in to Axess.

- *Login (Steps 3–10).* When the user wishes to authenticate to the WAS (Axess), the WAS creates a *request token* and redirects the UA (Stanford student's browser) to the WebKDC (weblogin), passing the token in the URL. The WebKDC authenticates the user (via a user name and password), stores a cookie in the UA, and redirects the user back to the WAS, passing an *id token* identifying the user in the URL. The WAS receives the *id token* as a query parameter in the WAS URL and verifies cryptographic properties of the token to authenticate the user. Finally, the WAS sets a cookie on the UA to establish the authenticated user session.

Questions

(a) (*2 points*) In step 8, the WebKDC (weblogin) sets a cookie on the user's computer. This allows WebAuth to skip steps 6 and 7 if the same user wants to visit another site protected by WebAuth, during the lifetime of this cookie. Does this convenience feature have any security advantages or disadvantages?

(b) (*2 points*) Why does WebAuth use a token (special value computed using cryptography) in a URL in step 9, instead of letting the WebKDC put the token in a cookie that the WAS can read?

(c) (*1 point*) What is a "secure" cookie?

(d) (*2 points*) Describe a potential or hypothetical attack that might be possible if this protocol did not use "secure" cookies?

(e) (*1 point*) What is the difference between a "secure" cookie and an HttpOnly cookie?

(f) (*2 points*) Would the security be improved if the cookies were also HttpOnly?

(g) (*2 points*) What is CSRF? Explain in a few short sentences.

(h) (*2 points*) List two kinds of CSRF.

(i) (*2 points*) How does the WAS (Axess) know that the UA that follows the link in step 9 is the same as the UA that made the initial request in step 3?

(j) (*2 points*) If you have a link and you want someone else to click on it, what sort of tricks could you use, on the web today? (name two)

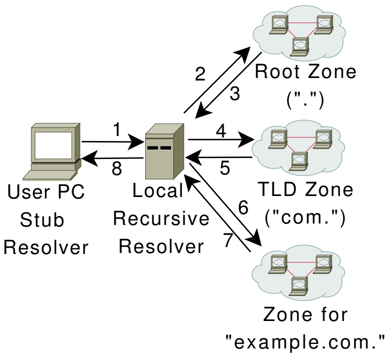
(k) (*2 points*) If you were able to trick another user into clicking on the link you received in step 9 (after following this protocol to log in to Axess, for example), what will happen when the user clicks on it?

(l) (*1 point*) Give the standard name for this form of attack.

(m) (*2 points*) How can you fix the protocol?

5. (*22 points*) .... Domain Name Security

The figure below shows how a DNS query "`www.example.com A?`" is resolved to IP address "`1.2.3.4`". The middle column of the table shows the responses under conventional DNS and the right column of the table shows the additional sets of Resource Records (RRs) that are sent under DNSSEC. Authoritative RRSets are in plain text and glue RRSets are in *italic*. The stub resolver is not expected to handle DNSSEC RRs, so none are sent to it. Starting with the DNSKEY of the root zone as the trust anchor, DNSSEC Reply 3 provides the DS to attest to the DNSKEYs of "com.", Reply 5 adds the DNSKEY of "com." and the DS to attest to the DNSKEY "example.com.", which is provided by Reply 7.



Reply	RRSets in DNS Reply	RRSets added by DNSSEC
3	*"com. NS a.gtld.net."* *"a.gtld.net. A 192.5.6.30"*	"com. DS" "RRSIG(DS) by ."
5	*"example.com. NS a.iana.net."* *"a.iana.net. A 192.0.34.43"*	"com. DNSKEY" "RRSIG(DNSKEY) by com." "example.com. DS" "RRSIG(DS) by com."
7	"www.example.com. A 1.2.3.4"	"example.com. DNSKEY" "RRSIG(DNSKEY) by example.com." "RRSIG(A) by example.com."
8	"www.example.com. A 1.2.3.4"	

Questions

(a) (*2 points*) When HTTPS (or SSL) is used, a server returns a certificate that allows the client to identify the server. How does HTTPS security depend on DNS?

(b) (*2 points*) How does the browser Same-Origin Policy (SOP) depend on DNS?

(c) (*2 points*) Do conventional packet-filter firewall policies depend on DNS? Explain.

(d) (*2 points*) Why does DNSSEC provide signed RRs for DNS "A" records? (Include the function of "A" records in your answer.)

(e) (*2 points*) Why does DNSSEC provide signed RRs for DNS "NS" records? (Include the function of "NS" records in your answer.)

(f) (*2 points*) Why does DNSSEC not provide signed RRs for DNS "glue" records? (Include the function of "glue" records in your answer.)

(g) (*2 points*) Why does DNSSEC provide "*authenticated* denial of existence"? What kind of attack is this intended to prevent?

(h) (*2 points*) Suppose that `ibm.com` deploys DNSSEC, but the DNS servers for `.com` do not. What benefits does this provide to users within IBM?

(i) (*2 points*) Again assuming that `ibm.com` deploys DNSSEC but the DNS servers for `.com` do not, are there security problems that *would be* solved if `.com` and the root zone "`.`" also deployed DNSSEC, but *are not* mitigated by deploying DNSSEC only within IBM? Explain.

(j) (*4 points*) DNSSEC also provides Sender Policy Framework (SPF) records. SPF allows administrators to specify which IP addresses are allowed to send e-mail from a given domain by creating a specific DNS SPF record in the public DNS. Mail exchangers then use the DNS to check that mail from a given domain is being sent from an IP address sanctioned by that domain's administrators. For example, a mail exchanger will drop an email message claiming to be from `gmail.com` but coming from an IP address which SPF says is not a `gmail.com` IP address.

i. (*2 points*) Briefly explain how you would implement SPF records in DNSSEC. Describe explicitly what information is stored in an SPF record and how it is retrieved.

ii. (*2 points*) Suppose a spammer who owns a single IP address wishes to send mail that appears to be coming from `gmail.com`. The spammer's IP address is not a valid IP address for `gmail.com`. The spammer can try to defeat SPF by sending packets to the mail server with a spoofed source IP address so that the mail appears to come from a valid `gmail.com` IP address. The message is sent using the standard mail protocol called SMTP which runs over TCP. The spammer will not receive any response from the server since the server will be sending its responses to the spoofed source IP address. Fortunately for the spammer an SMTP send mail transaction can be completed without seeing responses from the server. In other words, SMTP does not provide any defense against source IP address spoofing.
What is to prevent the spammer from defeating SPF this way? Think of lower levels of the networking stack.

6. (*15 points*) .... Control hijacking attacks

Stackshield is a stack overflow defense similar to stackguard and works as follows: When a function begins executing it makes a copy of the return address located in its stack frame to a shadow stack. When the function is about to return (i.e. just before calling `ret`) the program checks that the return address on the shadow stack is equal to the return address in its stack frame and terminates the program if not. Like Stackguard, Stackshield can add its checking code during compile time. Assume the shadow stack is located in some fixed location on the heap known to the attacker.

(a) (*6 points*) Give sample C code and a stack buffer overflow that defeats Stackguard but not Stackshield. Use the back of this page for extra space.

(b) (*6 points*) Give sample C code and a stack buffer overflow that defeats Stackshield but not Stackguard. Use the back of this page for extra space.

(c) (*3 points*) How would you strengthen stackshield to defend against your attack from part (b)?