Raluca Ada Popa
Spring 2018

CS 161
Computer Security

Midterm 2

PRINT your name: ____________________, ____________________
(last) (first)

I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that academic misconduct will be reported to the Center for Student Conduct.

SIGN your name: ____________________

PRINT your class account login: cs161-____________________ and SID: ____________________

Name of the person sitting to your left: ____________________

Name of the person sitting to your right: ____________________

You may consult one sheet of paper of notes. You may not consult other notes, textbooks, etc. Calculators, computers, and other electronic devices are not permitted.

If you think a question is ambiguous, please come up to the front of the exam room to the staff. If we agree that the question is ambiguous we will add clarifying assumptions to the central document projected in the exam rooms. Write your student ID on the top of every page.

You have 80 minutes. There are 7 questions, of varying credit (133 points total). The questions are of varying difficulty, so avoid spending too long on any one question.

Do not turn this page until your instructor tells you to do so.

Problem 1 ***True or False*** **(20 points)**

Answer True or False for each of the questions below. You do not need to explain your answers.

(a) True or False: Javascript running on `example.com/index.html` cannot manipulate webpages on `me.example.com`.

○ True ○ False

(b) True or False: If `example.com` loads Javascript from `me.example.com/evil.js`, then `me.example.com` can manipulate webpages on `example.com`.

○ True ○ False

(c) True or False: It is impossible for a cookie set on `example.com` to be accessed by Javascript running on `me.example.com`. (Assume `example.com` is not vulnerable to XSS.)

○ True ○ False

(d) True or False: A stateless packet filter can drop packets belonging to a connection whose first packet payload (the first byte) starts with the character "x".

○ True ○ False

(e) True or False: A DNS server that receives multiple UDP packets from the same client can be assured (with high probability) that these packets are in the order that they were sent. (Assume no attackers.)

○ True ○ False

(f) True or False: If an attacker guesses the IPs, ports, TCP sequence numbers and TLS sequence numbers in a TLS connection, then they can successfully inject traffic into the connection.

○ True ○ False

(g) True or False: If you join an unsecured network with an on-path eavesdropper, then the attacker could hijack all of your HTTP requests to `www.google.com`. (Assume all caches are empty.)

○ True ○ False

(h) True or False: Without dropping packets, an on-path attacker can still stop you from successfully creating a TLS connection with `https://google.com`. (Assume that `google.com`'s IP address is in your DNS cache.)

○ True ○ False

(i) True or False: A man-in-the-middle attacker can successfully impersonate `https://google.com` if they forge a DNS response for `google.com`.

○ True ○ False

(j) Say that we have two detectors. We combine them in series, by only triggering an alert if both detectors detect an attack. True or False: The new false positive rate is always less than or equal to the false positive rate of both detectors.

○ True ○ False

Problem 2 ***A/B Testing*** **(27 points)**

For each of the questions below, select all of the technologies which would work for the scenario described. **You may select multiple choices, or none of the choices.**

(a) We know the IP address of a host, and we want to lookup the MAC address. (Assume we are already established on the network.)

□ ARP □ DHCP

□ DNS □ HTTP

(b) Alyssa is about to open her laptop, connect to unsecured Starbucks WiFi, and load the URL `http://squigler.com/feed`. A man-in-the-middle on her local network wants to steal her Squigler cookies. (Assume that Squigler does not implement any defences. All caches are empty.)

□ ARP Spoofing □ A reflected XSS on `/feed`

□ UDP Spoofing □ A stored XSS on `/feed`

□ DHCP Spoofing □ A buffer overflow on `squigler.com`

□ TCP Spoofing □ A root CA compromise

(c) The user is visiting a website vulnerable to an XSS attack. Mark any technology that helps prevent the XSS attack or mitigate its effects.

□ Content-Security Policy □ HTTPS

□ VPN □ DNSSEC

□ Secure cookie flag □ Prepared statements

□ Escape user input □ HttpOnly cookie flag

(d) We want to create an intrusion detection system which will work on a new class of never-before-seen attacks.

□ Anomaly-based □ Specification-based

□ Signature-based □ Behavioral-based

(e) We want to detect any attempted accesses to the file `/etc/passwd`.

□ HIDS □ NIDS

(f) We want to detect any malware sent as email attachments.

□ HIDS □ NIDS

(g) Considering an on-path/man-in-the-middle attacker between a DNS resolver and a DNS server, we want to prevent a DNS spoofing attack in which the DNS resolver receives an incorrect IP address for a DNS lookup on a domain name to the DNS server. Select all technologies that can prevent the attacker from modifying the contents of the DNS reply from the DNS server.

□ TLS □ Firewall

□ TCP □ HIDS

□ UDP □ DNSSEC

(h) Consider an attacker stole the password database from a server and wants to reverse Alice's password P in particular. Assume Alice chose P by choosing one word at random from the English dictionary, and the attacker knows this. Assume that hashes are stored alongside passwords in the database. Select all ways the server can store Alice's password in the password file that would prevent an attacker from determining Alice's password.

□ Hashed password ($H(P)$)

□ Salted hash of P, with same salt per user

□ Salted hash of P, different salt per user

□ Hash of P using a slow hash (100 applications of SHA-256)

□ Encrypted P using AES-CBC

(i) As in the question above, but now assume Alice chose P to be a random string from a large set (2^{200} possibilities). Select all ways the server can store Alice's password in the password file that would prevent an attacker from determining Alice's password.

□ Hashed password ($H(P)$)

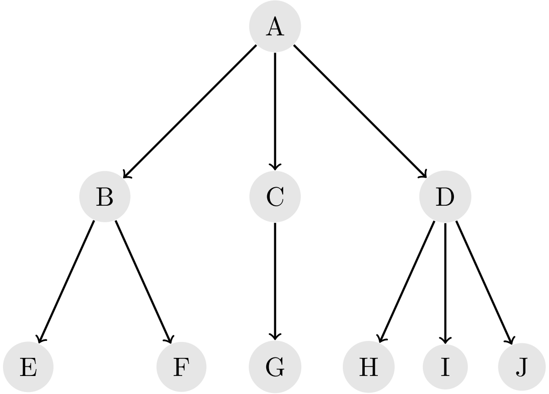
□ Salted hash of P, with same salt per user

□ Salted hash of P, different salt per user

□ Hash of P using a slow hash (100 applications of SHA-256)

□ Encrypted P using AES-CBC

Problem 3 ***DNS Tree*** **(20 points)**



This tree describes a name server hierarchy. The node `A` is a top level domain (like `.com`, `.edu`), and each child name server controls a subdomain of its parent. Each letter corresponds to both the nameserver and the name of the domain. For example, an uncached recursive query for `https://inst.E.B.A` would result in queries to NS A, then NS B, then NS E.

Assume this is the complete nameserver hierarchy for `A`. No other nameservers under `A` exist apart from the ones depicted.

(a) Which one of the following nameservers could theoretically provide the most number of additional records in its response?

○ B ○ D

○ C ○ Not enough information

(b) If we use the most basic version of DNS, in which a resolver accepts any response over any domain from any nameserver, and NS D is compromised, which domains are safe from cache poisoning? (Assume D is contacted during resolution.)

(c) What domains should D be allowed to give additional records for to protect against cache poisoning?

(d) Now suppose the nameserver hierarchy operates on DNSSEC. During resolution, the resolver sends a query for `inst.G.C.A` to nameserver C.

i. What public key should be included in C's response? Choose one.

○ inst ○ C

○ G ○ A

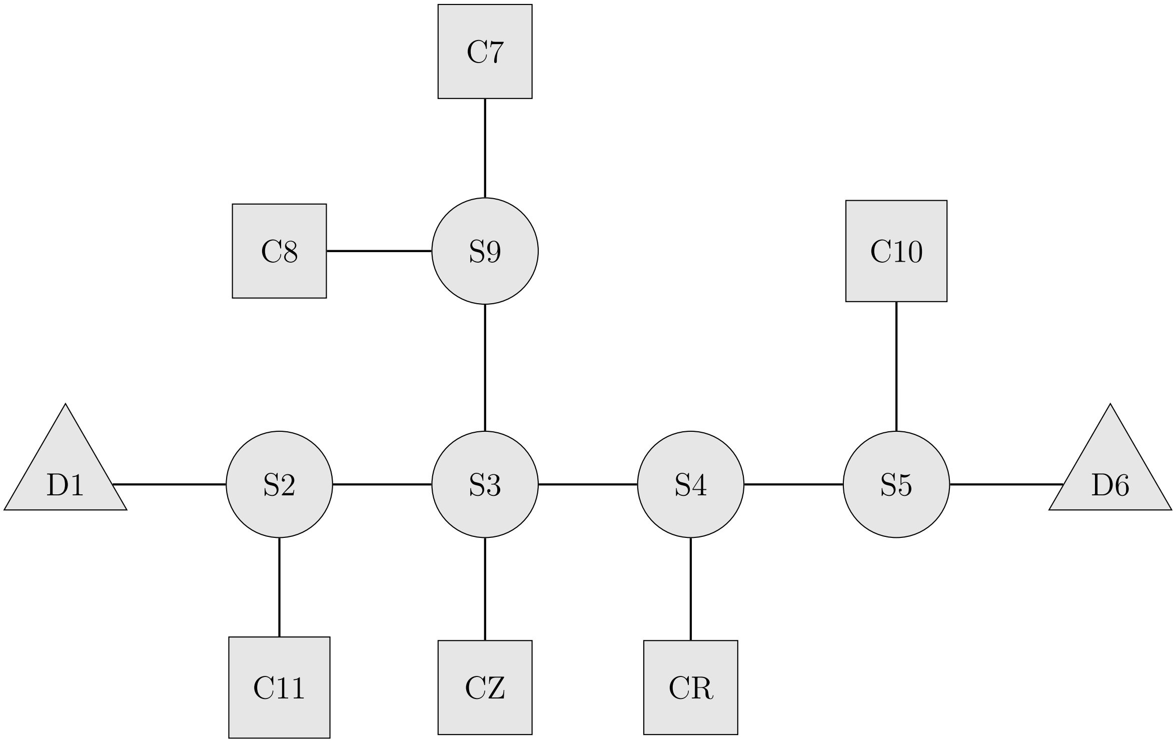
ii. If the root key is compromised, can the resolver trust NS C's response?

SID ____

○ Yes ○ No

Problem 4 ***Fun with L2*** **(20 points)**

Tux has gone undercover and wants to steal data from his enemy, **Rex**, who has plugged into the network with his computer labeled **CR**. Assume that the network is configured as such:



Note the tree-structure of the network.

Tux can choose **one** of the following options:

1. Taking control of some number of the DHCP servers (marked as triangles)
2. Taking control of some number of the switches (marked as circles)
3. Taking control of some number of the computers (marked as square)

Assume that if Tux has control of a switch, he has the power to drop, intercept, and modify packets. If Tux has control of a computer, he can send spoofed packets out to the network. However, the network is unpredictable and packets can arrive out of order, so you may not assume that the order of packets is tied to the number of links. For example, the packet from 6 may reach 11 before the packet from 1.

For each of the possible options, select the devices that Tux has to control to accomplish his mission. Tux wants to have to control the minimum number of devices. If there are multiple possible solutions with the same size, break ties by the solution **with the smallest numerical device**. For example, if both switch sets $\{2, 5, 9\}$ and $\{3, 4, 9\}$ are valid solutions, you should bubble $\{2, 5, 9\}$. Consider each option alone, namely, that the devices in that option are the only devices to be compromised.

(a) Suppose Tux wants to give a fake IP address to Rex, who is just joining the network. Tux wants this attack to work 100% of the time.

i. Option 1: DHCP Servers

□ D1 □ D6

ii. Option 2: Switches

□ S2 □ S3 □ S4 □ S5 □ S9

iii. Option 3: Computers

□ C7 □ C8 □ C10 □ C11

(b) Tux wants to give the fake IP address to Rex, but this time, Tux is okay with the attack working only part of the time.

i. Option 1: DHCP Servers

□ D1 □ D6

ii. Option 2: Switches

□ S2 □ S3 □ S4 □ S5 □ S9

iii. Option 3: Computers: Purposely omitted.

(c) Tux wants his computer (located at CZ) to be sent **all** of Rex's outgoing Layer 3 (IP) traffic (but doesn't care about Rex's Layer 2 traffic). He intends to do this by spoofing DHCP packets. Which of the four steps of the DHCP protocol should Tux aim to spoof?

(d) Tux wants the above attack to work 100% of the time. Which devices should he control? (Same rules as above)

i. Option 1: DHCP Servers

□ D1 □ D6

ii. Option 2: Switches

□ S2 □ S3 □ S4 □ S5 □ S9

iii. Option 3: Computers

□ C7 □ C8 □ C10 □ C11

Problem 5 ***Blind Signatures*** **(15 points)**

A *blind signature scheme* is a digital signature scheme where the contents of the original message are hidden from the person signing the message. Specifically, we define the following cryptographic primitives:

1. $\textsc{KeyGen}()$: returns a public key pk and a secret key sk.
2. $\textsc{Sign}(sk, m)$: signs m, as in traditional signature schemes.
3. $\textsc{Verify}(pk, s, m)$: verifies the signature s on the message m, as in traditional signature schemes.
4. $\textsc{Blind}(m, r)$: takes in a message $m \in [0, N]$ and a random number r. It returns a "blinded" message m'.
5. $\textsc{Unblind}(s', r)$: when applied to a valid signature s' on a blinded message m' with the original randomness r, $\textsc{Unblind}$ produces a valid signature s on the original message m.

For the purpose of this question, we will assume that our messages, signatures and randomness are all encoded as numbers modulo some N. Assume that all math occurs modulo N.

If Alice wants Bob to blindly sign the message m, she can perform the following protocol:

1. Alice generates a random number r with $0 \le r < N$.
2. Alice computes $m' \leftarrow \textsc{Blind}(m, r)$ and sends m' to Bob.
3. Bob computes $s' \leftarrow \textsc{Sign}(sk, m')$ and sends s' to Alice.
4. Alice computes $s \leftarrow \textsc{Unblind}(s', r)$.
5. Alice checks $\textsc{Verify}(pk, s, m)$ to make sure that Bob is giving her a real signature.

A signature scheme is **valid**, if, for every correct run of Steps 1–4, it verifies correctly in Step 5 (except possibly with some negligible probability of failure).

Moreover, we say that it is **unforgeable** if Alice cannot create a signature on a message m for which she did not follow the protocol above, similarly to regular digital signatures.

We consider the following two possible definitions for blindness:

1. **Blindness I** (Random messages): Alice randomly chooses two messages m_0 and m_1. Alice flips a coin b to select either m_0 or m_1, and sends a blinded message m'_b to Bob. Alice then shows Bob both m_0 and m_1. Bob wins if he can identify which of m_0 and m_1 was blinded. If Bob cannot win with probability significantly greater than $\frac{1}{2}$, we say that the scheme satisfies **Blindness I**.
2. **Blindness II** (Bob chooses): Bob chooses two messages m_0 and m_1, and gives both to Alice. Alice flips a coin b to select either m_0 or m_1, and sends a blinded message m'_b to Bob. Bob wins if he can identify which of m_0 and m_1 was blinded. If Bob cannot win with probability significantly greater than $\frac{1}{2}$, we say that the scheme satisfies **Blindness II**.

Now we will consider some candidates for blind signature schemes. For each of the schemes below, identify which of the properties it supports. In all the schemes below, $\textsc{KeyGen}, \textsc{Sign}$ and $\textsc{Verify}$ are identical to the RSA signature scheme as presented in class, **except no hash is applied to the message before signing or verification**. The number N is set to be the public key. If a scheme is not a valid blind signature scheme, you do not need to mark its other properties.

[Quick RSA reminder: $sk = d$, $\textsc{Sign}(sk, m) = H(m)^d \mod N$, $\textsc{Verify}(pk, s, m) : s^3 \mod N \overset{?}{=} H(m) \mod N$.]

(a) $\textsc{Blind}(m, r) = mr^3$ and $\textsc{Unblind}(s', r) = s'r^{-1}$.

○ Valid scheme ○ Invalid scheme

□ Blindness I (Random messages) □ Unforgeable

□ Blindness II (Bob chooses)

(b) $\textsc{Blind}(m, r) = H(mr^3)$ and $\textsc{Unblind}(s', r) = s'H(r^{-1})$, where H is a cryptographic hash function.

○ Valid scheme ○ Invalid scheme

□ Blindness I (Random messages) □ Unforgeable

□ Blindness II (Bob chooses)

(c) $\textsc{Blind}(m, r) = H(m)r^3$ and $\textsc{Unblind}(s', r) = s'r^{-1}$, where H is a cryptographic hash function.

○ Valid scheme ○ Invalid scheme

□ Blindness I (Random messages) □ Unforgeable

□ Blindness II (Bob chooses)

Problem 6 ***Blue and Red Teams*** **(16 points)**

Give only one answer to each question.

(a) Answer the following short-answer questions about network security defenses.

i. Explain the meaning and purpose of the rule `ext allow tcp *:80 -> *:* if ACK set`.

ii. In class, we saw an attack where a packet containing something "bad" (the word "root") was split over multiple packets to evade detection. Name one technology that can detect the correct byte sequence received by the destination.

(b) Answer the following short-answer questions about web attacks.

i. Boogle uses the following PHP code to display what a user has searched for.

```
$SEARCHTERM = $_GET['QUERY'];
echo "You searched for: <b>$SEARCHTERM</b>";
```

What query should Alyssa enter to make the site display an alert box saying hacked?

ii. Boogle's website performs the following database query for every search query:

```
INSERT INTO query_log VALUES ('$SEARCHTERM');
```

where $SEARCHTERM is the search term that a user searches for. Moreover, Boogle checks the syntax of SQL queries before they are executed, and will not execute them if their syntax is invalid (i.e., quotes/parenthesis are closed incorrectly, missing semicolon separating queries). Boogle also does not allow the command "`--`" (SQL comment) in queries because it saw in CS 161 that it can be exploited by hackers.

Boogle also has a table `prohibited_users` where it stores (in a column called `user`) the name of each user who is not allowed to access certain parts of its site. Eve knows that her name 'Eve' is in that database and wants to remove it. She has access to the form that triggers the insert query above. What should Eve enter in the search box to do so? (Keep in mind Boogle's filter described above.)

Problem 7 ***New TLS*** **(15 points)**

You are in charge of creating the new TLS specification! You have received the following suggestions by email. **Evaluate** each suggestion by determining if it is secure or insecure, and then **explain** your answer.

(a) Consider Diffie-Hellman TLS. It is possible that the connection dies due to network issues. SUGGESTION: Let the client restart the connection by sending the plaintext PS to the server. The server will keep track of all recently used PS values and their corresponding keys, allowing for easy resumption. Evaluate and explain.

(b) Consider Diffie-Hellman TLS. SUGGESTION: Rather than having the server choose g and p and send them, have the client decide on the values as part of the ClientHello message. The server no longer sends g and p, and only sends $g^a \mod p$ along with a signature. The server assumes that the client chooses appropriate values for g and p. Evaluate and explain.

(c) Consider RSA TLS. SUGGESTION: Rather than sending $\{PS\}_{K_{\text{server}}}$, we have the client choose the server keys C_s, I_s. It then sends $\{C_s, I_s\}_{K_{\text{server}}}$ to the server. Then the server chooses the client keys C_b, I_b and sends $\{C_b||I_b, \text{MAC}_{I_s}(C_b||I_b)\}_{C_s}$ to the client. Evaluate and explain.