CS 378 - Network Security and Privacy
Spring 2017

FINAL

May 3, 2017

DO NOT OPEN UNTIL INSTRUCTED

YOUR NAME: _____

Collaboration policy

No collaboration is permitted on this exam. Any cheating (*e.g.*, submitting another person's work as your own, or permitting your work to be copied) will automatically result in a failing grade. The Computer Sciences department code of conduct can be found at `https://www.cs.utexas.edu/academics/conduct`.

Final (125 points)

Problem 1 (24 points)

Circle only <u>one</u> of the choices **(4 points each)**.

1. **TRUE** **FALSE** DNS queries are authenticated by 16-bit random transaction IDs.
2. **TRUE** **FALSE** Same-origin policy in modern Web browsers says that a script can access Web resources only from the same server, protocol, and port as the script itself.
3. **TRUE** **FALSE** An SSL connection using an Extended Validation SSL certificate provides more security against an eavesdropper than one which uses a self-signed certificate.
4. **TRUE** **FALSE** Recall that an RSA modulus n is a composite number, which is a product of two large primes. If someone discovers an efficient algorithm for computing the greatest common divisor of two composite numbers, then breaking RSA will become feasible.
5. **TRUE** **FALSE** To use an Application Level Gateway, applications such as web browsers and FTP clients must be modified accordingly.
6. **TRUE** **FALSE** If an attacker acquires a copy of a websites SSL/TLS certificate, and uses DNS cache poisoning to direct a victim to an attacker controlled server, the attacker can successfully impersonate the website.

Problem 2 (6 points)

Describe at least **two** changes that could be made to the C **compiler** to prevent buffer overflow attacks. Explain why these defenses would be effective.

Problem 3

Consider a stateless packet filtering firewall installed at the gateway of a corporate network. Assume that all traffic to and from the network flows through the firewall. The format of a firewall rule is as follows:

```
Interface, Action, SourceIP, SourcePort, DestIP, DestPort
```

Problem 3a (5 points)

Can the packet filter block all external attempts to connect to a Web server located at a particular address within the corporate network, but permit FTP access to the same server? If yes, what would the firewall rule(s) look like? If no, why not?

Problem 3b (5 points)

Can the packet filter block all email messages containing the string `V1AGRA` to a particular client within the corporate network? If yes, what would the firewall rule look like? If no, why not?

Problem 3c (6 points)

List **three** different network attacks that even a stateful firewall cannot protected against.

Problem 4

Recall that DKIM and SPF are two defenses against spam.

Problem 4a (6 points)

Describe an attack on SPF that does not work against DKIM.

Problem 4b (5 points)

I have used CAPTCHA breaking techniques to acquire thousands of Gmail accounts, with which I intend to send spam. Would the SPF defense against spam help? How about DKIM? In both cases, why or why not?

Problem 5 (8 points)

In the context of email spam, what is a "graylist"? How does it work, and why is it effective?

Problem 6

You are careful, and ensure that you only ever visit websites over HTTPS. In fact, you have rendered your browser incapable of making HTTP requests, meaning you don't need to worry about attacks like sslstrip or mixed content. You access `https://www.awkwardwebsite.com` (hosted at IP 1.2.3.4), log in, and view some awkward videos.

Problem 6a (6 points)

For each of the following, state whether or not it is possible, and justify your answer:

1. A passive network attacker can infer your login credentials to awkwardwebsite.com
2. A passive network attacker can infer that you visited awkwardwebsite.com
3. A passive network attacker can infer that you visited some website hosted at 1.2.3.4

Problem 6b (5 points)

How about in the face of an active network attacker?

Problem 7

The Same Origin Policy is defined over the (*domain, protocol, port*) triple.

Problem 7a (5 points)

The primary objective of most Man in the Browser attacks is stealing online banking credentils. What is a Man in the Browser attack? Why do the Same Origin Policy and SSL/TLS not protect you from them?

Problem 7b (5 points)

Given the security guarantees of SSL/TLS, including preventing attackers from injecting scripts into HTTP responses, would protections based on the Same Origin Policy still be necessary if *all* HTTP connections used SSL/TLS? Why or why not?

Problem 8 (10 points)

Recall the format of service tickets in Kerberos. When the client C requests a ticket for some network server V from the ticket-granting service (TGS), TGS sends, encrypted under their pairwise symmetric key, the following:

- Fresh session key k_{cv} to be used between C and V.
- Identity of V.
- Current timestamp.
- Ticket, encrypted under V's symmetric key k_v. It consists of:
  - Session key k_{cv} (same as above).
  - Identity and network address of C.
  - Identity of V.
  - Current timestamp.

Observe that all information encrypted under k_v is already known to the client C. What is the purpose of encrypting it?

Problem 9 (10 points)

Suppose that every packet observed by a network-based intrusion detection system (NIDS) belongs to one of the following mutually exclusive categories: legitimate (88% of all traffic), known worm (3%), distributed denial of service (4%) or port scan (5%).

The NIDS correctly classifies all known-worm packets. A DDoS packet is classified as DDoS with probability 85%, as a known worm with probability 10%, and as a legitimate packet with probability 5%. A port-scan packet is classified correctly with probability 90%, and misclassified as a legitimate packet with probability 10%. A legitimate packet is classified as legitimate with probability 94%, and misclassified as belonging to any of the three attack categories with equal probability.

If the NIDS announces that a particular packet belongs to a known worm, what is the probability that this packet is **not** a legitimate packet? Show your calculations.

Problem 10

Consider the following variant of RSA encryption. Recall that an RSA public key is a pair (n, e). To encrypt some message m, first generate a fresh random value r of the same length as m. Use r as if it were a one-time pad to encrypt m (*i.e.*, let $s = m \oplus r$), and then encrypt r using plain RSA (*i.e.*, let $t = r^e \mod n$). The ciphertext is the (s, t) pair.

Problem 10a (4 points)

How does decryption work in this scheme?

Problem 10b (5 points)

Is this encryption scheme secure against the chosen-plaintext attack? Explain.

Problem 11

Being security conscious, I encrypt my files with a very strong encryption scheme. An encrypted file is indistinguishable from uniformly random bits.

Being a nerd, I know that 1 kilobyte is 2^{10} bytes, and 1 megabyte is 2^{20} bytes.

Problem 11a (5 points)

A particularly unpleasant virus has been spreading. Luckily, the people in who wrote my signature-based malware detector have discovered that every file infected with this virus always has the four consecutive bytes (`0xBAADC0DE`) in it somewhere. These four bytes have been loaded into the detector as a signature.

How large could a file be before I would expect it to start triggering a false positive? Show your calculations.

Problem 11b (5 points)

More careful analysis results in the discovery that in fact there are 8 consecutive bytes that are always present in an infected file (`0xBAADC0DE15BAAAAD`). How would updating the signature affect your answer to the previous question?