CS 161
Computer Security

Dawn Song
Fall 2012

Practice Questions 2

1. Network Security

   Usually, the DNS protocol runs over UDP. However, it is also possible for DNS to use TCP.

   (a) Suppose you are using your laptop on an open wireless network and an attacker is within range of the wireless network, so the attacker can eavesdrop on all your traffic and inject forged packets. **Circle one** of the following that best describes the threat the attacker poses:

      1. The attacker can successfully inject a spoofed DNS response if your laptop uses UDP for all of its DNS queries, but not if it uses TCP for all of its queries.
      2. The attacker can successfully inject a spoofed DNS response if your laptop uses TCP for all of its DNS queries, but not if it uses UDP for all of its queries.
      3. The attacker can successfully inject a spoofed DNS response if your laptop uses either TCP or UDP for its DNS queries.
      4. The attacker cannot successfully inject spoofed DNS responses.

      Answer: (3) The attacker can successfully inject a spoofed DNS response if your laptop uses either TCP or UDP for its DNS queries.
      Both TCP and UDP are prone to injection attacks if the attacker can see all the traffic. So, all the communication (including application level protocols other than DNS) are vulnerable to injection attacks in this situation.

   (b) Suppose you access the Internet over a secured Ethernet network, so that the attacker cannot eavesdrop on your traffic, but the attacker can still inject forged packets. You can use either TCP or UDP for your DNS queries. Assume that the relevant TCP implementations choose Initial Sequence Number (ISNs) uniformly at random, and that the relevant DNS implementations do not implement source port randomization. Regarding Kaminsky-style blind spoofing of DNS replies, **circle one** of the following that best describes the threat the attacker poses:

      1. When you use TCP for your queries you are safer (harder to attack) than when using UDP.

2. When you use UDP for your queries you are safer (harder to attack) than when using TCP.
3. You are equally vulnerable to the attack whether you use UDP or TCP.
4. In this scenario, you are not vulnerable to the attacker regardless of whether you use UDP or TCP.

Answer: (1) When you use TCP for your queries you are safer (harder to attack) than when using UDP.
For a successful attack, the attacker needs to create a valid forged packet, as well as get the contents (DNS replies) of the packet right. Without the knowledge of Initial Sequence Number, it is hard to forge a valid TCP packet. However, without source port randomization, it is easy to forge a UDP packet which looks like as if it arrived from the DNS server.

(c) Suppose we could deploy a mechanism that would ensure IP source addresses always correspond to the actual sender of a packet. In other words, suppose it is impossible for an attacker to spoof source addresses. **Circle all** of the following threats that this mechanism would completely eliminate. By eliminate a threat, we mean that the anti-spoofing mechanism would suffice to prevent exploitation of the threat, without any additional mechanisms or assumptions.

1. Buffer overflow attacks
2. Cross-site request forgery (CSRF) attacks
3. TCP SYN flooding
4. TCP RST injection
5. None of the above

Answer: TCP RST injection.
Buffer overflow and CSRF do not rely on address spoofing. TCP SYN flooding will still be an possible if multiple machines (a botnet, for example) starts attacking a single victim. However, an attacker will not be able to launch a SYN flooding attack just from a single machine.

(d) Again suppose we could deploy a mechanism that would ensure IP source addresses always correspond to the actual sender of a packet. **Circle all** of the following threats for which this mechanism would eliminate at least some common instances of the attack but not all instances. Eliminate an attack instance refers to preventing that attack instance from succeeding, without any additional mechanisms or assumptions.

1. Buffer overflow attacks
2. Cross-site request forgery (CSRF) attacks
3. TCP SYN flooding
4. TCP RST injection
5. None of the above

Answer: TCP SYN flooding.

(e) Which of the following is true of viruses/worms? **Circle all** that apply.

1. Polymorphic viruses are harder to detect with signature-based techniques than metamorphic viruses.
2. Metamorphic viruses require public-key cryptography.
3. You can write a worm that uses multiple different techniques to spread.
4. None of the above.

Answer: (3) You can write a worm that uses multiple different techniques to spread.
A great example is Stuxnet, which used 4 infection vectors.

(f) Suppose you have a technology available that will prevent any buffer overflow attack. If you deploy it everywhere, which of the following best describes its effectiveness? **Circle just one**.

1. It would prevent all types of worms from propagating.
2. It would slow down the propagation of all types of worms but not fully prevent the propagation of any.
3. It would prevent the propagation of some types of worms, but not all types.
4. It would not help in preventing worms from propagating, but would slow down the propagation of some types of worms.
5. It would not help in preventing worms from propagating nor in slowing down their propagation.

Answer: (3) It would prevent the propagation of some types of worms, but not all types.
There are worms which use other infection vectors than just buffer overflows, e.g. unprotected administrator accounts which allow remote login.

(g) A random-scanning worm spreads (**circle just one**):

1. with linear speed until it reaches critical mass, after which it propagates at an exponentially increasing rate.

2. exponentially quickly at the beginning, but with the rate decreasing as more and more systems are infected.
3. at a quadratically increasing rate throughout the worms propagation.
4. at an exponentially increasing rate until all susceptibles are infected.

Answer: (2) exponentially quickly at the beginning, but with the rate decreasing as more and more systems are infected.

2. Web Security

(a) When visiting a website, such as a bank's website, which of the following is a necessary part of preventing a man-in-the-middle attack?

(a) An HTTPS connection

(b) A security image

(c) A CAPTCHA

○ (a) only

○ (b) only

○ (c) only

○ Both (a) and (b)

○ Both (b) and (c)

Answer: (a) only. HTTPS provides end-to-end cryptographic security, so this alone is sufficient. In a man in the middle, an attacker could communicate with the bank website itself to get your security image. A CAPTCHA provides no client security; it is a way for servers to differentiate humans from machines.

(b) In the following PHP code, in which line is there a potential XSS attack, assuming all sanitizer functions work correctly and all variables are user inputs?

```
 1  <?php
 2  echo ‘<p>Hello, ’ . sanitizeHTML($username) . ‘</p>’;
 3  echo ‘<p>The homepage for user id ’ .
 4       sanitizeNumber($userid) . ‘ is:</p>’;
 5  echo ‘<p><a href=   ’ . sanitizeHTML($homepage).
 6       ‘   >homepage</a></p>’;
 7  echo ‘<p><a href= myprofile.php >’ .
 8       ‘Return to profile of ’ .
 9       sanitizeHTML($username) .
10       ‘.</a></p>’;
11  ?php>
```

○ Line 2

○ Line 4

○ Line 5

○ Line 9

○ There is no XSS

Answer: Line 5. This variable is output to the attribute context, not the general HTML context. For example, this does nothing to stop the text `javascript:` from being placed in the `href` attribute.

The rest of the calls to `sanitizeHTML` are in the correct context, while the call to `sanitizeNumber` does not allow anything through that could be interpreted as code.

(c) In the `trusted.com` website, there are a number of references to external URLs at `untrusted.com`. For each of the following HTML elements that appear in the `trusted.com` website, *when the external resource is downloaded*, specify whether it is executed in the `trusted.com` or the `untrusted.com` origin.

1. `<a href="untrusted.com">` ____________________ **Answer: untrusted.com because a new window is opened for link.**
2. `<script src="untrusted.com">` ____________________ **Answer: trusted.com because this loads a script and runs it in the current window.**
3. `<iframe src="untrusted.com">` ____________________ **Answer: untrusted.com because this loads the content in a new frame which has it's own origin.**
4. `<style src="untrusted.com">` ____________________ **Answer: trusted.com because this loads the source as style in the context of the current window.**

(d) You are visiting a banking website, `http://www.americasbank.com`. After logging in, a session is established with the server with a random 8 bit session ID in the cookie. Unfortunately, Mallory, a network attacker, is able to hijack your session with the bank and transfer out a large sum of money. Which of the following changes does the bank need to do to prevent such attacks in the future and provide the most flexibility? **Circle all that apply (leave blank for none). Point awarded only if all the correct options (and no others) are circled**.

○ Use SSL/TLS.

- ○ Increase the random session ID length.
- ○ Check the IP address of the connection. If it is different from the previous IP address used with the given session ID, reject the connection.
- ○ Create a new, random session ID every 5 minutes.
- ○ Require the user to change their password at least once a month.

Answer: Increase random session ID length and use SSL/TLS. +2 point for both, 0 points otherwise. If you do not increase the session ID length, an attacker could go through all possibilities to find the user's session ID because it's so short. If you do not use SSL/TLS, an eavesdropper could observe the session ID on the wire and use it for herself.

You cannot rely on IP addresses because this would prevent legitimate users, such as those with a laptop or dynamic IP address assignment. Creating a new, random session ID ever 5 minutes would still allow an attacker to guess or eavesdrop on the session ID during the 5 minute period, which they then could repeat. Changing passwords once a month does not affect the session ID, and additionally occurs rarely.

(e) Are the following URIs same origin?

1. http://www.example.com:80/index.html
2. http://www.example.com/index.html

- ○ Yes
- ○ No

Answer: Yes. HTTP runs on port 80 by default, so the ports, domains, and protocols match, so by the same origin policy, they are the same.

(f) I go to a page on `http://www.example.net` and log in. The person who wrote the login page is my friend, and I know he always makes sure to set the form action (the target uri for form submission) to an HTTPS URI. Which one of the following options is correct? **Circle all that apply (leave blank for none). Point awarded only if all the correct options (and no others) are circled.**

- ○ A network attacker cannot read my password since it is always sent to an HTTPS URI

- ○ A network attacker can read my password because the form is submitted using a HTTP GET, which means the password is sent as part of the URI.
- ○ A malware attacker can get my password.
- ○ A web attacker can read my passwords using framebusting.

Answer: A malware attacker can get my password because she has software installed on the machine that can do anything it likes to the browser.

A network attacker can read the password, but not for the reason listed. A POST request sent via a form with an HTTPS URL is secure from a network attacker except that the initial page GET request was made via HTTP, not HTTPS. Thus a network attacker could have already modified the page, perhaps adding in a JavaScript key logger or modifying the POST URL to be HTTP.

Framebusting is unrelated to this type of attack.

(g) A web application firewall is a software program that sits on the network, next to the web application server and looks at all HTTP Requests going to the server. It is used to detect XSS attempts. Which of the following attack attempts could be detected by a web application firewall interposing on all requests to the `www.example.com` web server? **Circle all that apply (leave blank for none). Point awarded only if all the correct options (and no others) are circled**.

1. `http://www.example.com/postComment.php` with POST body `<script>doEvil()</script>`
2. `http://www.example.com/post.php?comment=<script>doEvil()</script>`
3. `http://www.example.com/search.php#!?=in=db&query=<script>doEvil</script>`
4. `http://blog.example.com/post.php?comment=<script>doEvil()</script>`

Answer: 1 and 2 only. Superfluous options gives zero points. In 3, everything after the # is not sent to the server; these are fragment URLs, and the browser does not send them in the server. In 4, the web application firewall sits at `http://www.example.com`, not at `http://blog.example.com`

(h) Prof. Evil provides all the members of CalTopia access to Zion, the centralized servers holding a distributed Badoop Filesystem for storing data. Users can ssh in and see their files and/or create new files. The linux kernel ensures the permissions setup. For example, files of `/home/profevil/` are all readable only by the `profevil` user and not by `minion420` user. Prof. Evil also creates a web based UI to view existing files. In this website, the UI requires that you login with your username and password. The web server runs as group `www,` and all user files are given group `www.` The server validates the login credentials with the OS. Finally, the web app looks up the owner of the file in question and makes sure the owner matches the logged in user. For example, if a file is not readable by `minion420`, then the WebUI will refuse to display it to him. Which of the following is correct regarding the check that the owner of the file matches the logged in user? **Circle all that apply (leave blank for none) Point awarded only if all the correct options (and no others) are circled**.

○ The server code doesnt need to do this; the OS kernel takes care of it automatically via the permissions setup. They should remove the check for efficiency.

○ The server code doesnt really need to do this, but its a good defense in depth mechanism.

○ The server code needs to do this, as otherwise minion420 could read profevils files.

○ The server code needs to do this because the OS kernels implementation might have a bug.

Answer: Only option 3 receives +2 points. Any other option, or selecting superfluous options results in zero points.

(i) `BCS.com` wants to add social networking to its website using gracebook.com. For this, it needs to accept post-messages from gracebook.com subdomains. The following code does this check:

```
window.onmessage=function(e){
if(e.origin.indexOf('.gracebook.com') != -1){
//trust the message
}
```

The String `indexOf` method is defined as:

> The indexOf method returns the index within the calling String object of the first occurrence of the specified value, returns -1 if the value is not found.

We only want to accept messages from gracebook.com and all its subdomains. Is the check sufficient? If not, give a counter example (i.e., a possibly attacker controlled domain that will be trusted). **Answer: No. 0 points for no example. 3 point for a correct example. One possible answer is :**`foo.gracebook.com.attacker.com`

3. Mobile Security

   Which one of the following mechanisms is **NOT** an integral component required for Android application isolation?(**circle one**)

   ○ Application code signing

   ○ Android permission system

   ○ Linux users

   ○ Linux process isolation

   Answer: Android permission system