Popa & Weaver
Spring 2019

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 and may further result in partial or complete loss of credit.

SIGN your name: _____

PRINT your SID: _____

Name of the person sitting to your left: _____

Name of the person sitting to your right: _____

You may consult **two** double-sided, handwritten sheet of paper of notes. You may not consult other notes or textbooks. Calculators, computers, and other electronic devices are not permitted.

Bubble every item completely. Avoid using checkmarks or Xs.
If you want to unselect an option, erase it completely and clearly.

For questions with **circular bubbles**, you may select only one choice.

- ○ Unselected option (completely unfilled)
- ● Only one selected option (completely filled)

For questions with **square checkboxes**, you may select any number of choices (including none or all).

- ■ You can select
- ■ multiple squares (completely filled).

If you think a question is ambiguous, please come up to the front of the exam room to the TAs. If we agree that the question is ambiguous we will add clarifications to the document projected in the exam rooms.

You have 110 minutes. There are 9 questions of varying credit (116 points total).

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

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

You don't need to provide an explanation to any part of this question.

(a) TRUE or FALSE: TCP sequence numbers prevent replay attacks within the same TLS session. (Assume less than 2^{31} bytes of data has been transferred.)

○ TRUE ● FALSE

> **Solution:** TLS does not encrypt or provide integrity on the TCP layer since TLS is built on TCP.

(b) TRUE or FALSE: A single message in TLS can be split into many TCP packets, and remains secure.

● TRUE ○ FALSE

> **Solution:** As above–TLS is layered on top of TCP and so TCP can do weird things independent of TLS.

(c) TRUE or FALSE: `www.example.com` can set a cookie with the flag `HttpOnly`. Then, this cookie can only be accessed through HTTP, but cannot be accessed through HTTPS.

○ TRUE ● FALSE

> **Solution:** As above–TLS is layered on top of TCP and so TCP can do weird things independent of TLS.

(d) TRUE or FALSE: `www.example.com/c` can set a cookie with the flag `path = /a` in a user's browser. Then, when the user visits the site `www.example.com/a/b`, the user's browser will send this cookie.

● TRUE ○ FALSE

> **Solution:** Two parts of the cookie policy: (1) domains can set to any other path and (2) any subpath of the path will receive a cookie.

(e) TRUE or FALSE: Say that `example.com` uses both DNSSEC and HTTPS. If the DNSSEC KSK of `example.com` is compromised, data received from `https://example.com` is not assured to be confidential.

○ True ● False

(f) True or False: Say that `example.com` uses both DNSSEC and HTTPS. If the TLS private key of `example.com` is leaked, data received from `https://example.com` is not assured to be confidential.

● True ○ False

Solution: DNSSEC does not provide confidentiality, TLS does.

(g) True or False: Say that a user sends a DNS query asking for `nx.example.com`, but this domain does not exist. One advantage of NSEC3 over NSEC is that NSEC3 hides the domain name that does not exist, *i.e.*, `nx.example.com`.

○ True ● False

Solution: NSEC3 hides (somewhat) the *existing* domains of the domain name.

(h) True or False: A banking website requires the user to attach their password as a form field in every HTTPS request to the website. If the password is incorrect, the bank ignores the request. Assume that the bank accepts only HTTPS connections. If the user's password is sufficiently high-entropy, this method prevents CSRF.

● True ○ False

Solution: This provides protection against CSRF because a web attacker does not know the password.

(i) True or False: Randomizing the source port used by DNS queries can help prevent on-path attackers from spoofing DNS replies.

○ True ● False

Solution: An on-path attacker can see the source port in the DNS reply, so randomization provides no additional security.

(j) True or False: An on-path attacker who successfully spoofs a DHCP reply can become a man-in-the-middle for all victim traffic to the Internet.

● True ○ False

Solution: Yes, they can spoof the IP of the gateway router.

Problem 2 ***Wildcard DNSSEC*** **(11 points)**

In this question, we discuss a variant of DNSSEC that supports wildcards.

We define a *wildcard* domain as a domain that matches many subdomains.

- For example, the wildcard domain `*.google.com` matches *all* domains under `google.com`, including `mail.google.com` and `drive.google.com`. Here, the star * indicates a wildcard.

We define a *non-wildcard* domain as a domain with no wildcard, such as `maps.google.com`.

- **Importantly,** non-wildcard domain records take *priority* over the wildcard domain records.
- For example, if there are two records:

* .google.com	5.6.7.8
mail.google.com	1.2.3.4

then the DNS server should respond `1.2.3.4` as the IP address of `mail.google.com`.

(a) In classical DNSSEC, if a user asks for the IP address of `mail.google.com`, and there is a *non*-wildcard record, the DNS server will return:

1. the IP address of `mail.google.com`.
2. the signature of the record containing the IP address.

◇ **Question**: What is the type of the record (*e.g.*, A, NS) that contains the information above?

1. the IP address of `mail.google.com`...
   - ... is in a record of the type _____ (write the record type).
2. the signature of the record above...
   - ... is in a record of the type _____ (write the record type).

Solution: A, RRSIG (4 points)

(b) We now modify the DNSSEC protocol to support wildcards, as follows:

- Consider a user who asks the IP address of the domain `abc.google.com`. There is *only* a wildcard record that matches `abc.google.com`, as follows:

  `*.google.com 5.6.7.8,`

- The server will return a record that consists of:

  `(abc.google.com, 5.6.7.8),`

  and *a signature* over this record.

However, this design is not good because it involves *online signing*; that is, the server cannot precompute the signature.

◇ **Question:** List one drawback of having online signing in DNSSEC. (write less than 10 words)

Solution: Answer one of the following:

1. online signing adds latency and consumes more resources from the DNS server

2. online signing requires having DNS keys on the DNS servers (greater risk of compromise)

3. can make the server more vulnerable to DoS/DDoS

(3 points)

(c) To remove the online signing, we can have the DNS server instead do the following:

- Return the wildcard record, which consists of:

  `(*.google.com, 5.6.7.8),`

- Return a signature over the record above.

A client who asks `abc.google.com` will receive this response. The client will believe that:

- No non-wildcard record matches the query.
- Only the wildcard captures this domain.

◇ **Question:** Is this design secure? If yes, explain why. If not, explain how it could be made secure without requiring online signing (max 15 words).

○ Yes, it's secure ● No, it's insecure

Solution: This design is insecure. Say `mail.google.com` is a domain where there is a non-wildcard record that matches it. The attacker can return the wildcard response above to the victim client, and the victim will believe `5.6.7.8` is the IP address for `mail.google.com`. But the IP address should be `1.2.3.4`, as discussed earlier in this problem.

To be made secure, the server needs to provide a signed statement that proves that the requested domain is not an existing non-wildcard domain. One possibility is providing a signed NSEC (or NSEC3) record. (4 points)

If the student did not write "with signature" or "signed" or "RRSIG on NSEC", only 2 points.

Problem 3 ***Online Banking*** **(16 points)**

In an online banking system, each customer has a unique username and a secret numerical PIN. To access the banking website, a customer logs in to this web system using their username and PIN.

Suppose the login script uses the following PHP code:

```
$user = escape_sql($_GET['username']);
$pin = $_GET['PIN'];
$query = "SELECT * FROM Users WHERE user = '$user' AND pin = '$pin'";
$results = $db->executeQuery(query);
if ($results->numRows != 1) {/* login fails */}
else {/* login succeeds as $user */}
```

Here, the `escape_sql` function escapes all quotes, dashes, and semicolons. You may assume that SQL injection cannot be performed from any input that has been sanitized with `escape_sql`.

(a) Mallory obtains the source code of the login script and notices that it is vulnerable to SQL injection.

◇ **Question:** Describe *what input* Mallory should use (*e.g.*, `$_GET['username']`, `$_GET['PIN']`) to exploit this vulnerability in order to drop the table `Users`.

`$_GET['username']` = _____

`$_GET['PIN']` = _____

Solution: Mallory can drop the `Users` table by passing anything as the username and `"'; DROP TABLE Users; --"` as the PIN; it results in the following query being executed:

```
SELECT * FROM Users WHERE user = '' AND pin = ''; DROP TABLE Users; --
```

The student should use "`;`" for separation (2 points) and '`--`" to end the SQL query (or properly start a "`'`" which will be ended by the "`'`" in the query) (2.5 points).

This question thus has 4.5 points in total.

(b) Mallory knows a rich user whose username is "`alice`".

◇ **Question:** Explain *what input* Mallory should use for SQL injection in order to log in to the online banking system as Alice, *without* knowing Alice's PIN.

`$_GET['username']` = _____

`$_GET['PIN']` = _____

Solution: Mallory can log in as Alice by passing in `"alice"` as the username and `"' OR user = 'alice'; --"` as PIN. This results in the following query being executed:

```
SELECT * FROM Users WHERE user = 'alice' AND pin = '' OR user = 'alice'; --'
```

Another possibility is having the PIN be `"' OR user = 'alice"`, as the "`'`" before alice will match the ending "`'`" in the query. (this part has 4 points)

If the student instead uses Alice as the username but adds an OR condition that's always true at the end, the SQL query will still work, but the number of rows in the result is likely not 1. In that case, the student will only receive 1.5 points.

It is very important to use Alice as the username. If not, minus 1.5 points.

(c) The bank decides to fix the SQL injection bug by constraining the PIN that the user enters to be an integer, as follows.

- Change the HTML of the login page:
  - From:

```
<form action="/login.php" method="POST">
<p>Username: <input type="text" name="username" /></p>
<p>PIN: <input type="text" name="PIN" /></p>
<p><input type="submit" value="Login" /></p>
</form>
```

  - To:

```
<form action="/login.php" method="POST">
<p>Username: <input type="text" name="username" /></p>
<p>PIN: <input type="number" name="PIN" /></p>
<p><input type="submit" value="Login" /></p>
</form>
```

- The `<input>` element with "`type=number`" will be treated differently by the web browser. The web browser will prevent the user from entering non-numerical data into this input field.
- ***No*** change to the PHP script.

◇ TRUE or FALSE: Does this fix prevent the SQL injection in parts (a) and (b)?

○ TRUE ● FALSE

You don't need to provide an explanation.

Solution: Mallory can still perform the attacks. The requirement that the PIN is a number is enforced by the browser, but it should be enforced by the server. Mallory can forge HTTP request and send the same string as the PIN. (2.5 points)

(d) The bank later did a major re-design of their website. Unfortunately, the new version was vulnerable to a CSRF attack. Mallory notices that she can exploit it by having another user make a GET request like:

```
/transfer?amt=100&to=Mallory
```

and the user who makes this GET request will send $100 to Mallory.

For each of the choices below, mark if it defends against CSRF. Assume that aside from the `/transfer?` endpoint, no other part of the bank is vulnerable to any web attacks.

◇ **Select** 0 to 5 options.

- ☐ Disable JavaScript from executing on the bank's website via content security policy (CSP).
- ☐ Add a new request parameter "`from=`" to the `/transfer?` endpoint. If "`from`" does not match the name of the currently logged in user according to the session cookie, reject the request.
- ■ When a user logs in, send them a new cookie called `Token`, which consists of 128 random digits (different for each user). When a user makes a request to transfer money, the bank uses JavaScript to retrieve the cookie and add it as a query parameter "`token=`" to the `/transfer` endpoint. The bank checks that the `token` query parameter matches what the cookie was originally set to for this user.
- ☐ When a user logs in, send them a new cookie called `Token`, the same one above. When the user makes a request to the `/transfer?` endpoint, the bank checks that cookie sent by the user matches what the cookie was originally set to for this user.
- ■ Reject any request to the `/transfer?` endpoint where the `Referer` is not the bank's website.

Solution: 5 points total. 1 point for each correct marking.

Problem 4 ***The Subtle TLS*** **(10 points)**

This question talks about a ***modified*** RSA TLS protocol. Recall in RSA TLS, the client sends R_b to the server, and then the server replies with R_s to the client. The cipher and integrity keys are generated by putting R_b, R_s, and the premaster secret PS together into a PRNG, like $PRNG(R_b \parallel R_s \parallel PS)$.

For each part of the question, assume an attacker with the following capabilities:

- The attacker is a man-in-the-middle.
- The attacker also controls a website `evil.example.com`, with a valid HTTPS certificate. The user may connect to this site while browsing the Internet.

(a) Let us assume that in RSA TLS, we make the following change. We generate the pre-master secret by $PS = R_b \oplus R_s$, where $\oplus$ is bit-wise XOR.

TRUE or FALSE and Explain: This modified protocol preserves the integrity of RSA TLS.

○ TRUE ● FALSE

◇ **Explain *concisely***:

_____

_____

Solution: No because a MITM can calculate PS and derive the integrity keys, since R_b and R_s are sent in the clear. (3 points)

(b) Let us assume that in RSA TLS, we make the following change. Instead of providing "$R_b \parallel R_s \parallel PS$" as input to the PRNG, both parties provide "$R_b \oplus R_s \oplus PS$" as the only input to the PRNG. That is, the cipher and integrity keys will depend only on $PRNG(R_b \oplus R_s \oplus PS)$.

TRUE or FALSE and Explain: This preserves the security of TLS against replay attacks.

● TRUE ○ FALSE

◇ **Explain *concisely***:

_____

_____

Solution: Yes. In RSA TLS, an attacker replays by pretending to be the client and sending out the same encrypted PS. Note that the client is the first party to send out the random nonce R_b, so there is no way for the client to adaptively choose R_b based on the server's nonce R_s. Therefore, $R_b \oplus R_s \oplus PS$ will be a new random; as a result, different keys will be derived.

(c) Let us assume that in RSA TLS, we make the following change. Rather than generating the PS randomly, the client begins with an initial value PS. For each TLS connection the client makes, it simply increments the PS as $PS \leftarrow PS + 1$.

TRUE or FALSE and Explain: This preserves the confidentiality of RSA TLS.

○ TRUE ● FALSE

◇ **Explain *concisely***:

_____

_____

Solution: Client can connect to attacker's server, then attacker can decrypt PS and derive the other values of PS used (in the past and future), then derive cipher keys for other connections. (3 points)

Problem 5 ***WPA2 Security*** **(13 points)**

Recall the WPA2 handshake protocol based on pre-shared keys, as follows:

- The pre-shared key (PSK) is computed from the passphrase (*i.e.*, the Wi-Fi password) and network SSID (*i.e.*, the name of the Wi-Fi network).
- Pairwise transient key (PTK) is determined from the ANonce, SNonce, and the pre-shared key (PSK) using a pseudorandom generator.
- The client derives the encryption and MIC keys from the PTK.

(a) TRUE or FALSE: The WPA2 protocol described above provides forward secrecy.

○ TRUE ● FALSE

You don't need to provide an explanation.

Solution: No. Everything but the PSK is public (ANonce and SNonce), so knowing the PSK allows an attacker to decrypt past messages. (3 points)

(b) Alice connected her computer to a Wi-Fi access point and accessed a few websites. An eavesdropper recorded the ANonce, SNonce, and all other messages in Alice's connection.

With the information collected above, the eavesdropper can brute-force the passphrase. We consider that at this moment, the eavesdropper guesses that P might be the correct passphrase.

◇ **Question:** How can the eavesdropper confirm whether or not P is the correct passphrase *without* connecting to the Wi-Fi access point? (answer concisely)

_____

_____

Solution: The eavesdropper can generate the MIC key and check whether the MIC generated is correct under this key. (3 points)

(c) A man-in-the-middle attacker knows the PSK of a Wi-Fi access point. Alice's laptop has been connected to this access point and is visiting `https://www.chase.com/`. Alice visited this website and logged in by providing her password.

◇ TRUE or FALSE: A man-in-the-middle attacker can see Alice's banking password.

○ TRUE ● FALSE

◇ **Explain *concisely*:** _____

Solution: HTTPS provides end-to-end confidentiality. (3 points)

(d) In WPA2 Enterprise the device authenticates to an authentication server over TLS to generate a unique PSK for the user before the WPA2 handshake. Yet, in the real world, users usually accept any certificate blindly since there is no notion of "name" (unlike for a web server) which the client can automatically validate.

◇ True or False: In this case, the protocol is secure against a man-in-the-middle attacker.

○ True ● False

◇ True or False: In this case, the protocol is secure against a passive eavesdropper.

● True ○ False

You don't need to provide an explanation.

Solution: 2 points each, 4 points in total

Problem 6 ***DNS*** **(14 points)**

(a) Injecting spoofed packets as an off-path attacker in TCP is much harder than in UDP. Even though TCP has a higher security guarantee, DNS often does not use TCP because TCP has a much higher *latency*.

◇ **Question:** Compared with UDP, what is the *chief* reason why using TCP for DNS has a higher latency? (answer within 10 words)

_____

Solution: TCP handshake adds a round trip. (3 points)

(b) Alice wants to access Berkeley's diversity advancement project DARE, `dare.berkeley.edu`. Her laptop connects to a wireless access point (AP).

Alice worries that a hacker attacks the DNS protocol when her laptop is looking for the IP address of `dare.berkeley.edu`. Assume that DNSSEC is not in use.

◇ **Question:** Which of the following can attack the DNS protocol and have Alice's browser obtain an incorrect IP address for DARE? (Select 0 to 8 options.)

- ■ The laptop's operating system.
- ■ The laptop's network interface controller.
- ■ The wireless access point.
- ■ An on-path attacker on the local network.
- ■ The local DNS resolver of the network.
- ■ The root DNS servers.
- ■ `berkeley.edu`'s DNS nameservers.
- ■ An on-path attacker between the local DNS resolver and the rest of the Internet.

Solution: 4 points. 0.5 points per marking.

(c) Now assume that `berkeley.edu` implements DNSSEC and Alice's recursive resolver (but not her client) validates DNSSEC.

◇ **Question:** Which of the following can attack the DNS protocol and have Alice's browser obtain an incorrect IP address for DARE? (Select 0 to 8 options.)

- ■ The laptop's operating system.
- ■ The laptop's network interface controller.
- ■ The wireless access point.
- ■ An on-path attacker on the local network.
- ■ The local DNS resolver of the network.
- ■ The root DNS servers.
- ■ `berkeley.edu`'s DNS nameservers.
- □ An on-path attacker between the local DNS resolver and the rest of the Internet.

Solution: Any on-path attacker can see or modify the DNS traffic. 4 points. 0.5 points per marking.

(d) An attacker wants to poison the local DNS resolver's cache using the Kaminsky attack. We assume that the resolver does not use source port randomization, so the attacker will likely succeed.

In the Kaminsky attack, the attacker asks the resolver for a *non-existing* subdomain of UC Berkeley, *e.g.*, `stanford.berkeley.edu`, instead of asking for an *existing* domain like `dare.berkeley.edu`.

◇ **Question:** What is the advantage of asking for a non-existent domain compared to asking for an existing domain? (answer within 10 words)

_____

_____

Solution: When you fail, you can keep trying with another nonexistant name/race until win!

(Note, caching alone is not sufficient, because you do have caching of NXDOMAIN too. The big thing is "race until win". (3 points))

Problem 7 ***Same-Origin Policy*** **(13 points)**

(a) TRUE or FALSE: Setting "secure" flag on a cookie protects it from a network attacker eavesdropping on an insecure HTTP connection.

● TRUE ○ FALSE

You don't need to provide an explanation.

Solution: True. The secure flag means that the cookie can only be transmitted through an HTTPS connection. (2 points)

(b) TRUE or FALSE: After a successful XSS attack, JavaScript can access all cookies set by the website it attacked.

○ TRUE ● FALSE

You don't need to provide an explanation.

Solution: False. With the 'HttpOnly' flag, the web server can prevent some cookies from being read by JavaScript. (2 points)

(c) Which of these URIs have the same origin as "`http://same.origin.com:80/a.htm`" according to same origin policy? (choose 0 to 4 options)

☐ `http://origin.com:80/a.htm` ■ `http://same.origin.com:80`

■ `http://same.origin.com:80/a.htm/b` ☐ `ftp://same.origin.com:80`

Solution: 0.75 per option. In total, 3 points.

(d) If a page loads a JavaScript file from some other site, this JavaScript file takes the origin of...

Choose one option:

● The page that loaded it ○ The site that hosts the JavaScript file

Solution: 3 points.

(e) Same-origin policy is very useful in preventing many web attacks. Yet, it also inconveniences for web developers – different domains cannot talk to each other.

◇ **Question:** Provide a *specific* solution for the web developers to *conveniently* enable JavaScript in different domains' webpages to *conveniently* talk to each other. (answer less than 10 words)

_____

Solution: Use JavaScript postMessage function. It is okay to write Postmessage, PostMessage, postmessage, postmsg, PostMsg, postMsg, or Postmsg. (3 points)

If the student says “use the backend”, no points because we mentioned “conveniently” in the question.

But, if the student just says "there is a function in JavaScript that allows ..." and makes sense, the student only gets 1 point, as the question said “specific solution”.

If the student answers “sendmsg”, “senddata”, no points.

Problem 8 ***HTTP TRACE method*** **(7 points)**

Web servers can support another type of HTTP requests, TRACE, as follows.

A TRACE request is like a GET request or a POST request. It simply has the web server echo back the HTTP request sent by the client. *Importantly*, this method will echo back the entire request, including all the cookies sent by the client.

Assume that neither web servers nor web browsers set any restriction preventing the use of TRACE requests. In particular, JavaScript can send a TRACE request and receive its response.[1]

(a) Mallory knows that `victim.com` is vulnerable to an XSS attack. Mallory also knows that this website stores a session cookie on the user's browser.

However, Mallory's injected JavaScript is unable to access that session cookie.

◇ **Question:** What is a likely reason for which Mallory's injected JavaScript code failed to access the session cookie? (answer in less than 10 words)

_____

Solution: The session cookie is protected by the "HttpOnly" flag. (2 points)

(b) Explain how Mallory's injected JavaScript can steal the cookie and send the cookie to Mallory's personal website `evil.com`.

You don't need to provide the specific JavaScript code; rather, you only need to provide the high-level ideas. Please arrange your answer in a few steps (no more than three steps).

◇ **Outline:** Mallory's injected JavaScript, which consists of no more than three steps:

1. _____

2. _____

3. _____

(Answer concisely, put only a single line of text above the dashes, and do not exceed the space.)

Solution: 1. send a TRACE request to some random webpage in `victim.com`.

2. read its response.

3. send the response to Mallory's personal website.

In total 5 points.

not using the TRACE request – no point.

[1] These days, browsers now all block JavaScript from sending TRACE requests because of this particular attack.

forgetting to send the response to Mallory's personal website – 3.5 points.

Problem 9 ***Firewalls and DDoS*** **(12 points)**

(a) TRUE or FALSE: A stateless firewall can block in-bound **TCP** connections on destination port 80 to devices on the internal network, while still allowing these devices to make out-bound connections using destination port 80.

● TRUE ○ FALSE

Solution:

(b) TRUE or FALSE: A stateless firewall can block in-bound **UDP** connections on destination port 53 to devices on the internal network, while still allowing these devices to make out-bound connections using destination port 53.

○ TRUE ● FALSE

(c) TRUE or FALSE: SYN flooding attacks can be effectively prevented by rate-limiting the number of TCP connections from a given IP address.

○ TRUE ● FALSE

Solution: 2 points each above.

(d) Consider the following implementation of SYN cookies. During the TCP handshake, the implementation sets the sequence number of the `SYN ACK` packet to be the first 32 bits of $\text{HMAC}_k(t)$, where t is the time rounded to the nearest second and k is a secret key known only by the server.

◇ **Question:** Explain why such a design of SYN cookies makes it easier for an off-path attacker to spoof TCP packets.

_____

_____

Solution: Off-path attackers could make connections to server, record sequence numbers used (3 points)

(e) What additional piece of information could you include in the MAC in order to fix the problem above?

◇ **Answer *concisely*:**

_____

Solution: Client IP address (also client port) (3 points)

Figure 1: An amazing XSS polyglot payload

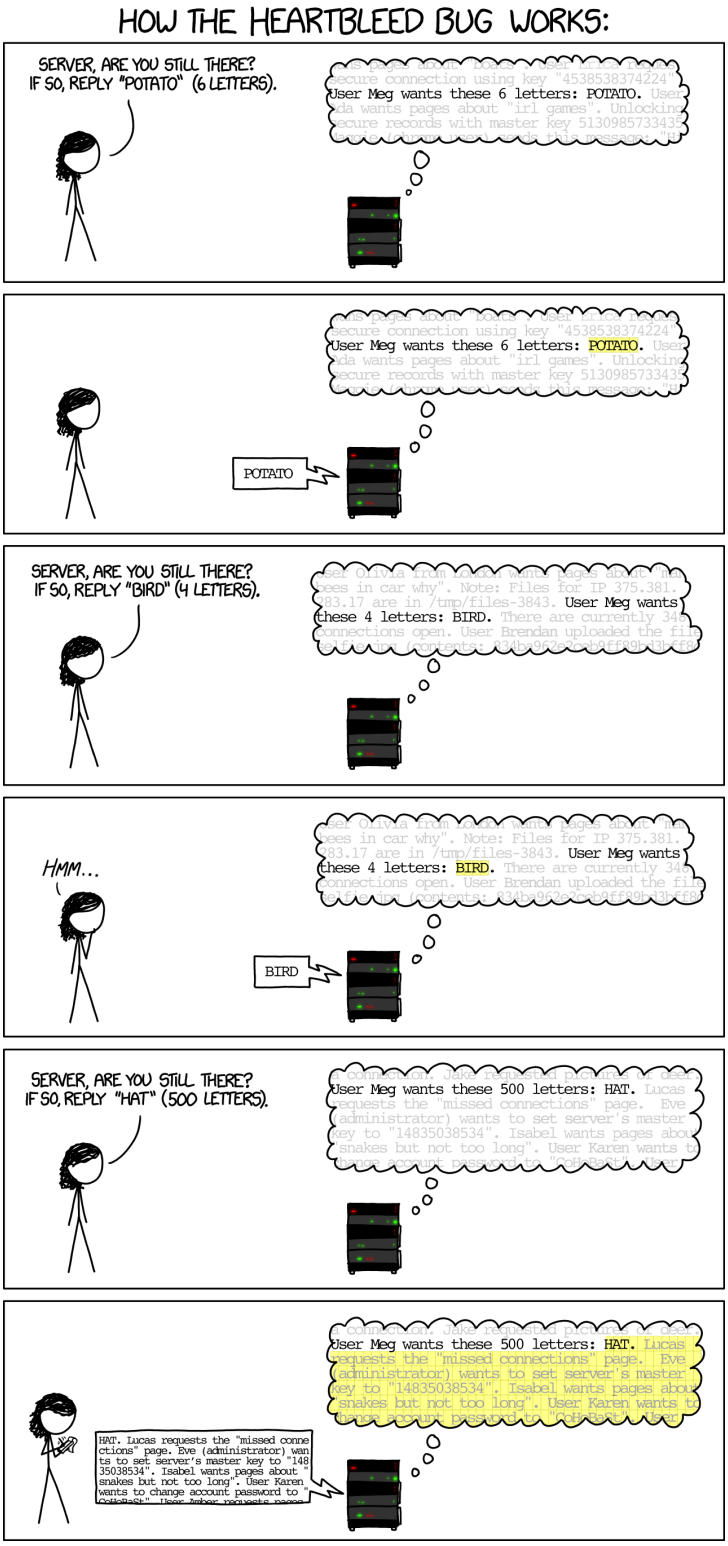

Figure 2: XKCD Explains Heartbleed