Computer Security

2018 Exam 3 Review

Paul Krzyzanowski

Rutgers University

Spring 2018

Grades

	Exam 1	Exam 2	Exam 3
Average	78.4	72.1	69.6
σ	12.3	9.9	12.3
Highest	97	95	96
Lowest	40	45	39
Top 10%	≥ 94	≥ 85	≥ 82
Top 20%	≥ 90	≥ 81	≥ 78
Bottom 10%	≤ 62	≤ 60	≤ 53
Bottom 20%	≤ 67	≤ 64	≤ 59

Approximate grades

A	83	77	74
B+	77	71	68
B	71	66	62
C+	65	61	56
C	58	56	50

Question 1

Which of the following is *not* an example of two-factor authentication?
(a) Password and a code sent to your phone via SMS.
(b) Access card and PIN.
(c) Fingerprint and a retina scan.
(d) All of these are examples of two-factor authentication.

Two-factor = two *different* factors

Fingerprint & retina scan both use biometrics

(a) Something you know (password) & something you have (phone)

(b) Something you have (card) & something you know (PIN)

Question 2

Salt in a hashed password:

(a) Makes it virtually impossible to use a brute-force search to guess a password.
(b) Obscures the ability to see that multiple users have the same password in a password file.
(c) Turns normal passwords into one-time passwords.
(d) Ensures that passwords have special characters in addition to alphanumeric text.

(a) No. The password remains unchanged. It makes it difficult to use a table of pre-computed passwords.

(b) Yes. Each user has a random salt, so hash(password + salt) will yield two different values even if the passwords match

(c) No.

(d) No.

1 point for (a)

Question 3

The *challenge-handshake authentication* (CHAP) protocol relies on
(a) Sending an encrypted password to the server.
(b) Showing that you can decrypt data sent by a server.
(c) Using a trusted third party to handle user authentication.
(d) Proving that you have a secret value that is shared with the server.

(a) Encrypted data of any kind is never sent.

(b) No. You're proving you can generate the same *f(challenge, key)*.

(c) No.

(d) Yes. Both sides generate *f(challenge, key)*

Question 4

The *Time-based One-Time Password* (TOTP) protocol:

(a) Relies on a shared secret between the client and server.
(b) Allows an administrator to control when a user can log in.
(c) Provides a time limit for the number of login attempts.
(d) Enables an administrator to set an expiration time for user passwords.

$$\text{password} := \textit{hash}(\text{secret\_key}, \text{time}) \ \% \ 10^{\text{password\_length}}$$

Both sides can generate the same password because they know the secret.

(b, c) This has nothing to do with the protocol.

(d) An expiration time for the shared secret needs to be managed outside of the protocol. Time is granular to 30 seconds – and can be changed in some implementations but that's a *granularity*, not an *expiration time*.

2 points for (d)

Question 5

A list of *hashes* is often used in an application signature to:

(a) Enable the user to pinpoint what modifications have been made to the application.

(b) Validate the integrity of the software even if some of the hashes are maliciously modified.

(c) Allow an operating system to check the integrity of the software as pieces of it are loaded into memory.

(d) Validate different parts of an application: code, data, stack, heap.

(a) This can be done but is never used that way. A user's system simply cares that an app has been compromised.

(b) No. The hashes are not redundant

(c) Yes – per-page hashing avoids the need to scan the entire app before executing it.

(d) No. Only static components are hashed.

Question 6

Biometric authentication algorithmically differs from other forms of authentication because it:

(a) Compares images instead of passwords.
(b) Uses data that cannot be shared.
(c) Relies on thresholds rather than exact matches.
(d) Provides a far higher degree of security.

(a) Not necessarily – depends on the biometric
Also, images are pre-processed (e.g., identify minutia on fingerprint)

(b) Biometric data can be stolen but cannot be easily shared
– *but this is not an algorithmic differentiation!*

(c) Yes – fuzzy matches

(d) No.

1 point for (b)

Question 7

A problem with *CAPTCHA* is that:

(a) Computer vision algorithms have been improving rapidly.
(b) It is easy for an attacker to try various combinations of text.
(c) Results are not sent over a secure link.
(d) The answer can be found directly in the JavaScript code of the challenge.

(a) Yes. ML-based vision algorithms can reach levels of human competence

(b) No. Need to resubmit a page each time, which will cause a new CAPTCH to be generated

(c) Depends on the implementation – they should be but it doesn't matter – they're single use

(d) No

Question 8

No CAPTCHA reCAPTCHA is a variation of CAPTCHA that:

(a) Presents a puzzle to solve instead of text to decode.

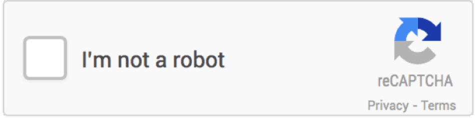
(b) Looks at a user's activity on the web page and server-side data to decide if it is a human.

(c) Provides alternatives to users, such as recognizing images or transcribing audio.

(d) Allows scripts as well as humans to interact with websites.

No CAPTCHA reCAPTCHA:

asks users to check a box stating that *I'm not a robot*



(a) No. The puzzle (text or other) is a fallback in case No CAPTCHA fails.

(b) Yes – server-side reputation management + JavaScript metrics

(c) No.

(d) No - that would defeat the point of CAPTCHA!

Question 9

A CAM (Content Addressable Memory) *overflow attack* on an Ethernet switch requires sending Ethernet frames:
(a) With varying fake source addresses.
(b) To a huge set of destination addresses.
(c) At a rate faster than the switch can process.
(d) That are malformed to indicate a length longer than the actual payload.

(a) Yes. The CAM table stores a mapping of
[MAC address] → [switch port]
A large # of source MAC addresses will overflow this table

(b) No – destination addresses don't affect the table

(c) No - that might cause the switch to drop frames but not overflow the CAM table

(d) No

Question 10

A key problem with both *ARP* and *DHCP* is that:
(a) Neither queries nor responses are encrypted.
(b) Clients have no way of authenticating themselves.
(c) A client might send a message to the wrong server.
(d) A client has no way of knowing who the authoritative server is.

(a) True, but that's not the key problem with the protocols

(b) No.
- ARP doesn't give out secret data
- *Authorization* may be useful before allowing a system to join your network and that's done at another time with other protocols

(c) Clients broadcast ARP and DHCP requests.

(d) Right. A client has no way of knowing if a response comes from a legitimate server or an imposter.

Question 11

ARP cache poisoning attacks can be reduced by:

(a) Configuring a switch to disallow ARP responses from systems not designated as ARP servers.

(b) Ignoring responses that are not associated with your request.

(c) Requiring responses to be signed.

(d) First establishing an encrypted channel to the server.

(a) There are no ARP servers
Every system is an "ARP server" – answering queries for its own IP address

(b) A system often accept *gratuitous ARP* messages – responses it sees on the network not associated with any request it made – to pre-populate its ARP cache

(c) No key infrastructure in place for validating this (+performance)

(d) How would the system know what server to connect to?

Question 12

SYN cookies were designed to:
(a) Provide a way for a client to authenticate a server.
(b) Create a shared secret between the client and server to encrypt traffic.
(c) Provide a time limit for establishing a TCP connection.
(d) Reduce the amount of state that a server sets up before finalizing a TCP connection.

(a) No. There is no authentication in a TCP connection setup

(b) No.

(c) No.

(d) Yes. The server delays allocating TCP state upon receiving a SYN
SYN/ACK sets a sequence number = f(secret#), which the client does not know
ACK from client must contain that #+1 for the server.
Server ensures it's talking with a client before allocating memory.

Question 13

The *Border Gateway Protocol*, *BGP*, is used to share routing information among ISPs. A security weakness with this protocol is:

(a) Hosts can bypass its advertisements and use alternate routes.
(b) An ISP can maliciously advertise better routes to divert traffic.
(c) It allows an attacker to impersonate an arbitrary host on the network.
(d) ISP routers that lose a shared key will not be able to communicate to external networks.

(a) Not really. An admin can always configure routes but that's by design.

(b) Yes. Malicious BGP messages can reroute traffic to other ASes (ISPs)

(c) Not directly. An ISP would need to know how to route the re-routed traffic to a malicious host

(d) There are no keys.

1 point for (c)

Question 14

Network tunneling is best described as:
(a) Sending a stream of packets over an encrypted communication channel.
(b) Relaying messages via a trusted third party.
(c) Signing all messages between two communicating hosts.
(d) Encapsulating one packet within another.

(a) Encryption is not necessary for tunneling

(b) No third parties are involved

(c) Signing is not necessary for tunneling

(d) Yes – tunneling is simply encapsulating other packets

Question 15

Unlike IPsec with the Encapsulating Security Payload, *SSL* and *TLS*:
(a) Encrypt messages in both directions.
(b) Are designed for point-to-point connections over TCP.
(c) Use a MAC to ensure message integrity.
(d) Rely on a trusted third party.

(a) So does IPsec/ESP

(b) Yes – IPsec is below the transport layer – over IP

(c) So does IPsec/ESP

(d) No third parties are involved.

Question 16

A *stateless screening router* is unlikely to be able to be configured to drop:

(a) TCP packets addressed to your mail server computer but accessing port 80 (HTTP).

(b) Any UDP packets from a set of IP addresses known to be untrusted.

(c) TCP packets to your web server that contain URLs with malicious syntax.

(d) UDP packets from the external network that are disguised with internal source addresses.

(a) Drop any packet where
protocol=TCP, dest_addr=10.11.12.13, port=80

(b) Drop any packet where
protocol=UDP, interface=external, source_addr=128.6.0.0/16

(c) URLs in HTTP requests will be in the data – requires deep packet inspection – may not even be in the first packet

(d) Drop any packets where
protocol=UDP, interface=external, source_addr=192.168.0.0/24

Question 17

A DMZ (demilitarized zone) is a subnet that contains:
(a) Systems offering Internet-facing services.
(b) No computers but acts as a barrier between the LAN and Internet.
(c) Internal hosts that may not be properly secured.
(d) Known malicious systems.

(a) The DMZ is a protected subnet for externally-facing services

(b) It has computers in it.

(c) No. That's the internal network.

(d) No.

Question 18

Signature-based intrusion detection systems (IDS):

(a) Scan incoming data to see if it matches known malicious patterns.
(b) Validate messages bidirectionally to ensure they conform to the right protocol.
(c) Detect deviations in network activity from known normal behavior.
(d) Drop all unsigned messages coming into the local network and add signatures to messages leaving the local network.

(a) Yes

(b) That's a protocol-based IDS

(c) That's an anomaly-based IDS

(d) This doesn't make sense

Question 19

Deperimiterization creates a problem in network security because:

(a) One system may run a virtual machine (VM) and host multiple operating systems.

(b) A single operating system may host secure and non-secure services.

(c) Trusted hosts are not confined to specific known networks.

(d) Network traffic may be seen by malicious parties.

(a) Not a problem … unless the VMs are outside of a network that can be protected

(b) Bad engineering … but that's not deperimiterization

(c) Hosts may move around: mobile devices, AWS services communicating with Azure services, …

(d) Applications can encrypt their traffic

1 point for (d)

Question 20

Denial of Service (DoS) *amplification* techniques rely on exploiting services where:

(a) Queries get forwarded to a larger number of hosts.

(b) Critical systems are taken out of service, causing systems that rely on them to die.

(c) Small queries generate large responses.

(d) Malware can infiltrate other systems to make them to participate in the attack.

(a) That's not DoS amplification.

(b) No. That's just DoS.

(c) Yes.
E.g., DNS query vs. response

(d) That's using malware to build a DDoS botnet

Question 21

Which statement is most accurate about Bitcoin?

(a) Each participant keeps a copy of all transactions since the beginning.
(b) Participants only keep a copy of uncommitted transactions.
(c) Each participant keeps a different portion of the ledger (transaction log).
(d) One server holds the master copy of the ledger but participants may cache recently used blocks.

(a) Yes. Each participating system keeps a copy of the entire blockchain so it can verify transactions.

(b) No.

(c) No.

(d) There is no master copy and no master server.

Question 22

In Bitcoin, a *proof of work* is performed to:

(a) Prove that a transaction has not been forged.
(b) Make it computationally extremely difficult to modify a block.
(c) Validate that the sender has sufficient coins for the transaction.
(d) Log the fact that a certain number of bitcoins have been created.

(a) Individual transactions are signed. Bob cannot create a transaction on Alice's behalf

(b) Yes. A malicious participant may try to delete Alice's transaction but that would require re-computing the proof of work for all blocks going back to that transaction

(c) No. That just involves traversing the list of transactions

(d) That's a side-effect – the reward for doing the proof of work. The logging of new coins occurs after 99 blocks have been added.

1 point for (d)

Question 23

A transaction is considered *confirmed* by a merchant:
(a) After a majority of participants approve the transaction.
(b) When the block that contains the transaction is added to the blockchain.
(c) After at least one participant approves it.
(d) After a certain number of additional blocks are added to the blockchain.

(a) A majority of participants do not need to approve the transaction. Each participant adds it to the list when the participant decides the transaction is valid.

(b) Not necessarily.

(c) No.

(d) Yes – typically 1 block for small transactions, 3 blocks for deposits and mid-size payments, and 6 blocks for large payments.

- To modify the past, you'd need to recompute proof of work #s for past blocks to reconstruct the longest chain
- 51% attack: to do this, you need >50% computing power of all participants

Question 24

If a web client at cs.rutgers.edu loads a web page from pk.org that downloads JavaScript from github.com, the JavaScript code on the page can access content (e.g., cookies) belonging to:

(a) pk.org
(b) github.com
(c) cs.rutgers.edu
(d) All of the above.

(a) Scripts follow the same-orgin policy.

JavaScript code executes with the authority of its frame's origin

If cnn.com loads JavaScript from jQuery.com, the script runs with the authority of cnn.com

Question 25

Typo in exam: *request* forgery, not *resource* forgery

Cross-site request forgery (*XSRF*) is a problem that occurs when:

(a) JavaScript on one page can access resources from a different site.

(b) A user clicks a maliciously placed link containing a command to a site that identifies the user via cookies.

(c) A server masquerades as another web site.

(d) A server presents cookies that are labeled for another site.

XSRF targets: sites where

- Site sets cookies that authenticate a user
- User requests are sent via the URL
  bank.com/transfer.jsp?amount=10000&from=202164&to=593144

- (a) No.
- (c) No. Any malicious link will suffice.
- (d) No. Cookies are sent to the correct site.

Question 26

Cross-site scripting (*XSS*) is an attack that allows an attacker to:
(a) Run JavaScript hosted from a different server than the web page.
(b) Run a script on a web page that accesses resources on a different site.
(c) Add JavaScript to a trusted web site.
(d) Run a script that replaces links on a page to point to malicious sites.

XSS is a code injection attack

A website allows user input in its pages and renders it as HTML

May be part of URL and the site will incorporate the arguments in its response

… or may be entered onto the page – e.g., forum responses

(a) No. The JavaScript gets run on the target web server.

(b) Not necessarily. It may just as easily do something directly on the site.

(c) Yes.

(d) Highly unlikely.

Question 27

Extended validation certificates are considered more secure than domain validated certificates because:

(a) They force a session to be established that is encrypted in both directions.
(b) They require two-factor authentication to establish a connection.
(c) The user has to authenticate with a password after an SSL session is established.
(d) The CA puts extra effort into validating the identity of the certificate holder.

(a) No. That's up to the site configuration – few expect clients to have certificates.

(b) No.

(c) No. That has nothing to do with EV certificates.

Question 28

The main mechanism that Android uses to *isolate applications* is:

(a) User IDs.
(b) Containers.
(c) Namespaces.
(d) Kernel-level sandboxes.

Unique Linux (Android) User IDs are assigned to each application.

(a) No containers are user.

(b) No namespaces are used – just directory permissions.

(c) No kernel-level sandboxing.
The Dalvik sandbox is used for the Dalvik VM but not for native code.

Question 29

The main mechanism that iOS uses to *isolate applications* is:

(a) User IDs.
(b) Containers.
(c) Namespaces.
(d) Kernel-level sandboxes.

Kernel-level sandboxing – essentially the same as in macOS – configures filename patterns, network access, privileged calls

Question 30

ARM's *TrustZone*:

(a) Uses hardware to speed up encryption, decryption, hashing, and key generation operations.

(b) Runs a separate operating system in isolated memory for security-sensitive features.

(c) Is a region of protected memory that is accessible only to privileged applications.

(d) Is a set of flags in the memory management unit to assign regions of memory to an application.

(a) Yes but that's in place with or without Trustzone

(b) Yes – separate execution environment
Protected memory, separate registers.

(c) No – Trustzone offers protected memory but it's accessible only to the code executing in Trustzone, not privileged apps under the main OS

(d) No.

Question 31

A DVD contains an encrypted movie. The decryption key is:

(a) Programmed into the player.

(b) Encrypted on the DVD with a master key that the player knows.

(c) Encrypted on the DVD via each of 409 player keys for various trusted manufacturers of DVD players.

(d) Obtained from a trusted server prior to playing the DVD.

Each movie is encrypted with a unique key.

Each family of players contains a unique key.

The movie key is encrypted with each of the player keys.

Question 32

A *null cipher*:
(a) Signs messages but does not encrypt them.
(b) Is a stubbed-out encryption function that performs no actual encryption.
(c) Encrypts with a key of all 0s with the hope that the adversary doesn't realize there is encrypted text present.
(d) Intermixes plaintext with non-relevant text.

It's not an encryption algorithm.

A null cipher hides the message within irrelevant data. You need to know where to look.

Question 33

Chaffing and winnowing:

(a) Shifts characters in text slightly to create a steganographic bit pattern.
(b) Encrypts data with a null cipher (chaffing) that a trusted receiver then decrypts (winnowing).
(c) Adds concealed data inside an image or audio file.
(d) Intermixes legitimate messages with proper MACs with non-relevant messages with invalid signatures.

Some messages are valid and some are not.

Each message is signed (e.g., HMAC)

Valid messages have a valid signature.

Invalid messages do not have a valid signature.

An eavesdropper cannot check.

The end