

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)

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)

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.

password := hash(secret_key, time) % 10password_length

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

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

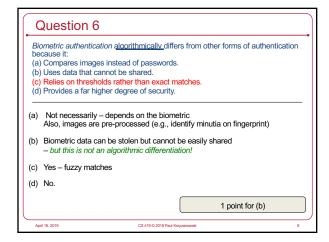
(d) An expiration time for the shared secret needs to be managed outside of the protocol. Time is granular to 30 seconds – and can be changed in some implementations but that's a granularity, not an expiration time.

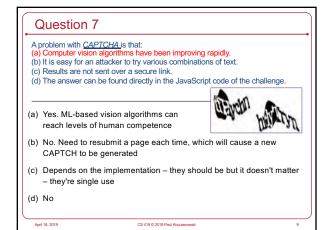
2 points for (d)

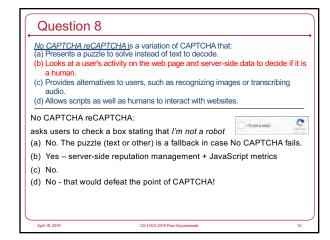
A list of <u>hashes</u> is often used in an application signature to:
(a) Enable the user to pinpoint what modifications have been made to the application.
(b) Validate the integrity of the software even if some of the hashes are maliciously modified.
(c) Allow an operating system to check the integrity of the software as pieces of it are loaded into memory.
(d) Validate different parts of an application: code, data, stack, heap.

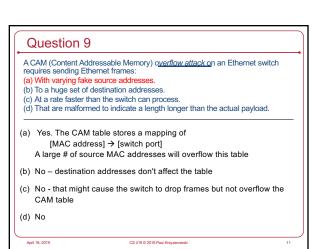
(a) This can be done but is never used that way. A user's system simply cares that an app has been compromised.
(b) No. The hashes are not redundant
(c) Yes – per-page hashing avoids the need to scan the entire app before executing it.
(d) No. Only static components are hashed.

CS 419 4/18/19

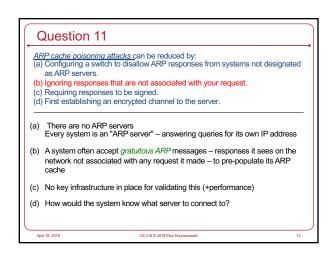








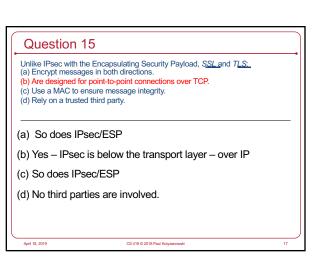
A key problem with both ARP and DHCP is that: (a) Neither queries nor responses are encrypted. (b) Clients have no way of authenticating themselves. (c) A client might send a message to the wrong server. (d) A client has no way of knowing who the authoritative server is. (a) True, but that's not the 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.



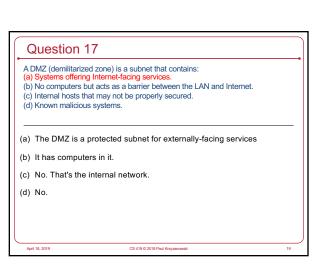
Question 12 SYN cookies were designed to: (a) Provide a way for a client to authenticate a server. (b) Create a shared secret between the client and server to encrypt traffic. (c) Provide a time limit for establishing a TCP connection. (d) Reduce the amount of state that a server sets up before finalizing 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.

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.

Question 14 Network tunnelina 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



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 16

CS 419 4/18/19

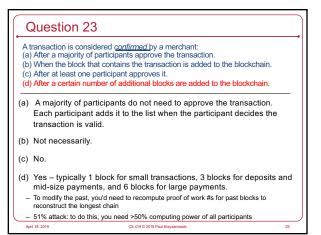
Question 18 Signature-based intrusion detection systems (IDS): (a) Scan incoming data to see if it matches known malicious patterns. (b) Validate messages bidirectionally to ensure they conform to the right protocol. (c) Detect deviations in network activity from known normal behavior. (d) Drop all unsigned 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

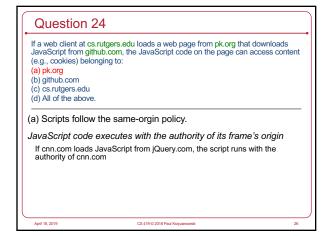
Question 19 Deperimiterization creates a problem in network security because: (a) One system may run a virtual machine (VM) and host multiple operating systems. (b) A single operating system may host secure and non-secure services. (c) Trusted hosts are not confined to specific known networks. (d) Network 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

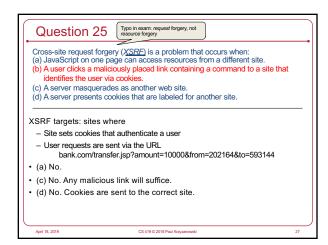
Question 20 Denial of Service (DoS) amplification techniques rely on exploiting services where: (a) Queries get forwarded to a larger number of hosts. (b) Critical systems are taken out of service, causing systems that rely on them to die. (c) Small queries generate large responses. (d) Malware can infiltrate 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

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.

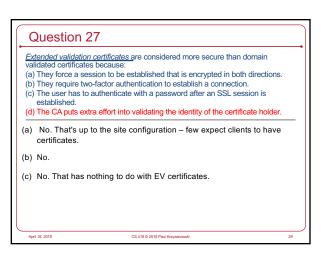
| In Bitcoin, a proof of work is performed to: (a) Prove that a transaction has not been forged. (b) Make it computationally extremely difficult to modify a block. (c) Validate that the sender has sufficient coins for the transaction. (d) Log the fact that a certain number of bitcoins have been created. (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. Apt 18,209 CS 419 6 2018 Pad Kryparwoods 24



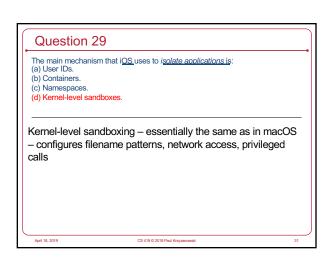




Cross-site scripting (XSS) is an attack that allows an attacker to: (a) Run JavaScript hosted from a different server than the web page. (b) Run a script on a web page that accesses resources on a different site. (c) Add JavaScript to a trusted web site. (d) Run a script that replaces links on a page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



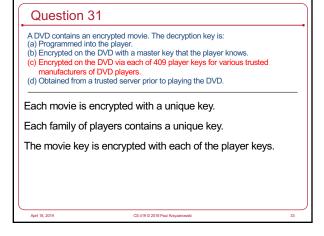
Question 28 The main mechanism that Android uses to isolate applications is: (a) User IDs. (b) Containers. (c) Namespaces. (d) Kernel-level sandboxes. Unique Linux (Android) User IDs are assigned to each application. (a) No containers are user. (b) No namespaces are used – just directory permissions. (c) No kernel-level sandboxing. The Dalvik sandbox is used for the Dalvik VM but not for native code.



4/18/19

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. April 10. 2019 CS 419 0 2018 Paul Kozyamowski 22

CS 419



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.

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

April 13.2019 CS419-0.2018-Pail (Roccarroscold) 36