Computer Security

10. Exam 2 Review

Paul Krzyzanowski

Rutgers University

Spring 2017

Question 1(a)

Suppose you come across some old text in the form

```
GEPPQ IMWLQ EIPWS QICIE VWEKS RIZIV QMRHL SAPSR KTVIG MWIPC LEZMR …
```

Also suppose you know it is English text and it is encrypted. How would you determine if the text was enciphered with a transposition or a substitution cipher?

- Perform a frequency analysis
- If the frequencies match those of English text then it is a transposition cipher

Question 1(b)

Suppose you come across some old text in the form

GEPPQ IMWLQ EIPWS QICIE VWEKS RIZIV QMRHL SAPSR KTVIG MWIPC LEZMR …

Suppose you decide you are looking at a substitution cipher. How would you determine if the text was encrypted is a monoalphabetic or polyalphabetic substitution cipher?

- Look at the frequency analysis
- If the frequencies match those of English text but for the wrong letters, then you have a monoalphabetic substitution cipher
- If the frequencies are close to uniform then you have a polyalphabetic substitution cipher

Question 2

What does it mean for a cryptographic hash to be collision-free?

- When people refer to a collision-free cryptographic hash, they mean it is *collision resistant*
- It is hard to find two messages M_1 & M_2 such that $H(M_1) = H(M_2)$

Pigeonhole Principle

If *n* items are put into *m* containers, with *n* > *m*, then at least one container must contain more than one item

- A hash is a fixed number of bits, which can hold a finite set of values
- You can have a larger, potentially infinite, set of inputs

Question 3

Why is the one-time pad not widely used even if it offers perfect secrecy?

- The key must be as long as the message and never reused
- The problem of secure communication has been replaced with the problem of secure key transmission

Question 4

If Alice has Bob's certificate (assume she has validated it already), how can she convince herself that she is talking to Bob?

Have Bob prove that he has the corresponding private key

Option 1

- Alice generates a random string (nonce)
- She asks Bob to encrypt it with his private key & send the results
- She decrypts the result using the public key in Bob's certificate

Option 2

- Alice generates a random string (nonce)
- She encrypts it with the public key in Bob's certificate
- She sends it to Bob & asks him to decrypt the message with his private key and send her the results

If the resulting message = original nonce, it's Bob

Question 5

A security advantage of using *containers* over using a combination of *cgroups*, *namespaces*, and *capabilities* is:

(a) Simpler configuration avoids comprehension errors.

(b) Containers provide better isolation.

(c) Containers are a form of sandboxing.

(d) Containers do not require administrative privileges.

Many containers are built on top of control groups, namespaces, and capabilities

(b) Namespaces provide great isolation

(c) Containers aren't any more a form of sandboxing than capabilities and namespaces

(d) They don't to run apps but neither to namespaces & cgroups. They require root privileges to install

Question 6

The basic mechanism that most sandboxes that handle native code rely on is :
(a) Interposing system calls.
(b) Capabilities.
(c) Namespaces.
(d) Control groups.

Intercept security-sensitive system calls (e.g., file, network operations) & validate them

Question 7

What can a *sandbox* do that namespaces, capabilities, and control groups cannot?

(a) Restrict file access based on if the file name matches a pattern.

(b) Disallow access to certain administrative system calls, such as setting the system time.

(c) Disallow an application from seeing any files or directories outside of one directory.

(d) Restrict the amount of disk space that an application uses.

Most sandboxes enable inspection & validation of filenames

(b) Capabilities do this

(c) Namespaces do this

(d) Control groups do this

Question 8

The *Java sandbox* does not include a :
(a) Container to isolate native methods called via the JNI (Java Native Interface).
(b) Security manager to ensure resource access conforms to policies.
(c) Class loader to restrict which classes can be loaded or overwritten.
(d) Bytecode verifier to validate Java bytecodes prior to execution.

You're on your own if you use JNI

Question 9

Most *virtual machine monitors* take this approach:

(a) examine and arbitrate: don't interfere with execution but lock access to a resource if an OS is using it.

(b) scan and translate: scans the instruction stream for privileged instructions and translates them to safe alternatives.

(c) replicated resources: allows privileged instructions to operate directly on replicas of shared resources.

(d) trap and emulate: emulate the operation of privileged instructions on shared resources.

Privileged operations trap to the VMM that then emulates their operation for that specific OS.

Question 10

A covert channel can sometimes be established between two virtual machines on one system because:
(a) Local memory copies can be used instead of an external network.
(b) A network connection can be set up that loops back to the same hardware.
(c) All traffic can be encrypted on any communication link.
(d) Activity on one virtual machine may affect system performance on another

---

If one system can influence something in another one, a process in a secured system may leak information to an unsecure one even if it is not permitted to do so.

Question 11

A distinction between a *worm* and a *virus* is that:

(a) Viruses are designed to replicate themselves and worms do not.

(b) Viruses are malicious while worms are benign.

(c) Worms are stand-alone programs and do not need to propagate via files or documents.

(d) Worms can hide inside an executable program while viruses hide in documents.

Question 12

When compared with regular viruses, a unique danger of *boot sector malware* is that it:

(a) Replicates onto other systems without human intervention.
(b) Can never be detected.
(c) Runs before the operating system can run any anti-malware software.
(d) Causes an alternate operating system to boot.

(a) Regular viruses do this.

(b) Sure it can: read the boot sector

(c) It can undo any patches anti-malware software put in place

(d) Unlikely unless one is installed … and the user will notice that the wrong OS is running.

Question 13

A program that allows an attacker to access a computer while hiding its presence is called a:
(a) Rootkit.
(b) Backdoor.
(c) Trojan horse.
(d) Masqueraded authenticator.

The key point is ***hiding its presence***.

Rootkits are programs that hide their presence – and often provide hackers with privileged access to a system

Question 14

A program that runs a useful task while also performing harm is a:
(a) Virus.
(b) Macro virus.
(c) Worm.
(d) Trojan horse.

A Trojan horse is a program with two purposes
- An overt purpose: the reason you install it
- A covert purpose: what it does behind the scenes (the malicious part)

In some cases, the overt purpose might be non-functional but it will be too late by the time the user realizes it

Question 15

This type of malware may run when a user opens a spreadsheet:
(a) Email virus.
(b) Macro virus.
(c) Worm.
(d) Trojan horse.

Spreadsheets & documents may contain embedded macros that run when the document is opened

Question 16

A *backdoor* is:

(a) A way to bypass the standard authentication mechanisms of software.

(b) A virus that allows an attacker to log into a computer.

(c) A covert communication channel that malware can use to communicate.

(d) A process that cannot be detected by the operating system.

Question 17

A *hypervisor rootkit* will:
(a) Embed itself within the operating system and bypass authentication requests.
(b) Install programs that enable an adversary to log in with administrative privileges.
(c) Run underneath the operating system to detect and log events of interest.
(d) Force the system to boot an alternate hacked version of the operating system.

A hypervisor is the lowest-level software, operating between the hardware and the operating system

- The OS does not know of its existence
- The hypervisor can intercept privileged instructions and all system interrupts

Question 18

Spear phishing differs from phishing attacks because:
(a) It uses pneumatic-powered spear guns.
(b) It is delivered via email rather than malicious web pages.
(c) It is delivered via malicious web pages rather than email.
(d) It is personalized to an individual target.

Question 19

A *virus signature* is:
(a) A hash of the code that makes up the virus.
(b) A portion of the code that makes up the virus.
(c) An encrypted hash of the virus code.
(d) Data that identifies the author of the virus.

Virus signatures have nothing to do with digital signatures. They're just a subset of code used for *signature scanning* by anti-malware software

Question 20

Kerckhoff's Principle states that:

(a) A cryptosystem should be secure even if everything except the key is public knowledge.

(b) To maximize security, the cryptographic algorithm should be shared with as few people as possible.

(c) True cryptographic security is unattainable in practice.

(d) The security of a system is exponentially proportional to the length of the key.

Public algorithms, private keys

Question 21

Suppose that you can crack a 56-bit key in one day. How long would it take to crack a 112-bit key?
(a) Two days.
(b) 56 days.
(c) 3,136 days.
(d) 197 trillion years.

You don't need to do exact math here – you have 56 more bits
Each bit doubles the search time.

56-bit key = 1 day ⇒ 57-bit key = 2 days

1 extra bit ⇒ ×2; 2 extra bits ⇒ ×4; 3 extra bits ⇒ ×8

10 extra bits ⇒ $\times 2^{10} = \times 1024$ days

20 extra bits ⇒ $\times 2^{20} = \times$ ~1 million days ⇒ *this is way more than 3,136*

30 extra bits ⇒ $\times 2^{30} = \times$ ~1 billion days

56 extra bits ⇒ $\times 2^{56} = 7.2 \times 10^{16}$ days $= 7.2 \times 10^{16}$ days $= 197 \times 10^{12}$ years

Question 22

The ciphertext of each block is a function of all previous plaintext blocks for the message with this mode:
(a) Electronic codebook (ECB).
(b) Cipher block chaining (CBC).
(c) Counter (CTR).
(d) None of the above.

(a) ECB = straight block-by-block encryption
(b) CBC =
new block = E_K(plaintext $\oplus$ previous ciphertext)
(c) CTR =
new block = E_K(counter) $\oplus$ plaintext

Question 23

Which algorithm does not rely on one-way functions?
(a) RSA.
(b) AES.
(c) Diffie-Hellman.
(d) SHA-2.

(a) RSA is based on the difficulty of factoring large products of primes. The modulus, $n = pq$

(b) AES is just a set of substitutions & permutations

(c) Diffie-Hellman is based on the discrete log problem

(d) Hash functions are irreversible and hence one-way

Question 24

To send a message to Bob, Alice would encrypt the message with:
(a) Alice's private key.
(b) Alice's public key.
(c) Bob's private key.
(d) Bob's public key.

She has to encrypt it in such a way that only Bob can decrypt it.

The only thing Bob has that nobody else does is his private key

Question 25

A *hybrid cryptosystem* uses:

(a) Different algorithms for each direction of data transmission.

(b) A public key algorithm to transmit a key and a symmetric algorithm for the data.

(c) Two levels of encryption for increased security: data encrypted with a symmetric algorithm is then encrypted with a public key algorithm.

(d) A symmetric algorithm to transmit the data and a public key algorithm to transmit a hash of the data.

Hybrid cryptosystem:

Public key cryptography for transmitting a key

Symmetric cryptography for communication

Question 26

What was the main problem discovered with the Needham-Schroeder protocol (assume Alice talks to Bob)?

(a) It relied on RSA keys that were not long enough to be secure.

(b) Anybody can impersonate the server, Bob, and obtain a session key.

(c) Its use of timestamps enables attacks on time synchronization.

(d) Another client who decrypted an earlier session key can impersonate Alice.

It was vulnerable to replay attacks that would allow an attacker to replay the initial communication that tells Bob what his session key is

Question 27

When Alice receives a Kerberos ticket to talk to Bob, it can be decrypted:
(a) Only by Alice & Kerberos.
(b) Only by Bob & Kerberos.
(c) Only by Alice, Bob, & Kerberos.
(d) Only by Alice and Bob.

Kerberos has all the keys
The *ticket* is the *sealed envelope* that Alice cannot decode

Question 28

What is a role of *Certification Authorities* (CA)?

(a) To establish a shared secret key between two parties.

(b) To relay messages securely.

(c) To bind a public key to a specific user or service.

(d) To distribute public/private key pairs.

Question 29

A *block cipher based MAC* (CBC-MAC) is:

(a) A message hash encrypted with a symmetric algorithm using cipher block chaining: $E_K(H(M))$.

(b) The last block of a message encrypted with a symmetric algorithm using cipher block chaining.

(c) A hash of a message that was encrypted with a symmetric algorithm using cipher block chaining: $H(E_K(M))$.

(d) A series of block-level hashes, with the output of each hash XORed with the next block of text.

Question 30

For Alice to *sign* a message for Bob, she would encrypt the message with:
(a) Alice's private key.
(b) Alice's public key.
(c) Bob's private key.
(d) Bob's public key.

Alice has to do something that nobody else can do.

Question 31

Salt in a password hash:
(a) Guards against dictionary attacks.
(b) Encrypts the password in the password file.
(c) Guards against using precomputed hashes.
(d) Speeds up password checking by storing a hash of the password in the password file.

The end