Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.033 Computer Systems Engineering: Spring 2010

Quiz III Solutions

I Multiple-Choice Questions

1. [6 points]: With respect to the paper "The Recovery Manager of the System R Database Manager" by Gray *et al.* mark each of the following statements true or false.

(Circle True or False for each choice.)

A. True / False Before modifying a "shadowed" file, the entire file is copied, and only the "current" version is modified: the shadowed version is not changed.

Answer: False. Only the modified pages are copied.

B. True / False Before any modified pages can be written to disk, the COMMIT log record for the transaction must be forced to disk.

Answer: False. Dirty records can be written to disk, since the log contains sufficient information to UNDO those operations.

C. True / False After a checkpoint is written to disk, System R discards all log records that precede the checkpoint record.

Answer: False. The log must contain all records for any active transactions. If a transaction that was started before the checkpoint is still running, the log can only be truncated up to that point.

D. True / False After saving a shadowed file (making the current version the new shadow version), the old shadow versions of the modified pages can be safely marked as free and reused.

Answer: True. These pages are no longer referenced by the new shadow, and if the shadow file is saved correctly, they will never be used by recovery.

2. [6 points]: In class we learned that DNS is an example of an eventually consistent system. Which of the following statements about DNS are true?

(Circle True or False for each choice.)

A. True / False If DNS is initially configured to resolve name N to IP address A, and is later reconfigured to resolve N to IP address B, clients looking up N after this reconfiguration may continue to receive A as an answer for lookups of N.

Answer: True. DNS may continue serving the old answer for a while from caches or from secondary nameservers that have not updated from their primaries.

B. True / False When there are no network partitions, DNS lookups see changes to DNS records immediately.

Answer: False. Same reasons as the previous answer.

C. True / False When consistency has been achieved, a given DNS name resolves to exactly one IP address.

Answer: False. A given name can have multiple DNS IP address records.

3. [6 points]: Indicate which of the following statements about Ross Anderson's paper "Why Cryptosystems Fail" are true.

(Circle True or False for each choice.)

A. True / False Anderson argues that discussing security failures openly improves security.

Answer: True. His position is that system designers must be able to learn from previous failures, in order to fix them.

B. True / False Most of the security failures described are a result of compromised cryptographic protocols.

Answer: False. Most of the failures are caused by people bypassing the security technology.

C. True / False Anderson suggests that system designers must consider the operation of the equipment as part of the security of the system.

Answer: True. Since many failures in this paper are caused by people using a "secure" system incorrectly, operating the system is a critical part.

D. True / False The "dual control" concept, where two individuals must collaborate to perform a function, is inconvenient and is sometimes bypassed by people wanting to save time.

Answer: True. While requiring two individuals to collaborate can provide good security, since both must collude to "break" the system, it can also be inconvenient. Anderson discusses two examples: a bank removing dual control to save time and money (although it seemed to lead to 10X fraud), and bank managers happily giving ATM technicians the crypto keys, so they do not have to wait while a machine is being serviced.

E. True / False Anderson suggests that if we had a set of "perfectly secure" components, a system composed of these components would also be perfectly secure.

Answer: False. Many security problems in this paper are caused by the interactions between components.

4. [6 points]: Indicate which of the following statements about the ObjectStore system described in the paper by Lamb et al (reading 18) are true.

(Circle True or False for each choice.)

A. True / False ObjectStore stores persistent objects in essentially the same format as the ordinary transient C++ objects.

Answer: True. It uses the same instruction sequences to access fields of both kinds of object, and the only difference between the disk format and the RAM format is possible adjustments to the values of the pointers (swizzling).

B. True / False To get transactions that are atomic with respect to other transactions that execute concurrently, ObjectStore requires the programmer to lock an object before using it, by explicitly invoking acquire, unlike an ordinary relational database system.

Answer: False. ObjectStore uses the VM system to detect reads and writes to objects and acquires the necessary locks automatically.

C. True / False If transaction T1 touches objects A and B and no others, and transaction T2 touches objects C and D and no others, then ObjectStore's locking system ensures that T1 and T2 can run concurrently.

Answer: False. The unit of locking is a VM page, so if A or B happens to be on the same VM page as C or D, T1 and T2 will have conflicting locks.

D. True / False In ObjectStore an object of type T can be either persistent or transient, and the choice is made when the object is created.

Answer: True. The formats are the same and the VM system rather than the compiled instructions takes care of bringing persistent objects into memory.

5. [6 points]: Indicate which of the following statements about the paper "Hints for Computer System Design" by Lampson (reading 26) are true.

(Circle True or False for each choice.)

A. True / False According to the paper (and the doctrine of 6.033), simplicity of design and interface are always the highest priority.

Answer: False. Simplicity is worth a lot, but if you really need performance or fault-tolerance then you have to design for them, even though it will make things more complicated. Simplicity is not a substitute for getting it right.

B. True / False DNS' hierarchical lookups are a good example of following the hint to use brute force.

Answer: False. Brute force would be a single server.

C. True / False "Keep secrets of the implementation" and "Dont hide power behind an interface" are hints that are often at odds with each other.

Answer: True. The secret that you keep may sometimes be the only way to release power. But a good interface like the file system needn't hide power, especially if its augmented with hints about prefetching and future needs for space.

D. True / False "Keep basic interfaces stable" is a hint that severely obstructs progress.

Answer: False. Almost always it's possible to move forward while providing compatibility with old interfaces. In the extreme, virtual machines make it possible to run an entire old OS, and protocols like X-windows that remote the user interface make it easy to mix access to old and new worlds.

6. [6 points]: With respect to the paper "Manageability, availability and performance in Porcupine: a highly scalable, cluster-based mail service" by Saito et. al, which of the following statements is true?
(Circle True or False for each choice.)

A. True / False A given user's mailbox fragments are all stored on the same node.

Answer: False. Each mailbox fragment may be located on a different node, to permit very large mailboxes to be distributed across multiple machines.

B. True / False When a failed node recovers after being down for a day the mailbox fragments it stores are brought back up to date from logs on the other nodes.

Answer: True. A recovering node contacts the cluster and asks other nodes which updates it missed.

C. True / False The mailbox fragment list is hard state that keeps track of the nodes that contain a user's mailbox.

Answer: False. It is soft state, i.e. it can be reconstructed from other information.

D. True / False It is possible that membership services will break a cluster into two disconnected groups of nodes in the presence of certain unusual network failures.

Answer: True. The paper notes that this may lead to inconsistent state, and Porcupine is designed to accommodate that.

7. [6 points]: Given the context of the paper "Exploiting Underlying Structure for Detailed Reconstruction of an Internet-scale Event" by Kumar et. al which of the following statements are true?
(Circle True or False for each choice.)

A. True / False Witty generated packets with random forged source IP addresses, and thus network telescopes were able to detect replies from victim hosts to the IP addresses fabricated by Witty.

Answer: False. The paper only makes sense assuming that the telescopes are able to see correct source addresses in the packets that Witty generates.

B. True / False The Witty worm exploited a buffer overflow in the ICQ client in Microsoft Windows.

Answer: False. It infected ICQ analyzers.

C. True / False It is possible to operate an Internet wide network telescope by telling your computer to listen for the packets addressed to the address range you wish to monitor.

Answer: False. You can't ordinarily ask to see packets addressed to destinations other than your IP address.

D. True / False The origin of "Patient Zero" was determined by carefully observing which host was first observed at the network telescope.

Answer: False. It wasn't the first host observed. It was different in that it was clearly running slightly different software (e.g. generating different patterns of random target IP addresses).

8. [6 points]: Indicate the truth or falsehood of each of the following with respect to Ken Thompson's paper "Reflections on Trusting Trust."

(Circle True or False for each choice.)

A. True / False Thompson believes that self-reproducing programs shouldn't be trusted.

Answer: False. While his "trusting trust" attack hides in programs that produce programs, he doesn't say anything about this making them more or less trustworthy.

B. True / False A Trojan horse like the one Thompson describes could not have been hidden in a compiler for a more modern language like Java.

Answer: False. There is nothing language specific about this attack.

C. True / False The Trojan horse Thompson embedded in the login program could have been found by looking at the machine instructions being executed by the CPU.

Answer: True. It might be difficult, but the back door does appear in the binary executable code.

D. True / False A programmer can prevent the type of attack Thompson describes by writing all of his or her programs in assembly code.

Answer: False. It would be just as easy to create a similar bug in the assembler used to translate the assembly code to machine code.

Buffer overrun, as explained in the paper "Beyond stack smashing: recent advances in exploiting buffer overruns" by Pincus and Baker, can overwrite the procedure return address that is stored on the stack. To avoid this problem, Betty writes a compiler that generates code that maintains two different stacks. One is the usual kind but without the procedure return address, and a second stack, stored in a different part of memory, is used only for the procedure return address.

9. [6 points]: Indicate whether each statement is true or false with respect to Betty's compiler.

(Circle True or False for each choice.)

A. True / False Betty's compiler avoids the classic stack smashing exploit as explained in papers previous to Pincus and Baker, by such hackers as AlephOne and DilDog.

Answer: True.

B. True / False Betty's compiler avoids the Arc injection exploit outlined in the paper.

Answer: True.

C. True / False Betty's compiler avoids the Function Pointer clobbering subterfuge exploit outlined in the paper.

Answer: False.

D. True / False Betty's compiler avoids the Exception-handler hijacking exploit outlined in the paper.

Answer: False.

E. True / False Betty's compiler avoids the Heap smashing exploit outlined in the paper.

Answer: False.

II Two-Phase Commit

Suppose you are running a transactional 3 node log-based storage system that is using two-phase commit as described in class.

Node 1 is the coordinator, and node 2 and 3 are just workers.

After awhile, node 1 crashes, and the log on its disk looks as follows (here ... indicates some number of UPDATE operations; you can assume in the following three questions that transactions do not UPDATE any of the same records, and that every transaction updates at least one data item):

```
BEGIN T1
BEGIN T2
...
ABORT T1
BEGIN T3
...
COMMIT T3
```

10. [6 points]:

For each of the following transactions, indicate whether the coordinator will end up considering the transaction to have committed or aborted, or whether the information in the log is not sufficient to decide.

(Circle committed, aborted, or can't tell for each transaction.)

A.	T1	Committed	Aborted	Can't tell **Answer: Aborted**
B.	T2	Committed	Aborted	Can't tell **Answer: Aborted**
C.	T3	Committed	Aborted	Can't tell **Answer: Committed**

Node 1 recovers, and resumes processing transactions. After awhile, node 2 crashes, and the log on its disk looks as follows (assume you know nothing about the coordinator's state other than what is implied by the following):

```
BEGIN T4
...
PREPARE T4
BEGIN T5
BEGIN T6
...
PREPARE T6
...
COMMIT T4
```

11. [6 points]:

For each of the following transactions indicate whether node 2 will end up considering the transaction to have committed or aborted, or whether the information in the above log is not sufficient to decide.

(Circle committed, aborted, or can't tell for each transaction.)

A.	T4	Committed	Aborted	Can't tell **Answer: Committed**
B.	T5	Committed	Aborted	Can't tell **Answer: Aborted**
C.	T6	Committed	Aborted	Can't tell **Answer: Can't tell**

Ben Bitdiddle notices that two-phase commit workers have to write two log records for every transaction (a PREPARE record and a COMMIT or ABORT record.) Ben proposes a protocol called *Ben's 2 Phase Commit (B2PC)*. In B2PC, the messages and operation of the protocol are identical to the two-phase commit protocol we learned in class. The only difference is that, when a transaction commits, the workers do not write a COMMIT record to the log (they do, however, still write ABORT records to the log.) When scanning the log during recovery, workers assume that a transaction they prepared actually committed unless an ABORT record for the transaction appears in the log. The coordinator still logs COMMIT records as in the original protocol.

12. [8 points]: Which of the following statements about B2PC protocol are true? Assume that "correct transactional behavior" means the outcome (i.e., COMMIT or ABORT) of the transaction would be the same as in the unmodified 2PC protocol.

(Circle True or False for each choice.)

A. True / False In the absence of crashes or other faults on any of the nodes, B2PC provides correct transactional behavior.

Answer: True. If there are no faults, B2PC is the same as 2PC.

For the following choices, assume that the workers or coordinator may crash, and that there are no other faults in the system.

B. True / False In this case, B2PC provides correct transactional behavior.

Answer: False. If a worker crashes after responding to a prepare message, and restarts, it will assume the transaction committed; but some other worker could have responded "no" to the prepare, in which case the transaction would have actually aborted.

C. True / False Assume the coordinator remembers the outcome of all transactions forever. Suppose Ben modifies the protocol described above so that, when recovering from a crash, workers contact the coordinator for the outcome of any prepared transaction that does not have an ABORT record in their log. In this case, B2PC would provide correct transactional behavior.

Answer: True.

D. True / False Suppose Ben modifies the protocol described above so that workers do write COMMIT records, but the coordinator omits them. In this case, B2PC would provide correct transactional behavior.

Answer: False. If the coordinator crashes just after sending out COMMIT messages, it will assume the transaction aborted after it restarts.

III Trusting Ted's Terrific Telegraphic Text Teamware

Theodore is developing a collaborative editor called Ted's Terrific Telegraphic Text Teamware, or TTTTT. Theodore plans to have lots of enthusiastic customers, and he knows that he will have to add new features and issue new releases constantly. He is mulling over the problem of how his customers can verify the authenticity of each new release. A release consists of a single executable file, called ttttt-V.exe (where V is the version number), so the problem boils down to each customer being able to verify that their ttttt-V.exe file has the contents that Theodore intended. Your job is to give Theodore advice about three different authentication schemes he is contemplating.

Theodore's first scheme is to calculate a hash of the content of each release, and to post the hash along with the release on his web site. He uses a cryptographic hash function as described in section 11.2.3 in the course textbook. For each release, Theodore calculates $h_V = H(\text{“TTTTT”} + V + R_V)$, where R_V is the content of the release file ttttt-V.exe and $+$ indicates string concatenation. Theodore does all his development, compiling, and computing of hash values on his laptop, which only he has access to. He posts the release and h_V on his web site at these URLs:

```
http://ted.com/ttttt-V.exe
http://ted.com/V-hash.dat
```

He tells all his customers to fetch both files, and to check for authenticity by comparing the fetched hash value with $H(\text{“TTTTT”} + V + R'_V)$, where R'_V is the contents of the ttttt-V.exe file they fetched. If the two are equal, the customer should accept the new release. If they are not equal, the customer should reject the release.

13. [8 points]: Indicate the truth or falsehood of each of the following statements about Theodore's first scheme.

(Circle True or False for each choice.)

A. True / False An attacker capable of modifying the packets of the customer's transfer of the ttttt-V.exe file could do so in a way that would cause the customer's ttttt-V.exe file to be different from what Theodore intended, but would cause the customer to nevertheless accept the release.

Answer: False. If the attacker can only modify the ttttt-V.exe file, then he will not be able to produce a different file that matches the hash.

B. True / False It would be easy for an attacker who can read and modify any files on Theodore's server to cause customers to accept a release that has different contents from what Theodore intended.

Answer: True. The attacker can install any ttttt-V.exe file on the server, and install a matching hash file.

C. True / False If a customer sees a “mirror site” of TTTTT that is not affiliated with Theodore, consisting of (for example) http://mirror.com/ttttt-V.exe and http://mirror.com/V-hash.dat, it would be just as safe to fetch those files and compare hashes as it would be to fetch the files from Theodore's web site.

Answer: False. Using a mirror opens the user up to an additional set of attacks in which whoever operates the mirror can serve incorrect (but matching) ttttt-V.exe and hash files.

Theodore's second scheme is to generate authentication tags for each software version using a shared-secret message authentication code (MAC), as described in the textbook in sections 11.3.3, 11.3.4, and 11.3.5. Theodore generates a separate key K_i for each of his customers, and contacts each customer on the telephone to give them their K_i. For each new release V, Theodore calculates an authentication tag $T_{V,i}$ for each of his customers by calling SIGN("TTTTT" $+ V + R_V$,K_i), where R_V is the content of ttttt-V.exe. Theodore does all his development, compiling, and computing of MAC values on his laptop, and he stores the K_i keys only on his laptop; only Theodore has access to this laptop. Theodore posts the release file and all the tag files on his web site, at these URLs:

```
http://ted.com/ttttt-V.exe
http://ted.com/V-1.tag
http://ted.com/V-2.tag
...
```

He tells his customers to each fetch the new release and their tag from the web site and to call VERIFY("TTTTT"$+$ $V + R'_V$,T,K_i), where R'_V is the content of the ttttt-V.exe file they fetched and T is the content of the V-i.tag file they fetched. If the call returns ACCEPT, the customer should accept the new release; otherwise the customer should reject the new release.

14. [9 points]: Indicate the truth or falsehood of each of the following statements about Theodore's second scheme.

(Circle True or False for each choice.)

A. True / False An attacker capable of modifying the packets of the customer's transfer of the ttttt-V.exe file could do so in a way that would cause the customer's ttttt-V.exe file to be different from what Theodore intended, but would cause the customer to nevertheless accept the release.

Answer: False. VERIFY would return REJECT.

B. True / False It would be easy for an attacker who can read and modify any files on Theodore's server to cause customers to accept a release that has different contents from what Theodore intended.

Answer: False. The attacker would not be able to generate V-i.tag files that caused VERIFY to ACCEPT.

C. True / False If a customer sees see a "mirror site" of TTTTT that is not affiliated with Theodore, consisting of (for example) http://mirror.com/ttttt-V.exe and http://mirror.com/V-i.tag files, it would be just as safe to fetch those files and verify with VERIFY as it would be to fetch the files from Theodore's web site.

Answer: True. Even a malicious mirror operator would not be able to generate V-i.tag files that caused VERIFY to ACCEPT.

Theodore's third scheme is to configure his web server to use SSL with a certificate from a well-known certificate authority. Theodore's server holds a particular private key, and the certificate says that whoever knows that private key is the rightful owner of the DNS domain ted.com. Theodore tells his customers to fetch new releases from

```
https://ted.com/ttttt-V.exe
```

Theodore tells his customers that they do not have to take any special steps to check the authenticity of the software, since, if you tell a web browser to connect to https://*servername*, it will use SSL to check that the server can prove ownership of a certificate for *servername* from a well-known authority. Theodore tells his customers to check that the right URL appears in their browsers' URL box. SSL (also known as TLS) was described in lecture, and you can also read about it in section 11.10 of the textbook.

15. [9 points]: Indicate the truth or falsehood of each of the following statements about Theodore's third scheme.

(Circle True or False for each choice.)

A. True / False An attacker capable of modifying the packets of the customer's transfer of the ttttt-V.exe file could do so in a way that would cause the customer's ttttt-V.exe file to be different from what Theodore intended, but would cause the customer to nevertheless accept the release.

Answer: False. SSL protects the transfer against modification.

B. True / False It would be easy for an attacker who can read and modify any files on Theodore's server to cause customers to accept a release that has different contents from what Theodore intended.

Answer: True.

C. True / False If a customer sees a "mirror site" of TTTTT that is not affiliated with Theodore, consisting of (for example) https://mirror.com/ttttt-V.exe, it would be just as safe to fetch those files via SSL as it would be to fetch the files from Theodore's web site.

Answer: False. The browser would raise no error (after all it might really be connecting to the real mirror.com), but the mirror server could serve any data it liked, including incorrect data.

End of Quiz III