Computer Security 1, COMPGA01 and COMPM062, 2012-13 **ANSWERS, CONFIDENTIAL!**

Answer ALL questions. 2.5 hours.

Correct answers should be short, straightforward and clear, rather than long and obscure.

Marks for each part of each question are indicated in square brackets

Calculators are NOT permitted

Question 1. Questions about theory and mathematical models.

1. Given a poset $A, \leq$ give an exact **mathematical** definition of GLB (defined as a certain element of a certain set satisfying certain properties). Explain the acronym GLB.

   [5 marks]

   Answer: GLB = **G**reatest **L**ower **B**ound. Defined as the biggest element which is dominated by the arguments, mathematically: $GLB(a,b) = Max_{\leq}\{x \in A | x \leq a \text{ AND } x \leq B\}$

2. Consider the set of integrity levels $L = \{low, admin, kernel\}$, where $kernel > admin > low$. Furthermore consider the set of categories Cat containing

   $$Cat = \{Management(M), HR/Admin(H), Auditors(A), Traders(T), ITPersonnel(I)\}.$$

   Compute the exact number of security classes in the Biba product lattice.

   [5 marks]

   Answer: $3 \cdot 2^5 = 96$

3. In the Biba model explain what is the semantics or our understanding of what it means for a file to be at a high integrity level in some lattice. What can we say about the information flow in the strict Biba model? Can information flow between two unrelated levels in the lattice?

   [5 marks]

   Answer: High level: the data is accurate, reliable and not contaminated. In Biba model the information flow goes down, and only within pairs related in the product lattice.

THIS IS THE ANSWERS PAPER, NOT TO BE PRINTED, CONFIDENTIAL

4. Let $\mathbb{N}, |$ be a poset where $|$ is the division relation. Is it a lattice? Please justify your answer. Compute $LUB(18,16)$ and $GLB(18,16)$ if they are defined.

[5 marks]

Answer: It is a partial order and also a lattice because GLB and LUB always exist and are known as GCD and LCM. $LUB(18,16) = 16 \times 9 = 144$. $GLB(18,16) = 2$.

5. Which relations are NOT transitive? Any number between 0 and all solutions can be selected.

   (a) Having a shared factor for integers

   (b) Each number n is $\geq$ itself and 1 and no other number.

   (c) The relation "dominated by" in a BLP lattice

   (d) Relation on sets commonly known as $\subset$: being subset of another set or equal.

   (e) We say that computer1$\Rightarrow$computer2 IF computer1 knows the MAC address of computer2 and stores it in a local ARP cache.

   (f) Relation $a|b$ or a divides b for natural integers.

   (g) Equality of fractions like $3/6 = 1/2$ is they are equal as real numbers.

[7 marks]

Answer: Exactly two correct answers or relations which are not transitive:

a) Having a shared factor for integers.

e) Computer1$\Rightarrow$computer2.

All the other are transitive.

6. Which tasks can be performed by a Turing machine in constant time? Select zero, one or several answers.

   (a) Deciding whether a 10000-bit long string of 0s and 1s is generated with by pseudo-random number generator with some (unknown) 64-bits seed.

   (b) Deciding whether a given program in Java halts within 2^{64} steps

   (c) Factoring a 1024-bit SSL/TLS RSA public key

   (d) Finding a collision for MD5 hash function as implemented in Unix or other OS in order to forge a security update: produce two updates with the same signature.

   (e) Deciding whether a program contains a buffer overflow exploit.

   (f) Blocking all viruses in such a way that they are always prevented from doing any harm.

[6 marks]

Answer: Right answers are a,b,c,d,f. A finite computation with some 2^K steps but K being fixed is acceptable. It is easy to block viruses if we restrict functionality.
e is not possible due to the Rice Theorem.

[Total 33 marks]

Question 2. Questions about OS and computer security.

1. Why should the current directory "." not be added to UNIX PATH? Give one reason why this would be a bad idea.

   [3 marks]

   Answer: Many reasons, for example, if one could try to fool a system process running as root into calling some program hack.exe in the current directory, then hack.exe will be running with root privileges!

2. Explain and/or draw a picture to show how a compiled C program will call a sub-routine, what is the relative position of the return address and how this can be exploited to run arbitrary code. Is it a good idea to wipe the stack to remove the local variables and data (which would remain in RAM), just before leaving the routine?

   [10 marks]

   Answer: When a program will call a sub-routine, it will write on the top of the stack the parameters of the routine, then the return address, (then a saved pointer to a bottom of the stack; which is not so important) and then in the space above, local variables will be created and used when running this "C" language routine. It is important to see that the return address is **below** the data, this is on our picture with a stack growing upwards, it is a higher address in (virtual) memory space. Data and pointers on the stack can be manipulated using PUSH and PULL instructions and when the routine terminates the stack level should be the same. However the data are NOT wiped because many routines are fast and cannot afford to even read or write the stack space they allocate and free later (like buffers which are most of the time longer than necessary and wiping them would very badly affect the speed of programs). **Bad idea**. If we overwrite these local buffers with longer data downwards, we can **overwrite the return address** of the sub-routine to make it jump, after it finishes, to a memory location chosen by the attacker. If in addition this location contains machine code injected by the attacker into the same buffer, this code will be executed.

3. Explain DEP and NX bits. Why or how it became an 'NX bit' to be set and not an 'X bit' to be disabled?

[5 marks]

Answer: Since i386, W/R permissions exist at the **page table** entry level, 4 K pages typically, but only very recent computers are able to prevent the execution (NX bit, Never eXecute) from a given individual memory page, and this type of protection was for years implemented in form of for example software patches. Now it is a hardware mechanism and existed since ever for i64 CPUs, but then Pentium 4 Prescott and AMD adopted this for general-public computers around 2004. In order to prevent the execution with the NX bit, and for the **compatibility** reasons with existing software and hardware, by default X is enabled, and it must be explicitly disabled: the NX bit must be set to prevent execution from this page. This is called Data Execution Prevention (DEP) and works only when the OS supports it (and using 64-bit page tables with so called PAE memory addressing mode enabled, roughly speaking Windows XP SP2 and later and Linux kernel since release 2.6.8.).

4. Give one example of one special privilege of the OS Kernel under Linux or Windows.

[2 marks]

Example of answer: Writing the boot sector of the current system drive.

5. In Windows many objects will have a ”security descriptor”. How does it work for files? Explain what is ACL and ACE. What kind of data does one ACE contain?

   Are these objects stored in the object's directory of the hard drive, in system directories, user data files, in RAM, or in the registry?

   What mechanisms prevent the user from reading some of sensitive administrative and system-level rights?

   [6 marks]

   Answer: In Windows implementation of Access Control Lists (ACL's) a file has a security descriptor with attributes which contains a list of Access Control Elements (ACE) objects. ACL is a linked list of ACEs. One ACE will contain, for one user, group or another type of SID = Security Identifier, (for example SID="local system") an encoding for the set of rights (R/W/X/Delete/Change the Owner etc) which is allowed for this SID. They are stored in the object's directory, together with the file, as file attributes. They are accessible and visible in totality only to the system. At runtime access to many attributes is simply denied or they will be invisible. However because they are stored on the hard drive various software will be able to read and decode them imperfectly. Security by obscurity: they are encoded in an obscure way, with enigmatic SIDs and unclear interpretation rules.

   [Total 26 marks]

Question 3. Questions about network security and SSL/TLS.

1. An IP datagram (one IP packet) has always the following security features: Select zero, one or several answers:
   (a) it is digitally signed
   (b) it is protected against accidental modification or transmission error
   (c) it is encrypted
   (d) it cannot be decrypted if we do not have the chaining value from the previous packet
   (e) it is protected against modification of source/destination IP address

   [5 marks]

   Answer: b) is correct: it is protected against accidental modification or transmission error (by a simple CRC). Exactly one correct answer.

2. Explain the term DMZ in the context of network security.

   [3 marks]

   Answer: DMZ stands for a De-Militarised Zone, sort of buffer between two territories. Typically it is a zone which is not inside the internal (secure and protected) network but a separate logical and physical zone. It hosts machines which will be communicating with the Internet and insecure outside networks: proxies, email public FTP and HTTP servers, etc. These machines are exposed to attacks, it is a sort of “electronic frontier” of an organisation.

3. Name two protocols which use SSL/TLS, and which are extensions of earlier insecure protocols. Explain very briefly what they do.

[2 marks]

Answer: HTTPS connects to web with data encryption and (mostly server to client) authentication, based on HTTP. Other examples are email protocols POP[S] and IMAP[S]. In fact IMAPS means "IMAP over SSL".

4. Assume that we want to consult our email via IMAPS. What is the necessary **trusted setup** condition which should not be forgotten at the moment of installing or configuring our client email system? Why and when this setup would in many cases not be needed?

[4 marks]

Answer: We need to have some **authentic verification keys** allowing the SMTP server to be authenticated, for example install once for all a certificate which contains such a key. If we don't, anybody on the local network can pretend to be this SMTP server, it will be able to capture our password, and much more. This is not needed if we have a working PKI (Public Key Infrastructure) deployed: if we use and email client which has some root CA certificates installed, and the SMTP server have obtained certificates for his public key. But if there is no PKI we need to do the job of the PKI.

5. Explain the Man-In-the-Middle attack on the Diffie-Hellman key exchange. How does SSL/TLS solve or tries to solve this problem?

[12 marks]

Answer: DH allows two entities to establish a common session key. The server selects x and transmits g^x modulo a large prime p. The client selects y and transmits g^y modulo p. The shared secret is g^{xy}. In the Man-In-the-Middle attack on the Diffie-Hellman the person in the middle will replace g^x by his own g^z for which he knows z. Also they will replace g^y by his own g^t for which he knows t. Then he needs to decrypt and re-encrypt all the messages with the correct key. This will not be detected in principle though there may be slight time delays. SSL authenticates all these messages (e.g. with added signatures).

6. Explain the two main stages in TLS: TLS Handshake and Secure Communication stage. What happens at each stage? How the authenticity of public keys is guaranteed?

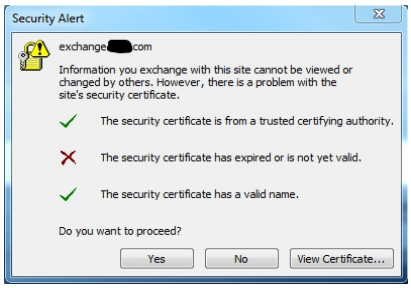
[5 marks]

Answer: In TLS Handshake we establish an ephemeral shared key using PK cryptography techniques, e.g. Authenticated Diffie-Hellman. Secure communication will be a fully encrypted and authenticated communication (like a tunnel with a guarantee of authenticity) sending arbitrary messages: arbitrary HTTP session. The public keys are authenticated with certificates: they are digitally signed by a CA.

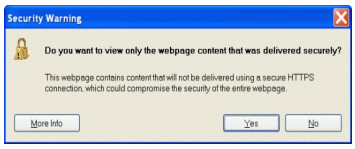
7. For each of the following **four** security alerts, provide a plausible explanation as to why these messages are displayed, and what are the risks if we accept.

[10 marks]

Answer to Q3.7.a. it could be that the certificate was issued for one domain but the server is hosted at another domain or it is redirected. Very frequently there are several domain names pointing to the same IP and the same server. It is good to view the certificates and compare the names if it is www6.yahoo.com and news.yahoo.com, there is no harm. Risks: if the two names belong to different entities we could have a total compromise: all our communications will be with the wrong server.



Answer to Q3.7.b. Expired: maybe not that serious, can check the key size. Risky only if this expired key can be broken today. If key size is not too bad and it was recently expired we may accept.



Answer to Q3.7.c. Some parts of the page displayed are not secure: not encrypted and not authenticated. The box suggests clicking YES, and it is a simple and secure solution. Otherwise we don't know which data are really coming from the right web server and which are authentic. We risk being fooled in many ways, for example we could end up clicking at a wrong link in a wrong place, for example an advertisement might suddenly display a button which directs us outside of the bank but looks like the real button, and we are used to just click at that sort of button.

Answer to Q3.7.d. Everything is almost fine. The local user or local browser has a policy not to trust this CA (Certification Authority). Depending on who set up this policy, the context and our perception of risk, we may chose to trust the CA in question for this session. However there is probably a reason why it is not trusted, it is better to do some research and reflect on why this CA is not trusted.

Moreover: OK to accept once, it is VERY dangerous to permanently load/install a CA to be trusted permanently. Once loaded, the browser will accept all certificates signed by this new CA. Is there any evidence that this comes from a trusted authority? It is better to NEVER install a new CA (approve a root public key signed by no other CA).

[Total 41 marks]

[Total For The Whole Exam 100 marks]

THIS IS THE ANSWERS PAPER, NOT TO BE PRINTED, CONFIDENTIAL

END OF PAPER