UNIVERSITY OF DUBLIN

TRINITY COLLEGE

FACULTY OF ENGINEERING, MATHEMATICS & SCIENCES

SCHOOL OF COMPUTER SCIENCE & STATISTICS

MSc in Computer Science Hilary Term 2011
CS7053

Security of Networks and Distributed Systems

EXAM SOLUTIONS

Date	**Location**	**Time**
19th April 2011	**LB1.07**	**9:30am**

Dr. Stephen Farrell

Instructions to Candidates

Please attempt 3 questions.
All questions carry equal marks.

EXAM SOLUTIONS

EXAM SOLUTIONS

Question text is in italics below.
`Answer outlines are like this.`

Question 1. (33 marks)

You are a senior software developer in a large software company tasked to produce a security review process for a new product development group that has just been established. The group will be developing web services based applications for the healthcare market, including handling of sensitive (patient) data. The security review process will be followed by designers and developers. You cannot expect that all people in the group will be familiar with security, and they will be constant under time-pressure to produce product deliverables. For the purposes of the answer, you can assume whatever general software development methodology you like is to be used.

The student should take proper account of the context – healthcare, a s/w dev company etc. The main point is to produce a reasonable process that approximates what might really be used.

a) Outline the security review process you would initially suggest, and how it fits with the general software development methodology you select. (15)

Marking is 5 for overall goodness, and 10 for the specifics of the process. The process should at least involve some security/risk analysis early in a development project and a later phase with security involved as part of testing/QA of some sort. There may be an intermediate stage where security specific product/technology issues is examined in detail. They may decide to follow some security methodology (e.g. OWASP, ISO). The process might be broken down like this:

Requirements phase: establish actual security requirements (e.g. read/write access to data according to some data model), a threat model (e.g. which kinds of bad actor are considered) and relevant roles and security/network/application management issues. Probably meet with developers and system owners.

Development phase: establish a security testing and reporting structure, with an incident handling process/team; as products/technologies are identified by developers, check for known security issues with those (e.g. in CVE databases); do a security review of design documentation as appropriate; possibly meetings with developers to review the above. Produce a report for the system owner on the above.

Testing phase: consider pen testing of system; do a security review of the implemented system as part of overall QA prior to deployment. Identify known issues and where possible mitigations (e.g. audit logs may need periodic checking for known remaining vulnerabilities, role administration may be partly manual) etc.

The process should have something specific to handling patient data, e.g. dealing with regulations like the US HIPAA or similar. Deciding to always deal with patient data via a productised subsystem that gets re-used might be a good feature of the process. Ensuring that fake data is used for testing (and never real data) is a good thing. Having a good split between developers and operational/customer staff is a good thing. There needs to be an inident handling scheme aligned with product release/patch cycles or that allows emergecy patching. Etc. etc.

b) How would you get from your initial suggestion for a process to a final, signed-off security process to be followed by the group? Describe the actions you would take and the interactions with others you would expect to occur before you get that final sign-off. (10)

Marking is 3 for overall goodness and then 7 for specifics. The specifics should include things like the following (roughly 2 marks for each such thing well described):

- Its a big company – make sure to use resources from elsewhere, e.g. a company CERT or other security experts; make sure to deal with incident handling

- Get external assistance with regulatory issues where necessary, including for other countries (assuming products are not for one marketplace)

- You have to get management signoff to insist on anything

- Need to get dev. managers to actually run the process so even if they're told to do it, they need to actually want it (convince them that doing less security QA leads to more work in the end)

- Develop the process iteratively; don't regard it as final first time its used.

- Learn from running the process; get feedback from developers and everyone concerned

b) How would the security process be maintained over time in order to handle changes to the group's activities, the legal and the business environment? (8)

Straight marks from 8 for goodness; things that might affect the process include selling into new markets; specific incidents may indicate process changes are needed; regulatory environment evolves (e.g. FDA make make rules for s/w); new product features might throw up new security testing or review requirments; tools might become better or cheaper; as developers get familiar with the process they will suggest changes and won't necessarily need so much handholding for security etc.

Question 2. (33 marks)

a) For any real Internet key exchange protocol (e.g. TLS, S/MIME, IPsec, Kerberos) describe the protocol in detail, including a description of the relevant configuration settings that an implementation would require. (15)

Marks-from-10 for a good description of the scheme, with e.g. the TLS handshake and application layer properly described. 5 marks for properly identifying and describing configuration issues such as trust anchor management, ciphersuite and compression settings; certification revocation settings (CRLs/OCSP) and intermediate CA cert access (LDAP); session management for resumed sessions is a good extra.

b) For your selected protocol, describe the main issues that would arise were it to be used in a large distributed network with some "normal" nodes but mostly consisting of many (i.e., millions), very small devices (e.g. motes) with limited processing power and memory and disrupted network connectivity. An example of such a network might be a country-wide seismic monitoring system but you can use any other example if that helps. (10)

Marks-from-4 for overall goodness; specific things that can be mentioned for roughly 2 more marks each:

- footprint – omit optional features on motes
- handshake - use TLS-PSK with provisioned secrets & session resume in TLS can help a lot here
- RSA key size - use of ECC for smaller key exchange packets
- CPU challenges – maybe invent more efficient ciphersuites for TLS application protocol
- revocation would be a problem - maybe use short-lived certs
- TCP could be an issue - use DTLS rather than TLS (as in CoAP)
- well-connected nodes might need load balancing/SSL-acceleration
- might need MITM proxies to segregate regions of nodes
- cost of certs - don't use commercial CA, setup own PKI

c) Describe two changes you would recommend for your selected protocol that would mitigate some of the issues you identified in part (b) of this question. Be specific as to whether you consider each recommended change in something that could immediately be used, or whether your recommendation would be to experiment with the relevant change. (8)

Straight marks from 8; see right hand side of above list

Question 3. (33 marks)

You are asked to design an interpersonal communications system for staff from the International Monetary Fund (IMF) who visit countries in financial difficulty to negotiate IMF loans for that country. The system should support "instant" (e.g. jabber/XMPP) and store-and-forward (e.g. email) messaging as well as file transfer both between the IMF staff visiting a country as well as between them and staff at IMF headquarters. The messages, files and the computer systems involved are all considered highly sensitive and governments, hackers and financial-system actors can all be considered as potential attackers. Local network providers (e.g. in-hotel WiFi, GSM operators) are not to be trusted. In the context of a particular IMF visit to a country, only a small number of IMF staff should have access to the details (messages, files) of the negotiation.

a) Describe the system you would design for this purpose, with a specific focus on the security elements. Your description should include security for the entire system, from the protection of the computer systems (e.g. laptops) to the applications and networks, and cover both application layer security functionality (e.g. confidentiality services) as well as other security components (e.g. audit, intrusion detection). Include system and network diagrams as appropriate. (15)

Straight marks from 15. The main thing here is that they should include the entire system incl. e.g. a particular OS (version) on laptop, disk encryption, backup and archival as well as the application layer stuff. Systems (laptops and servers) should have AV and local firewalls and generally be hardened. Applications could be web-based or more traditional separate clients, either can work. If web-based then they need to describe how they handle access control in the web app; if separate user agents they should say how those should be setup (e.g. to require s/mime for mail etc.). IM with e2e confidentiality and access control would be a nice extra rather than just jabber over TLS. Any "sharing" stuff (e.g. file transfer) should have some kind of access control and maybe a need-to-know type thing, probably even labelling. A traitor-tracing aspect to the system would be a nice extra. A VPN seems like a must here and should be included in addition to, and not instead of, application layer security. Local DNS etc. shouldn't be trusted, or only if DNSSEC used. There should be a firewall and IDS and some audit components. There should be a DMZ in the HQ setup with e.g. the VPN gateway. They should include some diagrams.

b) Describe how you would deploy and operate your system, that is, the processes required for adding and deleting computer systems and users and the processes you would follow to monitor the ongoing security of the system. Note that part (a) asks about the design of the system, but part (b) is concerned with operational issues. Your answers should not overlap significantly. (10)

Straight marks from 10. Should start from trustworthy suppliers; devices should be provisioned at HQ; probably re-imaged for each visit abroad with different crypto keys etc. each time. Systems should be pen-tested regularly. Ops people should be checking audit trails all the time, with alerts generated to on-call staff. Adding a user is fairly obvious, but when someone leaves a team/the system their access should be revoked quickly. Traffic patterns for remote staff should be checked. Laptop logs should be centralised to check that remote staff are obeying policy. Laptops should be forensically examined on return in case of breach (by e.g. host of the visit) or in case of staff misuse (e.g. connecting to open WiFi).

c) You are an IMF system administrator based at IMF headquarters who is a citizen of a country being visited by an IMF negotiation team. Your government pay you to break into the system described above, in order to get information as to the negotiation position of the visiting IMF team. How would you go about that? (8)

Straight marks from 8. Sneaky answers here are fine, e.g. just phone a person on the visit (or their spouse) and socially engineer them into telling you what's up or sending you files (claim the file is virus infected). Better if you don't get caught though, so arrange to get an infection to one of the visitor's laptops, detect that and get them to ship the laptop back for forensics, when you grab the data. Before the trip install trjoan s/w on laptops that uses some covert channel (e.g. timing, ARP messages, ICMP echo) to leak information in a way that's hard to detect. Higher marks here is they've done a good job earlier and not tried to make this bit easier for themselves.

Question 4. (33 marks)

a) Describe in detail the Domain Keys Identified Mail (DKIM) mechanism for associating an authenticated domain identifier with email messages. (15)

Straight marks from 10 for a good description. Should include the DKIM-signature header field and something about its values, the key record in DNS and how that's found and the rules for signature validation; describing how signing and verification are done in MTAs and not UAs is more or less mandatory. Extras for the addition 5 might include: some key management stuff e.g. delegated keys, DomainKeys vs. DKIM, ADSP and what that does, maybe even a contrast with SPF.

b) Describe the role that DKIM can play in countering phishing and SPAM and explicitly describe what DKIM <u>does not</u> provide in terms of countering phishing and SPAM. (8)

Straight marks from 8 for a good description. Main thing is to say that DKIM doesn't prevent or reduce SPAM but does associate a domain identifier with a (mail) domain so that whitelisting can be done with higher confidence and independent of e.g. MTA IP addresses and that might form the basis for either local or later broader reputation services e.g. VBR. Contrasting DKIM with SPF etc. here would be fine too. Saying that DKIM doesn't deal with bogus domains nor with friendly-name annotation of “From:” header fields is a good thing here. Saying that DKIM doesn't say that the message is “good” is good here.

c) Briefly describe two other commonly used mechanisms that aim to mitigate problems caused by phishing attacks distributed via email. (5)

Straight marks from 5 for a good description. Basically any anti-spam stuff here is fine – statistical analysis/AI/AV filtering etc. Black/block-lists would likely get a mention. Authenticating and rate-limiting outbound mail would be fine too.

d) You are a spammer. Describe two ways in which you would you attempt to get your SPAM delivered to users in systems that have deployed DKIM signature verification and the other mechanisms you described in part (c)? (5)

Straight marks from 5 here. Easiest is to zombie an authorized host and then its domain will happily add a DKIM signature and get by verification. Just buy your own domain. Get someone to send you a DKIM-signed message and replay that. Getting by the measures from part c will depend on them but is fairly obvious. Higher marks here is they've done a good job on part (c) and not tried to make this bit easier for themselves.