UNIVERSITY OF DUBLIN

TRINITY COLLEGE

FACULTY OF ENGINEERING, MATHEMATICS & SCIENCES

SCHOOL OF COMPUTER SCIENCE & STATISTICS

MSc in Computer Science Hilary Term **2015**

CS7053

Security of Networks and Distributed Systems

CS7053-1

Date	**Location**	**Time**
TBD	**TBD**	**TBD**

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)

A financial services software product company are highly distributed, with most of their 500 employees working from home offices and only a few people located in 5 small offices. Locations are scattered around the world, some better connected and some less well connected. The company want to introduce a new system for collaboration that supports employee meetings (video/audio/chat) and file storage, including source code control. The company are evaluating various service providers who offer solutions in this space and have tasked you do help them evaluate the security properties of the various bids they have received from service providers, including having you be a part of the team communicating with the bidders.

Answer outlines are like this.

a) Describe the most relevant risks (at least 3) you see for the new system, including consideration of their impact and likelihood of occurrence, and outline the countermeasures you would expect that the company and/or their chosen service provider would need to impelment to mitigate those risks. (18 marks)

The student should take proper account of the context – financial s/w, distributed staff and a 3rd party cloudy provider. The main point is to describe a reasonable set of threats and mitigations that approximates what might really be used. GOT HERE

The main thing here is to describe risks with their impact and an estimate of probability of occurrence and to concentrate on the more significant of those. The impact and probability can be take any value without losing marks, but for something odd (e.g. if they considered date leakage low impact) they will need more justification for saying that. If the impact/probability are fairly obviously right, less needs to be said. Marking is out of 5 points for each good one and with 3 more for overall goodness. Possible risks would include:

- Top requirement is data-integrity for source code
- Confidentiality for business data, including meeting content and meta-data
- Pervasive monitoring is also a threat
- Leaking of staff personally identifying data and/or customer information
- Data leakage via stolen/lost laptop
- Data leakage via the 3d party
- Denial-of-Service attacks on the 3rd party
- Hackers breaking in to or defacing the 3rd party

- 3rd party staff or other tenants abusing the infrastructure from within
- Cloud-providers monitoring site traffic and data for their own benefit or for the benefit of competitors or governments
- Sales or customer service staff working for competitors
- etc. etc.

b) What other questions would you suggest the company ask about the service provider's systems and operational processes? (10 marks)

As above basically in terms of marking scheme. This could be more or less focused on security, which is fine, OPS issues are good to ask about too. Possible questions could include:

- Describe your operations security model
- Have you done tiger-team testing
- Describe your risk analysis model
- What SLAs (five-9's etc) do you offer
- What other tennants use this and how to you separate them/us
- How do you report security issues found to your customers (i.e. us)
- etc etc

c) What countermeasures would you put in place to detect or prevent a dishonest developer employee from misbehaving? (5 marks)

As above basically in terms of marking scheme. Dishonest dev may want to sell code or plans or install trojan. Dectection is going to be far easier, and involve logging/auditing, but logs would need a clever back-end processing setup for stuff to be actually spotted. Backups that are not under dev control may be needed to revert possible trojans. Prevention based around not letting devs have root on their hosts is impractical. Code review before commit is needed.

Question 2. (33 marks)

(a) For any real Internet key exchange protocol (e.g. TLS, S/MIME, IPsec, Kerberos) describe the key management and application data protection aspects of the protocol in detail, including a description of how you would test which protocol features have been deployed in the wild. (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 describing a real use case such as IMAP/TLS. Deployment measurement via alexa-top-N, zmap, qualsys etc and calling out the things they tend to measure.

(b) What are the main barriers to deployment that are likely with your chosen protocol? (10)

So much from which to choose! :-(2 marks for each good point made + 2 for overall goodness (expecting 3-4 different points), possibles:

- admin/opex costs
- ignorance
- lack of browser hardfail
- http -> https content transitions hard sometimes
- CPU (not really)
- middleboxes
- ...

(c) Desribe and justify how you think your chosen protocol might evolve in the coming deacde? (8)

Most likely here is TLS1.3, where its evolving towards fewer RTTs, getting rid of old/not-so-good crypto, more protection for meta-data where possible etc. For non-TLS cases, evolution will be slower but along similar lines. If they say that a decade is too long to ask about that's a bonus.

Question 3. (33 marks)

You are asked to design a system for summoning and paying for taxis (like Hailo) or private car rides (like Uber). The system should have a web-based user interface as well as smartphone apps for end-users. Before the trip starts, the system provides the end-user with some basic information about the car/taxi and driver that are assigned to them. During or after the trip, the end-user can input comments on the driver/car. End-users can have an "account" that is pre-paid and billed for the trip, or can use cash or card-based payments. Drivers and taxi-company owners will have a separate interface that allows them to view statistical and payment information. The system is to be operated by a third party company who are not a taxi-company but who take a small commission for each ride that is arranged via the system.

a) Outline the overall design for such a system (include a network diagram) and describe the security requirements the system must meet. (10)

This should be a fairly straightforward web-like distributed system. Separating this system from the usual distributed DB (at least as the SQL-level or equivalent) is a near MUST. There could be shared storage beneath though as that might be the best way to get reliability. The only unusual here is that entered comments are UGC and hence need moderation or management.

b) Describe, in detail, the security solution you would propose to meet those requirements. (15)

Given a good set of requirements (marking will be lenient in terms of allocating marks to this part for text in the answer to (a)) this is pretty straightforward too, basically use good mechanisms for the confid., integ and AAA requirements. Audit and moderation should be part of the solution too though.

c) Describe a non-obvious way in which you (as a system developer) might leave in place some feature that benefits you financially and that is hard to detect. (8)

Obvious is easy here (endless pre-paid credit) but gets fewer marks, maybe 3. Non-obvious is better. An example might be to blackmail drivers into cheating with you by making their meters record trips badly so less commission is paid than ought be the case, but where you make it look like the driver is the one cheating (by turning on/off equipment or comms. Interfaces) and where you and the driver share the underpayment. (And you stay anonymous to the driver via whatever method you choose.)

Question 4. (33 marks)

a) Describe the architecture and key management hierarchy of DNSSEC. (15)

5 for arch; 5 for key hierarchy and 5 for overall goodness. This is a straightforward “do they know it” part of the question; they should cover all the basics as taught in class

b) How will DNSSEC deployment impact on registrars, registries and applications? (8)

2 marks for how well they cover apps; 2 each for registrars and registries ops; 2 for overall goodness; points that could be made

- End-hosts will need to have trust points for the DNS root for their resolvers and will need to know DNSSEC
- Some applications will need to include their own resolver and cache in order to be sure that DNSSEC was used (e.g. a TLS implementation in a browser using DANE). Maybe that'll be an interim measure, maybe long term.
- Applications will need to start using some kind of API to tell them if names have been resolved securely.
- Middleboxes (e.g. home gateways) will need to pass DNSSEC results without messing with stuff.
- Registrars will need to include DNSSEC in their “create a domain” Uis and figure out how (or if) to charge for that.
- Registries and authoritative servers will need to do key management including roll-overs and will need to figure out validity periods for signatures (RRsig duration).
- As above for DNS within enterprises.
- ...

c) Describe ways in which the DNS can leak private information and how one might mitigate that with changes to DNS protocols, implementations and deployments. (10)

They should note that DNS data being public is not the same as the act of accessing to that data being public. 3

for covering lack of confid; 3 for covering QNAME minimisation; 4 for overall goodness; -1 if they mix up DNSSEC with this. Bonus is they get the diff between stub and recursive resolver and authoritative server and how that that affects all this and/or things like Google's public DNS servers hoovering up access info.