Exam Code

NDS106 SECURITY ANSWERS/MARKING SCHEME

Question text is bold, italic, like this.

Answer/marking scheme text is normal, like this.

Question 1:

You are hired as an IT security specialist by a start-up "web 2.0" company developing a new social networking site similar to Flickr, Bebo or MySpace. The company have not previously thought much about security. Your first task for the company is to analyse what they've done so far and to suggest security-related improvements.

Main thing is to give reasonable answers that show a proper understanding of security and risk, as being amongst many things that need to be considered. In this case, no-one would use a totally paranoid social network, (SN) so the appropriate level of security is less than e.g. for Internet banking.

(a) ***Name the top 5 security issues that you would first examine? For each, justify its inclusion on your top 5 list. (5)***

Max. 2 marks for each that's well justified, capped at 5.
Possibilities include:

- Privacy – does the system properly manage users' expectations of privacy (e.g. keeping buddy lists private); justification is that this is a common gotcha for SNs and that user's react badly when their privacy is unexpectedly disrespected.
- Proper use of HTTP/SSL for login pages and other sensitive information; justification is that logins should be more "secure" but developers often include cleartext logins as they're afraid of performance issues.
- Bookmarkable URLs –for content that is only supposed to be accessed "securely", is access to the relevant URLs properly controlled; justification is that developers often forget that URLs can be guessed/typed.
- Administrative interfaces – could an administrator easily leak the entire user DB? Justification is that developers often forget that administration (via the web) provides a point of attack.
- General network security – VPNs, F/W issues etc.; justification is that during development a site may be fairly open, but it needs to be hardened before going "live"
- Secure connections to partner sites; generally via VPN; justification as above.

Some more "advanced" ones:

- SQL insertion (and related XML equivalents) – many sites are vulnerable to such attacks & developers don't always include the required tests
- XSS attacks; web2.0/AJAX systems are often vulnerable to XSS attacks
- Javascript bugs
- Code signing

(b) ***For each of the top 5, how would you test whether or not the company's web site is sufficiently "secure"? (10)***

The key words here are “test” and “sufficiently”. Max three marks for each, capped at 10 (would expect uneven answers here).

Possible answers (there’s more than one for each of the top-5)

- Privacy – is there a privacy policy in place?
- Privacy – is PII handled as sensitive data?
- Use of HTTP/SSL – turn off port 80 and see what works/breaks
- Use of HTTP/SSL – various web probing tools, e.g stuff from http://sectools.org/web-scanners.html
- Generic: organise a penetration test, discuss results with management
- Admin interfaces: do all administrators have commensurate roles assigned with proper separation?
- SQL insertion: are all inputs from the web clients validated before being auctioned & how?

(c) ***For each of the top 5, if the test above “fails,” describe the technical, procedural and/or policy countermeasures you would recommend to be put in place. (10)***

Same marking scheme as for (b).

The answer should include an appropriate mixture of the technical, procedural and policy. For example, with PII, the answer should include proper use of SSL to protect data in transit, procedural separation of administrative interfaces and definition of a suitable privacy policy to which the company can actually adhere. For a pentest or similar, the answer should be to put in place a plan to fix the network, e.g. working on f/w rules, IDS, use of VPNs to partners etc.

(Note that the putative company are developing a web site, not a software product.)

Question 2:

(a) ***Describe, in detail, a practical application data security and key management protocol (e.g. TLS, Kerberos, or IPsec). Include a description of how application data is protected, as well as key management exchanges. (10)***

Marks from 7 for a straight forward correct description. Extra 3 as a bonus, e.g. for covering D-H ciphersuites in TLS or other less-used options or for good detail of how key derivation works (use of 2 hashes in TLS).

This calls for a straight description of the relevant protocol. Level of detail required: for TLS they should cover the handshake messages and say what’s in each; they should cover the application data protocol; they should define a ciphersuite. Marks will be deducted here for rampantly incorrect things, e.g. saying that a TLS server sends its private key.

(b) ***For your selected protocol, where and how should it be used by a company selling tangible goods (e.g. books, hardware) over the web? Include a network diagram, and consider that the overall system must scale so as to handle millions of users. (10)***

Marks as for part (a). Example of getting bonus marks would be considering that TLS endpoints might be in load balancers and not in the app servers.

Here the "where" is important, the "how" should mainly consider scaling and deployment issues (e.g. getting SSLcerts). For TLS, its mostly fairly obvious but the scaling issues would need to be well addressed (e.g. use of SSL accelerators), as would the use of TLS for checkout and payment pages. For the others, its important to place them correctly, e.g. Kerberos for administrators, VPN connections between servers and to/from partners.

(c) ***What are the three most likely (note: most likely, not most impactful) threats related to the selected protocol that remain in the setup described in part (b) above? (5)***

Straight marks from 5 for good answers. Possible answers would be related to:

- Phishing sites masquerading as an SSL server
- Administrative hosts being zombied
- Server compromise via application or OS zero-day threats
- Administrator access to key materials resulting in accidental leakage
- Key leakage via redundant hardware (after de-commissioning)
- Partner site compromise

Question 3:

You are asked to design an Internet based system to allow citizens who are living abroad to vote in their national elections. Voters cast their votes for candidates in their "home" constituencies, e.g. a Dublin-registered voter living in New York might vote for candidates in the Dublin-central constituency. The number of voters involved is very small compared to the overall population at home, who vote using a traditional paper-based method. (Note: you can assume any realistic electoral system; it need not be the Irish one.)

(a) Outline the system you would develop, emphasising security considerations. (Use diagrams where appropriate.) (10)

Marks from 7 for something reasonable that's well described. Bonus marks for better detail on specific aspects (e.g. getting into some detail as to how voter anonymity could be preserved or ways to make sure votes don't get changed after the fact).

Important thing here is to consider the security requirements involved, e.g. privacy, confidentiality, registration, authentication, etc. Since it's a very hard problem to solve, any reasonable approximation at a solution is acceptable for the purposes of the exam. Generally, they should consider the flow from a paper based registration, via some sort of online signup (maybe with some out of band paper based authentication), then voting and counting. "Side" issues that need attention include recording votes (something better than just a DB entry would be good), counting votes etc.

(b) How would you validate the system before the election? (5)

Straight marks from 5. Basic approach should be something akin to open-sourcing the system and also having an expert panel and some testing (both user testing and pentesting). Consideration of the political acceptability of the system could be included.

(c) ***Describe how the system would validate the votes cast, before, or as, they are counted, including any additional "sanity" checks you would put in place. (5)***

Straight marks from 5. Various forms of clever use of crypto will be suggested. The schemes don't need to be perfect – so long as they avoid basic crypto mistakes then they're acceptable here. Additional checks could include generating alerts for potentially "suspicious" voting patterns (e.g. odd hours given the TZ, over-use of source IP addresses, funny timing etc.) Some way to (almost) manually check those votes would be good as would some sampling of votes for additional checks.

(d) ***You are a clever dishonest candidate with full knowledge of how the system works and lots of time and money. How would you attack the system? (5)***

Straight marks from 5 for clever hacks. Potential to DoS a more successful opponent in the election is an obvious one.