### **3413ICT Network Security**

### **Workshop – 10A**

**Review Questions:**

1. Compare ‘information assurance’ and ‘security assurance’.

**Information Assurance**:

The practice of assuring information and managing risks related to the use, processing, storage, and transmission of information or data and the systems and processes used for those purposes. Information assurance includes protection of the integrity, availability, authenticity, non-repudiation and confidentiality of user data.

**Security Assurance:**

Process that helps design and implement software that protects the data and resources contained in and controlled by that software.

1. What is Service Level Agreement? Discuss a case study SLA draft (no more   
   than 5 points) between an ISP and the small-scale entrepreneur where quantifiable security measures should be part of the SLA. (Hint: time taken by the process of detection of intrusions, time taken by the response to intrusions, defined user roles, etc.).

A service level agreement is written statements specifying what services a company will provide. Case study:

-When an intruder is detected an individual will be alerted within 1 hour with phone number registered with.

-After alerting the customer, authorities will be contacted within 5 hours of the alert.

-Users must not breach copyright laws and offer copyrighted data either by web hosting services or through peer to peer sharing.

-Users will be throttled if downloading or uploading excessive amounts of data to ensure other customers have a fair share of data pipe and do not experience sluggish service (Denial of service).

-Users must adhere to e-Etiquette and respect the privacy of other e-Citizens. Suspicious behaviour such as repeated unsuccessful login attempts over the network will be flagged as suspicious behaviour and will result in a phone call to discuss the login failure attempts. Repeated behaviour will result in forwarding logs to authorities for further investigation.

1. Briefly explain following threat types and identify how each of them confidentiality, availability, integrity and non-repudiation:

* **Spoofing**: non-repudiation is compromised because someone can send messages spoofed as another individual.
* **Tampering**: Confidentiality is broken if they message can be tampered with because an unintended recipient has access to the message. Integrity of the message is also no longer valid if the message has changed.
* **Denial of service:** Directly affects the availability of a service. Can also affect non-repudiation if someone sends a message to a receiving party. The receiver of the sent message may never receive the message because they are flooded with other traffic and unable to receive or reply to the request.
* **SQL Injection:** Affects confidentiality because the attacker can gain privileges to sensitive information they would normally not have access to.
* **Cross Side Scripting:** The integrity of the website becomes violated once an attacker successfully pulls off an XSS attack. Code is executed from the client which embeds itself into the server, which is served to visitors of the site.
* **Session Hijacking:** Affects all of the above: confidentiality, integrity, non-repudiation and availability. Confidentiality is broken because the attacker has access to session data, which does not belong to them. Integrity is no longer provided when someone who has access to the session can alter data in the session. Availability can be denied if someone hijacks a session and kicks the other user off their session. Non-repudiation is no longer valid when someone can send messages on behalf of another individual.

1. What is threat modelling? Why appropriate threat model is important to design a security framework? Discuss with the help of examples.

Threat modelling is a way of diagnosing the severity and impact of exploitable code. For example exploits that provide escalated privileges would be classified as a high impact attack with severe consequences. It is important to have a threat model because it helps allocate resources (man hours, and money for example) to the appropriate places based on project budgets.

1. Briefly explain relative path traversal and SQL injection threats.

**Relative path traversal**: This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory. Modifying URL parameters for example to gain access to portions of a web site that are confidential.   
**SQL injection threats**: User input is poorly sanitized allowing an attacker to insert SQL code that is executed on a server that can cause a number of problems such as: allowing a user to insert a user into a database with administrative privileges, deleting data from a database, gaining access to data that is confidential.

1. Explain, with an example (include pseudo code) how ‘sanitise and declassify’ can assist security assurance.

Sanitise and declassify is a process where code is escaped from user input fields in order to prevent the execution of code on the server. Example pseudo code:

user\_name = GET from web page(‘user name’)

user\_name\_sanitize = sanitize\_input(user\_name)

1. How does ‘Attack Surface Analyser’ help in analysing threats to the Microsoft Windows?

Attack Surface Analyzer is all in one tool for analysis for a variety of things such as: registry changes, file permissions, developers can use the tool to view changes in the attack surface resulting from the introduction of their code on to the Windows platform, tool can be used to assess the attack surface changes by the installation of an organization's line of business applications, etc.

1. Briefly explain how ‘Veracode’ service helps to identify security flaws.

Veracode helps discover security flaws by scanning websites for vulnerabilities. Vercode offers a variety of different products on their website such as:  
-Web based platform

-Static code analysis

-Dynamic analysis

-Web Application Perimeter Monitoring

-Vendor application security testing

-Mobile application security

1. Discuss potential security vulnerabilities in an e-voting machine (Hint: Compuware report).

Potential security vulnerabilities in e voting according to the Compuware report include:  
-Third party components are not regulated and could introduce poorly written code that could potentially introduce errors.

-unencrypted contents of the smart card may be read and used to disrupt the election process

-PIN is assigned to each card nationwide. If PIN is learnt, user could gain access supervisor functions

-Ballot definition file is not encrypted.

-Key for encryption is hard coded in the system. Key could potentially be read from memory.

-Data is not encrypted over data link, interception is possible.

Etc

1. Explain how ‘Build Security In’ aims to enhance security and software assurance?

‘Build Security In’ enables engineers to build a system with security in mind from the design and architecture stage. This is very different from what happens the majority of the time where software is created with no emphasis on security. Security is later added as additional layers making it more difficult to integrate with the existing system when adding security features at a later point during development.

When security is built in from the start the requirements will have the policies written out making the process easier to implement because the design stage takes in consideration the different layers from things like design, coding, testing, management/acquisition, and deployment.

1. Give a critical review (200 words) of the paper “Comprehensive experimental analyses of automotive attack surfaces” by Checkoway etal.

This paper discusses vehicle security flaws, which are exploitable over a wireless medium such as wireless network, Bluetooth, and cellular radio. Historically car hacks were deployed through having prior physical access. This paper focuses on remote attacks that are more prominent in the newer vehicles because of wireless technologies becoming cheaper and therefore more widely available.

The author briefly describes the different technologies available inside vehicles, which are potentially exploitable. CD Players, Bluetooth, Remote Keyless Entry, tire pressure monitoring, RFID car keys, and Emerging Short range channels were some of the technologies mentioned.

The sections that followed got into more detail in regards to vulnerabilities and exploitation. For example, the media player firmware was reverse engineered and the source code was analysed. Through reviewing the source there were assumptions made on input length of a song as well as another code path through the WMA parser that allowed a buffer overflow attack to take place.

Another example is Bluetooth. Through reverse engineering access to the Engine Control Unit was gained and a program responsible for Bluetooth protocol stack was identified. The telematics system was custom built and poorly coded. There were a number of strcpy calls there were not checked during a Bluetooth configuration command. This meant that code could be executed through taking advantage of unchecked functions.

Overall, the paper was educational about the new world of connected vehicles. It provided an overview of the technologies available and the weaknesses discovered during the probing and reverse engineering process. Some details were provided in regards to how to vulnerabilities were discovered and executed, but not enough to recreate the scenario based on the paper. It was understandable from a computing professional and the general public could benefit as well from reading the paper and using the help of Google to define some technical terms.

**Hands-on Exercises:**

This week you need to complete the lab 4 (available on the course website). Please ensure that you have already completed the previous labs.