LECTURE 2

COMPUTER SECURITY: Measures and controls that ensure CIA of the information processed and stored by a computer. "cybersecurity" is often used interchangeably with "information security" because both focus on protecting the CIA Triad. We want computer security to be delivered inside the CIA triangle. Hence, we need the security service to either deliver one of the dimensions, or a combination of the dimensions.

CIA TRIAD

CONFIDENTIALITY

1) Data confidentiality: Protection of personal/sensitive information from being revealed to unauthorised users
2) Privacy: Assures individuals have control over their information. Hence, if anyone else wants data, they require consent and a notation of how data will be used.

INTEGRITY (Intact of data)

1) Data Integrity: Completeness and accuracy of data (maintain fidelity of data upon transfer or temperament, usually through checksum, backup or version control)
2) System Integrity: Assures the system performs its intended function uninterrupted, without manipulation (such as a compromised machine or a hacked website)

AVAILABILITY

1) Keeping the system accessible to authorised users, and systems are delivered without interruptions such as DoS or Ransomware.

SECURITY TERMS

Adversary or Attacker: An entity, individual, agency, state that attacks, or is a threat to a system, such as a hacker or governments

Black hat hacker: attempts to gain unauthorised access into a system.

White hat hacker: works with organisations to strengthen the security of the system.

Attack: An action, that has already happened, that compromises or breaches security of a system

1) Inside attack: Someone inside an organisation that explores something they aren't supposed to.
2) Outside attack: Someone outside an organisation that tried to exploit or gain resources and perform harm

Countermeasure: Preventative action or procedure to prevent an attack

Risk: The possibility that a threat will exploit a vulnerability with a harmful result

Security Policy: Set of rules that sets up security regulation

System Resource or Asset: Data, a service, a system capability (processing power, communication bandwidth)

Threat: The potential violation of security principles

Vulnerability: A flaw or weakness in a systems design, implementation, management, or operation that can be exploited

Loss of Security: Loss of confidentiality, loss of integrity, loss of availability

LEVELS OF IMPACT DUE TO LOSS OF SECURITY

Low:	Minor damage or harm, Minor loss
Medium:	Serious adverse effect; significant damage or loss
High:	Catastrophic adverse effect; Major damage loss

Each dimension of the CIA, we have a low, medium, and high impact.

NETWORK SECURITY: a set of preventative measures that is supposed to prevent network infrastructure from a series of malicious use, to establish a secure environment so functions promised to users are carried out.

When discussing NETWORK SECURITY, we use OSI* Security Architecture

OSI* SECURITY ARCHITECTURE

1) Security Attack: An action that compromises security of the network (exchanged information)
2) Security Service: A countermeasure for a security attack (A program, algorithm, hardware implementation)
3) Security Mechanism: Is a program, machine, algorithm that actually implements security service.

TYPES OF SECURITY ATTACKS

1) **Passive**

   Release of Message Content: Intercepting a message and observes or listens to the message.

   Traffic Analysis: Intercepts a message and observes metadata (Frequency of message, packet size, period of send)

2) **Active**

   Masquerade: Impersonating someone else (Usually combined with another type of attack in order to gain information about the person they are impersonating, such as a REPLAY attack)

   Replay: Capturing a message without altering it for a Masquerade

   Message Modification: Changing the message to the receiver

   DoS: Overwhelming servers to disable a server from delivering its intended service by flooding it, resulting in it crashing or slowing down.

TYPES OF SECURITY SERVICES

Authentication: Verifying that someone is who they claim they are

Access Control: Regulates the access of resources

Confidentiality: Protection of data

Data Integrity: Keeping data untempered
Non-repudiation: Making system accessible to legitimate users
Availability: Non-interruption of service that is supposed to be delivered

Each type of security attack targets one of the security services.

Masquerade → Authentication / Access Control
Release of Message Content → Confidentiality (Message)
Traffic Analysis → Confidentiality (Meta-Data)
Message Modification / Replay → Data integrity
Denial of Service → Availability

TYPES OF SECURITY MECHANISMS

Encryption: Encoding messages so only authorised parties can read it
Digital Signature: Hash values and encryption to deliver authentication services
Access Control mechanism: Implemented techniques to enforce assess rights
Notarisation: Using a trusted party to assure safe data exchange
Password: Ensuring you are you and not someone else

Each type of security service targets one or more security mechanism

Encryption → Authentication, Confidentiality, Data integrity
Digital Signature → Authentication, Data integrity, Non-repudiation
Access Control → Access Control
Notarisation → Non-repudiation
Password → Authentication

CYBER SECURITY: A set of tools (policies, safeguards, training, etc) to protect a safe cyber environment.

Cyber Security vs Information Security:

Both refer to the CIA Triad, but Cyber Security is more specific to cyberspace and incorporates the human factor (Such as the psychological features of humans such as curiosity or laziness). Cyber Security also involves cybercrime, cyber bullying, and cyber terrorism.

New Zealand's Cyber Security Strategy:

- Exercising cyber resilience
- Having cyber capabilities
- Improving cyber security
- Increasing international cooperation

LECTURE 3

DPD protects US citizens within the EU. DPD was later replaced by GDPR which protects citizens outside the EU. GDPR also extends personal data to include audio, video, financial transactions, social media posts, IP address, IMEI numbers, browsing history and genetic information.

GDPR PURPOSES:

- Harmonises data privacy laws across EU
- Protects and empowers EU citizens data privacy
- Reshapes the way organisations across the region approach data privacy

In GDPR, Data refers to personal, de-identified, and anonymised data

GDPR TERMS

Data Subject: Any entity that has information that can be processed by someone else. EG: Me.
Data Controller: The one that determines the purpose of data collection. EG: UoA
Data Processor: An entity that engages with data on behalf of data controller EG: Google

GDPR PRINCIPLES

Lawfulness, Fairness, Transparency: Gathered data must be handled legitimately, impartially, and transparently.
Purpose Limitations: Data should not be used in ways unknown to clients
Data Minimisation: Only data directly relevant to a specified purpose should be collected
Accuracy: Data must be updated regularly
Confidentiality and Integrity: Only authorised entities can collect data, and data should be collected in a legal way
Accountability: Expectation that the subject has responsibility for what they do with their data.

GDPR KEY CHANGES

Increased Territorial Scope: Applied to all companies that handle EU data subjects
Penalties: Enforces monetary penalties (fines), up to 4% of annual global turnover of an organisation, or 20 million pounds
Consent: Request for consent in order to collect data, in a understandable and easily accessible form, and can be withdrawn

DATA SUBJECT RIGHTS

Breach Notification: Within 72 hours of a data breach, data subjects must be informed by the data controllers and processors

Right to Access: Confirmation has been to be issued by the data collector, including how and why data is being collected, ALSO, the controller shall provide a copy of personal data at any time upon subject request, for free and in electronic format. (This is a change from GDP to GDPR where data transparency and empowerment of data subjects is highlighted)
Right to be Forgotten: Entitles subjects to have controller erased, and enforces controllers to check that data is of public interest.
Data Portability: Empowers data subjects the right to retrieve their data, and the right to transmit data to another controller. This only applied to personal data, NOT anonymised data.
Privacy by Design: Data privacy should be embedded at the design (early) stage, which also enforces data minimisation, upon designing software, data collection should be minimised as much as possible.

LECTURE 4

IDENTIFICATION & AUTHENTICATION

Identification is the process in which a system entity provides its claimed identity. (UPI)
Authentication is the process of verifying an identity claimed by a system entity (Password)

TYPES OF PASSWORD VULNERABILITIES		COUNTERMEASURES
OFFLINE DICTIONARY	Gaining access to a dictionary of common passwords hash-converted and trialling and erroring these hashes against a password file.	● Prevent unauthorised access to password files (Intrusion detection to identify a compromise) ● Rapid reissuance of passwords
SPECIFIC ACCOUNT	Attacker guesses one person's password until it is correct	● Lockout mechanism after a number of failed attempts ● Progressively increase delay after each incorrect attempt.
POPULAR PASSWORD	Guessing easy / common passwords such as "QWERTY" or "12345" against a wide range of users.	● Enforcing complex password policies ● Scanning IP address / client cookies for submission patterns (Dynamic password policies) ● Intrusion Detection
PASSWORD GUESSING AGAINST SINGLE USER	Hackers know something personal about someone they know in real life, and use this knowledge to make an educated guess.	● Educating users, don't use personal information as your password ● Enforcing complex password policies
WORKSTATION HIJACKING	Attacker waits until a logged-in workstation is unattended	● Logging out after a certain period of inactivity ● Intrusion Detection to detect unusual behaviour
EXPLOITING USER MISTAKES	Mistakes such as writing down a password, or using social engineering tricks that lures a user to reveal a password, or organisation that used default passwords such as "stationery"	● Change default passwords ● Educate users on social engineering tricks
EXPLOITING SAME PASSWORD USE	Users who use the same password for multiple sites, an attacker gains access to one, and attacks multiple streams	● Educate users on choosing different complex passwords for different sites.
ELECTRONIC MONITORING	Communicating a password in plaintext is vulnerable to eavesdropping	● Never send a password in plaintext ● Technical solutions for secure transfer of passwords.

HOW PASSWORDS ARE PROTECTED

LOADING PASSWORD: SALT WITH HASH Salt is a random number, combined with a hashed password to complicate hash to plain text comparison. Hence to authenticate a user, you need a salt, a userID, and a hash value.

- Counters EXPLOITING USER MISTAKES, specifically, if someone uses the same password for multiple sites.
- Counters OFFLINE DICTIONARY as you need more than just a hash value, especially with a large salt increasing the difficulty of guessing.

MULTI-FACTOR AUTHENTICATION: On top of a password, use a smartcard, fingerprint, voice patterns, or behaviour analysis.

TYPES OF PASSWORD ATTACKS

Man-In-The-Middle Attack: An attacker intercepts communication between a victim and a service where they are trying to change their password. The attacker communicates between the victim and service. Victim think they are talking to service, but they're actually giving all their information to the attacker, the attacker masquerades the victim, talks to the service to provide credentials to get a password change.

LECTURE 5

Access Control: Granting access to authorised individuals to provide resources

ACCESS CONTROL REQUIREMENTS

1) Reliable Inputs: Access Control MUST be given to authorised users, using an ID and Password. For example, security systems at UoA should authenticate a UPI, and assign different interfaces whether a UPI belongs to a teacher or a student, and hence each have different accesses.
2) Principle of Least Privilege: Grants the minimum set of access rights to do a job, just enough to complete a job
3) Administrative Duties: Only a special person or department should be able to manage access rights including granting, revoking or updating access rights.

ACCESS CONTROL ELEMENTS r (right) s (subject) o (object)

1) Subject: An entity that can access objects (EG: a user)
2) Object: System resource (EG: Files, directories, etc)
3) Access Right: The relationship between a subject and their corresponding access of an object, as well as their permissions (read / write / execute)

ACCESS CONTROL SYSTEM

1) Access Control Function f(s, o, r): Looks up r for the combination of o and r.
2) Security Administrator: An entity that manages access rights
3) Auditor: Does not take a role in access control system, but monitor and inspects operation of access control (s, o and r)

ACCESS CONTROL MODELS

1) DAC: Resource owner decides who can access and what level of access. Vulnerable to Trojan Horses, and these mechanisms are only good for honest users.

   ACCESS CONTROL MATRIX (ROWS = ENTITY, COLUMNS = HOW THEY CAN ACCESS)

	FILE 1	FILE 2	FILE 3	FILE 4
ALICE	ORW		ORW	
BOB	R	ORW	W	R
CHARLIE	RW	R		ORW

   ACCESS CONTROL LIST (EG: We can extract four lists, one per file)

   FILE 1 → Specifies all users that have access to FILE 1 → Alice, Bob, Charlie

   CAPABILITY LIST (EG: We can extract three lists, one per entity)

   Bob → Read FILE1, Owns/Reads/Writes FILE2, Writes FILE3, Reads FILE4.

2) RBAC: Maps roles to access rights, so a manager will have more access compared to employees. RBAC supports complex access control and reduces errors in administration. It complies with the principle of Least Privilege, where we assign just enough privilege for each entity to get the job done.
   - User: A human
   - Permissions: approval of a mode of access to some object
   - Roles: Job title (Manager, clerk, CEO, etc)
   - Assignments: User-role (relationship between user and role) and role-perm (relationship between role and permission)
   - Session: Mapping users to roles (NOT ROLE AND PERMISSION)
   - Constraints: Assigning a user to a role, and containing what different users can access what exactly.

LECTURE 6

MALWARE: A set of instructions that run on your computer and make your system do something an attacker wants it to do.

MALWARE TYPES

1) Virus: An executable script that replicates itself in a computer
2) Worms: A computer program that runs independently and propagates itself over a network
3) Trojan Horse: Appears to have a useful function, but has a hidden function
4) Spyware: Software that collects information from computer and transmits it elsewhere (Monitors keystrokes, network traffic)
5) Ransomware: Stealing data and asking money for it to get it back
6) Botnets: Receive command from remote Command and Control servers, usually used for DoS

WAYS OF MALWARE TYPES

1) Document-Based Malware: Receiving an email, content is enabled by victim, and malicious code is downloaded and installed.
   - Malware embeds codes into documents, PDF, spreadsheets, files, etc
   - Can access password or infect email contacts

    - Overrides commands, damages computer, or further spreads malicious codes
    - Tampers with addings, deleting, and changing words or images in documents or hard drive
    - Infiltrate email, sending unsolicited emails to contact lists.
2) Advanced Persistent Threat (APT): Requires advanced tools and runs over an extended period, costing a lot to a victim.
3) Zero-Day: Realising there is an attack but not having time to protect or come up with a viable countermeasure
4) Backdoor: Any mechanism that bypassess a normal security check, allowing unauthorised access to data.
5) Social Engineering: Using inside knowledge about a system and psychological properties of humans, to trick a user to make security mistakes. (EG: Email Phishing)

MALWARE COUNTERMEASURES

- Anti-virus
- Phishing-detection system
- Blacklisting IPs
- Firewalls
- Malware Detection
- Machine Patching
- Detect and Prevent (Intrusion detection / intrusion prevention)
- Education for users

LECTURE 7

HUMANS: Are the weakest link and are chronically responsible for the failure of security systems. Humans are incapable of securely storing high-quality cryptographic keys and have unacceptable speed and accuracy. Humans like to bypass barriers, and even if password policies are put into place, humans will still find a lazier way to create a weak password.

STATIC PASSWORD RULES

Password policies that apply to everyone, such as lowercase/uppercase, numeral, punctuation, special characters, a minimum length, etc

DYNAMIC PASSWORD RULES

Password policies that involve a process first where they collect personal information such as address, background information, fullname, email, etc in order to design a dynamic password, where the policies cannot contain any of the given personal details.

USER SECURITY : INTERDISCIPLINARY

User security involves many theories, frameworks, models and methods from many backgrounds such as psychology, sociology, science, behavioural economics, HCI, etc

SECURITY VS USABILITY (HCI)

1) Security is not the primary interest of the user.
2) Many applications that handle security issues interrupt a user's primary task, and users may find it a distraction and would rather ignore it.
3) HCI looks at normal behaviours, while security looks at abnormal behaviours, which is overlooked in HCI.

What makes Usable Security Different:

1) Security and privacy problems are well-understood
2) It also looks at both types of users, legitimate and attackers.

Hence, it can pinpoint where users behave in an unpredictable way, can detect stress, pressure, tension, and whether users are attentive or not (lazy users). Hence, traditional HCI itself is insufficient in keeping a system safe. Security and Usability are better used together, to make Usable Security. It is about creating and designing a system that intersects security and usability.

Security (Strict)	**Usability / HCI (Optimistic)**	**Usable Security**
Human behaviour is secondary to security	Security is secondary to human behaviour.	Both human and security factors are primary constraints
Humans considered in their role as attackers	Concerned about human error, but not human attackers	Concerns about both normal users and adversaries
Involves threat models	Involved task models, mental models, cognitive models	Involves threat models AND task models
Focus on security metric	Focus on usability metrics	Considers usability and security metrics together
User studies rarely done	User studies are common	Both user and attacker studies

USER-SELECTED GRAPHICAL PASSWORDS

A new password policy was introduced where instead of a plain-text authentication, the user selects a picture of face that they have selected as their password. Here we are trading off some security to suability. This system is friendly for humans as it is easier to remember a face rather than a complicated string.

Security	**Usability**	**Usable Security**
● *What is the space of possible passwords?* ● *How can I make the password space larger to make the password harder to guess?* ● *How are the stored passwords secured?* ● *Can an attacker gain knowledge by observing a user entering her password?*	● *How difficult is it for a user to create, remember, and enter a graphical password?* ● *How long does it take?* ● *How hard is it for a user to learn the system?* ● *Are users motivated to put in effort to create good passwords?* ● *Is the system accessible using a variety of devices for disabled users?*	● *All the security/privacy and usability questions* ● *How do users select graphical passwords?* ● *How can we help them choose a password harder for attackers to predict?* ● *As the password space increases, what are the impacts on usability for human selections?*

THE HUMAN THREAT:

- Malicious humans
- Clueless humans
- Unmotivated humans
- Humans constrained by human limitations (Disability)

<u>When applying usability, we need to understand:</u>

- misleading / unhelpful user interface looks like
- requiring users to make decisions for which the user is not qualified to make
- Unreasonable amounts of attention / effort

HUMANS IN THE LOOP: A METHOD TO CONDUCT USABLE SECURITY

- Do they know they are supposed to do something?
- Do they understand?
- Do they know how to do it?
- Are they capable?
- Will they actually do it?

How to implements Usable security

1) Identify points where system relies on humans to perform security-critical functions
2) Find ways to partially / fully automate some of these tasks
3) Identify potential failure models for the remaining tasks (Using Human-In-The-Loop)
4) Find ways to prevent these failures.

EMAIL PHISHING RED FLAGS

- An "urgent" subject such as 'SOMEONE HAS YOUR PASSWORD!'
- Email greets the victim with their actual name 'Hi John'
- Email provides details such as a date, IP Address, and Location
- Email lures users to change their email password by clicking a click '<https://bit.ly/1PibSU0>, bit.ly is a shortening URL address service.

So in this example, with the increasing use of technology, attacks become creative and so we must treat it. We first identify that this system relies on humans to perform security-critical functions (1) as scanning for phishing emails can not be conducted in an automated manner (2), we should find ways to identify potential failure models (3), such as the 'Game Design Framework', which is a framework that sums costs and benefits of a security implementation, such as perceived threat, safeguard cost and effectiveness, to calculate human avoidance motivation and behaviour, to find ways to prevent these failures (4).

LECTURE 8

VULNERABILITIES: Are features that exist in applications that we can utilise. Acts as a prelude of an attack, so it can become a risk, but it can also be used to construct threats or risks that point to systems which we use to mitigate possible problems. The main source of vulnerabilities are inputs.

OWASP: Open Web Application Security Project: Looks for top 10 issues in web applications, updates the list every 3-4 years to see most common vulnerabilities that we can utilise when building programs to embed security. Each year, OWASP points can change, stay the same, replace, swap positions, merged, or even just be removed entirely.

TYPES OF OWASP VULNERABILITIES

1) **INJECTION**: Untrusted input gets processed as part of command or query. Attacker injects and executes unintended commands and can access unauthorised data.
    - Command Injection
    - Code / SQL Injection

      EG: Typing 'google.com' to get an IP Address. Injecting code where the input changes to 'google.com; uname -v', hence the lookup will generate the nslookup, and also execute the second part, returning all information of sensitive information.

      EG: SQL = SELECT id FROM users WHERE username='...' AND password = 'password' OR 1=1' Hence the code will always run and the attacker gains access to the corresponding username.

   COUNTERMEASURE: Separate data from application logic. Use parameterised queries, such as prevention = shlex.quote(domain_name) command = "nslookup{}".format(prevention)
2) **BROKEN AUTHENTICATION:** Incorrectly implementing authentication by allowing default, weak, or well-known passwords, or application session timeouts not set properly.

   COUNTERMEASURE: MFA, do not use default credentials for admin users, enforce strong password policies, limit or increasingly delay failed login attempts, securely use session ID and invalidate after logout, idle, and timeouts.
3) **SENSITIVE DATA EXPOSURE**: Not protecting financial, healthcare and personally identifiable information via encryptions, or cipher suites, hashes with salts.

   COUNTERMEASURE: Identify sensitive data as per privacy laws (GDPR), do not store data unnecessarily, protect it via encryption, delay factors and hashes and salt.
4) **XML EXTERNAL ENTITIES (XEE):** Older versions of XML are poorly configured and do not examine XML that use external resources, or XML's that try to direct to external resources. Directing external sources can reveal personal information, it can scan internal systems, remote code execution, and possibly a DoS if an endless loop is clicked onto unintendedly.

   COUNTERMEASURE: Patch and upgrade XML processors, Disable XEE, validate incoming XML
5) **BROKEN ACCESS CONTROL**: Bypassess RBAC / DAC and unauthenticated users access unauthorised resources, so non-admin accounts may access admin accounts.

   COUNTERMEASURE: Create a deny by default list, log access control failures, set API rate limits (restricting the number of requests to minimise harm from automated attacks)
6) **SECURITY MISCONFIGURATION:** Attackers attempt to exploit unpatched flaws, for example, an application that has detailed error messages, which tells the attackers the contents of the flaw.

   COUNTERMEASURES: Review and upgrade configurations, and an automated process to validate the effectiveness of configurations and settings, and test this in all possible environments.
7) **CROSS-SITE SCRIPTING (XSS):** An attack in which an attacker injects malicious executable scripts into the code of a trusted application or website. Attackers often initiate an XSS attack by sending a malicious link to a user and enticing the user to click it.

   COUNTERMEASURE: Separate untrusted data from browser content, and use frameworks that automatically escape XSS by design.

CVE & NVD DEFINITIONS

CVE: Common Vulnerabilities and Exposures (Identifiers for publicly known vulnerabilities)

NVD: National Vulnerability Database (Scores vulnerabilities using Common Vulnerability Scoring System CVSS)

LECTURE 9

Programmers face problems with embedding Privacy in Design because

- Practical Issues: Difficult to understand terms, such as 'PbD' which is a privacy policy which urges programmers to incorporate privacy into design. They do not understand the principle itself, and so there is a gap between privacy requirements and techniques.
- Privacy concepts may not work in software development, the technology requires expensive computational resources.
- Developers have personal opinions (bypassing obstacles) which take precedence over privacy requirements

We should educate software developers and enhance their coding behaviour through motivation.

LECTURE 10

PROGRAM ANALYSIS: Process of analysing the behaviour of programs for the goal of finding problems in code

STATIC ANALYSIS: Do not run programs, just read scripts, comments to draw a logic flow and identify the values of data. Tries to discover any vulnerabilities in execution through full coverage via checking ALL parses. It is PROGRAM-CENTRIC, and is a type of white box testing (You need to understand scripts). It is scalable, does not depend on computational power. It is vulnerable to accuracy issues providing false positives by misclassifying behaviour as misbehaviour.

DYNAMIC ANALYSIS: Run the program in the actual execution environment, and if vulnerabilities exist, they show up. Difficult to generate and test ALL possibilities. It is INPUT-CENTRIC and a type of black box testing.

CONCOLIC ANALYSIS: Static analysis ran, followed by dynamic analysis.

CONTROL FLOW GRAPH: Static analysis based on CFG, which shows all possible paths.

TYPES OF STATIC ANALYSIS

- **PROGRAM SLICING:** Reducing program to the minimum form that still produces the selected behaviour. The reduced program is a slice. It reduces cost / time, and is only an approximation of the original program to locate sources or errors more easily and faster. Generally, finding a slice is difficult and an unsolvable problem.
- **SYMBOLIC EXECUTION:** Instead of specific values of inputs, we have symbolic values, such as 'x' which represents a range. Both paths symbolically executed independently, forming an execution tree.

In static analysis, we are not observing the actual behaviour, we only observe path conditions. Although, in dynamic analysis, we observe actual behaviour, there might be too many options which is its limitation. Hence why concolic analysis is utilised.

LECTURE 11

CODE OBFUSCATION: Making a program confusing aimed to make reverse engineering harder for attackers.

LAYOUT OBFUSCATION	Layout = Non-executable part of the program (comments, debugging information, variable names) ● Hide comments / debugging information, rename variables to confuse etc.
DESIGN OBFUSCATION	Making design confusing, such as in a hierarchical, layered way, etc So attackers spend more time trying to understand the code, or not understand it at all. ● Splitting / Merging classes to make it hard for attackers to understand the idea of code. ● Reversing a class
DATA OBFUSCATION	Making exact values of data hard to guess ● Hide value of data via encryption, instead of coding exact input, use a hash function ● Variable splitting, split v into p and q, such that v = p xor q ● Changing lifetime of variables such as swapping local and global variables
CONTROL OBFUSCATION	Making the control flow of the code confusing ● Opaque Predicates: if/else statements are made harder to guess, for example instead of using if(A>0), use if (A>b^2+b % 2) or if(A>b[0]) ● Control Flow Flattening: break or change the structure of CFG

LECTURE 12

THE UNCONTROLLABLE NATURE OF THREATS: Threats are unavoidable (1) and we have no control over it (2). So we can only prevent threats.

TYPES OF THREATS

UNINTENTIONAL: Environmental, Human, Accidents, Failure

INTENTIONAL: Greed, Anger, Desire to Damage

COMMON ATTACKERS

- Criminals
- Advanced persistent threats (APTs)
- Vandals
- Saboteurs
- Disgruntled employees
- Activists
- Other nations
- Hackers

THREAT VULNERABILITY PAIR: Occurs when a threat exploits a vulnerability. Vulnerability provides a path for the threat that results in a harmful event or a loss. Both the threat and the vulnerability must come together in order to result in a loss.

EXAMPLE: An ex-employee is unsatisfied by a company and acts a threat as they are thinking of something bad to do. The vulnerability is that the system has not removed the ex-employee from the system. The threat-Action is that the employee accesses proprietary data.

US GOVERNMENT RISK MANAGEMENT INITIATIVES

NIST: Belongs to Department of Homeland Security

NCCIC: Department in government that discusses how to integrate security

US-CERT: Team in charge of spreading different documents issued by NCCIC

CVE: List of common Vulnerabilities

VULNERABILITY MITIGATION TECHNIQUES

Policies and Procedures	● How to assign more systems to more sources ● GDPR policy compliance
Documentation	● Document system resources, plans, etc
Training	● Educating and training sessions to teach all employees on recent data breaches and teach precautions
Separation of Duties	● For RBAC, separate roles from system resources
Configuration Management	● Patch softwares as soon as possible, ensure systems are updates
Version Control	● Keep back-ups of databases or documents to restore upon the case of an attack
Incident Response	● When outlining policy, outline step by step how to tackle these attacks
Continuous Monitoring	● Periodically review policies to ensure they are up to date
Technical Controls	● Any risk management tools
Physica Controls	● For example, preventing a computer room with a lock and key

MANAGING VULNERABILITIES

a THREAT utilises a VULNERABILITY which incurs a RISK which can result in some LOSS. Mitigation techniques control the flow from Vulnerability to risk, to dampen the pair to reduce a risk, and therefore a loss, or by blocking the source of the loss. Hence, vulnerability mitigation is about prevention.

RISK MITIGATION TECHNIQUES (FOR PUBLIC FACING SERVERS)

Remove or change defaults	● Remove or change admin default passwords such as 'stationery'
Reduce attack surface	● For injection, the attack surface is the code, it is observable by external users, thinking of how to reduce this surface
Keep systems up to date	● Keeping policies, patches, updates
Enable firewalls	● Blocks internal and external data transfers. Firewalls sit between transfers and observe packet sizes, and raise alarms when something looks unusual.

BEST PRACTICES FOR MANAGING THREATS	**BEST PRACTICES FOR MANAGING VULNERABILITIES**	**BEST PRACTICES FOR RISK**
- Create a security policy - Purchase insurance - Use access controls - Use automation - Input validation - Provide training - Use antivirus software - Protect the boundary	- Identify vulnerabilities - Match the threat-vulnerability pairs - Use as many of the mitigation techniques as feasible - Perform vulnerability assessments.	- Harden servers - Use configuration management - Perform risk assessments - Perform vulnerability assessments

PRIORITISING RISK

- Organisations have limited funds
- Threats cannot be eliminated
- All vulnerabilities do not result in a loss
- Identify important risks
- Use resources to identify current risks

LECTURE 13

COMPONENTS OF RISK MANAGEMENT

1) **RISK ASSESSMENT**
    a) **RISK IDENTIFICATION**
    b) **RISK ANALYSIS**
    c) **RISK PRIORITISATION**
2) **RISK CONTROL**
    a) **RISK MANAGEMENT PLANNING**
    b) **RISK RESOLUTION**
    c) **RISK MONITORING**

RISK MANAGEMENT PLAN

1	RISK IDENTIFICATION A list of threats, vulnerabilities, and recommendations, and once identified, form pairs of threat-vulnerabilities
2	RISK ANALYSIS Cost of risk + costs of recommendations = cost-benefit analysis Also, report all of the above, use risk statements to communicate a risk and resulting impact. Useful Tool: FishBone Diagram: Finds all possible causes of a result, such as attacker + no IDS + DoS + Open Ports on Firewall = System out of service. Profitability vs Survivability
3	RISK PRIORITISATION Useful Tool: Risk Management Plan using Risk = Threat * Vulnerability * Asset to calculate the weight of the risk in order to prioritise each risk.
4	RISK MANAGEMENT PLANNING Draw a schedule and make a living document of each recommended solution by breaking into steps with a timeline and detailed descriptions of the project. Useful Tools: Milestone Chart → Shows the dates/times of when project and subprojects should be completed by. Useful Tools: Gantt Chart → Shows a full project schedule, tasks that can overlap and cannot overlap are outlined.
5	RISK RESOLUTION
6	RISK MONITORING

LECTURE 14

STRIDE THREAT MODEL

TYPE	**DESCRIPTION**	**SECURITY CONTROL**
Spoofing	Threat action aimed at assessing and use of another user's credentials, such as username and password.	Authentication
Tampering	Threat action intending to change persistent data, such as records and the alteration of data in transit between two computers over a network	Integrity
Repudiation	Threat action aimed at performing prohibited operations in a system that lacks the ability to trace the operations	Non Repudiation
Information disclosure	Threat action intending to read a file that one was not granted access to, or to read data in transit.	Confidentiality
Denial of Service	Threat action attempting to deny access to valid users such as by making a web server temporarily unavailable or unusable	Availability
Elevation of privilege	Threat action intending to gain privileged access to resources in order to gain unauthorised access to information or to compromise a system	Authorisation

COUNTERMEASURES

1) In-Place Controls: In place in operational system with documentation
2) Planned Controls: Specified implementation data, details in planning documents
3) Control Categories: Procedural, technical, and physical

PROCEDURAL → Policies, Procedures, Security plans, Insurance, Awareness & Training
TECHNICAL → Login Identifiers, System logs, Firewalls
PHYSICAL → Locked doors, video cameras, Fire detection and suppression

SEVEN DOMAINS

1) User Domain
2) Workstation Domain (Computer)
3) LAN Domain (Hub)
4) LAN-to-WAN Domain (Router / Firewall)
5) Remote Access Domain (Broadband internet)
6) WAN Domain (Firewall, Mainframe, Application & Web, System/Application Domain)

LECTURE 16

PRIVACY: Privacy is more attached to personal space, to ourselves. It is NOT about data.

CONTEXTUAL INTEGRITY: Privacy depends on the context

PRIVACY AS A SOCIAL CONSTRUCT

Society (A group of people) affects people's views, including their views on privacy. Privacy therefore differs from society to society, for example, societies with greater education surrounding privacy would have different views from societies without it. This also holds for societies with regulation differences, or societal norm differences. Real life example: A family shared a phone, where messages, photos, data were all kept on one phone which was used collectively between family members. Contextually, if one member leaves the country, they now hold a weak belief in privacy, thinking it is okay to have private information shared.

LECTURE 17

PERSONAL DATA: Any information that is directly or indirectly relation to an identified or identifiable person. According to GDPR, vague data, such as 'Age range', is NOT personal data. Although, if data only has one person linked to it, such as 'Job Title CEO in company Starbucks' or 'Home address', it is considered personal data.

DIGITAL FOOTPRINT: Can be increased by intentionally sharing information about yourself on Facebook or Linked in for example. Hidden data attached to intentional data sharing activities, such as what ads you've clicked, or information you've shared to friends. Shopping websites can tell if you like to exercise if your browsing history is workout clothes. Every click leaves a mark, a footprint.

- Read privacy policies
- Use the privacy settings
- Ask to delete your dta permanently if you no longer using a service
- Think about what data to share
- What technologies can you use to protect (VPN, anonymous web browser, Incognito mode, ad blockers)

Proposed Privacy by Design by Ann Cavoukian

1	Proactive not reactive, preventative not remedial → Plan it from the start
2	Privacy as the default → Default settings should be privacy protective
3	Privacy embedded into design → Prioritises privacy in the product design
4	Full functionality, positive sum, not zero-sum → Should not be a trade off for the user
5	End-to-end security, lifecycle protection → Collection, process, share, store at each stage as long as the data exists.
6	Visibility and transparency → Communicate properly about the data handling practices
7	Respect for User privacy, keep it user-centric → Respect the use, be empathetic

THREAT MODELLING

Before identifying threats

1) Identify personal data handled by the product or service
2) Understand the flow of identified data

LINDDUN THREAT MODELING FRAMEWORK

LINKING	Data can be linked to learn more about the person even if it does identify a person.
IDENTIFYING	Identity is revealed through leaks, deduction, inference
NON-REPUDIATION	A person can not deny a claim
DETECTING	Arrive at a conclusion about a person through observation
DATA DISCLOSURE	Excessively collecting, sharing, processing, storing personal data

UNAWARENESS & UNINTERVENABILITY	Insufficient transparency, feedback to users or less involvement of the user
NON-COMPLIANCE	Violation of best practices, standards and regulations

LECTURE 18

PRIVACY ENHANCING TECHNOLOGIES (PET): If a security breach occurred, if the data was in plain-text, this affects privacy, but if it was encrypted or obfuscated, then privacy is still maintained even after the attack. PET's are technologies that allow the utilisation of personal data while at the same time reducing privacy risks (a win-win).

ANONYMITY: Removing the link between data and the data subjects

- Negative side: Cannot track for cyber-bullying, or black market activities

SOME TYPES OF PETs

- **Pseudonymisation**: processing data in such a way that it is not possible to attribute them to a specific person without the use of additional information. Separate the identifiable information in a mapping table, and use a fake identity, such as random numbers (RNG), Cryptographic Hashing, encryption, or a counter (increment per new entry).
    - Scalability issue: The more data entries, the larger the table
    - Multiple identifiers issue: At UoA, a person can be Tutor and Student at the same time, we don't want to reveal a link
    - Linkage Attacks: People can be uniquely identified using only ZIP, DoB, and sex. Hence we should use K-Anonymity.
- **K-Anonymity:** for every combination of identifying attributes in a dataset, there are at least “K minus 1” other people with the same attributes. K = Number of people in a group. Quasi-Identifiers = identifiers multiplied by a factor that together, identifies a person.
    - Cannot use this to identify people, only used for analysing patterns. (EG: Census data)
    - Can still be breached with background knowledge. Someone can get data from all systems, combine, and pinpoint individuals.
- **Differential Privacy**: ensures that an individual will experience no difference whether they participate in information collection or not. This means that no harm will come to the participant as a result of providing data.
    - Cannot use this to analyse numeric or financial data, as adding noise will not preserve its accuracy.
- **Federated Learning**: ensuring that data remains decentralised. Apple records voice for Siri. We train machines to not learn data, instead, we send models at the phone, hence, data does not leave your phone.
- **Homomorphic Encryption:** the conversion of data into ciphertext that can be analysed and worked with as if it were still in its original form.
    - Performance issues: takes time
- **Zero-knowledge Proof**: every bit of information is treated with complete confidentiality. Proving something without using a password, for example 'what is the sum of the last 3 digits of your password'
- **Synthetic Data**: artificial data generated to mimic real data.