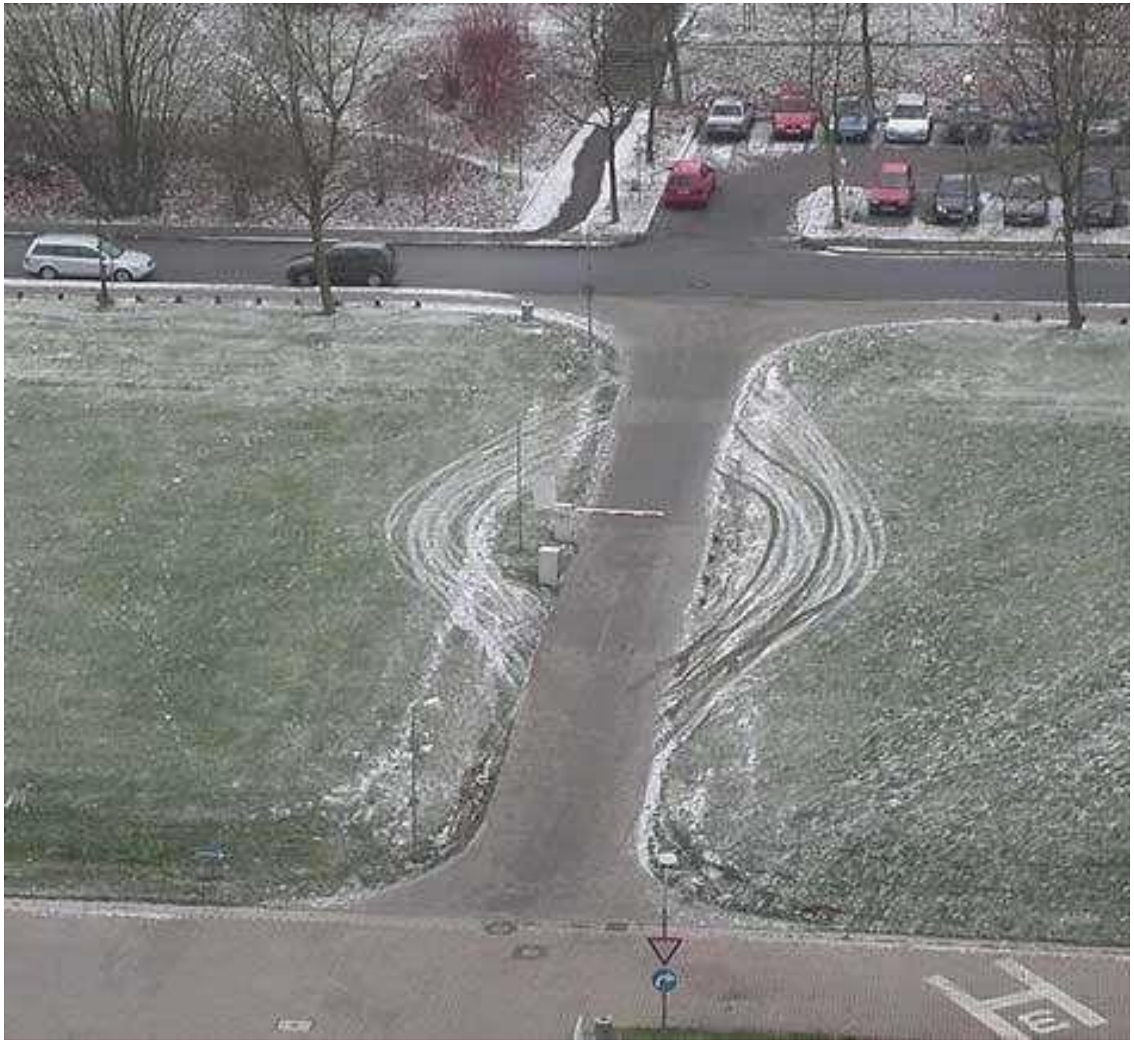
Application Security

L1 – Introduction to Applications Security

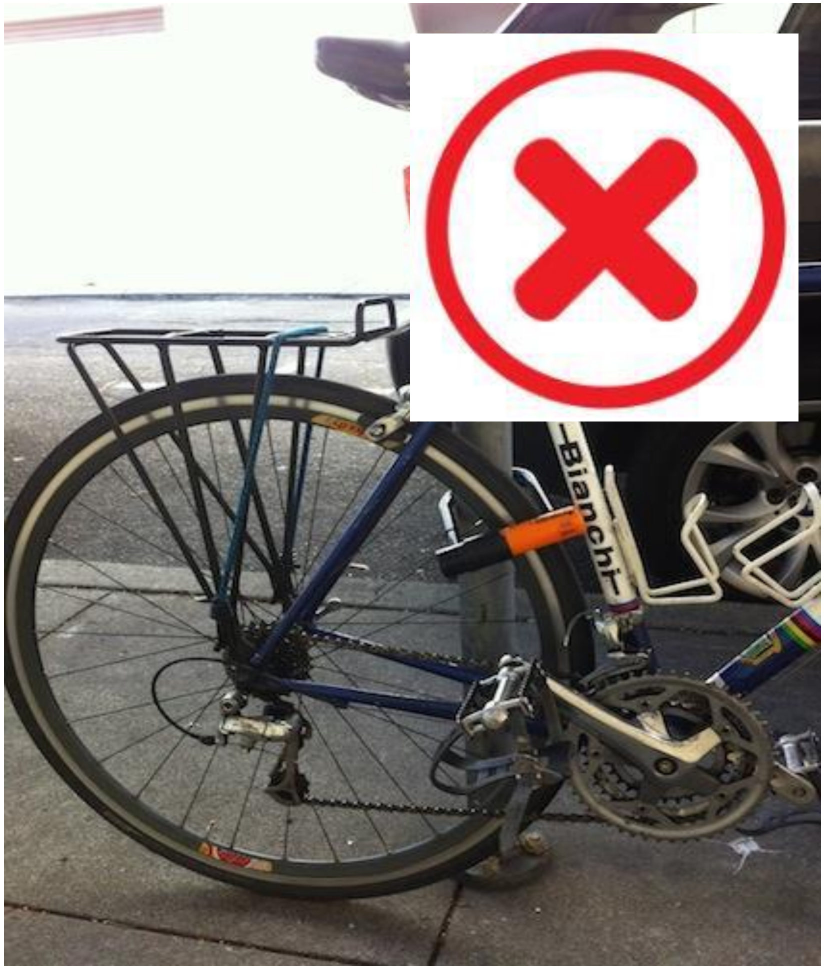
Nanyang Polytechnic

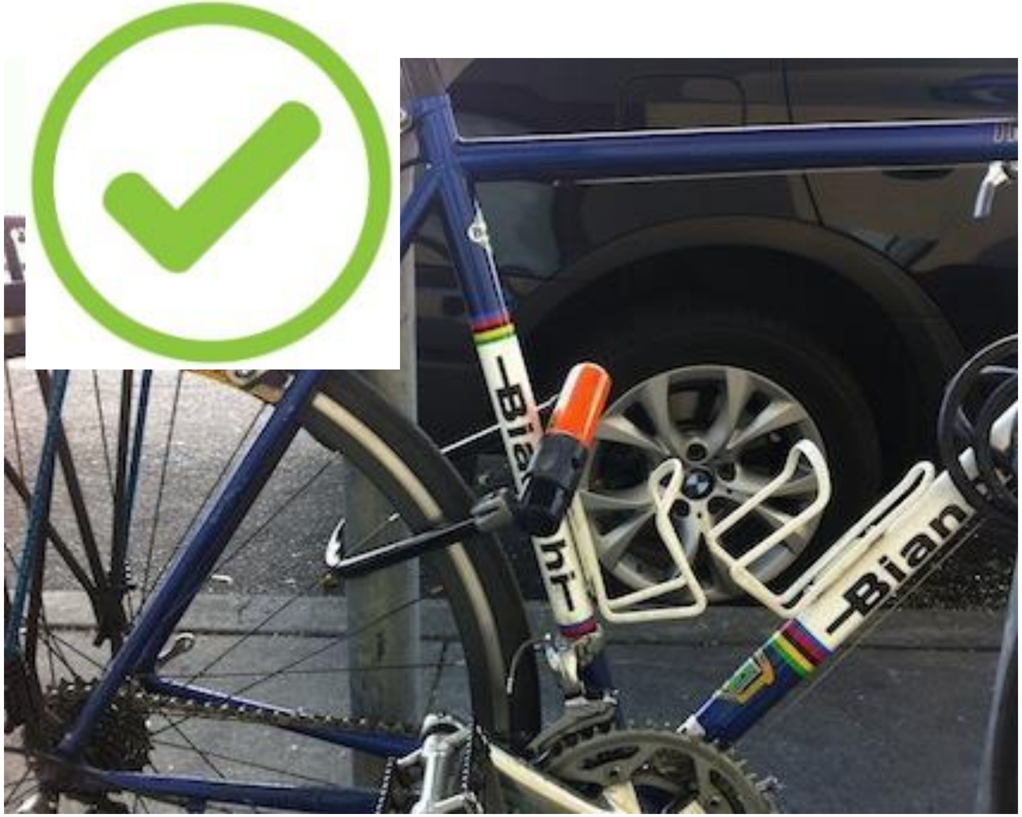
Some times the easiest way to break system security is often to go bypass it

Examples ?



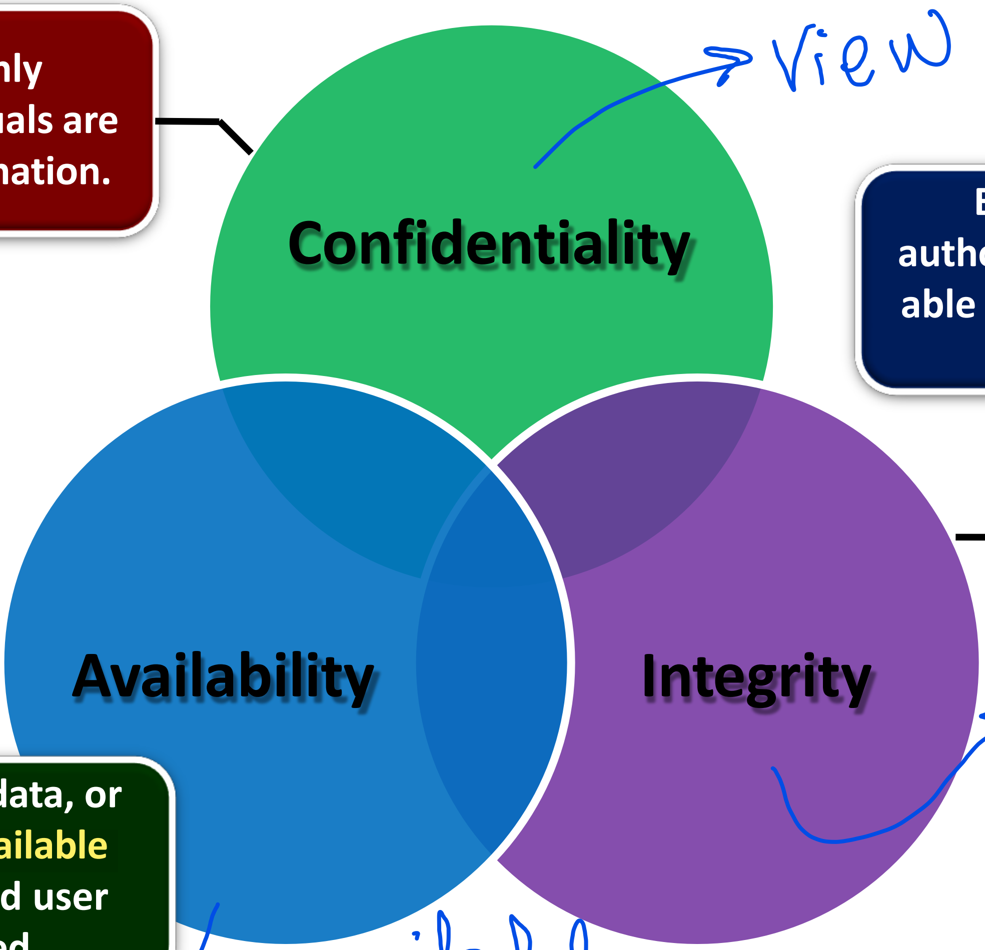
Securing your bike





Core Principles of Information Security (CIA)

Ensures that only authorized individuals are able to view information.



Confidentiality

→ view info

Ensures that only authorized individuals are able to change (or delete) information.

Availability

Integrity

→ change info

Ensures that the data, or the system, is available for the authorized user when required.

→ available for authorized users

CIA Extensions

The increased use of networks for commerce requires two additional security goals for the CIA of security.

Authenticity

↳ who he claims to be

- Ensures that an individual is who he claims to be.

Non-repudiation

↳ cannot deny
↳ sender sent

- Ability to verify a sender had sent the message to a recipient.
- Either party cannot deny later that they did not send or received the message.

Vulnerability and Threat

- Vulnerability: Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source. Repositories:
  - Common Vulnerabilities and Exposures http://cve.mitre.org/
  - Common Weakness Enumeration http://cwe.mitre.org/
- Threat: Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.

Source: SOURCE: FIPS 200, NIST IR 7298 Revision 2, *Glossary of Key Information Security Terms*

Quantifying Security

- Is your home secure ?
- Is the street secure at night ?
- Is the plane secure ?

❑ Can you quantify performance ? SPEED

❑ How do you quantify security ?

Security

- To protect means
  - Find weaknesses/flaws
    - Known weaknesses – based on history and testing
    - Unknown weaknesses – based on anomaly
  - Solution to the flaw
  - Prevent future occurrences
  - Detect attack pattern
  - Report the attack

Protection = Prevention + (Detection + response)

Prevent, Detect and Response

Protection = Prevention + (Detection + response)

Example of Preventive Measures = ?
Example of Detection Measures = ?
Example of Responsive Measures = ?

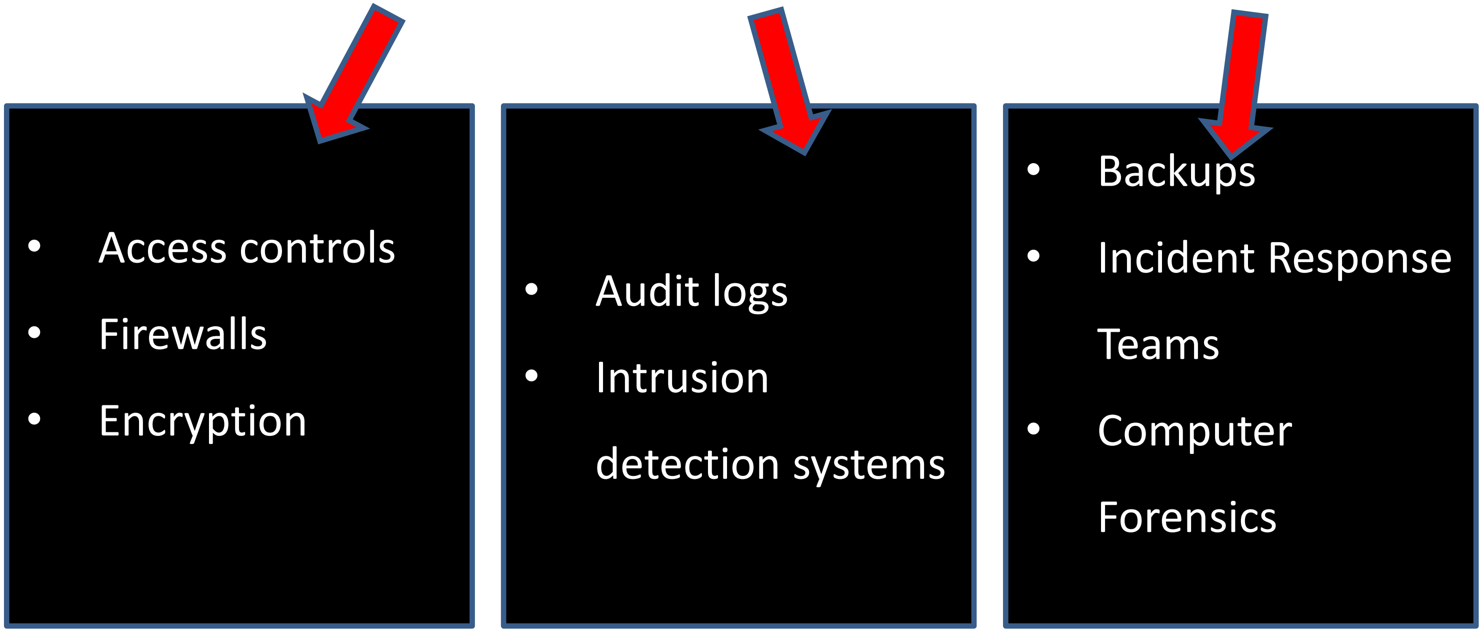
- Audit logs
- Intrusion detection systems

- Backups
- Incident Response Teams
- Computer Forensics

- Access controls
- Firewalls
- Encryption

Prevent, Detect and Response

Protection = Prevention + (Detection + response)



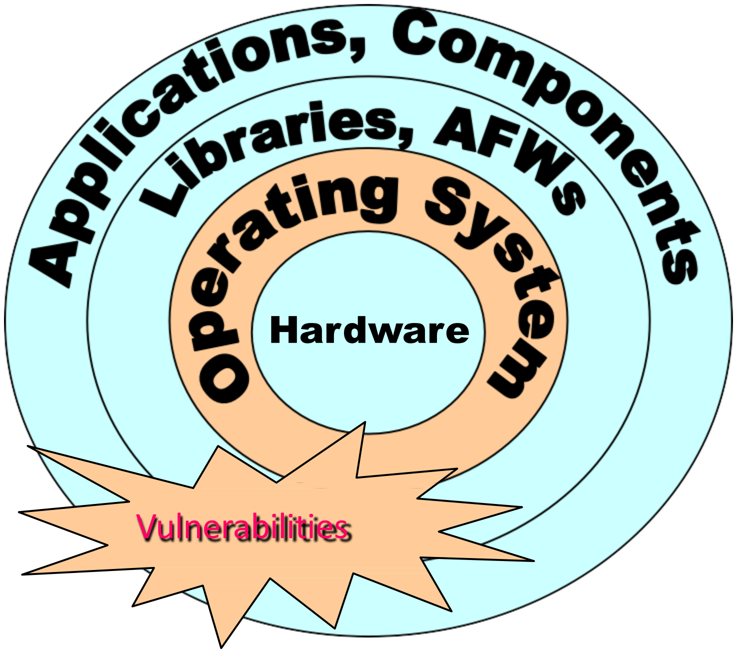
- Access controls
- Firewalls
- Encryption

- Audit logs
- Intrusion detection systems

- Backups
- Incident Response Teams
- Computer Forensics

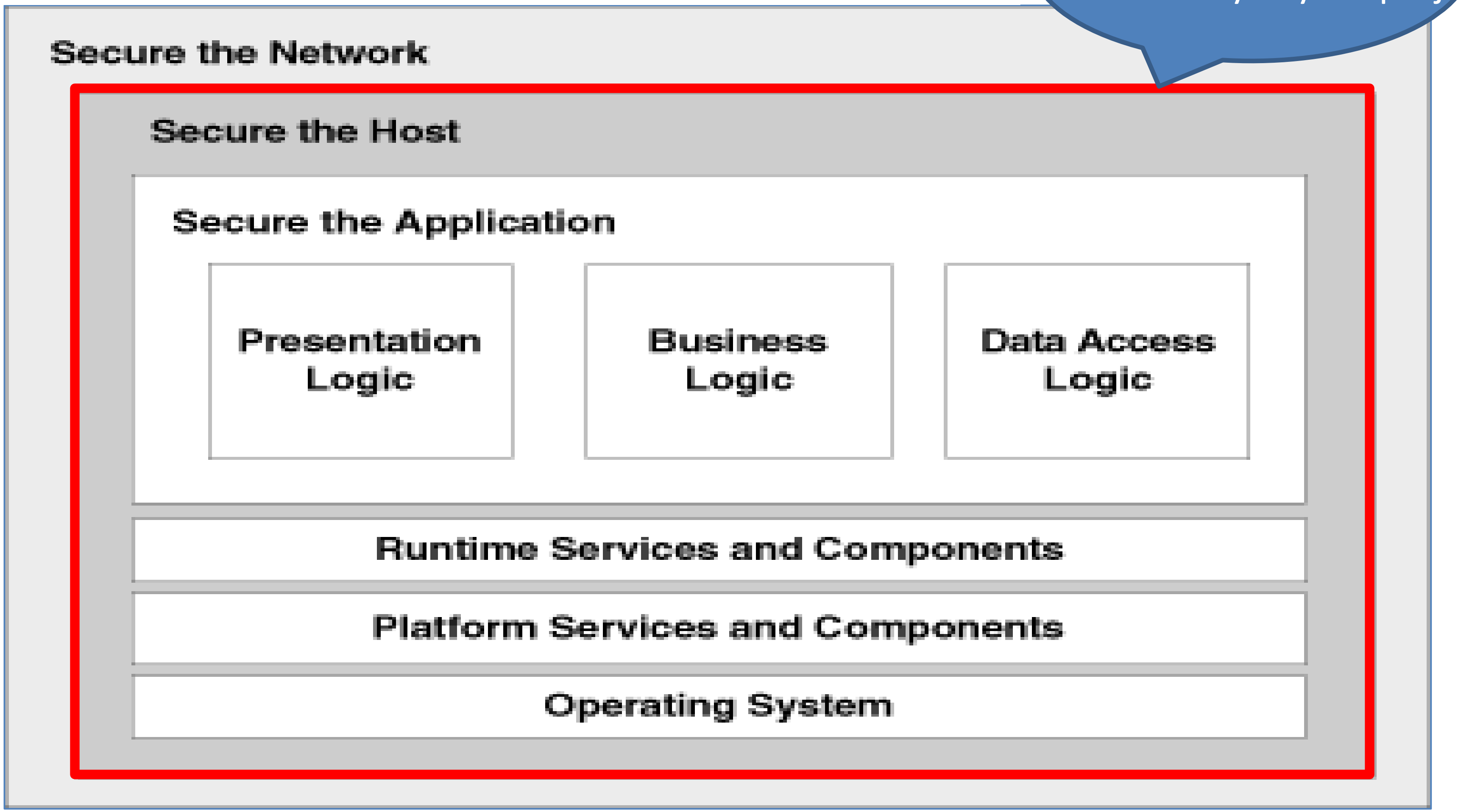
Systems & Application Security

- Operating Systems support the execution of applications, which uses system libraries, application frameworks
- Any vulnerabilities on the operating system results in potential threats against the applications running in the system.



Secure Your Network, Host, and Application

Implement Host and Web App security in your proj



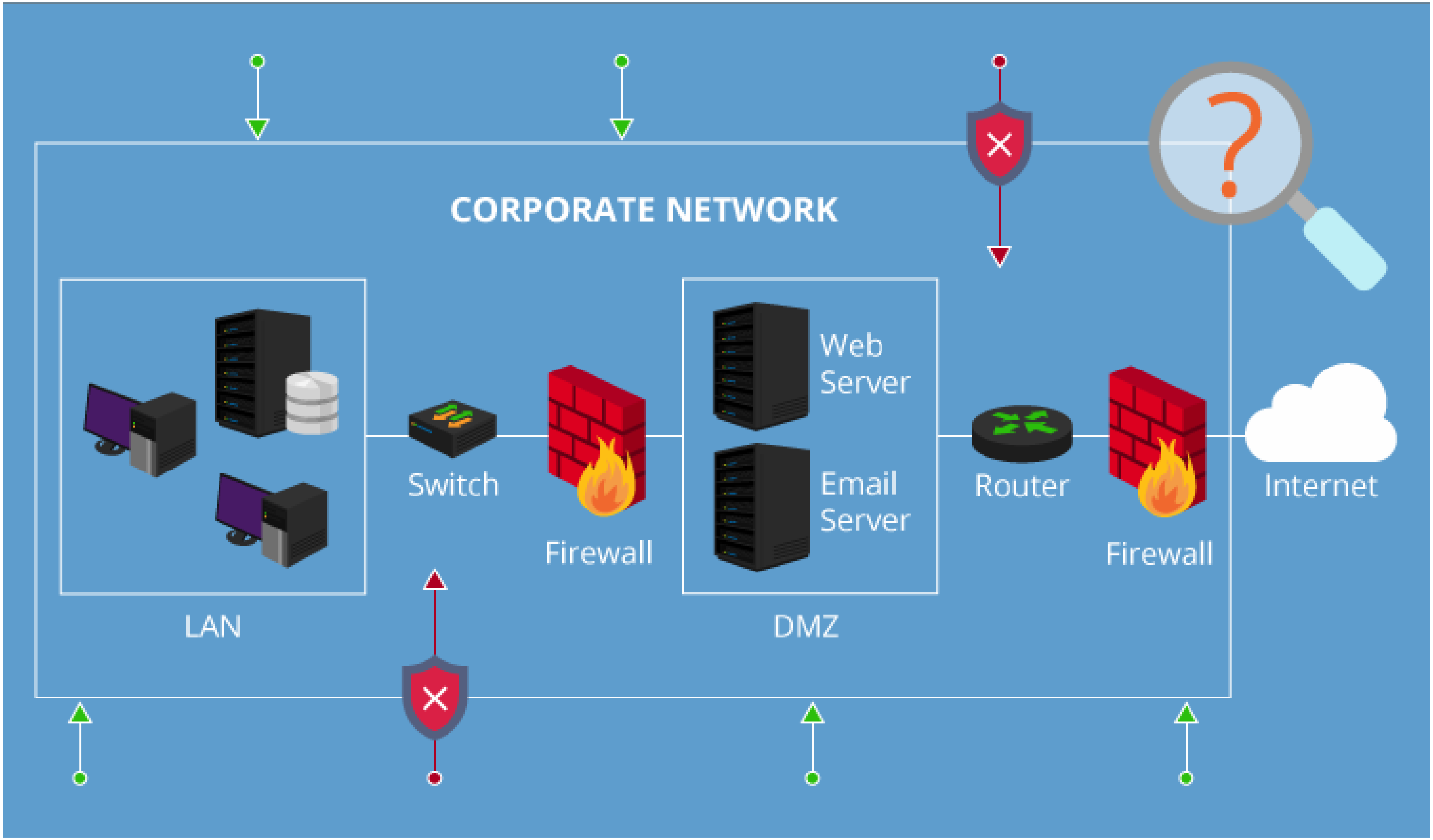
Source : msdn.microsoft.com

Securing Your Network

- Relies on secured network infrastructure

Component	Description
Router	Routers are your outermost network ring. They channel packets to ports and protocols that your application needs. Common TCP/IP vulnerabilities are blocked at this ring.
Firewall	The firewall blocks those protocols and ports that the application does not use. Additionally, firewalls enforce secure network traffic by providing application-specific filtering to block malicious communications.
Switch	Switches are used to separate network segments. They are frequently overlooked or overtrusted.

Typical Corporate Network



Project areas

Host Security (20%)

- Server Hardening (DB, Web server, OS)
- Account Policies
- Audit
- Malware

Web Security (80%)

- App hardening
- Security Features
- Secure coding
- Testing

ASP.NET

Preventive vs Detective

Preventive (Passive)

- Host
  - Firewall
  - Hardening
  - Secure Accounts
  - Log activities
  - Data Protection
- Web App
  - Write Secure code
  - Data Protection
  - Testing

Detective (Active)

- Host
  - IDS (inspect log files)
- Web App
  - ? Hint : Entry points
  - Anti bot (Captcha)
  - Account lockout
  - Authentication mechanism
  - etc

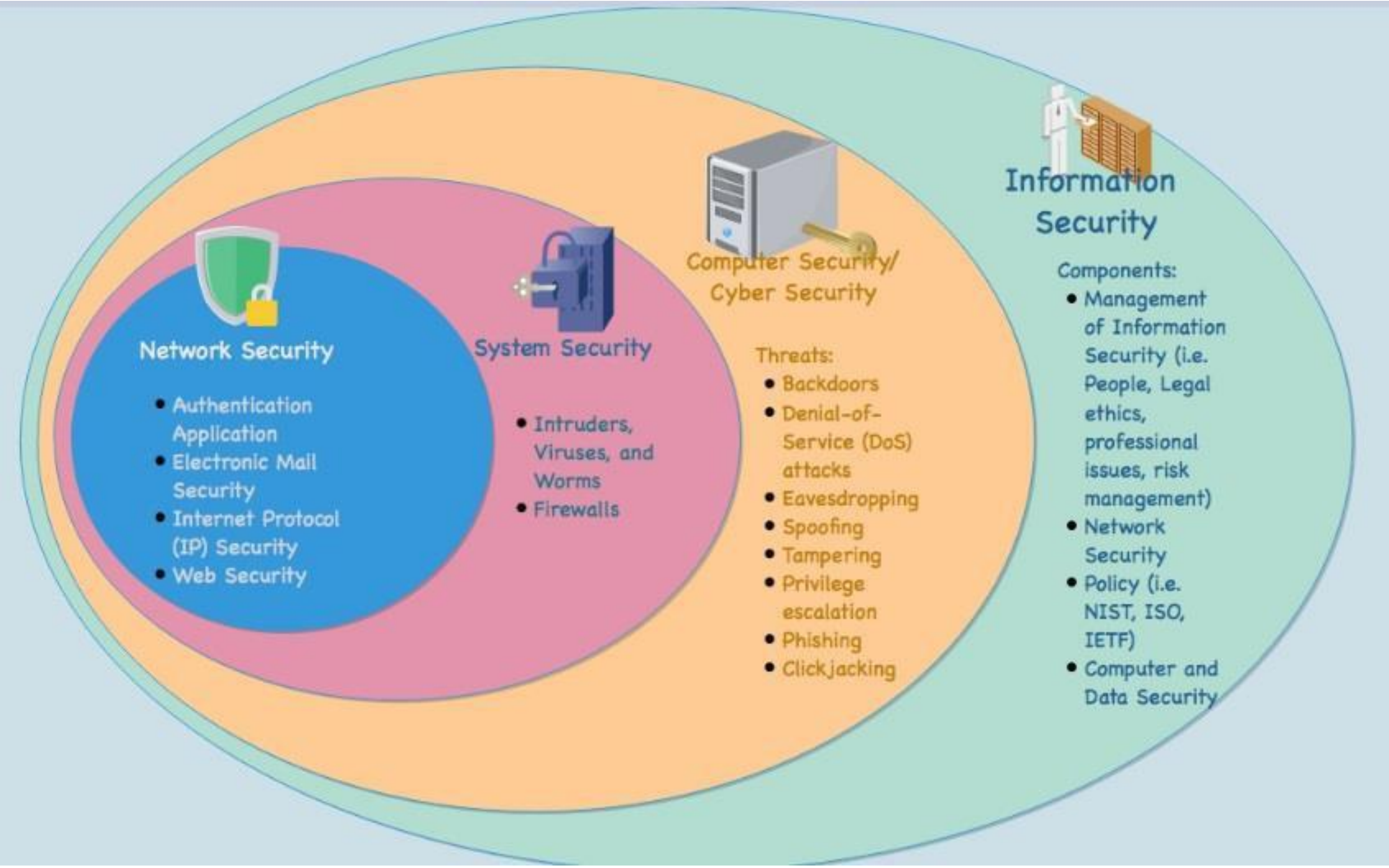
Securing Your Host

- Relies on Web server, application server, or database server configuration

Category	Description
Patches and Updates	When new vulnerabilities are discovered, exploit code is frequently posted on Internet within hours of the first successful attack. Patching and updating your server's software is the first step toward securing the server.
Services	The service set is determined by the server role and the applications it hosts. By disabling unnecessary and unused services, you quickly and easily reduce the attack surface area.
Protocols	To reduce the attack surface area and the avenues open to attackers, disable any unnecessary or unused network protocols.
Accounts	The number of accounts accessible from a server should be restricted to the necessary set of service and user accounts. Additionally, you should enforce appropriate account policies, such as mandating strong passwords.
Files and Directories	Files and directories should be secured with restricted NTFS permissions or perform necessary encryption.

Securing Your Host

Category	Description
Shares	All unnecessary file shares, including the default administration shares if they are not required, should be removed. Secure the remaining shares with restricted NTFS permissions.
Ports	Services running on a server listen on specific ports to serve incoming requests. Open ports on a server must be known and audited regularly to make sure that an insecure service is not listening and available for communication.
Auditing and Logging	Auditing is a vital aid in identifying intruders or attacks in progress. Logging proves particularly useful as forensic information when determining how an intrusion or attack was performed.



Network Security

- Authentication Application
- Electronic Mail Security
- Internet Protocol (IP) Security
- Web Security

System Security

- Intruders, Viruses, and Worms
- Firewalls

Computer Security/ Cyber Security

Threats:

- Backdoors
- Denial-of-Service (DoS) attacks
- Eavesdropping
- Spoofing
- Tampering
- Privilege escalation
- Phishing
- Clickjacking

Information Security

Components:

- Management of Information Security (i.e. People, Legal ethics, professional issues, risk management)
- Network Security
- Policy (i.e. NIST, ISO, IETF)
- Computer and Data Security

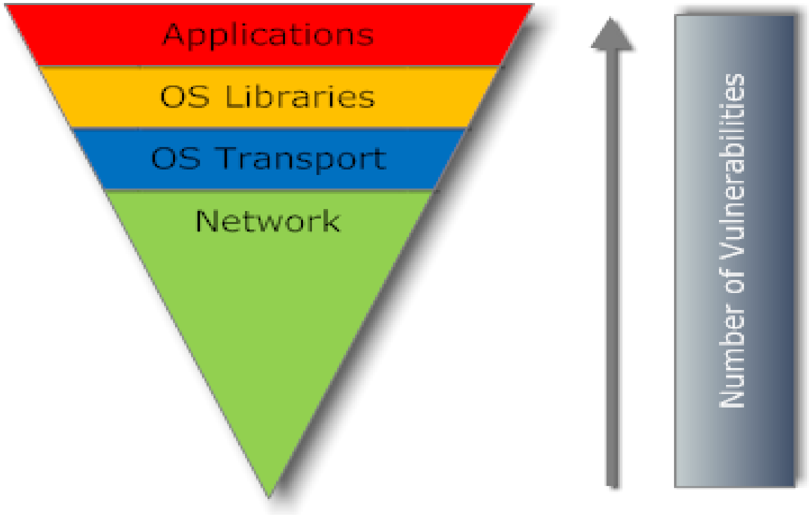
Securing Your Web Application

- Web application security is the process of securing confidential data stored online from unauthorized access and modification.
- The aim of Web application security is to identify the following:
  - Critical assets of the organization
  - Genuine users who may access the data
  - Level of access provided to each user
  - Various vulnerabilities that may exist in the application
  - Data criticality and risk analysis on data exposure
  - Appropriate remediation measures
  - *How to be used vs How to be misused*

Why Applications Security?

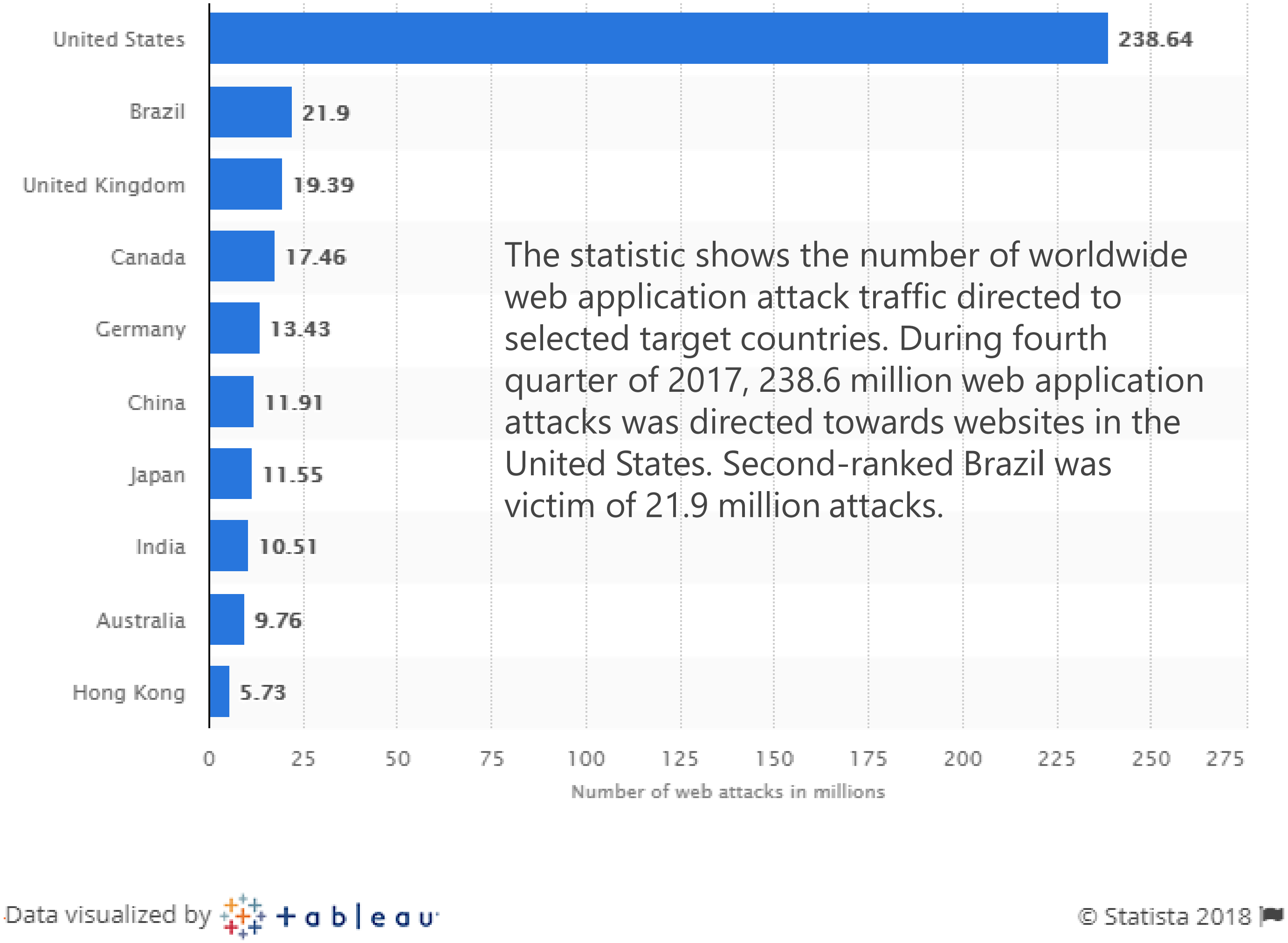
"….retired Rear Adm. Betsy Hight (vice president of Hewlett-Packard's cybersecurity practice). Hight said the network no longer is the primary target for attacks. Seventy percent of attacks now target applications, and that is a shift in the threat landscape that has not been adequately dealt with …..”

Source: http://gcn.com/articles/2011/04/04/security-threats.aspx



Application Vulnerabilities Exceed OS Vulnerabilities

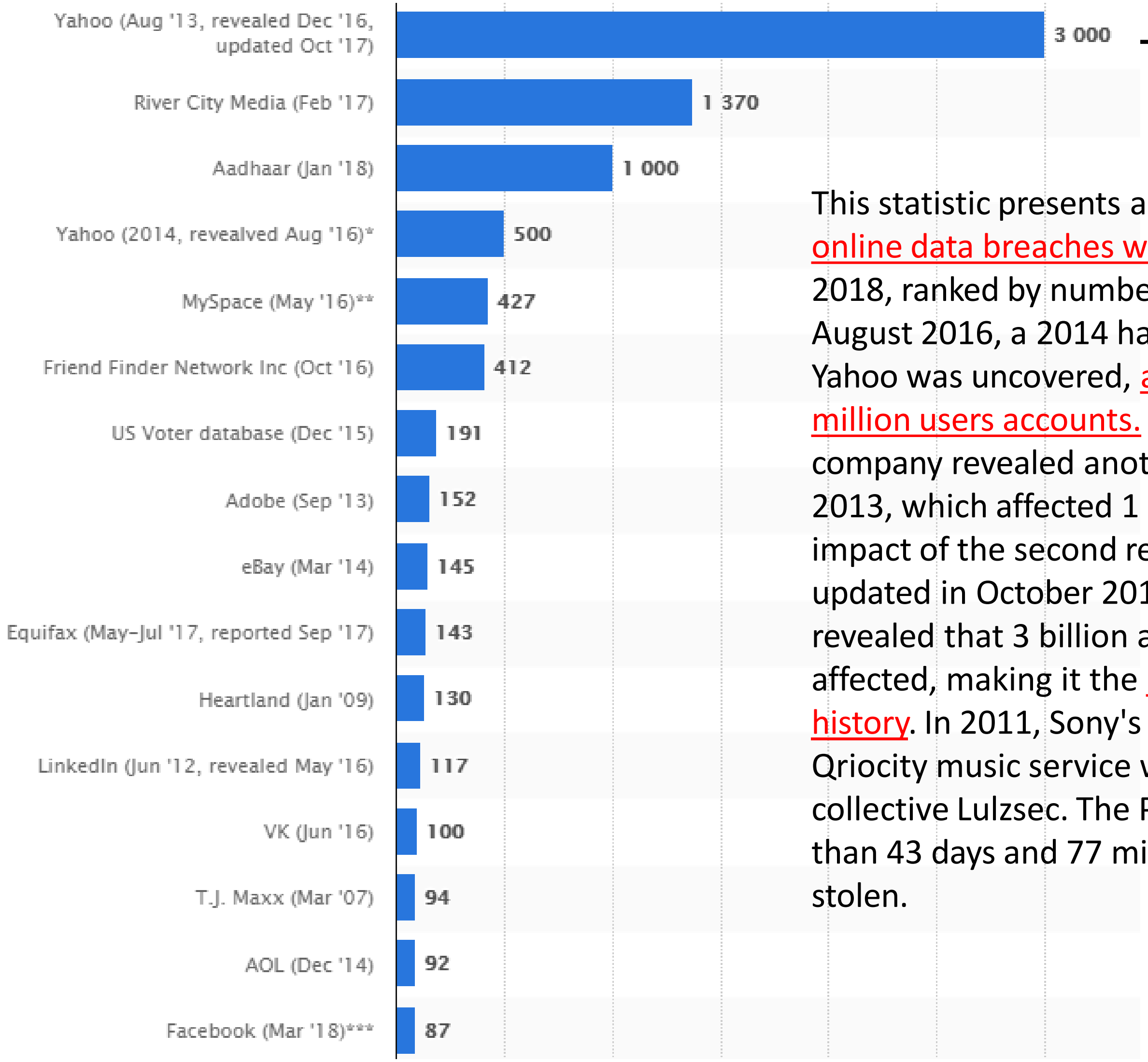
Number of global web application attacks during 4th quarter 2017, by target country (in millions)



The statistic shows the number of worldwide web application attack traffic directed to selected target countries. During fourth quarter of 2017, 238.6 million web application attacks was directed towards websites in the United States. Second-ranked Brazil was victim of 21.9 million attacks.

https://www.statista.com/statistics/440575/internet-application-attack-traffic-target-countries/

Number of compromised data records in selected data breaches as of March 2018 (in millions)



This statistic presents a selection of the biggest online data breaches worldwide as of March 2018, ranked by number of records stolen. In August 2016, a 2014 hack of online platform Yahoo was uncovered, affecting at least 500 million users accounts. In December 2016, the company revealed another hack dating back to 2013, which affected 1 billion user records. The impact of the second reported Yahoo hack was updated in October 2017, when the company revealed that 3 billion accounts had been affected, making it the largest data breach in history. In 2011, Sony's PlayStation Network and Qriocity music service were attacked by hacking collective Lulzsec. The PSN was offline for more than 43 days and 77 million data records were stolen.

https://www.statista.com/statistics/440575/internet-application-attack-traffic-target-countries/

Why Web Application Security Problems Exist?

- Root Cause:
  - Developers are not trained to write or test for secure code.
  - Network security (firewall, IDS, etc) alone does not help to protect the Web Application Layer. A combination of security methods are necessary!
- Current State:
  - Organizations test tactically at a late & costly stage in the SDLC.
  - A communication gap exists between security and development as such vulnerabilities are not fixed.
  - Testing coverage is incomplete.

Securing Your Application

- Create security profile to determine the security strength of an application.

Category	Description
Input Validation	How do you know that the input that your application receives is valid and safe? Input validation refers to how your application filters, scrubs, or rejects input before additional processing.
Authentication	"Who are you?" Authentication is the process where an entity proves the identity of another entity, typically through credentials, such as a user name and password.
Authorization	"What can you do?" Authorization is how your application provides access controls for resources and operations.
Configuration Management	Who does your application run as? Which databases does it connect to? How is your application administered? How are these settings secured? Configuration management refers to how your application handles these operational issues.
Sensitive Data	Sensitive data refers to how your application handles any data that must be protected either in memory, over the wire, or in persistent stores.

Securing Your Application

Category	Description
Cryptography	How are you keeping secrets, secret (confidentiality)? How are you tamperproofing your data or libraries (integrity)? Cryptography refers to how your application enforces confidentiality and integrity.
Parameter Manipulation	Form fields, query string arguments, and cookie values are frequently used as parameters for your application. Parameter manipulation refers to both how your application safeguards tampering of these values and how your application processes input parameters.
Exception Management	When a method call in your application fails, what does your application do? How much do you reveal? Do you return friendly error information to end users? Do you pass valuable exception information back to the caller? Does your application fail gracefully?
Auditing and Logging	Who did what and when? Auditing and logging refer to how your application records security-related events.

Nanyang Polytechnic

A Web Developer

C#, Python Scripting

Database

Testing

No sleep

Intuitive GUI

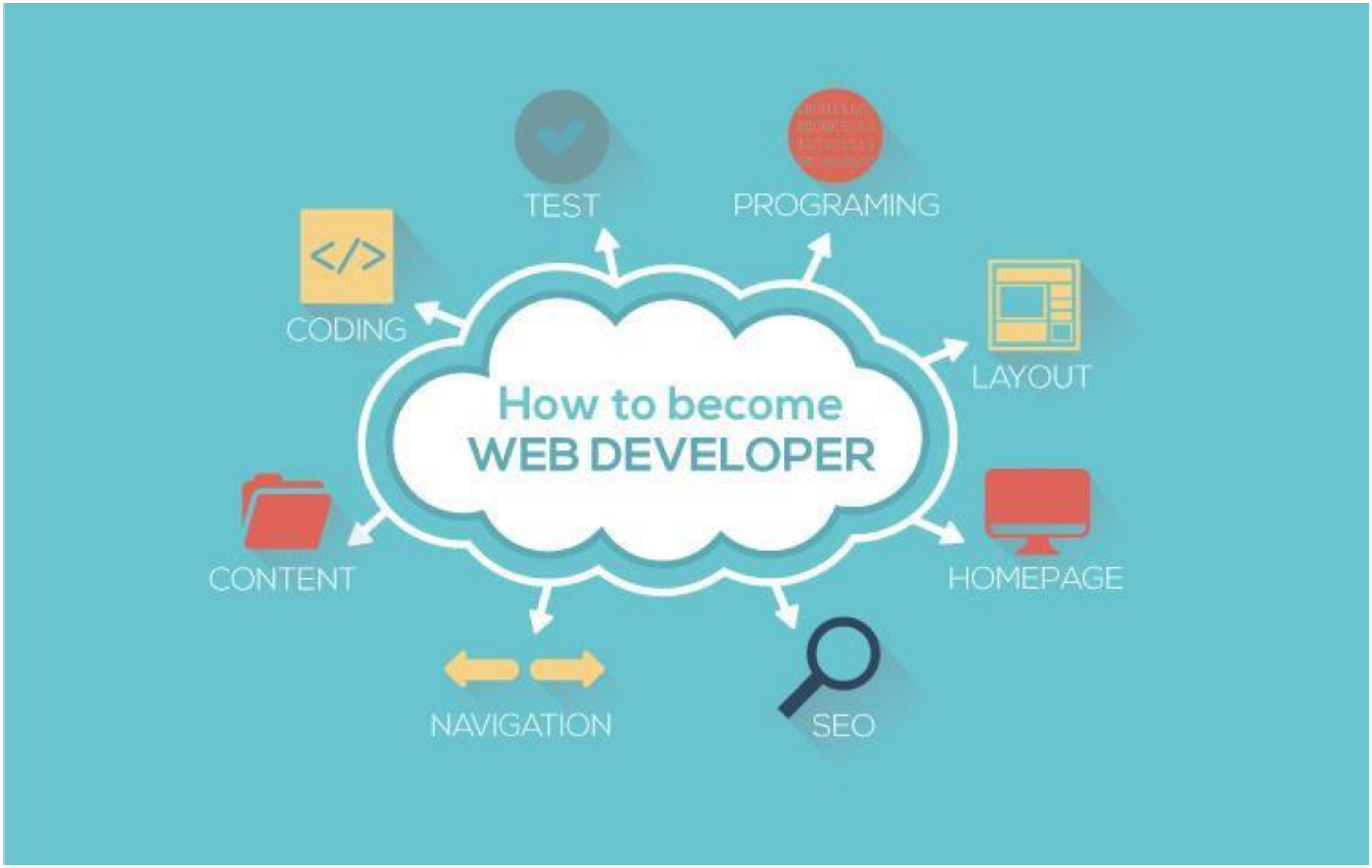
APIs

Documentation

Google

…

A Web Developer



WORLD WIDE WEB
TYPOGRAPHY
ASP
PHP
SCRI
STAND
INTER
SEO
XML
BUSINESS
DEVELOPER
ONLINE
ARCHITECTU
SERVE
CGI
BROWSER
CLIENT
JAVA
DESIGNER
CSS
WEB
USER
PROJECT
CONTENT
Security
DEVELOPMENT
PHP
COLORS
RESPONSIVE
CONCEPT
WEBSITE
HTML
SKILLS
DATABASE
MULTIMEDIA
PHICS
HOMEPAGE
STYLE
TERNET
VELOPMENT
SOFTWARE
BANNER
CODE
LAYOUT
FLASH
AJAX
MOBILITY
VOLUTION
USTRY
PROFESSIONAL
TREND
OPTIMIZATION
LAYOUT

SDLC

- *The **Software development life cycle** (SDLC) is a structure imposed on the development of a software product* -Wikipedia
- Traditional SDLC:
  - **Requirements**
  - **Design**
  - **Implementation**
  - **Testing**
  - Deployment
  - Maintenance

Secure Software **Requirements**

- Requirements for:
  - Confidentiality (e.g. all data in transit must be encrypted)
  - Integrity (e.g. all input must be validated against a set of allowable input)
  - Availability (e.g. availability must be 99.9999%)
  - Authentication (e.g. must have 2 or more factor of authentication)
  - Authorization (e.g. access to secret files restricted to users with secret or top secret clearance)
  - Audit/Logging (e.g. audit logs must be kept for 3 years)
  - Session Management (e.g. session id must be encrypted)
  - Errors and Exception Management (e.g. all exceptions are to be explicitly handled)
  - Etc…

Source: Official (ISC)^2 Guide to The CSSLP by Mano Paul

Secure Software **Design**

- Confidentiality – encryption algorithm, key size, digital certificates
- Integrity – hash functions, referential integrity
- Availability – redundancy
- Authentication – biometric, token
- Authorization – auditor role can perform read to all logs
- Auditing/Logging – what to log, validate integrity of logs by hashing the logs

Source: Official (ISC)² Guide to The CSSLP by Mano Paul

Secure **Design** Considerations

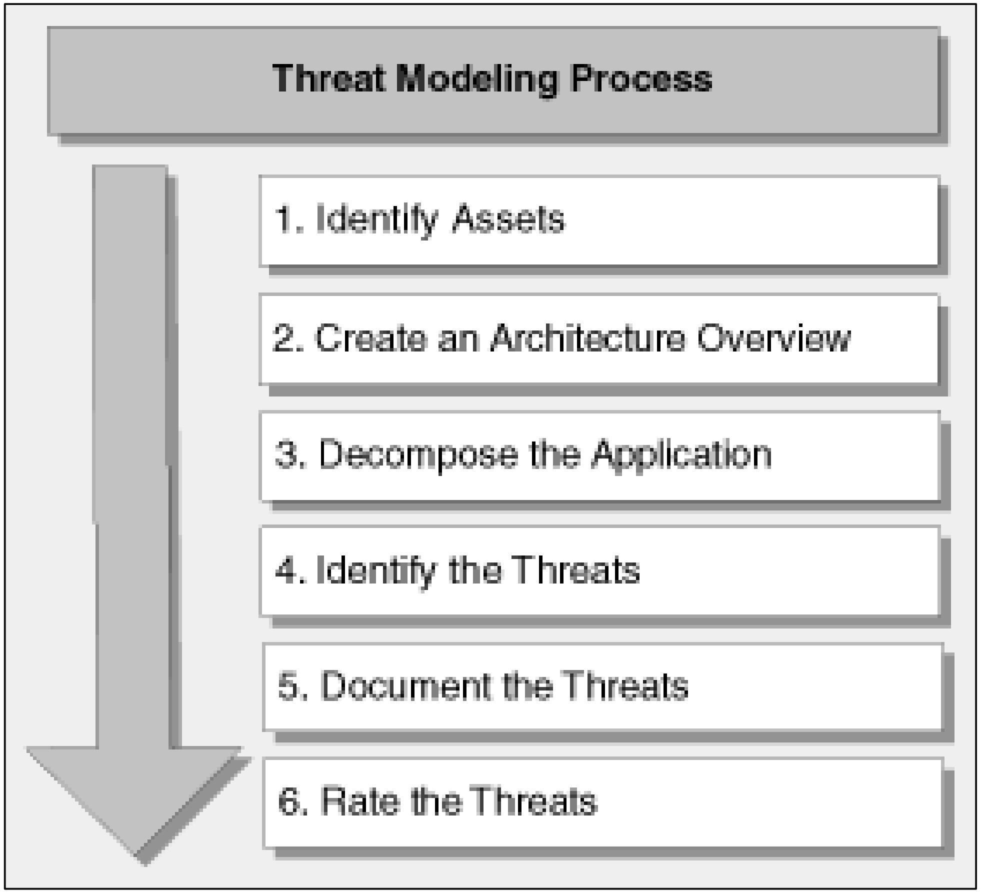
- Least Privilege
  - Allow each user/process minimum privileges to do his/its work.
- Separation of Duties
  - Programmer should be different from code reviewer
- Defense in Depth
  - Design software that will not easily break down just because one security mechanism has been broken.
- Fail Secure
  - Design your program to recover or terminate ‘gracefully’ upon any form of failure.
- Psychological Acceptability
  - Security protection mechanism should be easy to use

Other Secure **Design** Considerations

- Programming Language
- Data type, format, range and length
- Database Security (inference, aggregation)
- Interface (e.g. avoid remote logon by administrators)
- Interconnectivity (data from another system)

Secure **Design** Processes

- **Threat Modeling** to systematically identify and rate the threats that are most likely to affect your system.



Threat Modeling Process

1. Identify Assets
2. Create an Architecture Overview
3. Decompose the Application
4. Identify the Threats
5. Document the Threats
6. Rate the Threats

Secure Software **Implementation**

- CWE/SANS Top 25 Most Dangerous Software Errors
- OWASP top 10 vulnerability
- Common Software Vulnerabilities Category
- Defensive Coding Practices
- Secure Software Processes (during implementation)

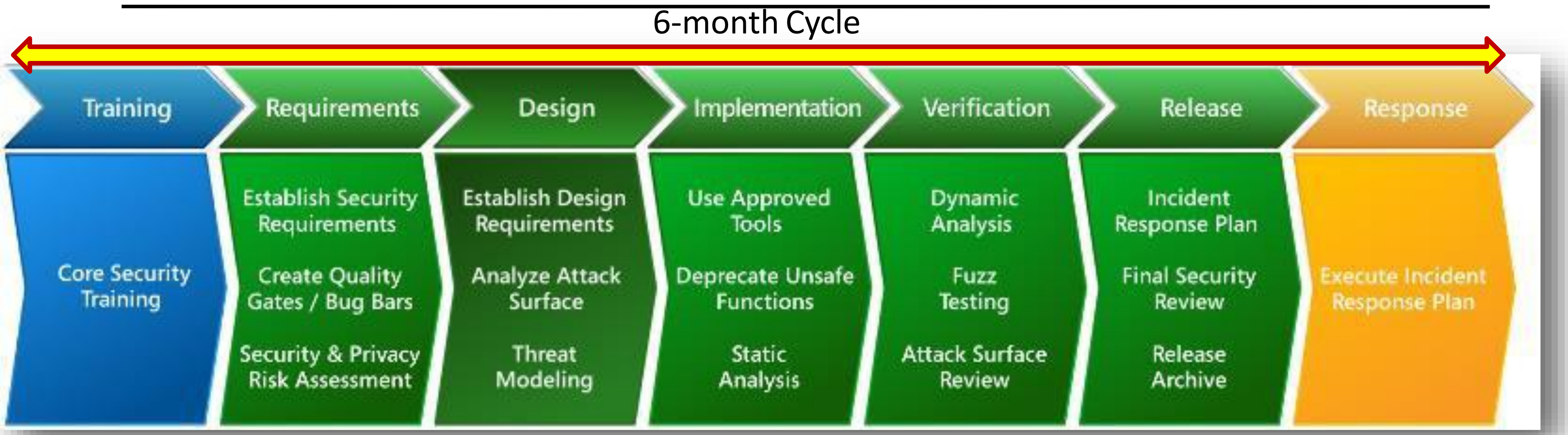
Secure Software **Testing**

- Types of Software Testing
  - Functional Testing
    - Unit testing, Integration testing
  - Recoverability Testing
    - Performance testing (load and stress test), Scalability testing
  - Security Testing
    - Test for resiliency of software
    - White box testing, black box testing, fuzzing, penetration testing, etc

Secure Coding Methodologies

- Microsoft Security Development Lifecycle http://www.microsoft.com/security/sdl/default.aspx
- Software Assurance Maturity Model (SAMM) http://www.opensamm.org/downloads/SAMM-1.0.pdf
- NIST Information Security in the System Development Life Cycle (SP 800-64) http://csrc.nist.gov/groups/SMA/sdlc/index.html
- Build-security In – Secure Software Development Lifecycle Process https://buildsecurityin.us-cert.gov/bsi/articles/knowledge/sdlc/326-BSI.html
- Others….

Microsoft Security Development Lifecycle



6-month Cycle

Training	Requirements	Design	Implementation	Verification	Release	Response
Core Security Training	Establish Security Requirements	Establish Design Requirements	Use Approved Tools	Dynamic Analysis	Incident Response Plan	Execute Incident Response Plan
	Create Quality Gates / Bug Bars	Analyze Attack Surface	Deprecate Unsafe Functions	Fuzz Testing	Final Security Review	
	Security & Privacy Risk Assessment	Threat Modeling	Static Analysis	Attack Surface Review	Release Archive	

- (SD3+C) - Secure by Design, Secure by Default, Secure in Deployment + Communications
- (PD3+C) – Privacy by Design, Privacy by Default, Privacy by Deployment + Communication
- A mandatory policy since 2004, 4 maturity model (Basic, Standardized, Advanced, Dynamic), Development focus
- Articulates policies for conformance & provides *tools*
- Security and privacy early and throughout the development process