# **Tech Talk: Application Security**

## IBM Application Security Services

IBM **Code**

Areej Essa
Cloud Developer & Advocate
IBM **Hybrid Cloud**



Content owner & creator: Areej Essa
Areej.Essa@ae.ibm.com

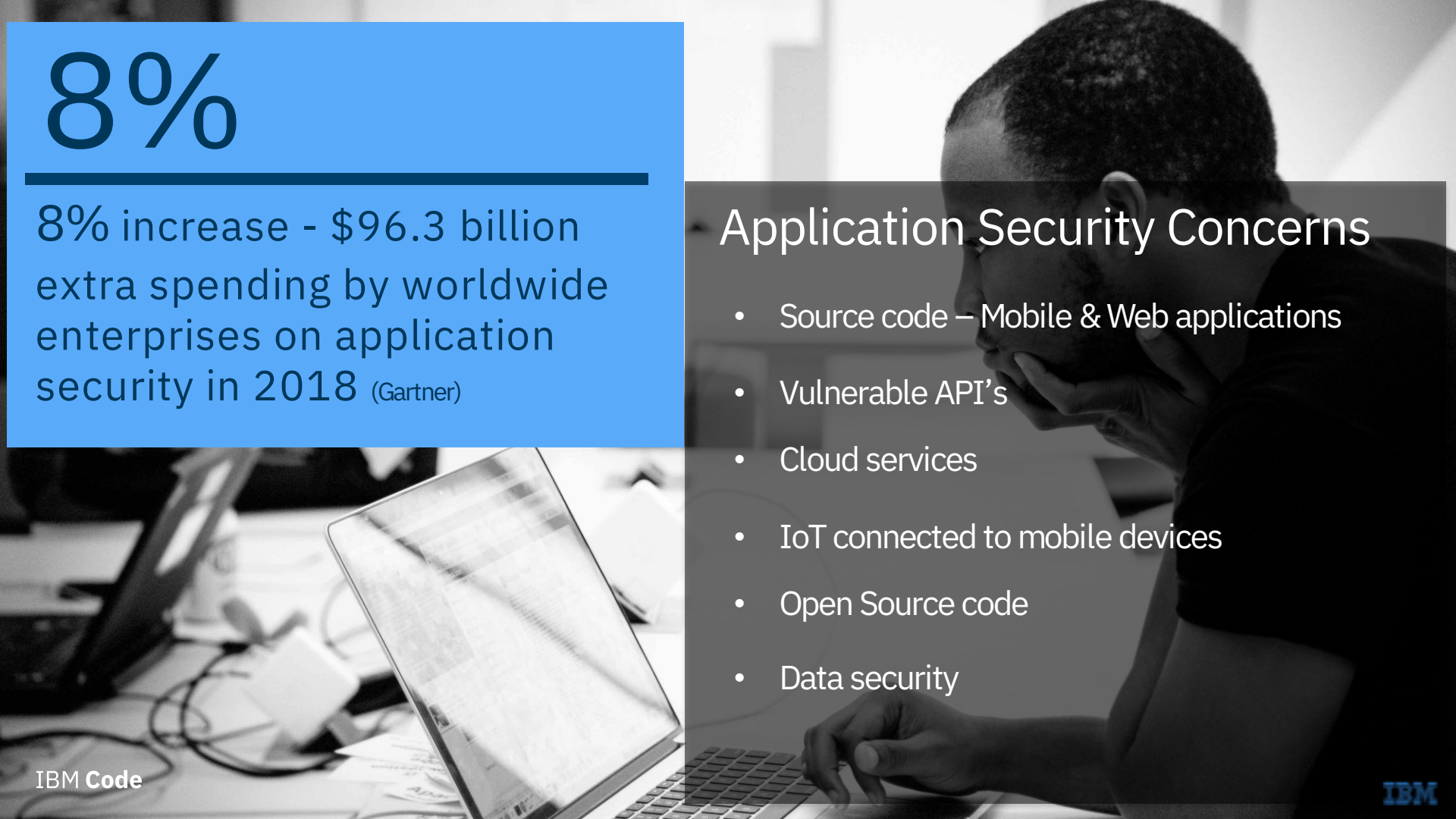
# Speaker

**Areej Essa**

Areej.Essa@ae.ibm.com

# AGENDA

- Application attacks incidents – who to blame?
- Application security concerns and best practices
- Security in Software Development Life Cycle
- DevSecOps
- Open source tools for secure code development



# 8%

8% increase - $96.3 billion extra spending by worldwide enterprises on application security in 2018 (Gartner)

## Application Security Concerns

- Source code – Mobile & Web applications
- Vulnerable API's
- Cloud services
- IoT connected to mobile devices
- Open Source code
- Data security

# An application is the gateway to an organization's data theft and attack executions on them.

**On September 2017, Equifax was attacked:**

145 million of its customer records were stolen

Over £42 million lost

Lost of its 100,000 customers

**Caused due to a vulnerability in their web application**

# An application is the gateway to an organization's data theft and attack executions on them.


Organizations do not perform any major testing methods prior to deployment

Organizations do not take basic security measures to remediate vulnerabilities

Organizations are influenced by pressure to release new apps quickly

Organizations did not know all the apps and databases currently active in their organizations

# Who is involved?

Vendor or seller

Developer

## Lack of awareness

### Common excuses (OWASP):

*“We never get hacked (that I know of), we don’t need security”*

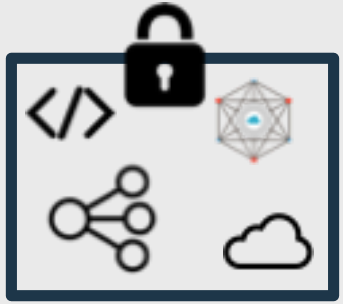
*“We have a firewall that protects our applications”*

*"We trust our employees not to attack our applications”*

## Vendor or seller

## Lack of knowledge

Deadlines



Skill set

No enablement

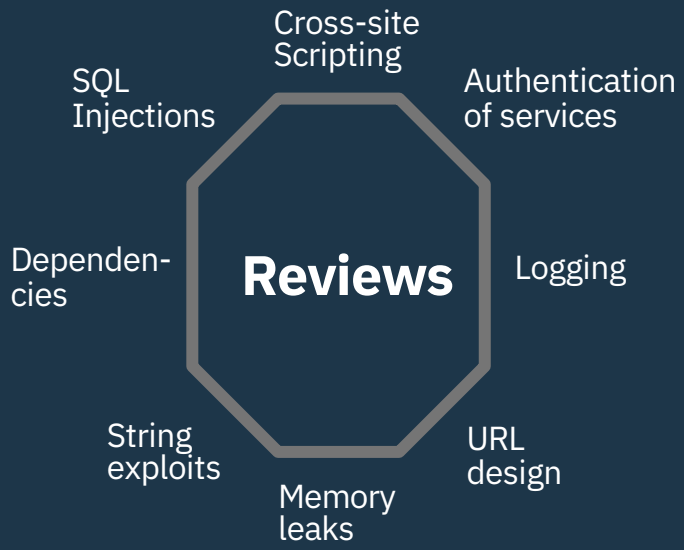
## Developer

IBM

# Application Security Concerns

- **Source code – Mobile & Web applications**
- Vulnerable API’s
- Cloud services
- IoT connected to mobile devices
- Open Source code
- Data security

# Secure Coding Practices: Source Code Review




*Bad code is the root cause of most web app breaches.*

- Code review run against set of rules to detect vulnerabilities in the code.
- Performed at early stages of SDLC.
- Performed by the developer along someone other than the developer.
- Can be performed manually or automated using tools.

# Secure Coding Practices: Source Code Review


- One of the most critical and commonplace security vulnerabilities for web applications.
- Allow SQL commands to be executed on the web server to directly interrogate the back-end database.
- Ability to return and steal tables of information, change records, or even delete the entire database.

***According to Gartner executive report, 2017:***

- *Of total breaches reported in 2017,* ***32%*** *were SQL injection errors.*
- *SQL injection ranked as the most critical vulnerability,* ***92%****, web application attack.*

# Secure Coding Practices: Source Code Review


- Use of a browser side script, to send malicious code to a different end user.
- Occur anywhere a web application uses input from a user in the output it generates without validating or encoding it.

***According to Gartner executive report, 2017:***

- *SQL injection ranked as the second most vulnerability,* ***38%,*** *web application attack.*

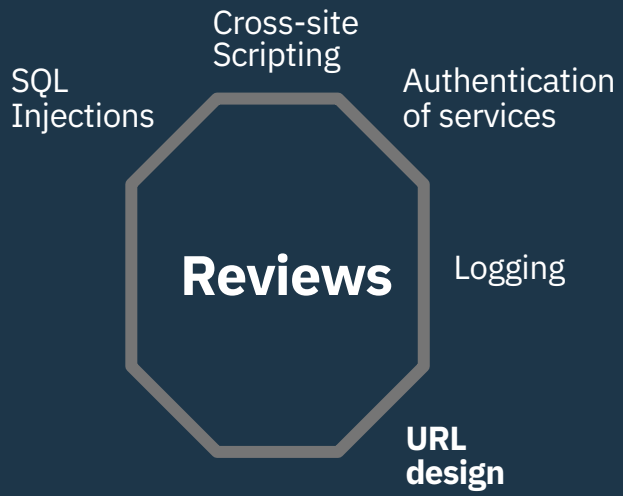
# Secure Coding Practices: Source Code Review


- Authenticating of services (API's) to the applications.
- Code must include the authentication of the service before sending and receiving requests.

# Secure Coding Practices: Source Code Review

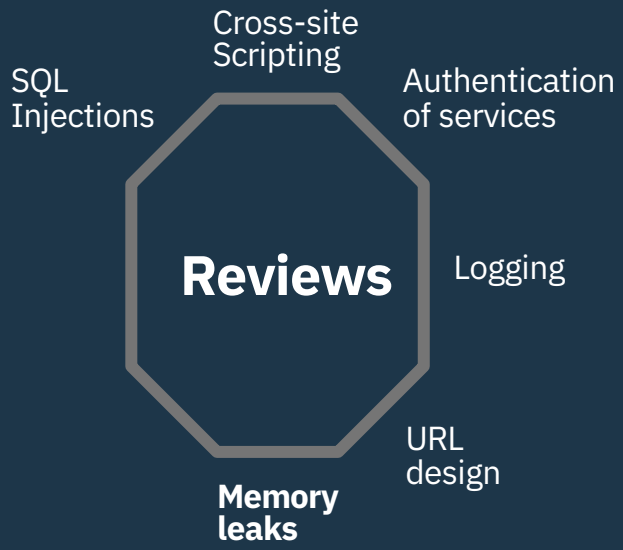

- **Logging of critical information such as:**
  - Timestamp.
  - Identity of the account/user that caused the event.
  - Source IP address associated with the request.
  - Event outcome (success or failure).
  - Description of the event.

# Secure Coding Practices: Source Code Review

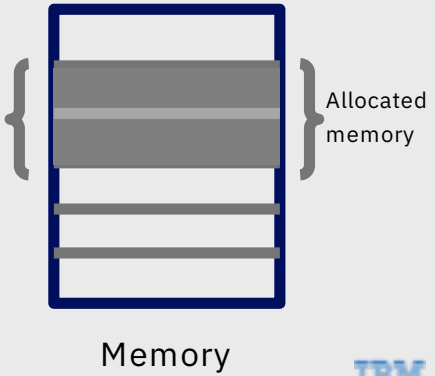



- Ranked in one of the top 10 web application attacks by OWASP.
- Application URL must not release sensitive information such as PII, API keys.
- Can cause "*forced browsing*".
- Attackers can successfully gain sensitive data.

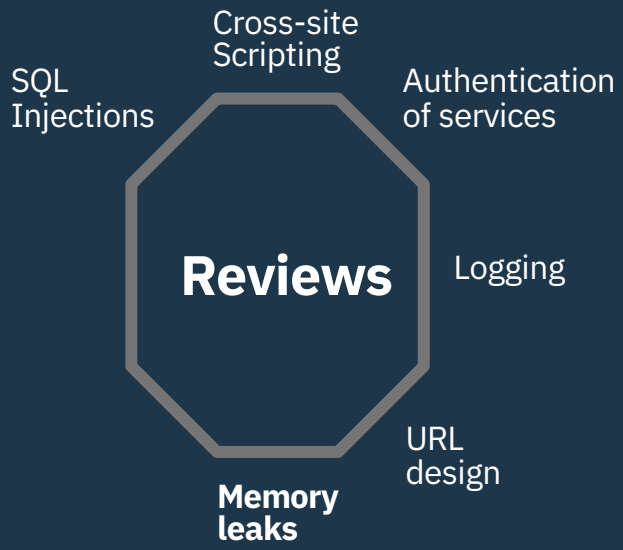
# Secure Coding Practices: Source Code Review



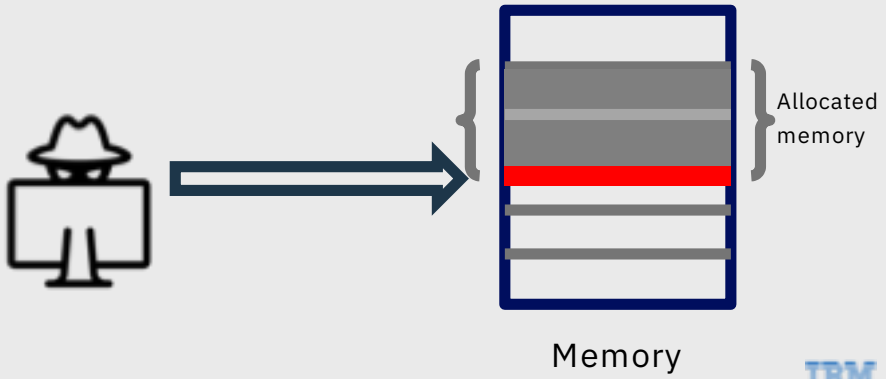

- Caused by memory buffer overflow.
- Occurs when a developer dynamically allocates memory space for a variable, but fails to free up that space before the program completes (takes all the space and program becomes unresponsive).
- Attacker submit data to an application larger than the allocated memory and overwrite valid data.



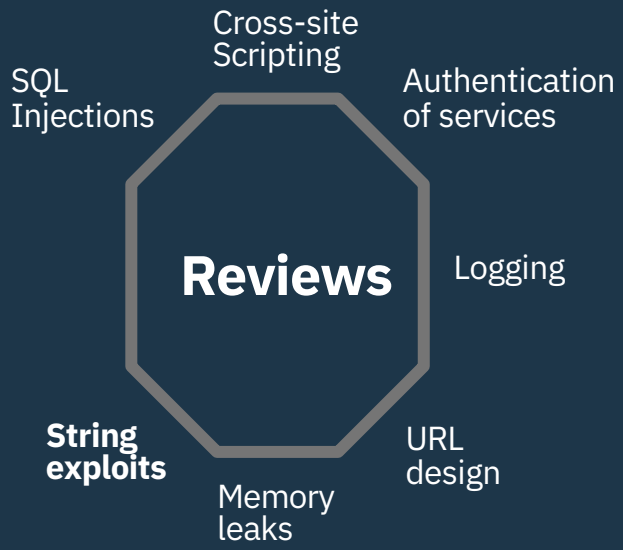
# Secure Coding Practices: Source Code Review




- Attacker submit data to an application larger than the allocated memory and overwrite valid data.
- New memory return address can allow the attacker to execute their malicious code.

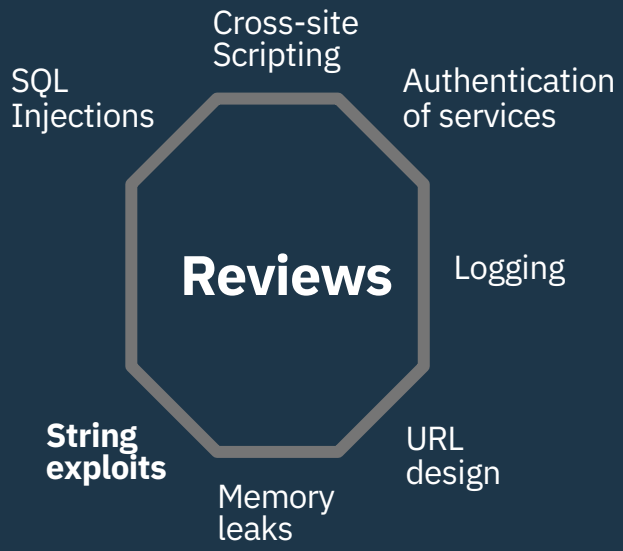


# Secure Coding Practices: Source Code Review

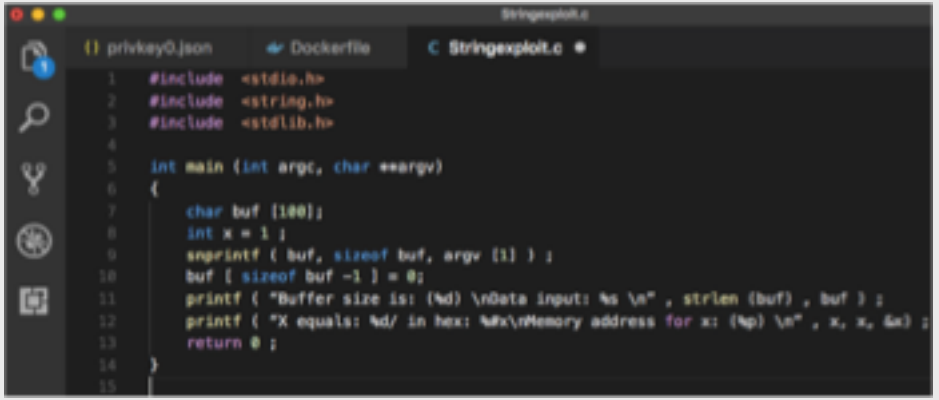



- Submitted data of an input string is evaluated as a *command by the application*.
- Attacker can execute code, read the stack, or cause a segmentation fault in the running application.

# Secure Coding Practices: Source Code Review




- Illustration: Showing contents of a memory address



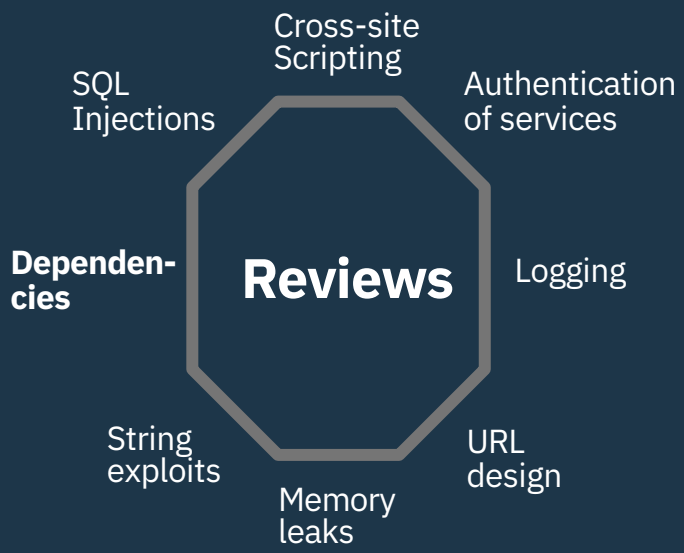

*Running:*
*./Stringexploit "myName %x %x"* returns memory contents too.

```
Buffer size is (14)
Data input : Bob bffff 8740
X equals: 1/ in hex: 0x1
Memory address for x (0xbffff73c)
```

Source: https://www.owasp.org/index.php/Format_string_attack

IBM

# Secure Coding Practices: Source Code Review




- Third-party libraries highly used for faster development can introduce vulnerabilities to the code.
- **90%** of software applications contain Third-party libraries. (Central Repository)
- Might contain vulnerabilities. For example: *GNU C Library* contained a buffer overflow vulnerability.

# Application Security Concerns

- Source code – Mobile & Web applications
- **Vulnerable API’s**
- **Cloud services**
- IoT connected to mobile devices
- Open Source code
- Data security

## Cloud Services API's


Developer/Application

Cloud services provider

Insecure cloud API's can pose a variety of risks related to confidentiality, integrity, availability and accountability.

# Application Security Concerns

- Source code – Mobile & Web applications
- Vulnerable API’s
- Cloud services
- IoT connected to mobile devices
- **Open Source code**
- Data security

**57%** of enterprises deploy open-source in their production environment

**50%** of enterprises have no formal approval process for open-source code

**53.8%** increase in the number of open source application library security in 2016

## Vulnerable Open Source Codes

- Accelerated development using open source
- Open source codes can contain vulnerabilities
- Use of un-scanned third-part libraries and tools
- Developers fail, forget or even dismiss the update of the latest version of the used libraries

***Examples of successful attacks (2016):***

*Nokogiri and LobXML* were attacked by DoS and private data disclosure.

# Application Security Concerns

- Source code – Mobile & Web applications
- Vulnerable API’s
- Cloud services
- IoT connected to mobile devices
- Open Source code
- **Data security**

# Data Security

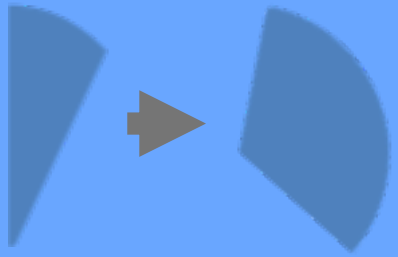
Every 2 years, enterprise will **double** its data

*(Gartner)*

Over the past 25 years, enterprises spent **trillions** on digitizing analogue data

Major focus is on **productivity**, overlooking their data security

The percentage of data breaches from a web application attack was **7%** in 2015 and grew to **40% in 2016.**

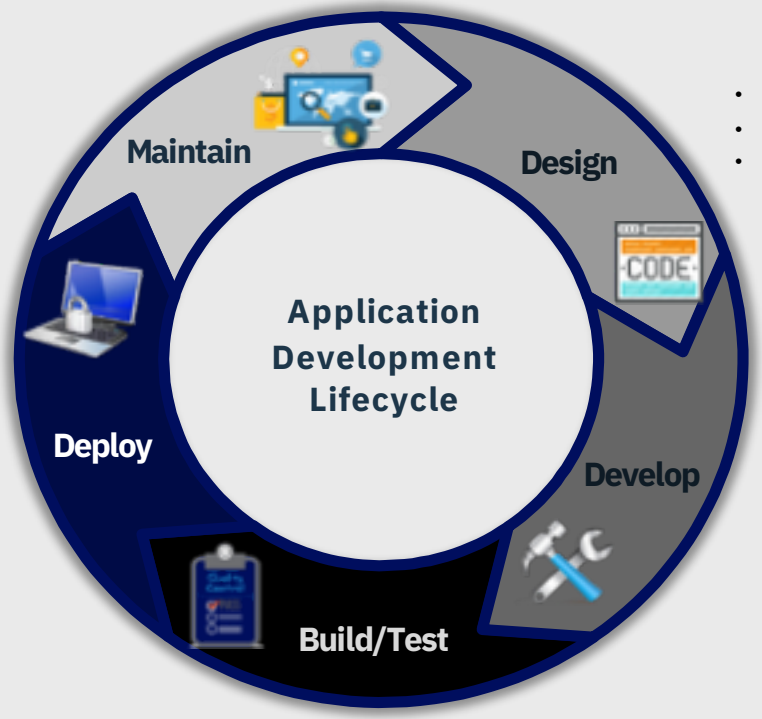


# Does an organization need to treat *all* of its the data in the same way?

- Not all the data need to be treated in the same way
- Critical and sensitive data needs to be more protected than ordinary data
- Saves cost and time, and effectively protects sensitive data

# Software Development Lifecycle - Issues

- Retrospective Fixes
- Cost to Fix at Production




- Lack of Security Requirements
- Inconsistent Development Standards
- Lack of Security by Design

- Time to Market vs. Security
- Accept or Remediate
- Platform Vulnerabilities

- Legacy Software Libraries
- Insecure Protocols
- Insecure Configuration
- Poor Development Practices

- Limited Security Testing Scope
- Large Numbers of False Positives
- Tool or Vendor Limitations
- Prioritization of Fixes



IBM

# DevSecOps

Dev | Sec | Ops


Automate core security tasks by embedding security controls and processes into the DevOps workflow.

# What is DevSecOps?

- Adds the element of Security to DevOps
- Embed concepts of Security in every step of SDLC
- Everyone is responsible for Security
- Use tools to automate various Security processes such as:
  - Code Review
  - Identity and Access Management
  - Vulnerability Scanning

# Benefits of DevSecOps

- Cost Reduction
  - Detect & fix Security issues during development phase
- Speed of Delivery/Recovery
  - Eliminate Security bottlenecks
- Secure by design
  - Automated code review, vulnerability scanning
- Secure iterative innovation
  - Innovate securely at speed & scale

# Common Practices of DevSecOps

- Risk Assessment
  - Threat identification
  - Strategies
- Secure Coding
  - High Cohesion & Loose Coupling
  - SAST – Static Application Security Testing
  - Code Analysis
- Secure Infrastructure (Cloud?)
  - Server updates
  - Backups

## Five Principles in DevSecOps

1. Automate security in
2. Integrate to “fail quickly”
3. No false Alarms
4. Build security champions
5. Keep operational visibility

### Key values of DevSecOps

- Greater speed and agility for security teams
- An ability to respond to change and needs rapidly
- Integrating security into the CI/CD pipeline ensures that security testing happens with every release
- More opportunities for automated builds and quality assurance testing
- Early identification of vulnerabilities in code
- Reduces costs and achieves agility

*Source: FIVE PRINCIPLES FOR Securing DevOps, VERACODE eBook*

IBM


# Open Source Tools for Secure Code Development

# Open Source Tools for Secure Code Development

- **Veracode:** Static analysis tool
- **Grafeas (IBM & Google):** Maintain proper security standards for software
- **Black Duck: (IBM Security Partner):** Automate the processes of securing and managing open source
- **Open source Analyzer:** Scans open source code
- **OWASP Code crawler:** Static code review tool
- **IBM AppScan:** Source code review
- **IBM Guardiam:** Data Security





# Code Lab: Test our application using ***IBM Application Security on Cloud***

## Let’s scan an application

**Create an IBM Cloud Account**: ibm.biz/devfest

**Go to IBM Application Security on Cloud:**
https://appscan.ibmcloud.com/AsoCUI/serviceui/home