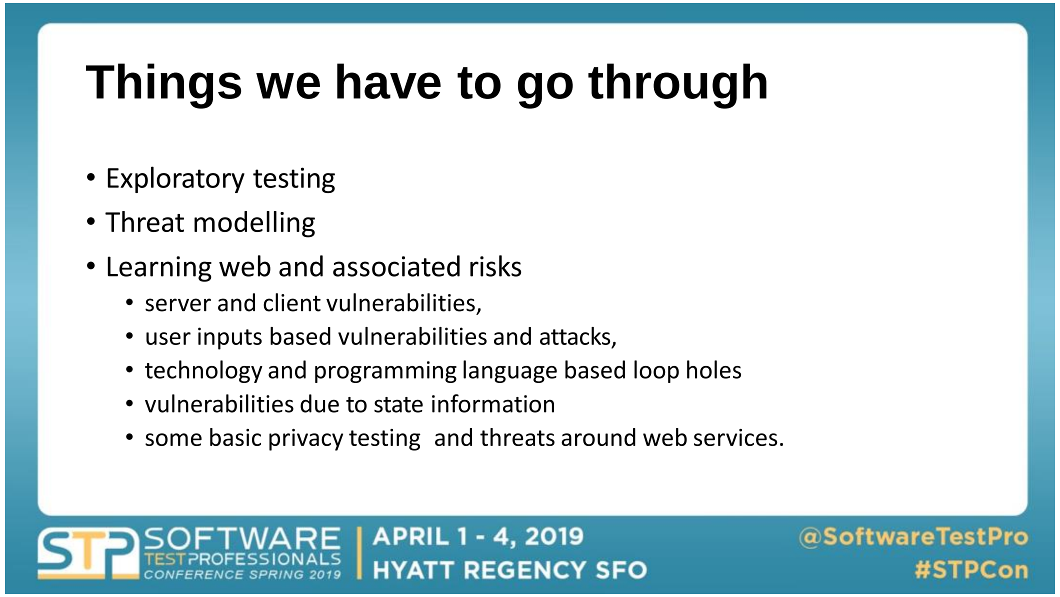
STP SOFTWARE
TEST PROFESSIONALS
CONFERENCE SPRING 2019

Breaking Web Applications

Smita Mishra, CEO, QAZone Infosystems

1

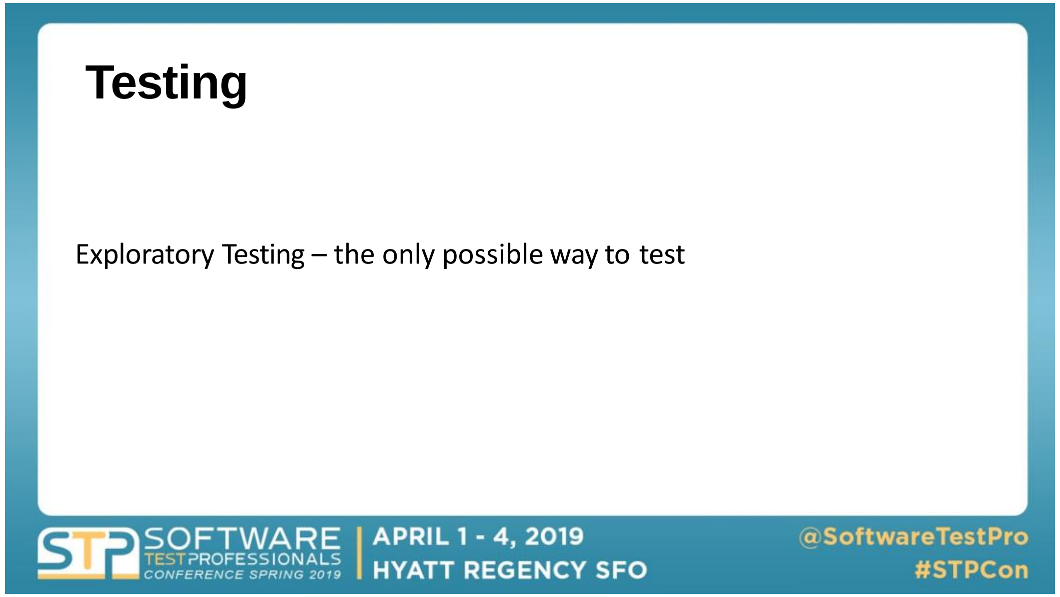


Things we have to go through

- Exploratory testing
- Threat modelling
- Learning web and associated risks
  - server and client vulnerabilities,
  - user inputs based vulnerabilities and attacks,
  - technology and programming language based loop holes
  - vulnerabilities due to state information
  - some basic privacy testing and threats around web services.

STP SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019
APRIL 1 - 4, 2019
HYATT REGENCY SFO
@SoftwareTestPro
#STPCon

2

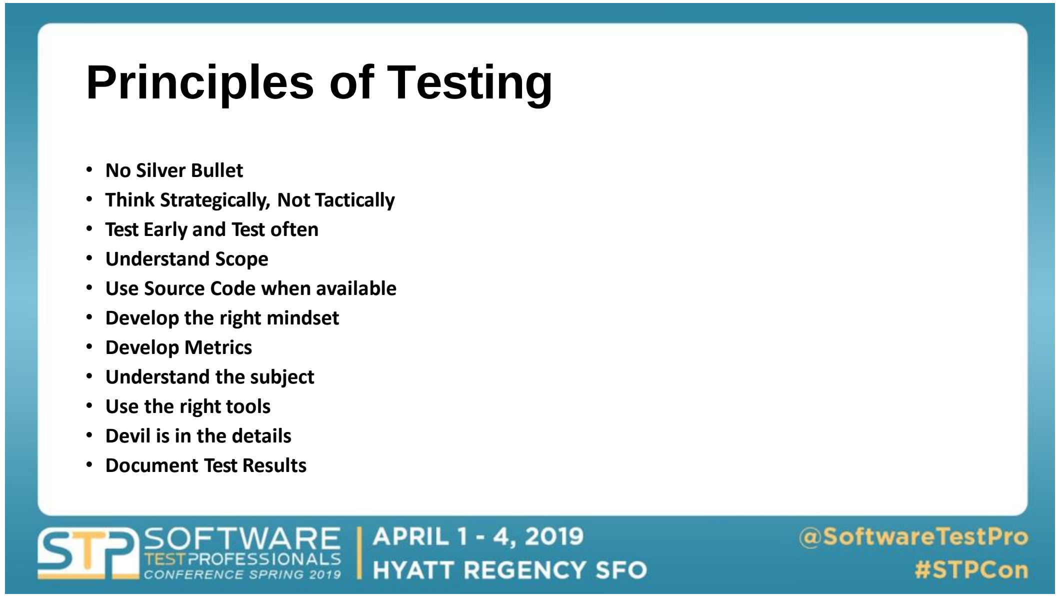


Testing

Exploratory Testing – the only possible way to test

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

3

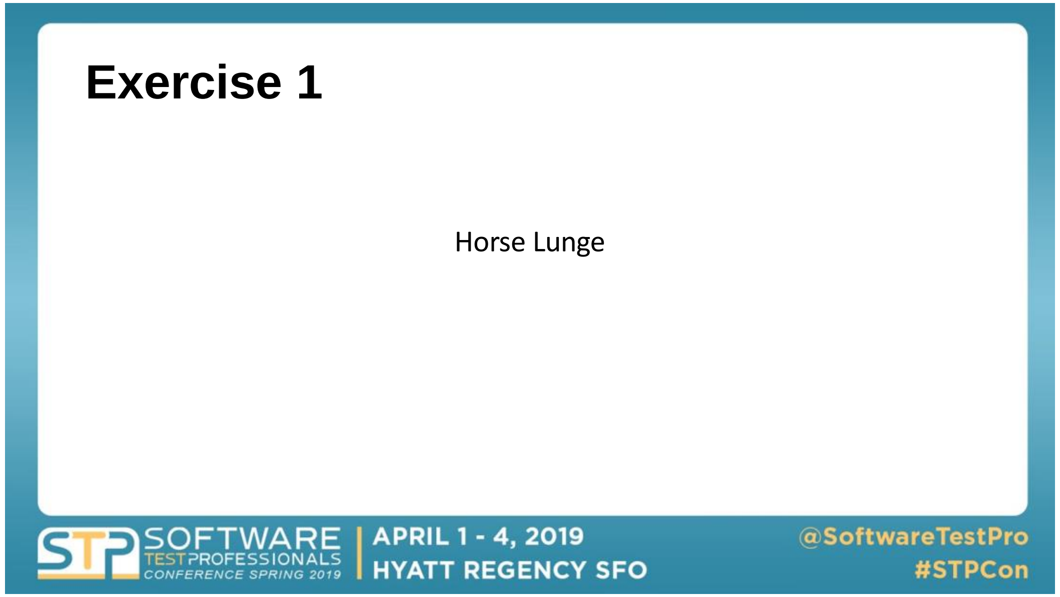


Principles of Testing

- **No Silver Bullet**
- **Think Strategically, Not Tactically**
- **Test Early and Test often**
- **Understand Scope**
- **Use Source Code when available**
- **Develop the right mindset**
- **Develop Metrics**
- **Understand the subject**
- **Use the right tools**
- **Devil is in the details**
- **Document Test Results**

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

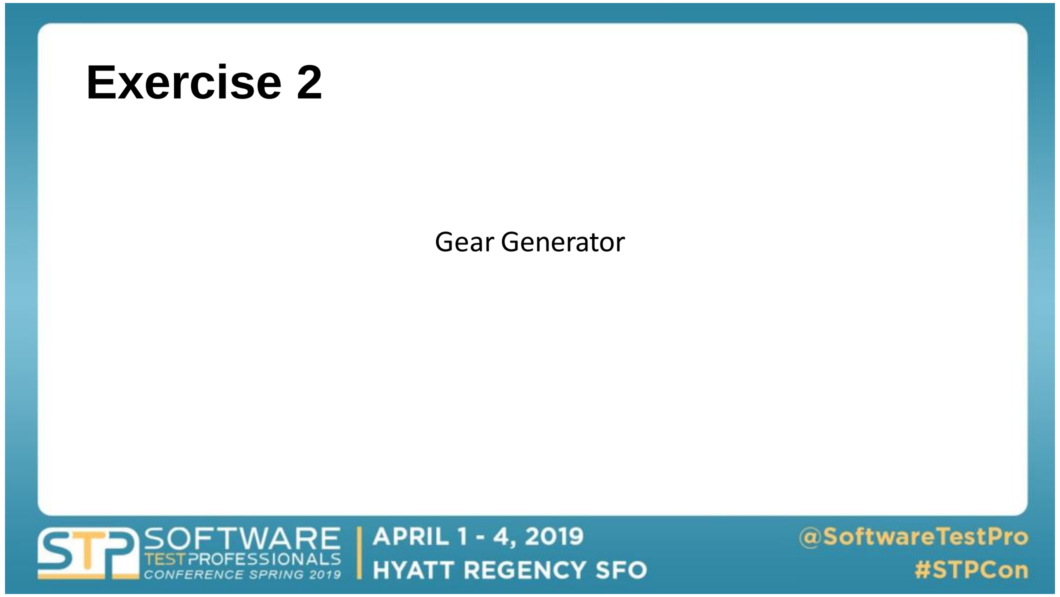
4



Exercise 1

Horse Lunge

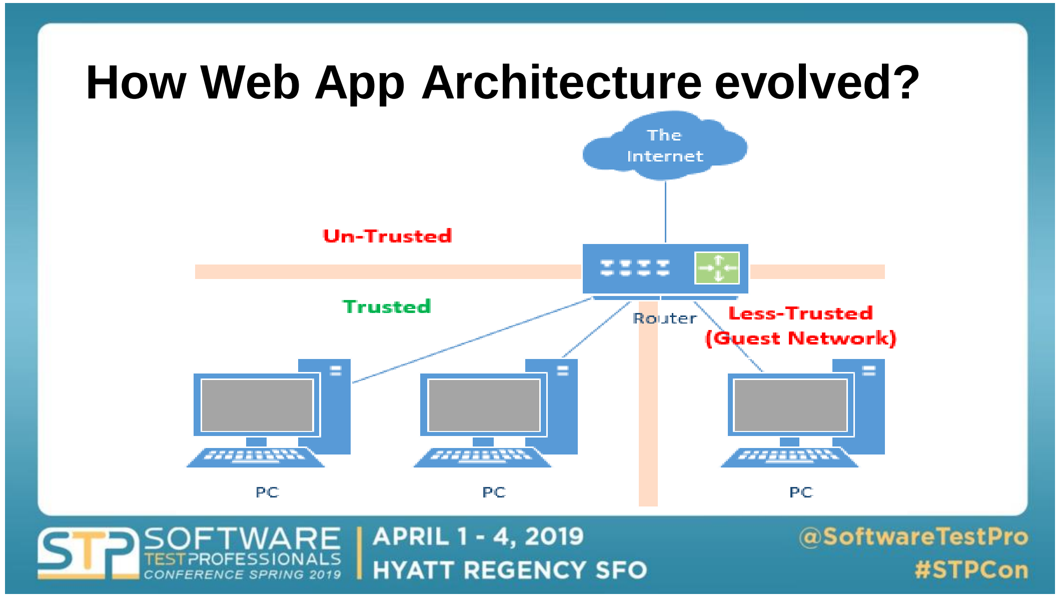
5



Exercise 2

Gear Generator

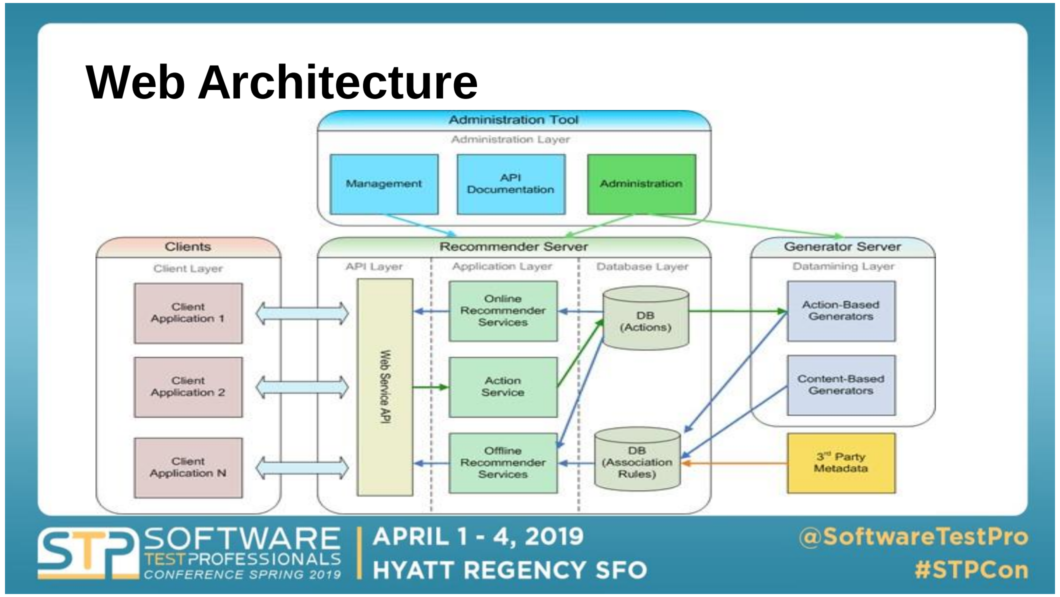
6



How Web App Architecture evolved?

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

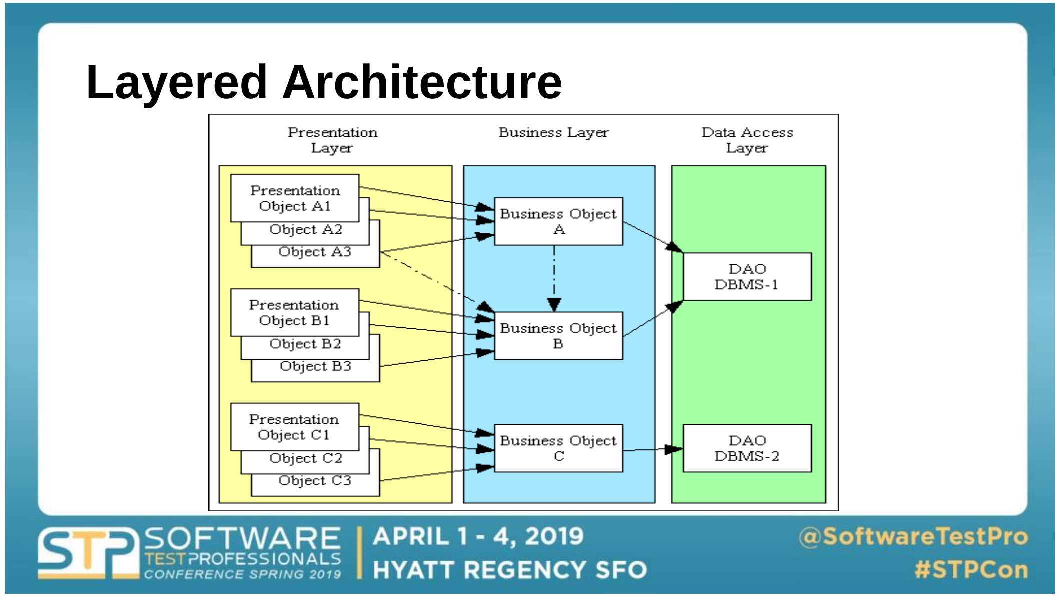
7



Web Architecture

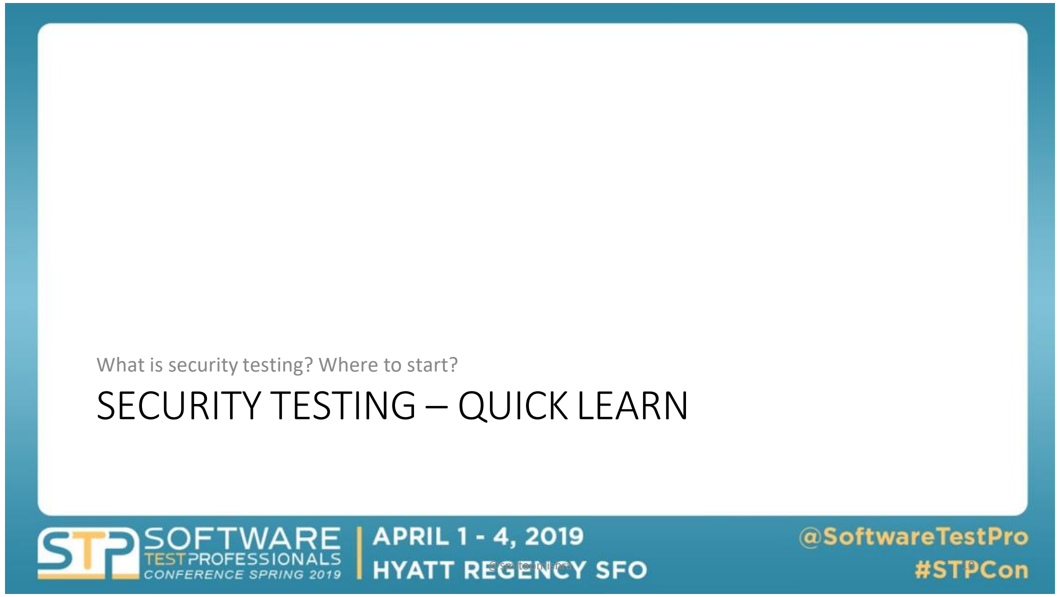
SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

8



Layered Architecture

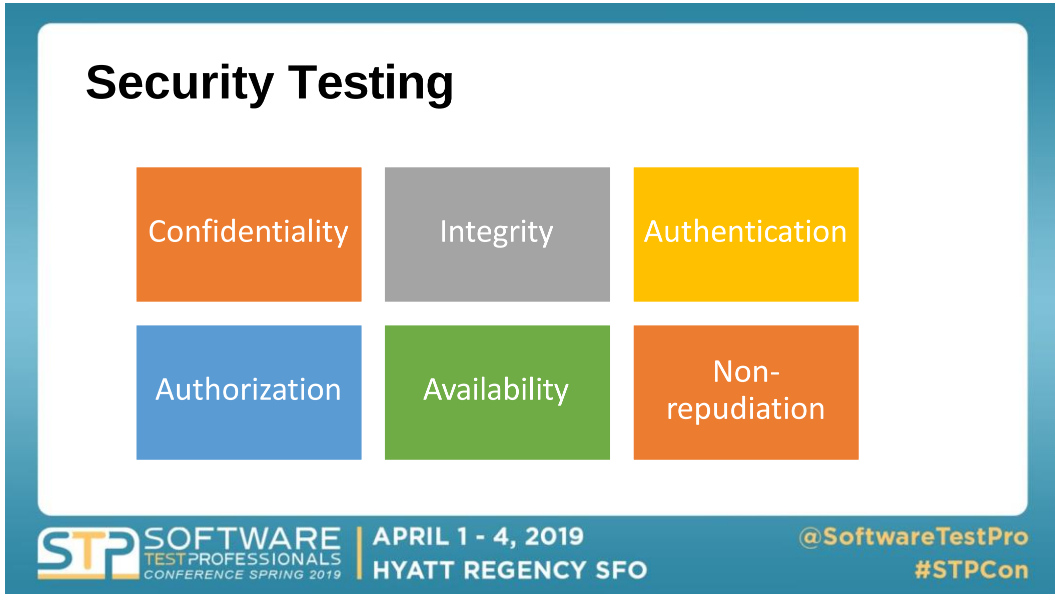
9



What is security testing? Where to start?

SECURITY TESTING – QUICK LEARN

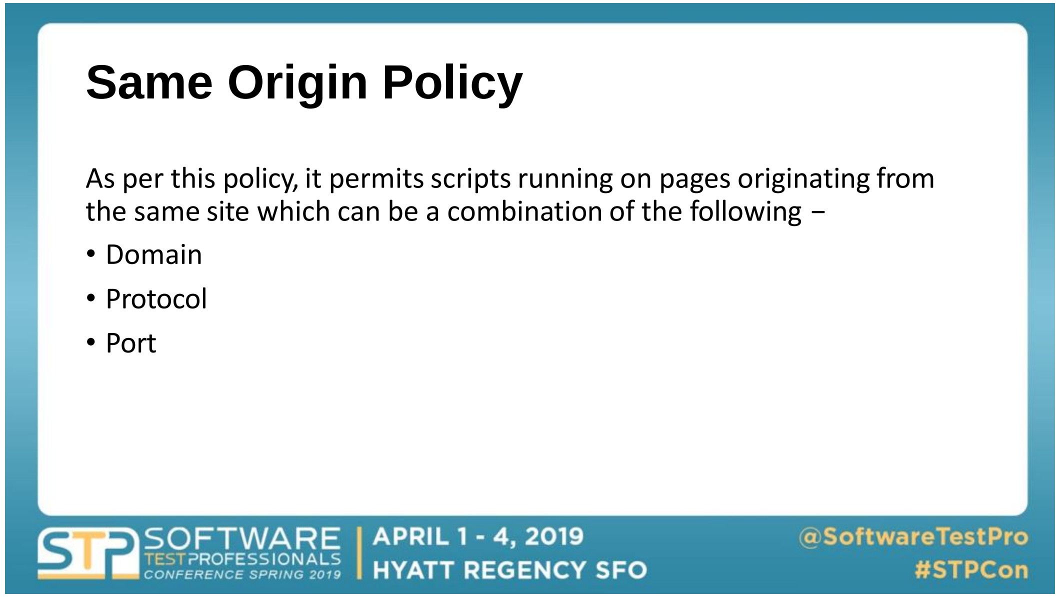
10



Security Testing

- Confidentiality
- Integrity
- Authentication
- Authorization
- Availability
- Non-repudiation

11

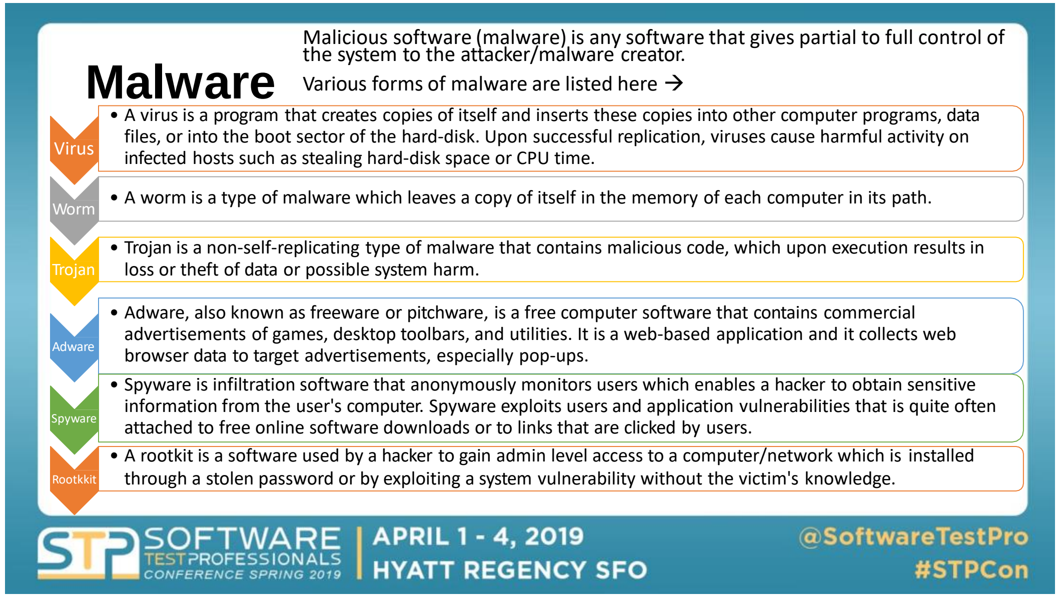


Same Origin Policy

As per this policy, it permits scripts running on pages originating from the same site which can be a combination of the following –

- Domain
- Protocol
- Port

12



Malware

Malicious software (malware) is any software that gives partial to full control of the system to the attacker/malware creator.

Various forms of malware are listed here →

Virus
- A virus is a program that creates copies of itself and inserts these copies into other computer programs, data files, or into the boot sector of the hard-disk. Upon successful replication, viruses cause harmful activity on infected hosts such as stealing hard-disk space or CPU time.

Worm
- A worm is a type of malware which leaves a copy of itself in the memory of each computer in its path.

Trojan
- Trojan is a non-self-replicating type of malware that contains malicious code, which upon execution results in loss or theft of data or possible system harm.

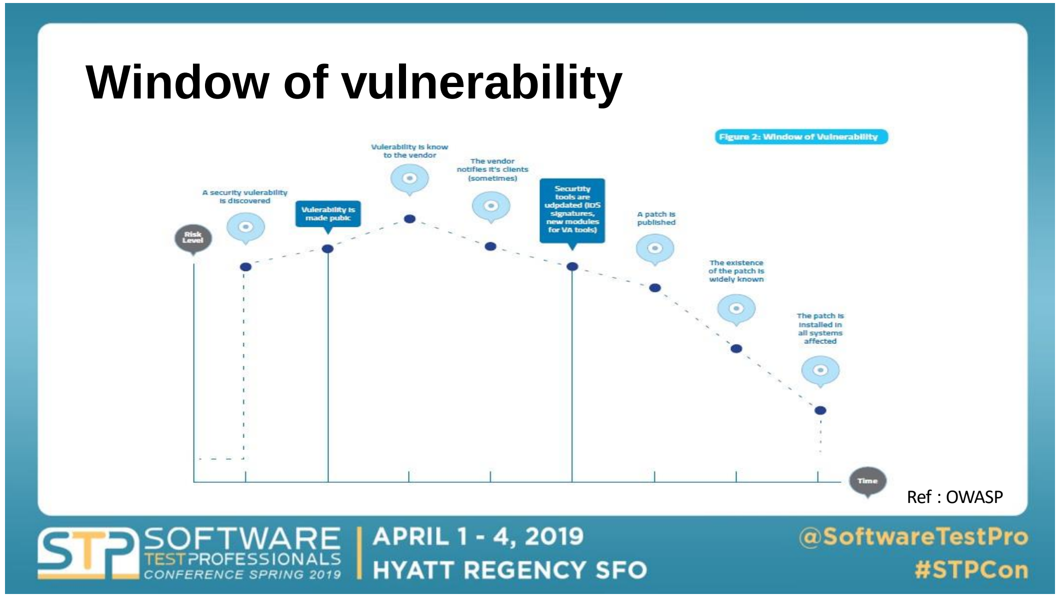
Adware
- Adware, also known as freeware or pitchware, is a free computer software that contains commercial advertisements of games, desktop toolbars, and utilities. It is a web-based application and it collects web browser data to target advertisements, especially pop-ups.

Spyware
- Spyware is infiltration software that anonymously monitors users which enables a hacker to obtain sensitive information from the user's computer. Spyware exploits users and application vulnerabilities that is quite often attached to free online software downloads or to links that are clicked by users.

Rootkkit
- A rootkit is a software used by a hacker to gain admin level access to a computer/network which is installed through a stolen password or by exploiting a system vulnerability without the victim's knowledge.

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

13



Window of vulnerability

Figure 2: Window of Vulnerability
Risk Level
A security vulerability is discovered
Vulerability is made pubic
Vulerability is know to the vendor
The vendor notifies it's clients (sometimes)
Securtity tools are udpdated (IDS signatures, new modules for VA tools)
A patch is published
The existence of the patch is widely known
The patch is installed in all systems affected
Time

Ref : OWASP

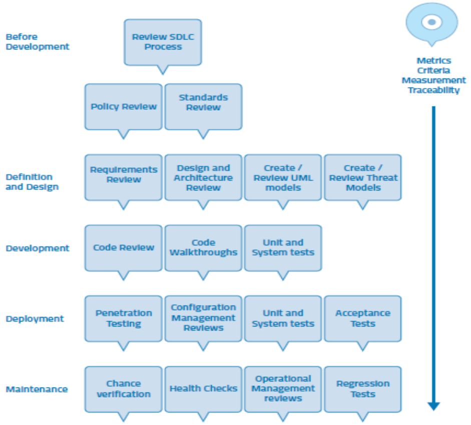
SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

14

Approach

- PTES – Penetration Testing Execution Standard
- OSSTMM – Open Source Security Testing Methodology Manual
- OWASP Testing Techniques – Open Web Application Security Protocol

OWASP TESTING FRAMEWORK WORK FLOW



Before Development

Review SDLC Process

Policy Review

Standards Review

Definition and Design

Requirements Review

Design and Architecture Review

Create / Review UML models

Create / Review Threat Models

Development

Code Review

Code Walkthroughs

Unit and System tests

Deployment

Penetration Testing

Configuration Management Reviews

Unit and System tests

Acceptance Tests

Maintenance

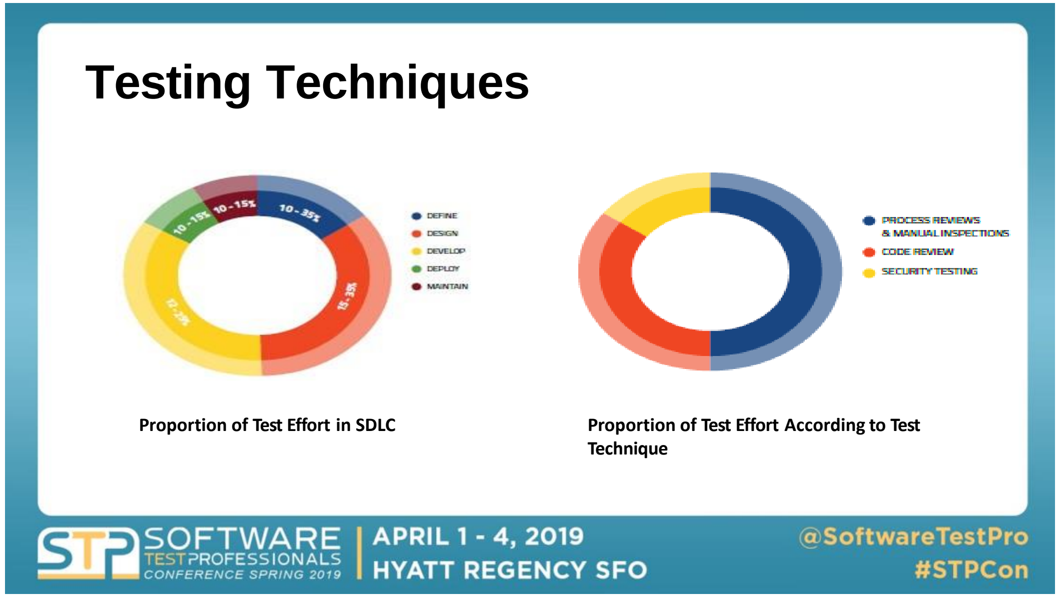
Chance verification

Health Checks

Operational Management reviews

Regression Tests

Metrics Criteria Measurement Traceability



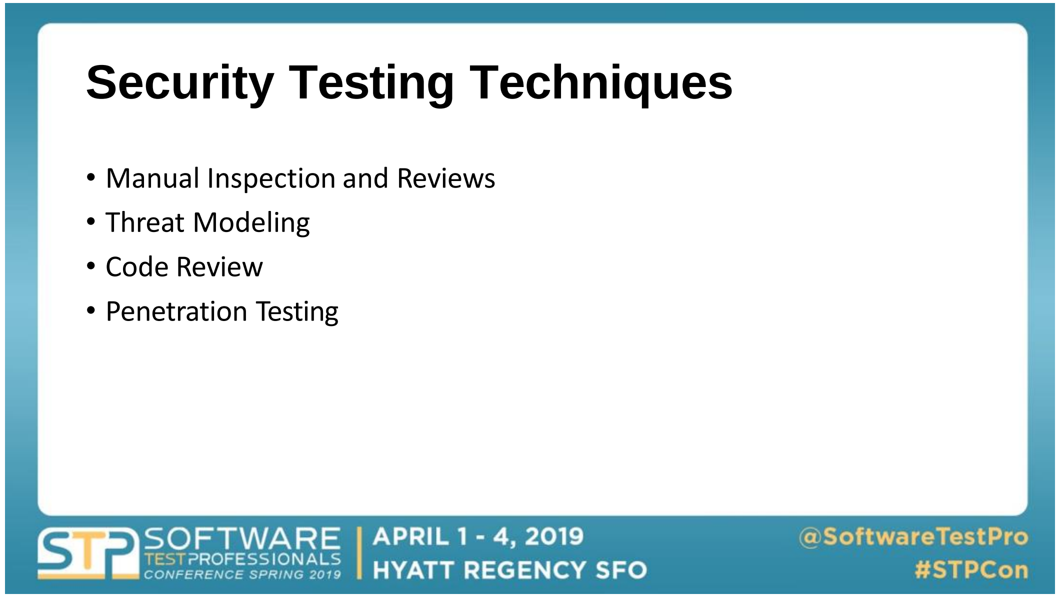
Testing Techniques

Proportion of Test Effort in SDLC

Proportion of Test Effort According to Test Technique

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

17

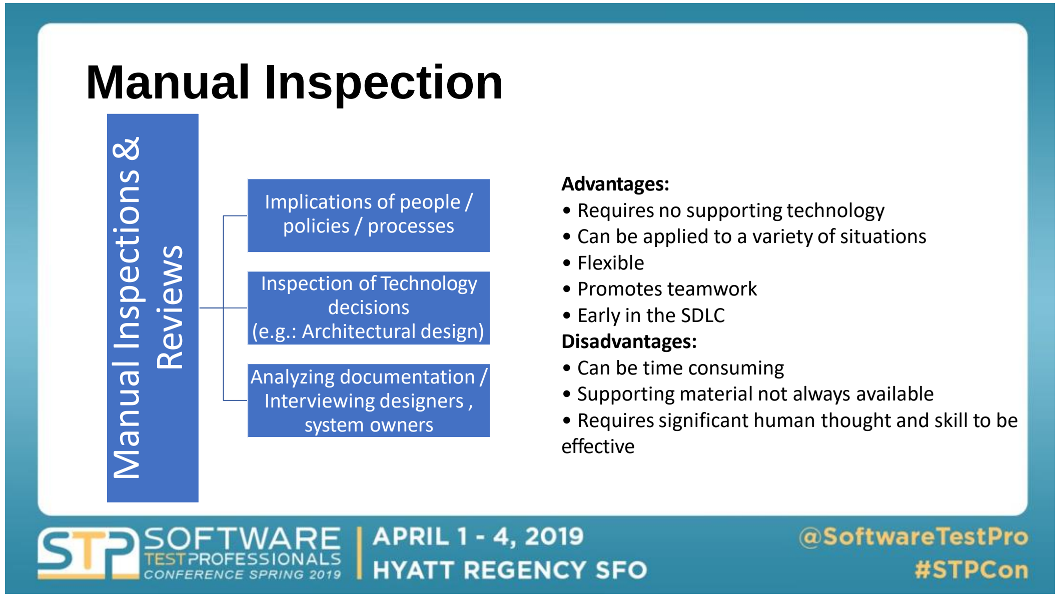


Security Testing Techniques

- Manual Inspection and Reviews
- Threat Modeling
- Code Review
- Penetration Testing

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

18



Manual Inspection

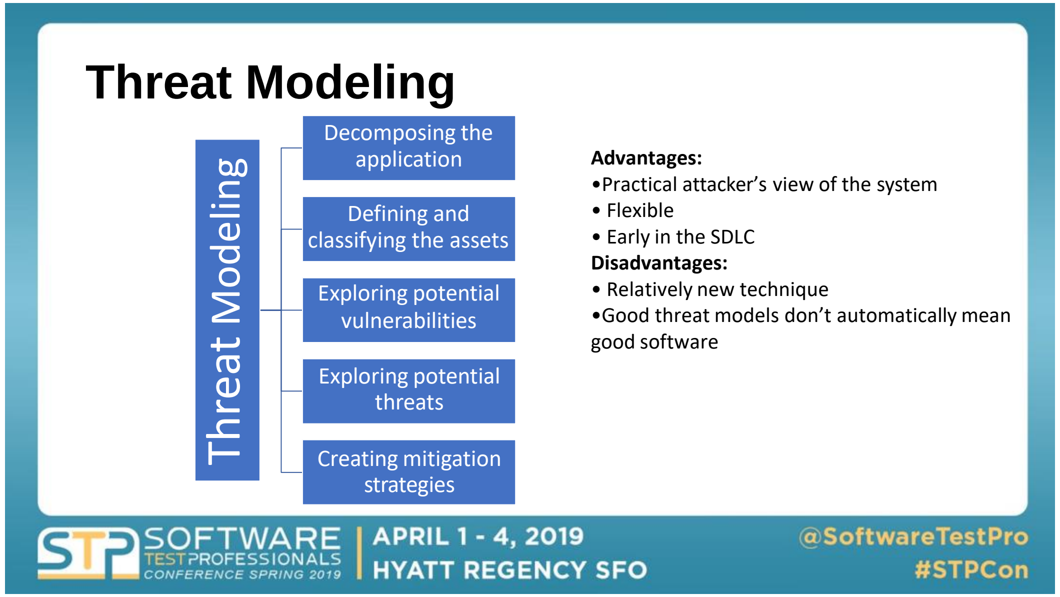
Manual Inspections & Reviews

- Implications of people / policies / processes
- Inspection of Technology decisions (e.g.: Architectural design)
- Analyzing documentation / Interviewing designers , system owners

Advantages:
- Requires no supporting technology
- Can be applied to a variety of situations
- Flexible
- Promotes teamwork
- Early in the SDLC

Disadvantages:
- Can be time consuming
- Supporting material not always available
- Requires significant human thought and skill to be effective

19



Threat Modeling

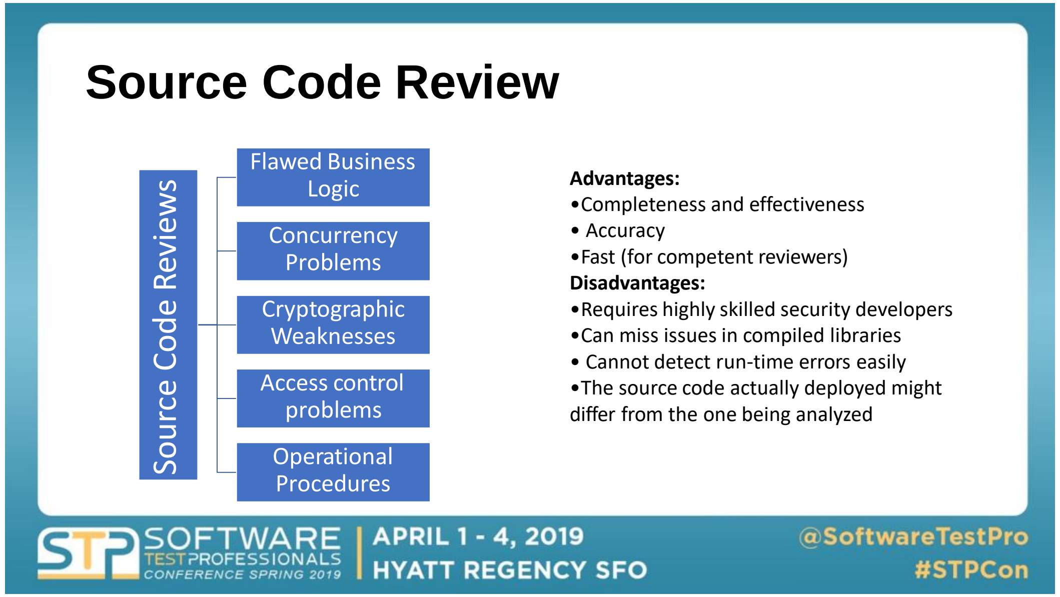
Threat Modeling

- Decomposing the application
- Defining and classifying the assets
- Exploring potential vulnerabilities
- Exploring potential threats
- Creating mitigation strategies

Advantages:
- Practical attacker's view of the system
- Flexible
- Early in the SDLC

Disadvantages:
- Relatively new technique
- Good threat models don't automatically mean good software

20



Source Code Review

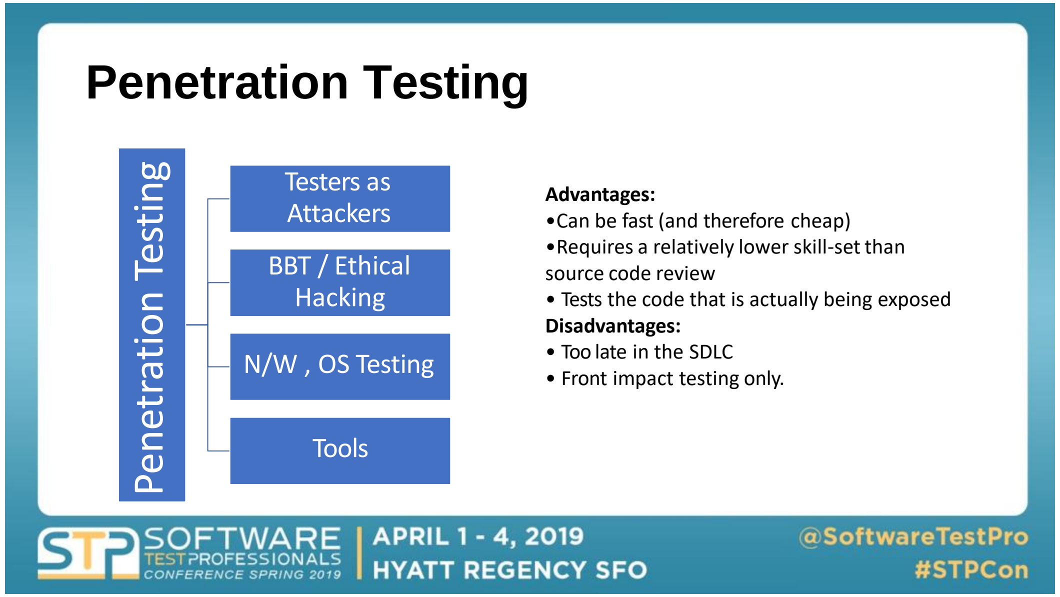
Source Code Reviews
Flawed Business Logic
Concurrency Problems
Cryptographic Weaknesses
Access control problems
Operational Procedures

Advantages:
- Completeness and effectiveness
- Accuracy
- Fast (for competent reviewers)

Disadvantages:
- Requires highly skilled security developers
- Can miss issues in compiled libraries
- Cannot detect run-time errors easily
- The source code actually deployed might differ from the one being analyzed

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

21



Penetration Testing

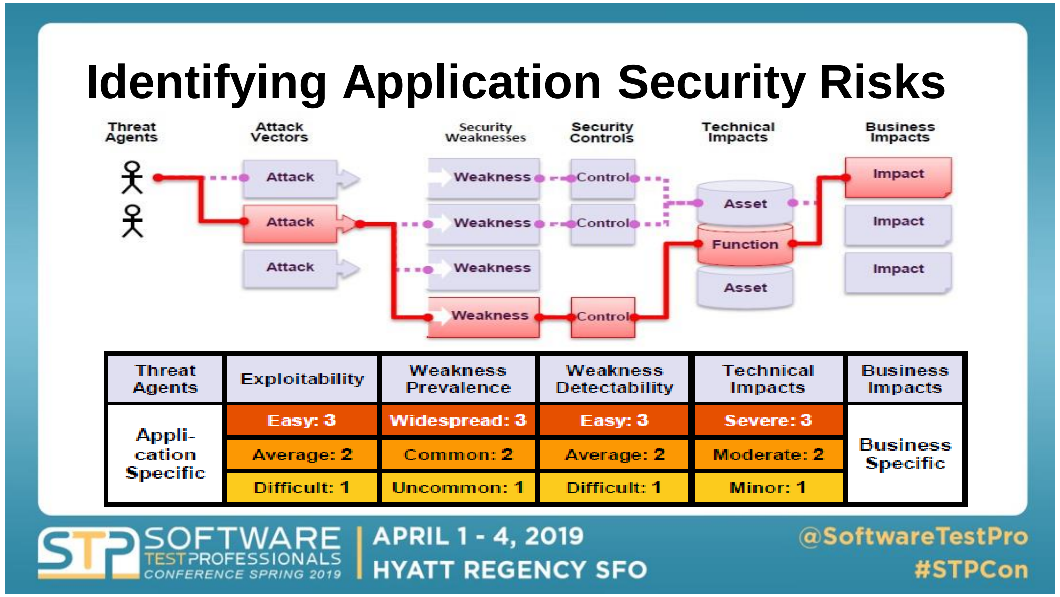
Penetration Testing
Testers as Attackers
BBT / Ethical Hacking
N/W , OS Testing
Tools

Advantages:
- Can be fast (and therefore cheap)
- Requires a relatively lower skill-set than source code review
- Tests the code that is actually being exposed

Disadvantages:
- Too late in the SDLC
- Front impact testing only.

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

22

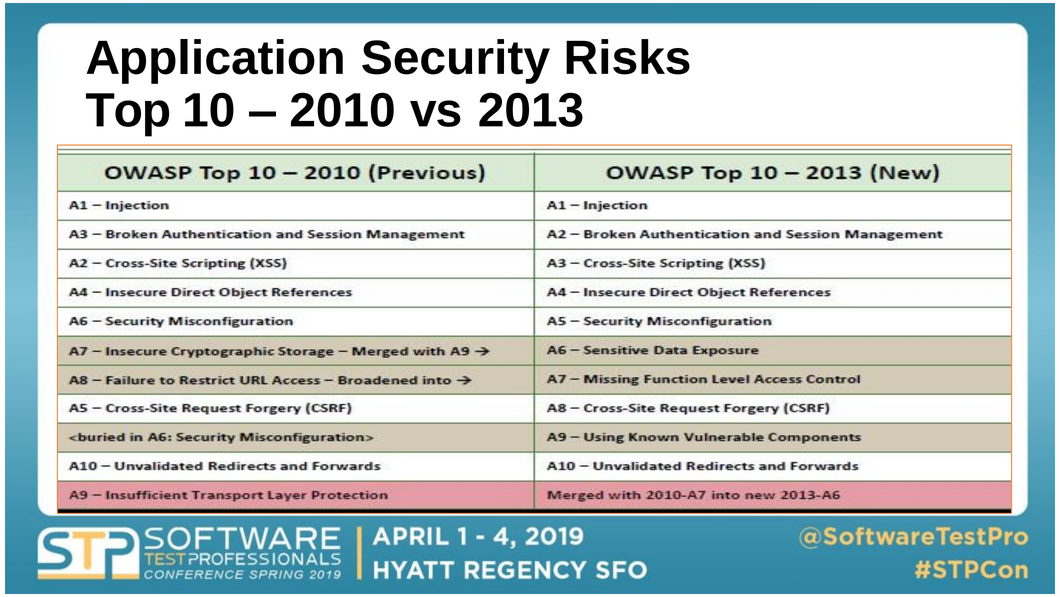


Identifying Application Security Risks

Threat Agents	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impacts	Business Impacts
Appli-cation Specific	Easy: 3	Widespread: 3	Easy: 3	Severe: 3	Business Specific
	Average: 2	Common: 2	Average: 2	Moderate: 2	
	Difficult: 1	Uncommon: 1	Difficult: 1	Minor: 1	

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

23



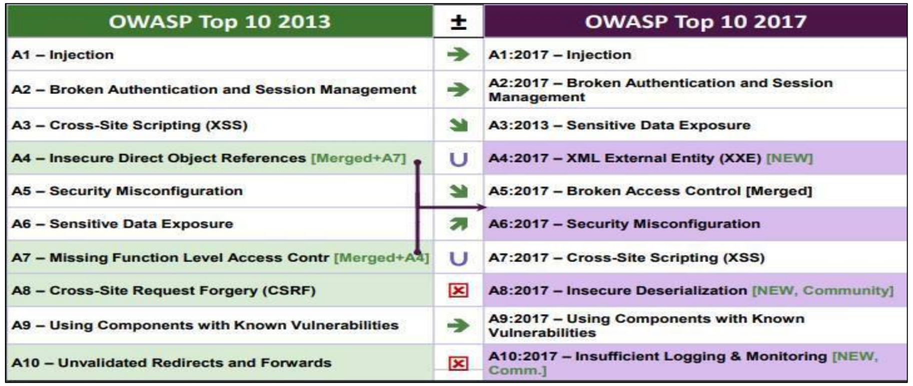
Application Security Risks Top 10 – 2010 vs 2013

OWASP Top 10 – 2010 (Previous)	OWASP Top 10 – 2013 (New)
A1 – Injection	A1 – Injection
A3 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References	A4 – Insecure Direct Object References
A6 – Security Misconfiguration	A5 – Security Misconfiguration
A7 – Insecure Cryptographic Storage – Merged with A9 →	A6 – Sensitive Data Exposure
A8 – Failure to Restrict URL Access – Broadened into →	A7 – Missing Function Level Access Control
A5 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
<buried in A6: Security Misconfiguration>	A9 – Using Known Vulnerable Components
A10 – Unvalidated Redirects and Forwards	A10 – Unvalidated Redirects and Forwards
A9 – Insufficient Transport Layer Protection	Merged with 2010-A7 into new 2013-A6

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

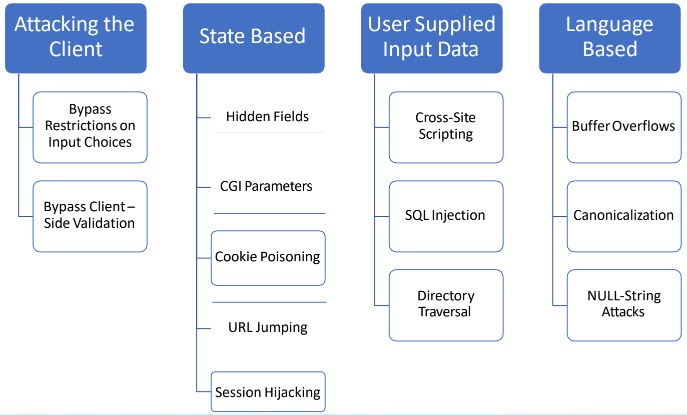
24

Top 10 vulnerabilities of 2013 vs 2017



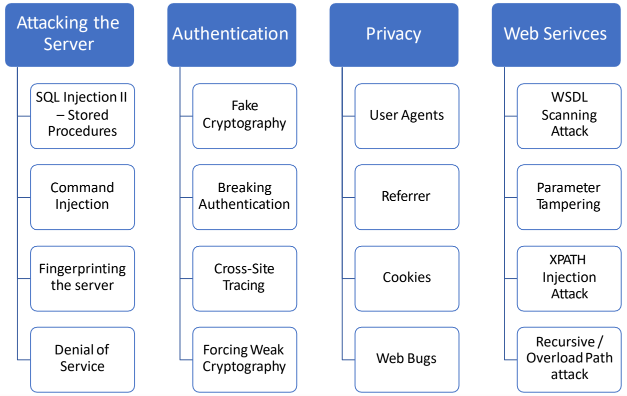
OWASP Top 10 2013	±	OWASP Top 10 2017
A1 – Injection	→	A1:2017 – Injection
A2 – Broken Authentication and Session Management	→	A2:2017 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)	↘	A3:2013 – Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	∪	A4:2017 – XML External Entity (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017 – Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017 – Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	∪	A7:2017 – Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017 – Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017 – Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017 – Insufficient Logging & Monitoring [NEW, Comm.]

Attacks Classification



Exercises

Attacks Classification

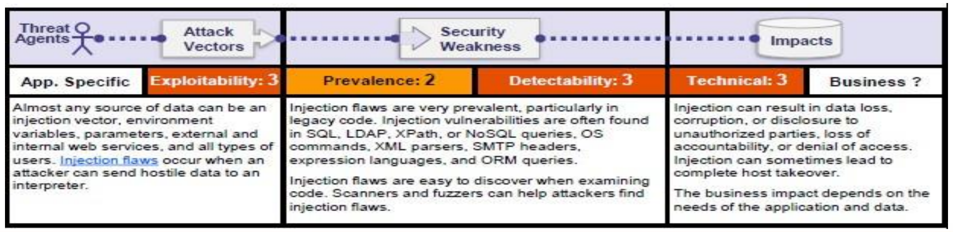


- Attacking the Server
  - SQL Injection II – Stored Procedures
  - Command Injection
  - Fingerprinting the server
  - Denial of Service
- Authentication
  - Fake Cryptography
  - Breaking Authentication
  - Cross-Site Tracing
  - Forcing Weak Cryptography
- Privacy
  - User Agents
  - Referrer
  - Cookies
  - Web Bugs
- Web Serivces
  - WSDL Scanning Attack
  - Parameter Tampering
  - XPATH Injection Attack
  - Recursive / Overload Path attack

OWASP Top 10
Application Security Risks–2017

A1:2017-Injection

- Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

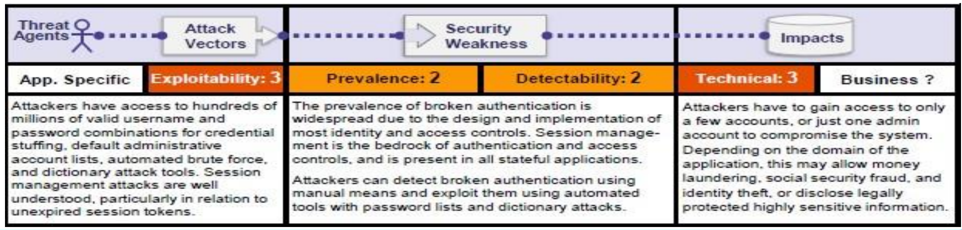

Threat Agents
Attack Vectors
Security Weakness
Impacts

App. Specific	Exploitability: 3	Prevalence: 2	Detectability: 3	Technical: 3	Business ?
Almost any source of data can be an injection vector, environment variables, parameters, external and internal web services, and all types of users. Injection flaws occur when an attacker can send hostile data to an interpreter.		Injection flaws are very prevalent, particularly in legacy code. Injection vulnerabilities are often found in SQL, LDAP, XPath, or NoSQL queries, OS commands, XML parsers, SMTP headers, expression languages, and ORM queries. Injection flaws are easy to discover when examining code. Scanners and fuzzers can help attackers find injection flaws.		Injection can result in data loss, corruption, or disclosure to unauthorized parties, loss of accountability, or denial of access. Injection can sometimes lead to complete host takeover. The business impact depends on the needs of the application and data.	

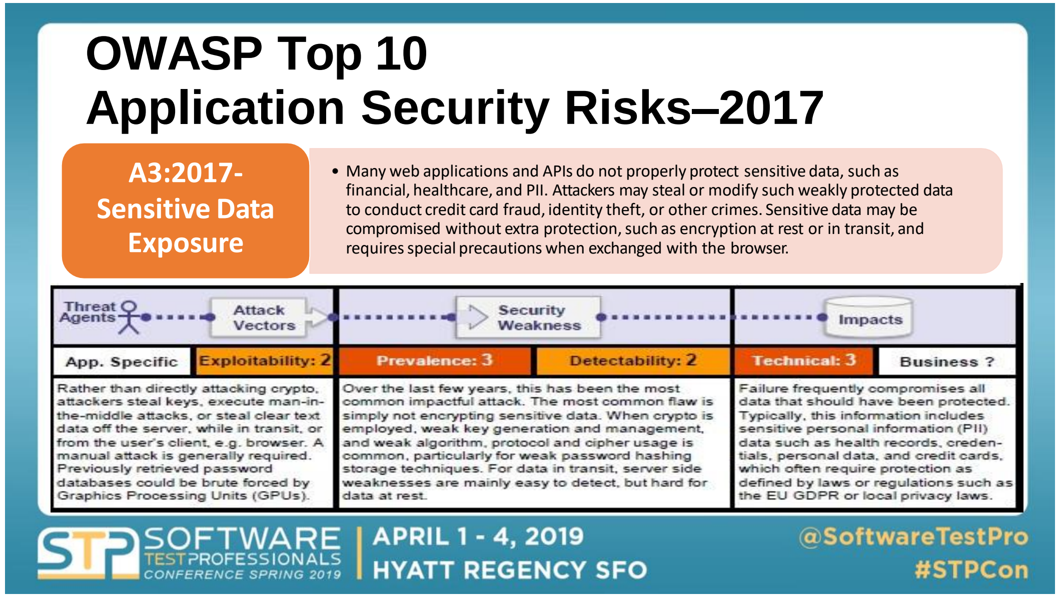
OWASP Top 10
Application Security Risks–2017

A2:2017-Broken Authentication

- Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.


Threat Agents
Attack Vectors
Security Weakness
Impacts

App. Specific	Exploitability: 3	Prevalence: 2	Detectability: 2	Technical: 3	Business ?
Attackers have access to hundreds of millions of valid username and password combinations for credential stuffing, default administrative account lists, automated brute force, and dictionary attack tools. Session management attacks are well understood, particularly in relation to unexpired session tokens.		The prevalence of broken authentication is widespread due to the design and implementation of most identity and access controls. Session management is the bedrock of authentication and access controls, and is present in all stateful applications. Attackers can detect broken authentication using manual means and exploit them using automated tools with password lists and dictionary attacks.		Attackers have to gain access to only a few accounts, or just one admin account to compromise the system. Depending on the domain of the application, this may allow money laundering, social security fraud, and identity theft, or disclose legally protected highly sensitive information.	



OWASP Top 10
Application Security Risks–2017

**A3:2017-
Sensitive Data
Exposure**

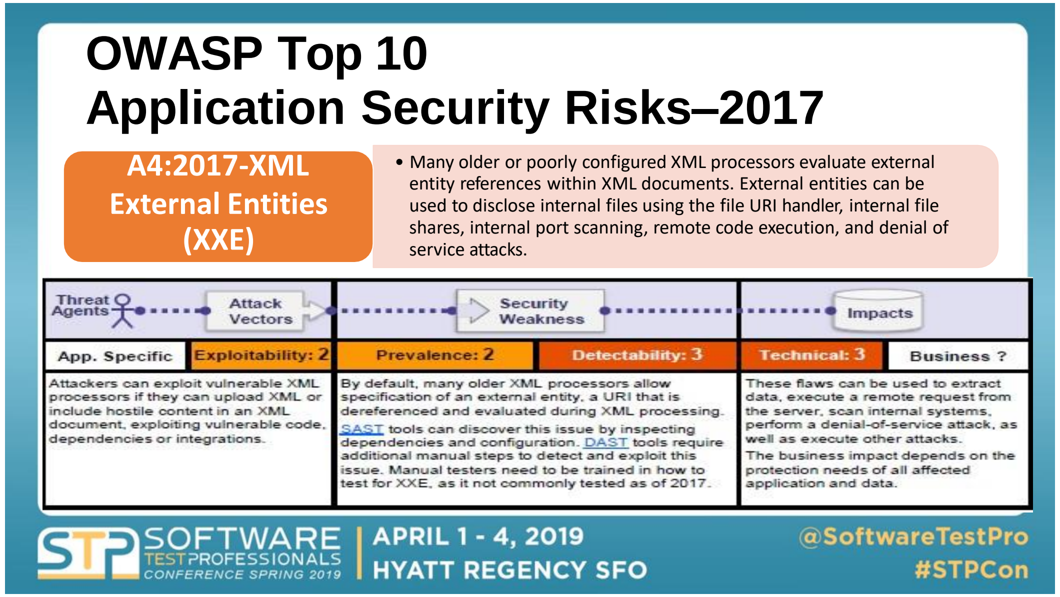
- Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

Threat Agents	Attack Vectors	Security Weakness		Impacts	
App. Specific	Exploitability: 2	Prevalence: 3	Detectability: 2	Technical: 3	Business ?
Rather than directly attacking crypto, attackers steal keys, execute man-in-the-middle attacks, or steal clear text data off the server, while in transit, or from the user's client, e.g. browser. A manual attack is generally required. Previously retrieved password databases could be brute forced by Graphics Processing Units (GPUs).		Over the last few years, this has been the most common impactful attack. The most common flaw is simply not encrypting sensitive data. When crypto is employed, weak key generation and management, and weak algorithm, protocol and cipher usage is common, particularly for weak password hashing storage techniques. For data in transit, server side weaknesses are mainly easy to detect, but hard for data at rest.		Failure frequently compromises all data that should have been protected. Typically, this information includes sensitive personal information (PII) data such as health records, credentials, personal data, and credit cards, which often require protection as defined by laws or regulations such as the EU GDPR or local privacy laws.	

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO

@SoftwareTestPro #STPCon

31



OWASP Top 10
Application Security Risks–2017

**A4:2017-XML
External Entities
(XXE)**

- Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

Threat Agents	Attack Vectors	Security Weakness		Impacts	
App. Specific	Exploitability: 2	Prevalence: 2	Detectability: 3	Technical: 3	Business ?
Attackers can exploit vulnerable XML processors if they can upload XML or include hostile content in an XML document, exploiting vulnerable code, dependencies or integrations.		By default, many older XML processors allow specification of an external entity, a URI that is dereferenced and evaluated during XML processing. SAST tools can discover this issue by inspecting dependencies and configuration. DAST tools require additional manual steps to detect and exploit this issue. Manual testers need to be trained in how to test for XXE, as it not commonly tested as of 2017.		These flaws can be used to extract data, execute a remote request from the server, scan internal systems, perform a denial-of-service attack, as well as execute other attacks. The business impact depends on the protection needs of all affected application and data.	

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO

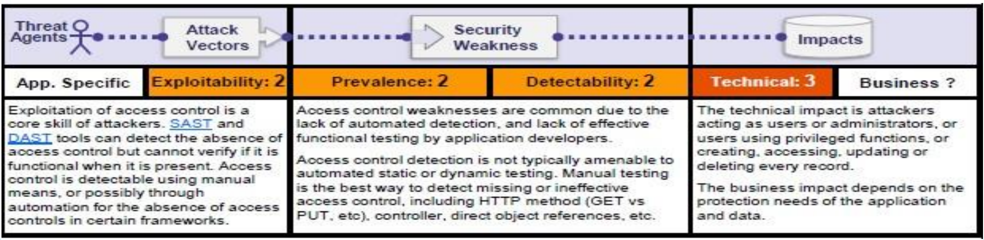
@SoftwareTestPro #STPCon

32

OWASP Top 10
Application Security Risks–2017

A5:2017-Broken Access Control

- Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

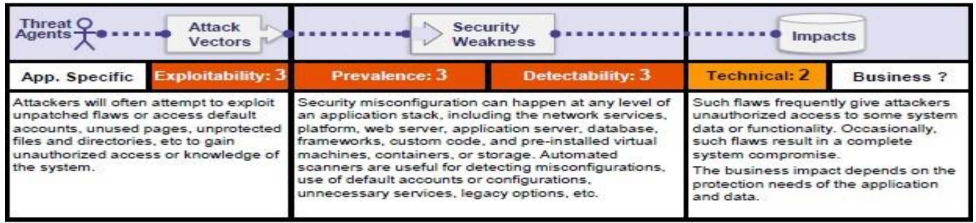

Threat Agents
Attack Vectors
Security Weakness
Impacts

App. Specific	Exploitability: 2	Prevalence: 2	Detectability: 2	Technical: 3	Business ?
Exploitation of access control is a core skill of attackers. SAST and DAST tools can detect the absence of access control but cannot verify if it is functional when it is present. Access control is detectable using manual means, or possibly through automation for the absence of access controls in certain frameworks.		Access control weaknesses are common due to the lack of automated detection, and lack of effective functional testing by application developers. Access control detection is not typically amenable to automated static or dynamic testing. Manual testing is the best way to detect missing or ineffective access control, including HTTP method (GET vs PUT, etc), controller, direct object references, etc.		The technical impact is attackers acting as users or administrators, or users using privileged functions, or creating, accessing, updating or deleting every record. The business impact depends on the protection needs of the application and data.	

OWASP Top 10
Application Security Risks–2017

A6:2017-Security Misconfiguration

- Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched and upgraded in a timely fashion.

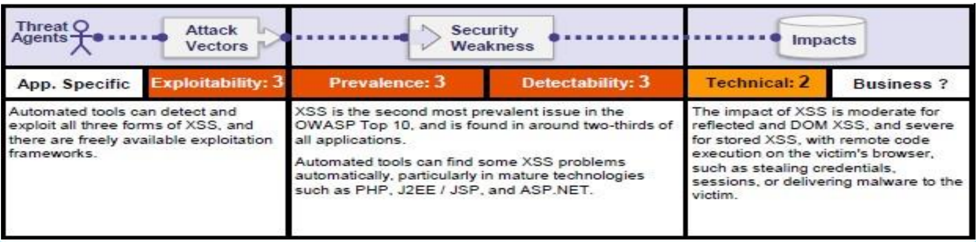

Threat Agents
Attack Vectors
Security Weakness
Impacts

App. Specific	Exploitability: 3	Prevalence: 3	Detectability: 3	Technical: 2	Business ?
Attackers will often attempt to exploit unpatched flaws or access default accounts, unused pages, unprotected files and directories, etc to gain unauthorized access or knowledge of the system.		Security misconfiguration can happen at any level of an application stack, including the network services, platform, web server, application server, database, frameworks, custom code, and pre-installed virtual machines, containers, or storage. Automated scanners are useful for detecting misconfigurations, use of default accounts or configurations, unnecessary services, legacy options, etc.		Such flaws frequently give attackers unauthorized access to some system data or functionality. Occasionally, such flaws result in a complete system compromise. The business impact depends on the protection needs of the application and data.	

OWASP Top 10
Application Security Risks–2017

A7:2017-Cross-Site Scripting (XSS)

- XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.



Threat Agents	Attack Vectors	Security Weakness		Impacts	
App. Specific	Exploitability: 3	Prevalence: 3	Detectability: 3	Technical: 2	Business ?
Automated tools can detect and exploit all three forms of XSS, and there are freely available exploitation frameworks.		XSS is the second most prevalent issue in the OWASP Top 10, and is found in around two-thirds of all applications. Automated tools can find some XSS problems automatically, particularly in mature technologies such as PHP, J2EE / JSP, and ASP.NET.		The impact of XSS is moderate for reflected and DOM XSS, and severe for stored XSS, with remote code execution on the victim's browser, such as stealing credentials, sessions, or delivering malware to the victim.	

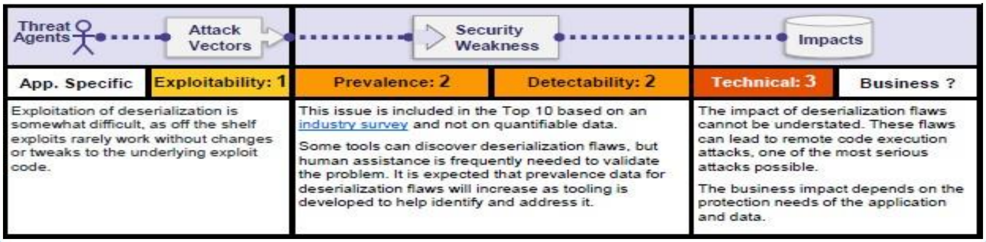
SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

35

OWASP Top 10
Application Security Risks–2017

A8:2017-Insecure Deserialization

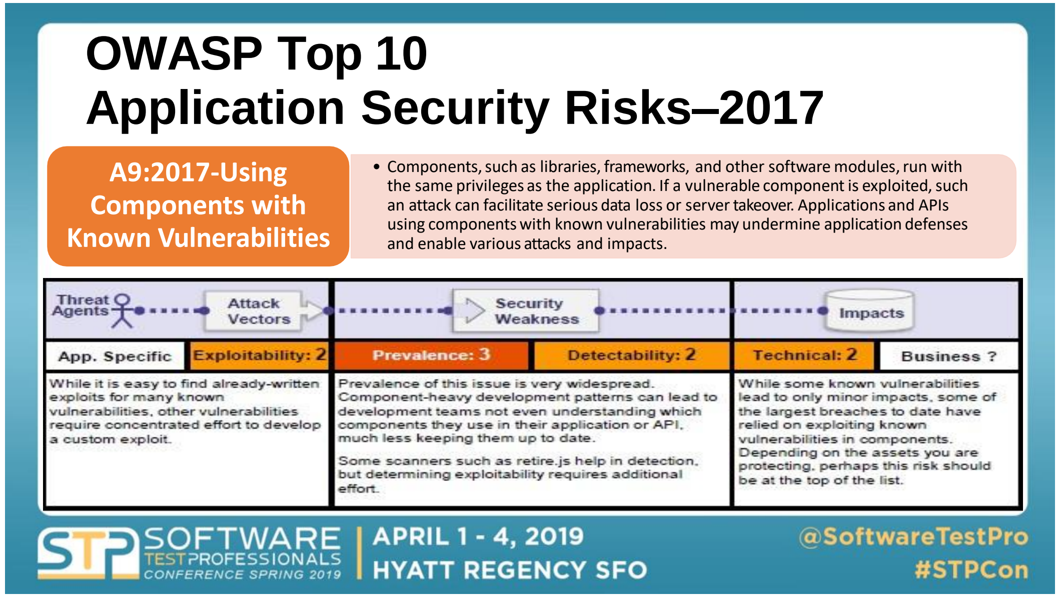
- Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.



Threat Agents	Attack Vectors	Security Weakness		Impacts	
App. Specific	Exploitability: 1	Prevalence: 2	Detectability: 2	Technical: 3	Business ?
Exploitation of deserialization is somewhat difficult, as off the shelf exploits rarely work without changes or tweaks to the underlying exploit code.		This issue is included in the Top 10 based on an industry survey and not on quantifiable data. Some tools can discover deserialization flaws, but human assistance is frequently needed to validate the problem. It is expected that prevalence data for deserialization flaws will increase as tooling is developed to help identify and address it.		The impact of deserialization flaws cannot be understated. These flaws can lead to remote code execution attacks, one of the most serious attacks possible. The business impact depends on the protection needs of the application and data.	

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

36



OWASP Top 10
Application Security Risks–2017

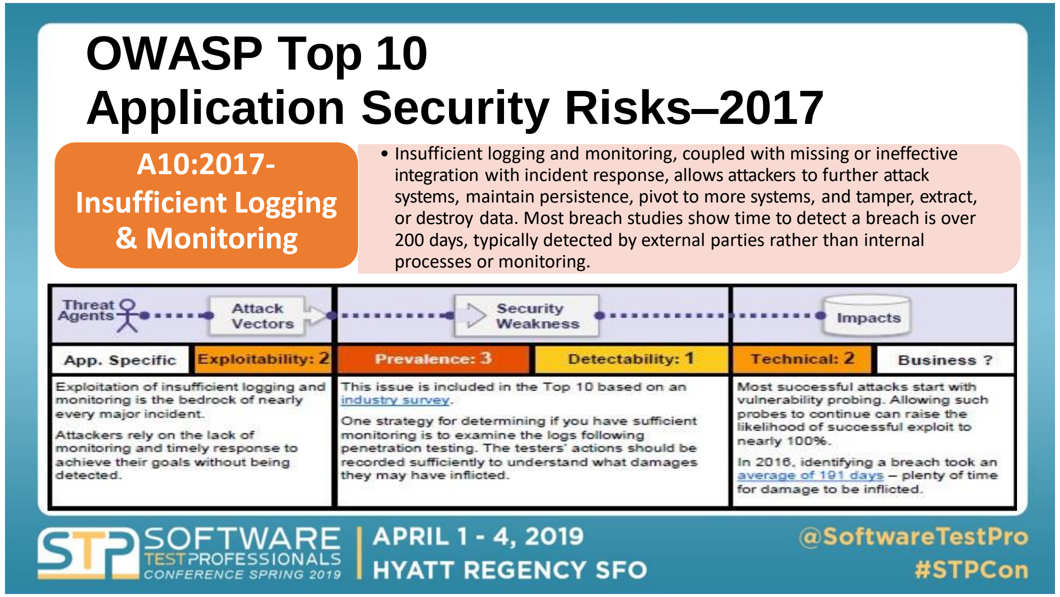
A9:2017-Using Components with Known Vulnerabilities

- Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

Threat Agents	Attack Vectors	Security Weakness		Impacts	
App. Specific	Exploitability: 2	Prevalence: 3	Detectability: 2	Technical: 2	Business ?
While it is easy to find already-written exploits for many known vulnerabilities, other vulnerabilities require concentrated effort to develop a custom exploit.		Prevalence of this issue is very widespread. Component-heavy development patterns can lead to development teams not even understanding which components they use in their application or API, much less keeping them up to date. Some scanners such as retire.js help in detection, but determining exploitability requires additional effort.		While some known vulnerabilities lead to only minor impacts, some of the largest breaches to date have relied on exploiting known vulnerabilities in components. Depending on the assets you are protecting, perhaps this risk should be at the top of the list.	

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

37



OWASP Top 10
Application Security Risks–2017

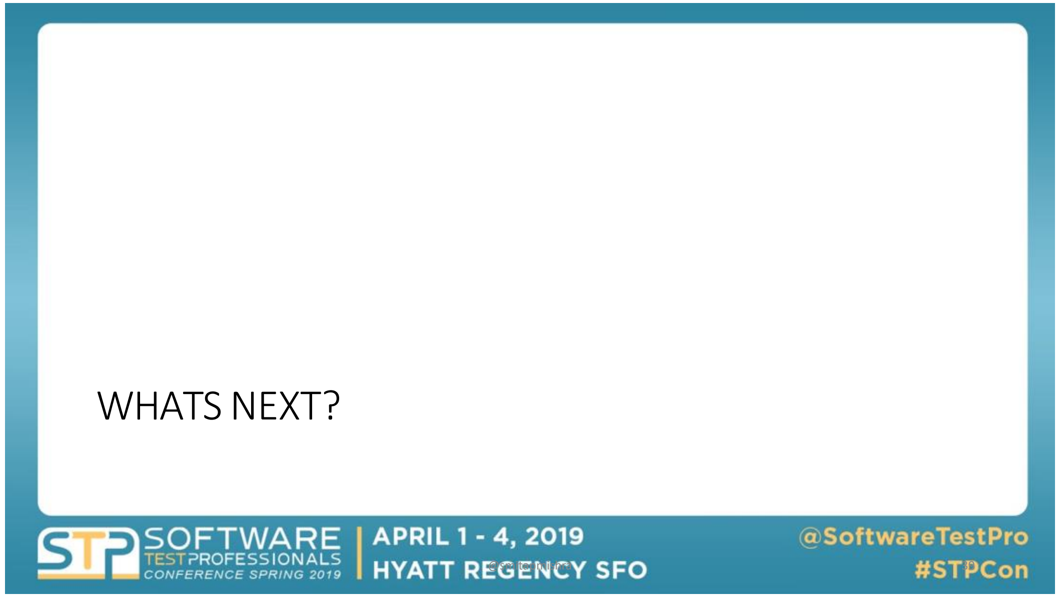
A10:2017-Insufficient Logging & Monitoring

- Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Threat Agents	Attack Vectors	Security Weakness		Impacts	
App. Specific	Exploitability: 2	Prevalence: 3	Detectability: 1	Technical: 2	Business ?
Exploitation of insufficient logging and monitoring is the bedrock of nearly every major incident. Attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.		This issue is included in the Top 10 based on an industry survey. One strategy for determining if you have sufficient monitoring is to examine the logs following penetration testing. The testers' actions should be recorded sufficiently to understand what damages they may have inflicted.		Most successful attacks start with vulnerability probing. Allowing such probes to continue can raise the likelihood of successful exploit to nearly 100%. In 2016, identifying a breach took an average of 191 days – plenty of time for damage to be inflicted.	

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

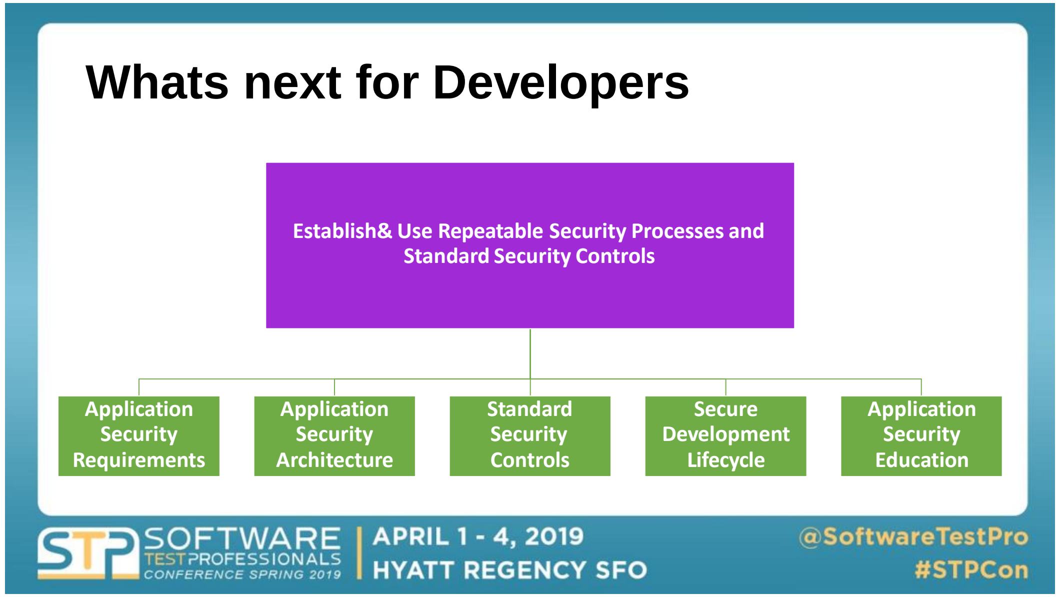
38



WHATS NEXT?

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

39



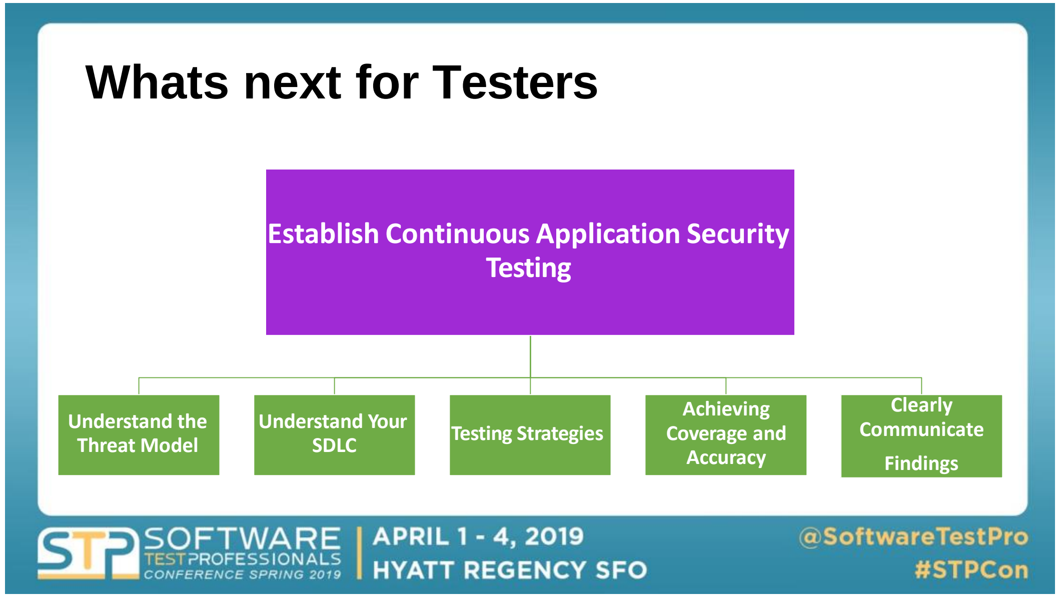
Whats next for Developers

Establish& Use Repeatable Security Processes and Standard Security Controls

- Application Security Requirements
- Application Security Architecture
- Standard Security Controls
- Secure Development Lifecycle
- Application Security Education

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

40



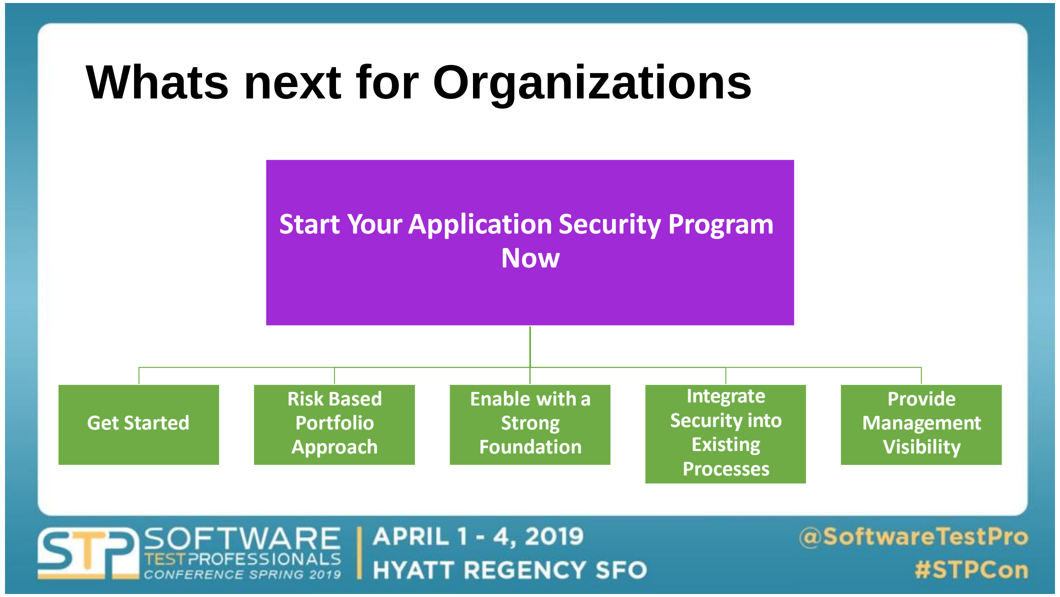
Whats next for Testers

Establish Continuous Application Security Testing

- Understand the Threat Model
- Understand Your SDLC
- Testing Strategies
- Achieving Coverage and Accuracy
- Clearly Communicate Findings

STP SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

41



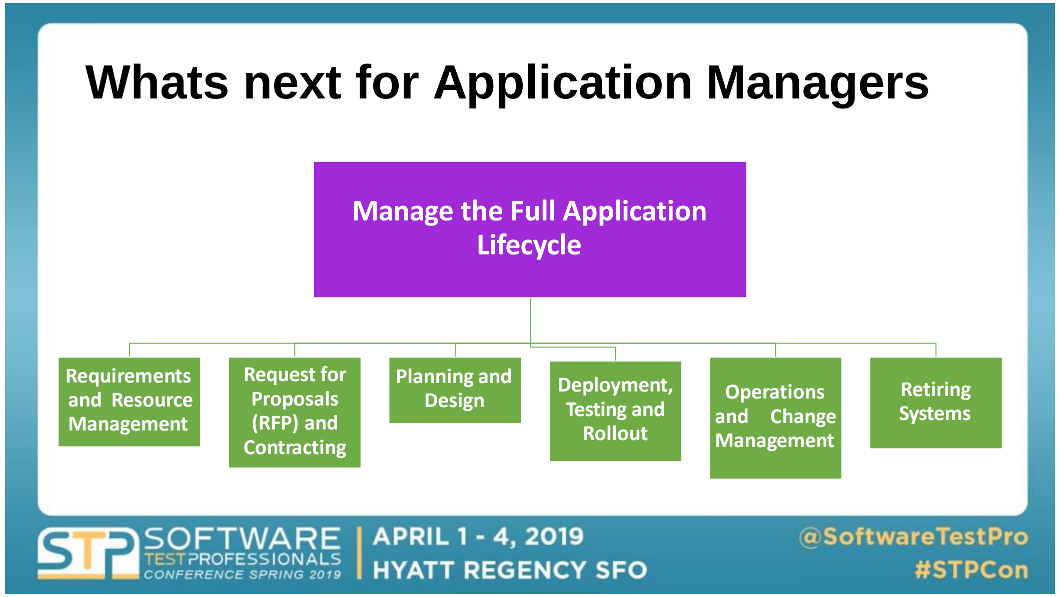
Whats next for Organizations

Start Your Application Security Program Now

- Get Started
- Risk Based Portfolio Approach
- Enable with a Strong Foundation
- Integrate Security into Existing Processes
- Provide Management Visibility

STP SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

42



Whats next for Application Managers

Manage the Full Application Lifecycle

- Requirements and Resource Management
- Request for Proposals (RFP) and Contracting
- Planning and Design
- Deployment, Testing and Rollout
- Operations and Change Management
- Retiring Systems

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

43

THANK YOU

STAY IN TOUCH

Smita.Mishra@qazone.in

Twitter : @smitapmishra

Linkedin: http://www.linkedin.com/in/smitapmishra

SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019 | APRIL 1 - 4, 2019 HYATT REGENCY SFO @SoftwareTestPro #STPCon

44