Fortify

Kickoff Meeting

Application Security Overview

Defining your goals, objectives and deliverables for

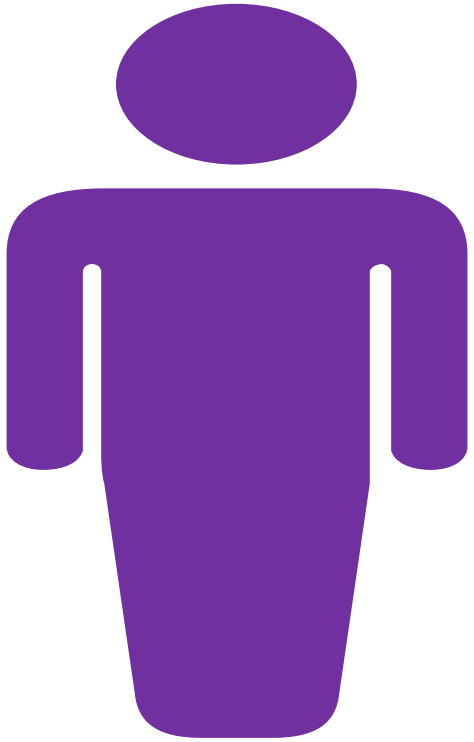
WebInspect and WebInspect Enterprise.

Fortify

Application Security - Overview

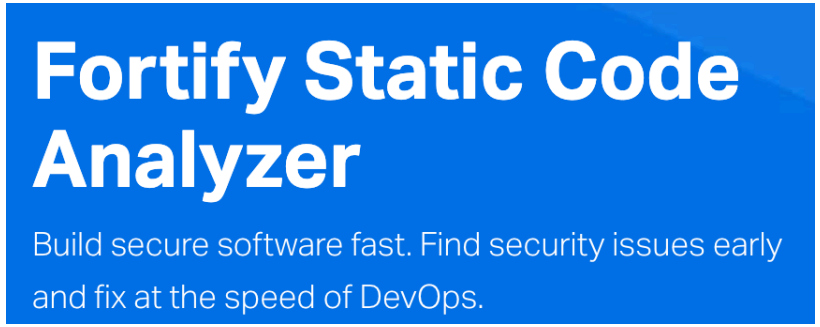
Software Security Assurance (SSA) is the process of ensuring that software is designed to operate at a level of security that is consistent with the potential harm that could result from the loss, inaccuracy, alteration, unavailability, or misuse of the data and resources that it uses, controls, and protects.

Application Security - Overview



The primary goal of Application Security is to **reduce your business risk** by eliminating vulnerabilities in your source code. For that reason, we are going to need a process that repairs the code.

Application Security – Automated Types



1. Dynamic Testing (DAST)

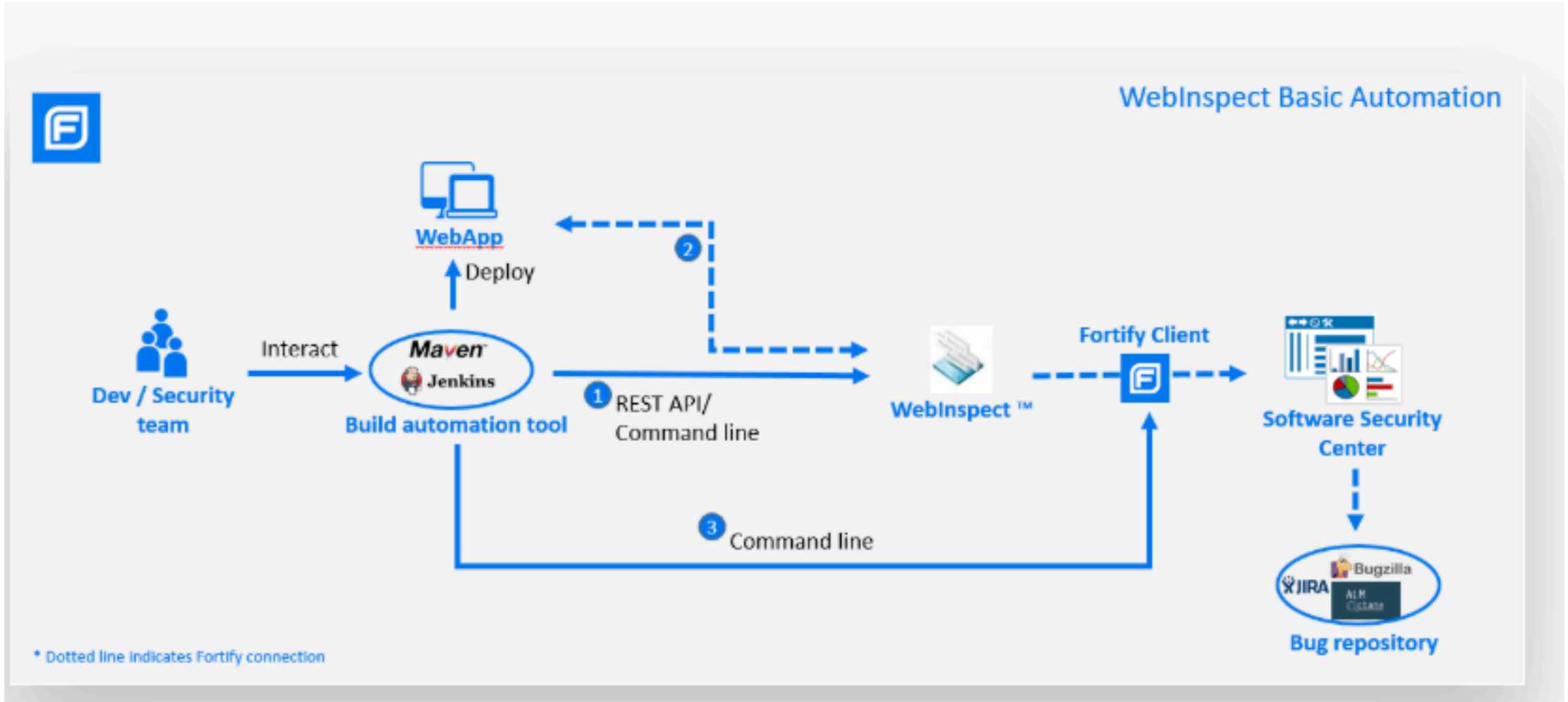
   This testing requires the application to be completed and deployed. Lower ROI then static testing since you pay developers twice.

2. Static Testing (SAST)

   This testing is performed as the developer writes the code. This allows fixes to be made before deployment.

   NOTE: The ideal is to perform both DAST and SAST.

Fortify

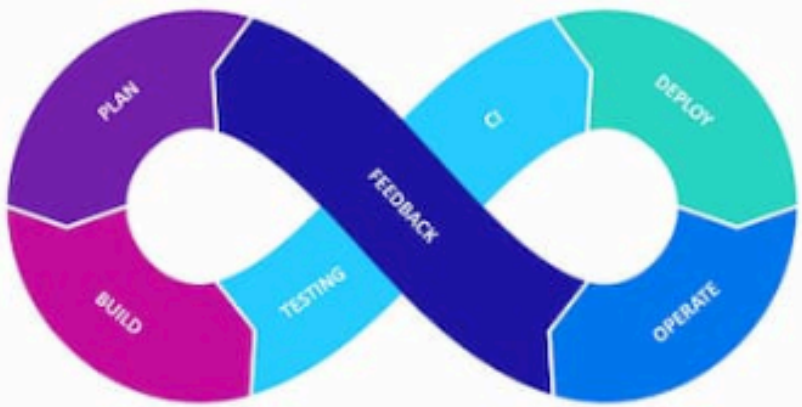

WebInspect Basic Automation
WebApp
Deploy
2
Interact
Maven
Jenkins
Dev / Security team
Build automation tool
1
REST API/
Command line
WebInspect ™
Fortify Client
Software Security Center
3
Command line
Bugzilla
JIRA
Bug repository
* Dotted line indicates Fortify connection

Fortify

Fortify Development Lifecycle Integrations

IDEs	SOURCE CONTROL	CI/CD SERVERS	TICKETING SYSTEMS
eclipse	GitHub	Jenkins	ALM OCTANE
Visual Studio	Bitbucket	Bamboo	Jira
IntelliJ IDEA		Azure DevOps	Bugzilla

OPEN SOURCE	BUILD TOOLS	DEVELOPER TRAINING
Sonatype		MICRO FOCUS
snyk	Maven	SECURE CODE WARRIOR
WhiteSource	Gradle	
BLACKDUCK		



Application Security - Overview

Tools do NOT fix problems, People do !

Tools can support people in finding problems and provide guidance for the human to fix.

Fortify

https://www.opensamm.org/

Software Assurance Maturity Model

The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization. The resources provided by SAMM will aid in:

- *Evaluating an organization's existing software security practices*
- *Building a balanced software security program in well-defined iterations*
- *Demonstrating concrete improvements to a security assurance program*
- *Defining and measuring security-related activities within an organization*

Fortify

BSIMM

What is BSIMM | Download the BSIMM | BSIMM Framework | Events | Resources

What people say about the BSIMM

Since 2008, the BSIMM has served as an effective tool for understanding how organizations of all shapes and sizes, including some of the most advanced security teams in the world, are executing their software security strategies. The current BSIMM data reflect how many organizations are adapting their approaches to address the new dynamics of modern development and deployment practices, such as shorter release cycles, increased use of automation, and software-defined infrastructure.

Jim Routh | Head of enterprise information risk management at MassMutual

Fortify

Application Security – Process - 1

https://www.owasp.org/index.php/Top_10-2017_What%27s_Next_for_Application_Managers

Manage the Full Application Lifecycle

Applications belong to the most complex systems humans regularly create and maintain. IT management for an application should be performed by IT specialists who are responsible for the overall IT lifecycle of an application. We suggest establishing the role of application manager as technical counterpart to the application owner. The application manager is in charge of the whole application lifecycle from the IT perspective, from collecting the requirements until the process of retiring systems, which is often overlooked.

Requirements and Resource Management

- Collect and negotiate the business requirements for an application with the business, including the protection requirements with regard to confidentiality, authenticity, integrity and availability of all data assets, and the expected business logic.
- Compile the technical requirements including functional and nonfunctional security requirements.
- Plan and negotiate the budget that covers all aspects of design, build, testing and operation, including security activities.

Request for Proposals (RFP) and Contracting

- Negotiate the requirements with internal or external developers, including guidelines and security requirements with respect to your security program, e.g. SDLC, best practices.
- Rate the fulfillment of all technical requirements, including a planning and design phase.
- Negotiate all technical requirements, including design, security, and service level agreements (SLA).
- Adopt templates and checklists, such as OWASP Secure Software Contract Annex.
  Note: The annex is for US contract law, so please consult qualified legal advice before using the sample annex.

Application Security – Process - 2

https://www.owasp.org/index.php/Top_10-2017_What%27s_Next_for_Application_Managers

Planning and Design

- Negotiate planning and design with the developers and internal shareholders, e.g. security specialists.
- Define the security architecture, controls, and countermeasures appropriate to the protection needs and the expected threat level. This should be supported by security specialists.
- Ensure that the application owner accepts remaining risks or provides additional resources.
- In each sprint, ensure security stories are created that include constraints added for non-functional requirements.

Deployment, Testing, and Rollout

- Automate the secure deployment of the application, interfaces and all required components, including needed authorizations.
- Test the technical functions and integration with the IT architecture and coordinate business tests.
- Create "use" and "abuse" test cases from technical and business perspectives.
- Manage security tests according to internal processes, the protection needs, and the assumed threat level by the application.
- Put the application in operation and migrate from previously used applications if needed.
- Finalize all documentation, including the change management data base (CMDB) and security architecture.

Application Security – Process - 3

https://www.owasp.org/index.php/Top_10-2017_What%27s_Next_for_Application_Managers

Operations and Change Management

- Operations must include guidelines for the security management of the application (e.g. patch management).
- Raise the security awareness of users and manage conflicts about usability vs. security.
- Plan and manage changes, e.g. migrate to new versions of the application or other components like OS, middleware, and libraries.
- Update all documentation, including in the CMDB and the security architecture, controls, and countermeasures, including any runbooks or project documentation.

Retiring Systems

- Any required data should be archived. All other data should be securely wiped.
- Securely retire the application, including deleting unused accounts and roles and permissions.
- Set your application's state to retired in the CMDB.

Fortify

Application Security Overview

Key Concepts

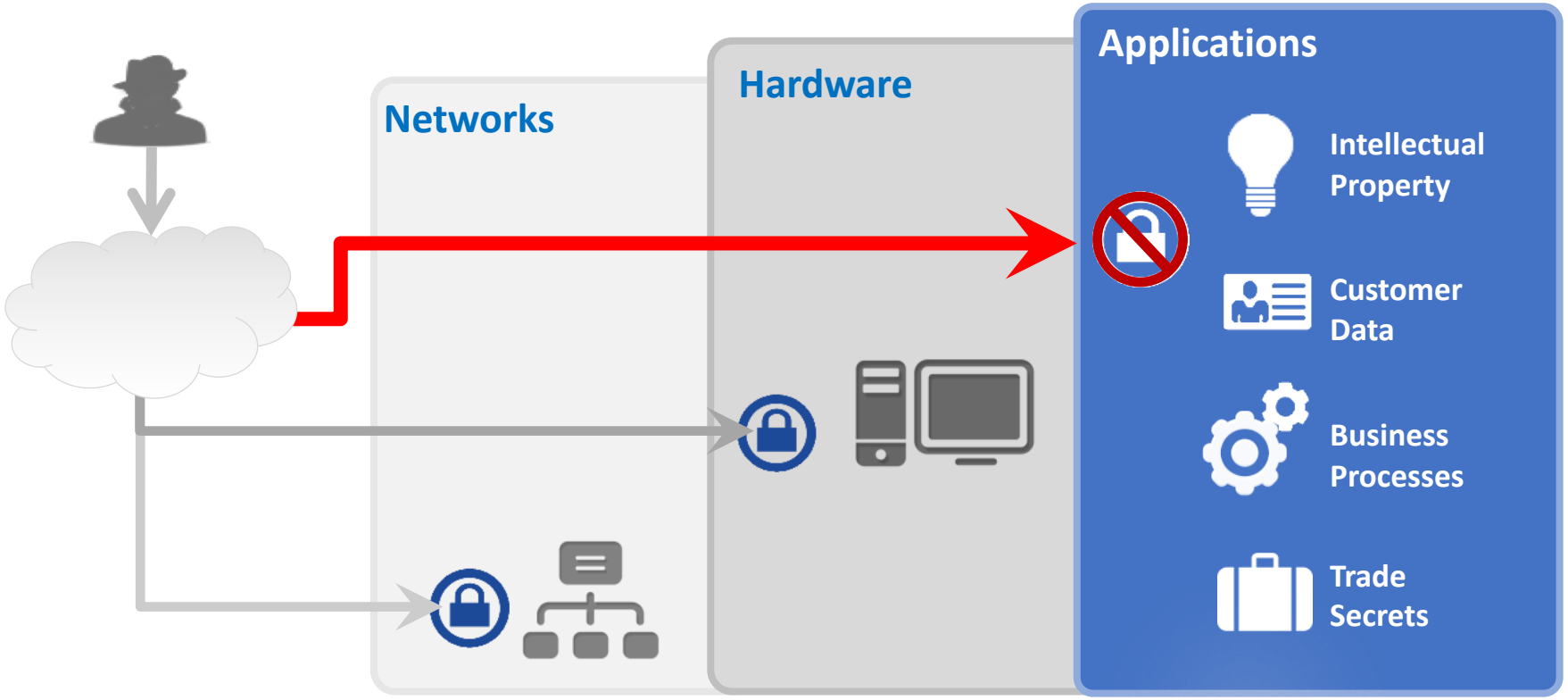
Application Security – Overview

https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (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]	U	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.]

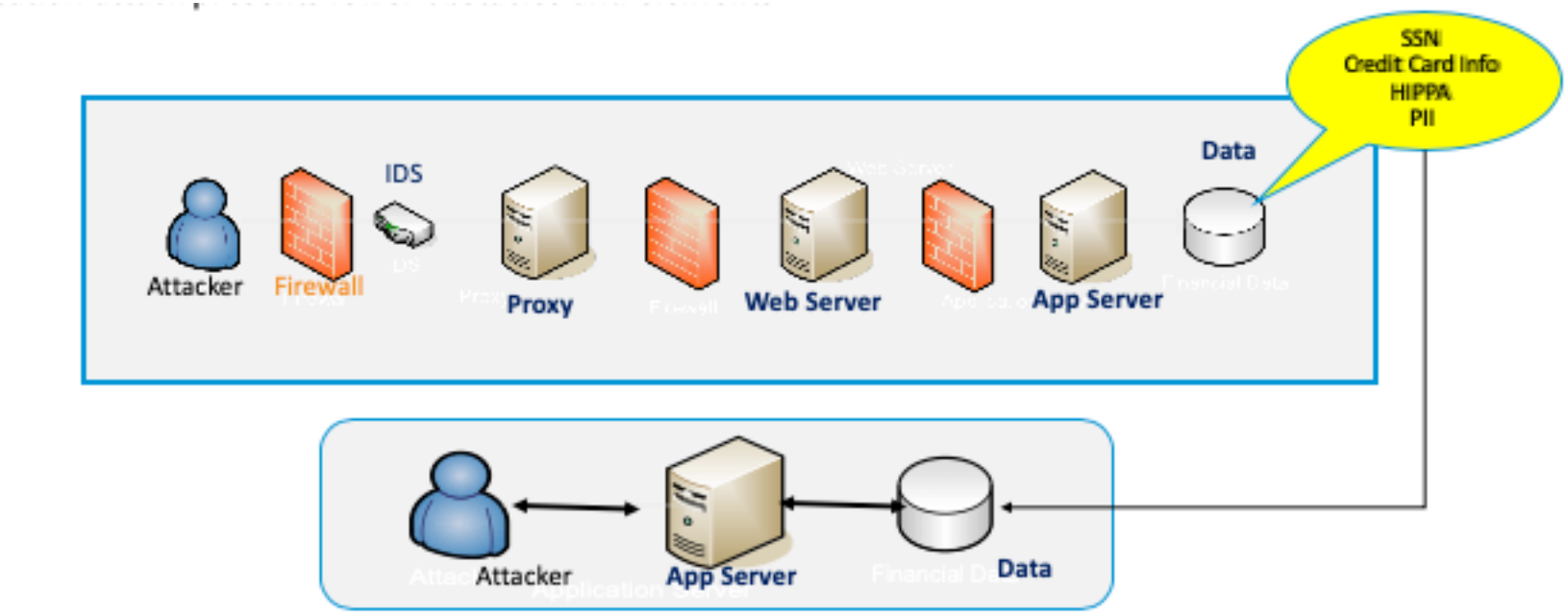
Cyber attackers are targeting applications

Over 80% of breaches occur through software applications



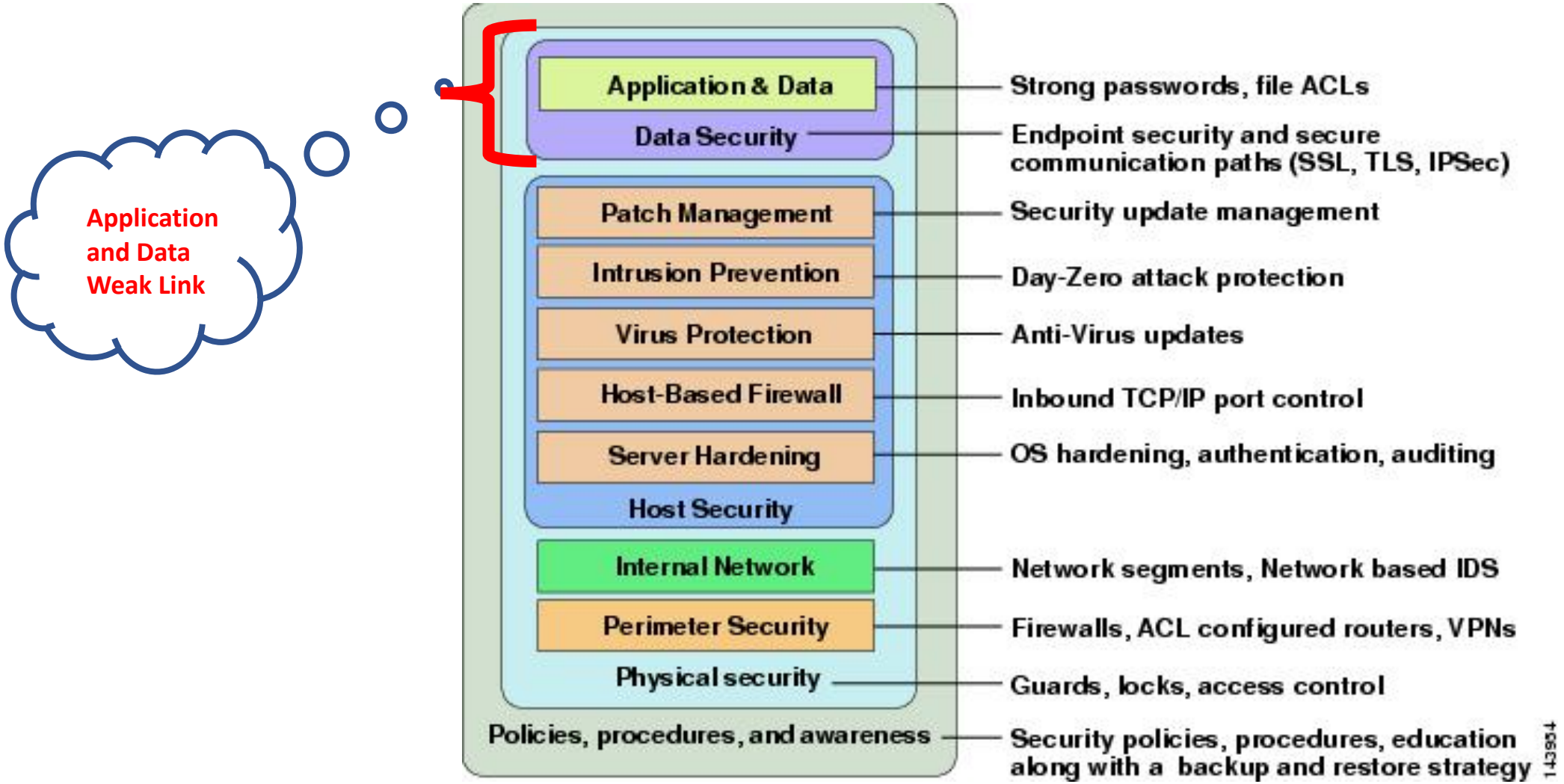
Application Vulnerabilities

- It is important to clarify that adversaries have fewer obstacles when performing an attack on code
- For example, an infrastructure attack presents all the following obstacles and elements
- An application attack presents fewer obstacles and elements



IDS = Intrusion Detection System

Defense in Depth



Fortify

Vulnerability –Example - What are you protecting ?

Do you create detailed threat models for your critical apps ?

https://www.owasp.org/index.php/Testing_Guide_Introduction#Threat_Modeling

To develop a threat model, we recommend taking a simple approach that follows the NIST 800-30 [11] standard for risk assessment. This approach involves:

- **Decomposing the application** – use a process of manual inspection to understand how the application works, its assets, functionality, and connectivity.
- **Defining and classifying the assets** – classify the assets into tangible and intangible assets and rank them according to business importance.
- **Exploring potential vulnerabilities** - whether technical, operational, or management.
- **Exploring potential threats** – develop a realistic view of potential attack vectors from an attacker's perspective, by using threat scenarios or attack trees.
- **Creating mitigation strategies** – develop mitigating controls for each of the threats deemed to be realistic.

Application Security - Overview

Tools do NOT fix problems, People do !

Tools can support people in finding problems and provide guidance for the human to fix.

Examples of AppSec Breaches

https://en.wikipedia.org/wiki/List_of_data_breaches

Wikipedia contributors. (2019, December 23). List of data breaches. In *Wikipedia, The Free Encyclopedia*. Retrieved 01:01, December 30, 2019, from https://en.wikipedia.org/w/index.php?title=List_of_data_breaches&oldid=932101229

WordPress	2018			hacked	[327]
Adobe Inc.	2019	7.5 million	tech	poor security	[11]
Amazon Japan G.K.	2019	unknown	web	accidentally published	[17][18]
2019 Bulgarian revenue agency hack	2019	over 5,000,000	government	hacked	[57]
Canva	2019	140,000,000	web	hacked	[59][60][61]
Capital One	2019	106,000,000	financial	unsecured S3 bucket	[62][63]
Desjardins	2019	2,900,000	financial	inside job	[93]
DoorDash	2019	4,900,000	web	hacked	[95]

Fortify

SQL Injection is Still and issue

Akorn Inc. Has Customer Database Stolen, Records Offered To Highest Bidder

cyberGRC *by Robert Westmacott - June 19, 2015*

33 0

CSO reports that a Hacker responsible for the recent Akorn breach says they compromised the company to teach them a lesson in security – GULP!

Akorn Inc., a niche pharmaceutical company Lake Forest, IL, has had a customer database with more than 50,000 records compromised by a hacker who is offering to sell the data to the highest bidder or back to the company, whichever comes first.

Fortify

Recent Security Incidents

Availability

http://www.wired.com/2015/05/hack-brief-theres-new-iphone-text-message-attack/

EMILY DREYFUSS 05.27.15 1:28 PM

HACK BRIEF: THERE'S A NEW IPHONE TEXT MESSAGE ATTACK

381

The Hack

YOU CAN CRASH an iPhone merely by texting it the exact right string of English and Arabic characters .

LATEST NEWS

ALEX DAVIES
Chevy's Taking Apple CarPlay and Android Auto to the Mass...
26 MINS

Fortify

Recent Security Incidents

http://www.wired.com/2015/05/hackers-hit-irs-access-100000-taxpayers-files/

Confidentiality
&
Integrity

ANDY GREENBERG 05.26.15 6:48 PM

HACKERS HIT THE IRS AND MAKE OFF WITH 100K TAXPAYERS' FILES

ANDREW HARRER/BLOOMBERG VIA GETTY IMAGES

Fortify

Data breach results in $4.8 million HIPAA

April 2014

settlemen

http://www.hhs.gov/news/press/2014pres/05/20140507b.html

Two health care organizations have agreed to settle charges that they potentially violated the Health Insurance Portability and Accountability Act of 1996 (HIPAA) Privacy and Security Rules by failing to secure thousands of patients' electronic protected health information (ePHI) held on their network. The monetary payments of $4,800,000 include the largest HIPAA settlement to date.

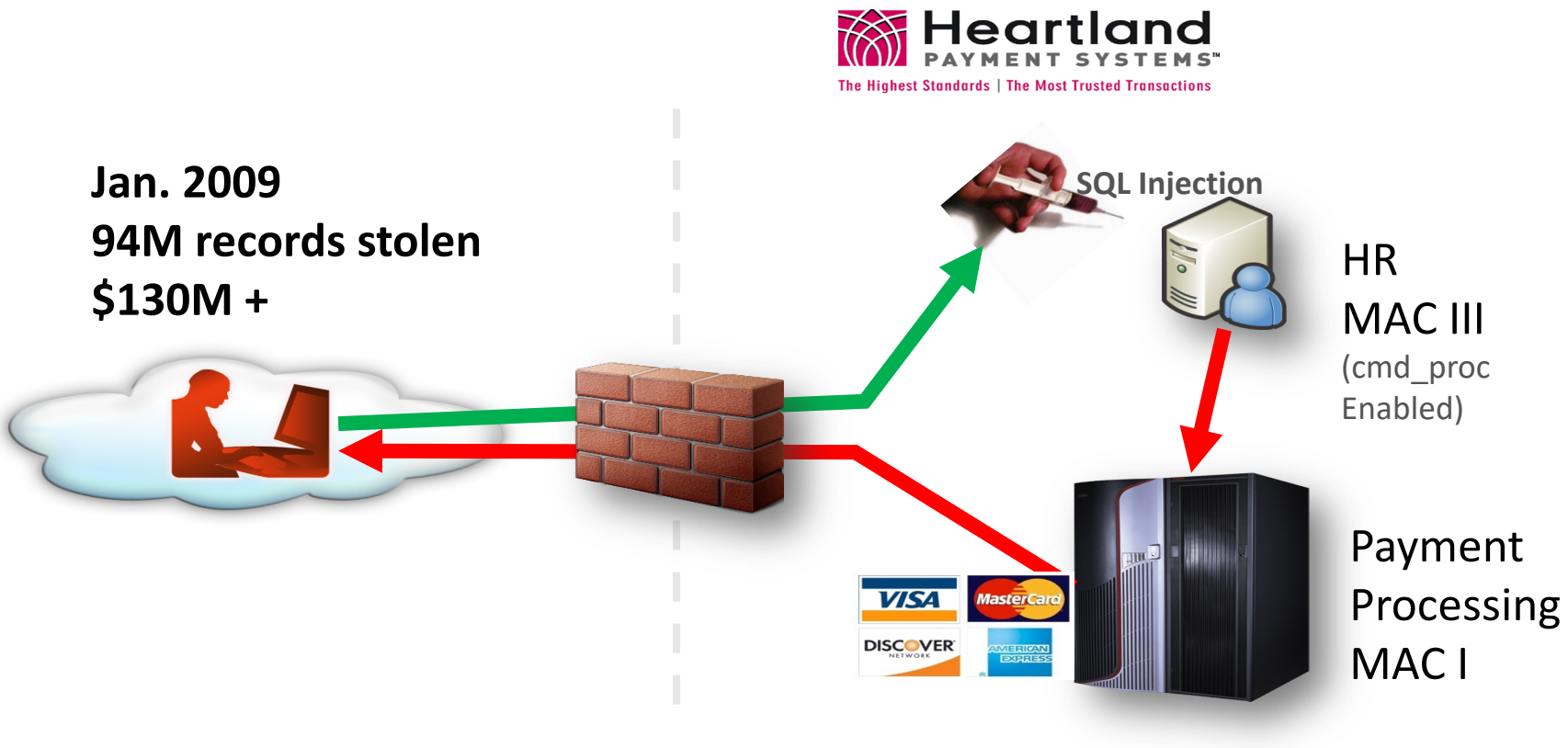
Cross Site Scripting Famous Example

PayPal, circa 2004 - 2006

- Steal credit card numbers
  1. Users access URL on genuine PayPal site
  2. Page modified via XSS attack to silently redirect user to external server
  3. Fake PayPal Member log-in page
  4. User supplies login credentials to fake site
- Exploitable for two years

Injection Attack Costly Example

Heartland Payment Systems – Jan, 2009



Fortify

Authentication Design Flaws

Stolen passwords to blame for OPM breach; director may take the fall

by
Michael Heller
Senior Reporter

Published: 25 Jun 2015

The OPM director told a Senate hearing that passwords stolen from a contractor led to the OPM breach. Now, her job is on the line and the number of breached records could be on the rise.

Fortify

Insider Threats

6 Types of Insider Threats – And How to Snuff Them Out

The Rogue Employee
AKA: Shadow IT, Rogue IT
Description: They have many aliases, but one definite goal – to take valuable data and leverage it into monetary gain, revenge or even some revolutionary crusade. Watch out for these ones in particularly – they are usually well-trained and can penetrate network defenses without setting off alarms.

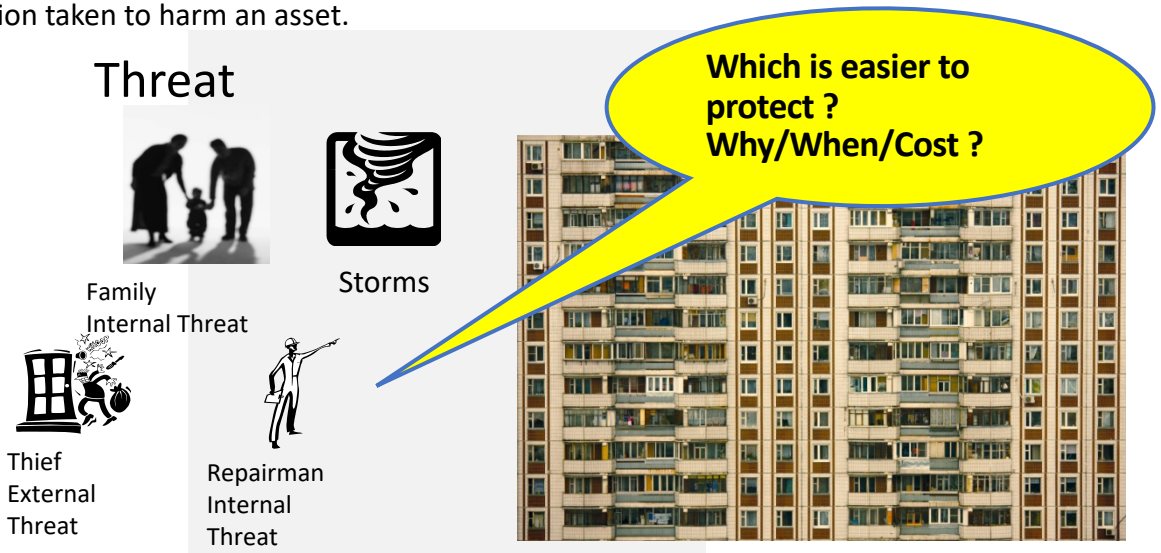
Fired or Disgruntled Worker
AKA: Pinch a Penny from a 1 million Transactions
Description: Think Office Space – where workers on their way out devise a way to rip off the company. With revenge driving their rage, they may even create malware to syphon off trade secrets, financial information or even customer information.

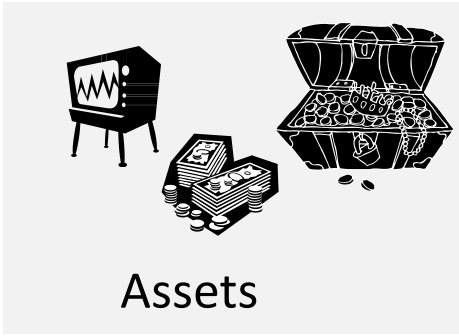
House Exercise:

LIST THE POTENTIAL ATTACKS/SAFEGUARDS

Attack (or exploit). An action taken to harm an asset.

House is like a
Desktop Computer





Apartment is like a
VM/Cloud/Shared
Computing

Fortify

Application Security – Overview
Key Security Concepts

- **Confidentiality**
  - ✓Prevent the disclosure of information to unauthorized individuals or systems.
- **Integrity**
  - ✓Prevent data modifications that are untraceable.
- **Availability**
  - ✓ Ensure the information is available when it is needed

Fortify

Application Security – Overview

https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

Technical Impact Factors

Technical impact can be broken down into factors aligned with the traditional security areas of concern: confidentiality, integrity, availability, and accountability. The goal is to estimate the magnitude of the impact on the system if the vulnerability were to be exploited.

Loss of confidentiality
How much data could be disclosed and how sensitive is it? Minimal non-sensitive data disclosed (2), minimal critical data disclosed (6), extensive non-sensitive data disclosed (6), extensive critical data disclosed (7), all data disclosed (9)

Loss of integrity
How much data could be corrupted and how damaged is it? Minimal slightly corrupt data (1), minimal seriously corrupt data (3), extensive slightly corrupt data (5), extensive seriously corrupt data (7), all data totally corrupt (9)

Loss of availability
How much service could be lost and how vital is it? Minimal secondary services interrupted (1), minimal primary services interrupted (5), extensive secondary services interrupted (5), extensive primary services interrupted (7), all services completely lost (9)

Loss of accountability
Are the threat agents' actions traceable to an individual? Fully traceable (1), possibly traceable (7), completely anonymous (9)

Application Security - Overview

Application security encompasses measures taken to improve the security of an application often by finding, fixing and preventing security vulnerabilities.

Key Terms

- **Asset**. Resource of value such as the data in a database, money in an account, file on the filesystem or any system resource.
- **Vulnerability.** A weakness or gap in security program that can be exploited by threats to gain unauthorized access to an asset.
- **Attack** (or exploit). An action taken to harm an asset.
- **Threat**. Anything that can exploit a vulnerability and obtain, damage, or destroy an asset.

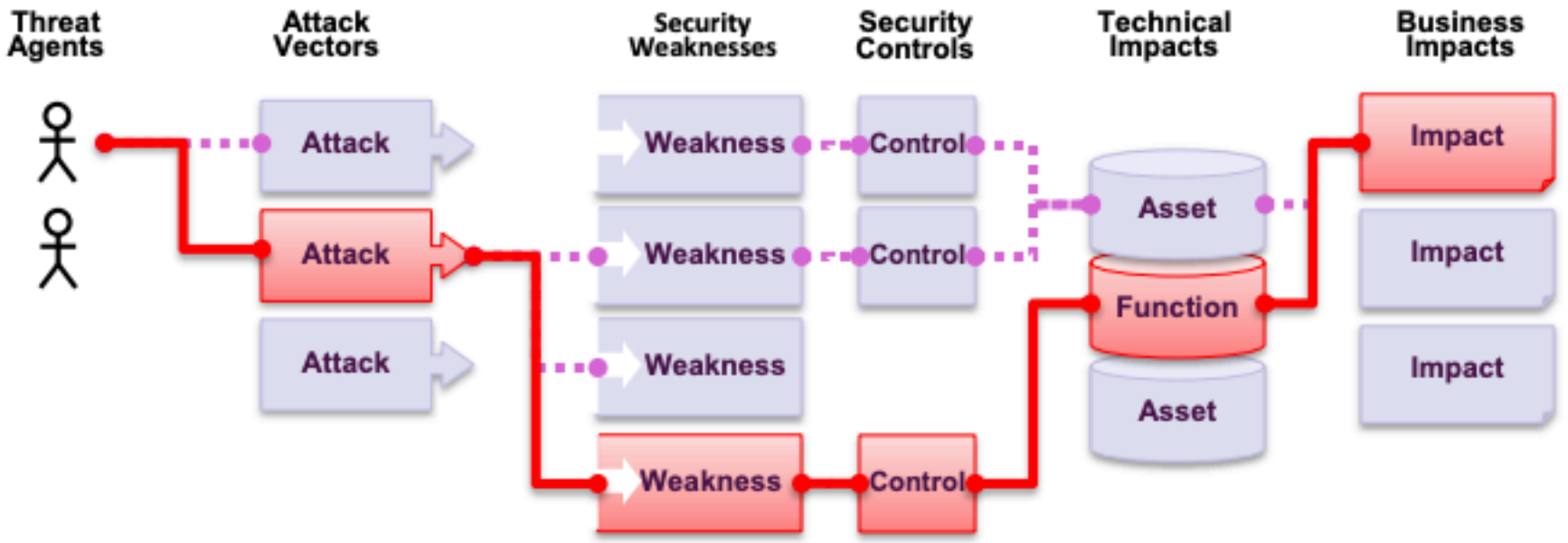
Wikipedia contributors. (2019, September 12). Application security. In Wikipedia, The Free Encyclopedia. Retrieved 23:32, December 29, 2019, from https://en.wikipedia.org/w/index.php?title=Application_security&oldid=915255773

Application Security – Overview

https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

What Are Application Security Risks?

Attackers can potentially use many different paths through your application to do harm to your business or organization. Each of these paths represents a risk that may, or may not, be serious enough to warrant attention.



Sometimes these paths are trivial to find and exploit, and sometimes they are extremely difficult. Similarly, the harm that is caused may be of no consequence, or it may put you out of business. To determine the risk to your organization, you can evaluate the likelihood associated with each threat agent, attack vector, and security weakness and combine it with an estimate of the technical and business impact to your organization. Together, these factors determine your overall risk.

Dynamic Web Site Testing

A dynamic application security testing (DAST) tool is a program which communicates with a web application through the web front-end in order to identify potential security vulnerabilities in the web application and architectural weaknesses.

Wikipedia contributors. (2019, December 2). Dynamic application security testing. In Wikipedia, The Free Encyclopedia. Retrieved 23:31, December 29, 2019, from https://en.wikipedia.org/w/index.php?title=Dynamic_application_security_testing&oldid=928924351

Fortify WebInspect is a DAST tool.

Dynamic Web Site Testing Phases

1. **Baseline/Initial Scans**

   This is the scan that is the initial Web ScanS performed to see the current state.

   NOTE: a comprehensive baseline scan may include many scan versions and is an iterative process until all potential threat paths have been addressed:

   1. Scan without Login Macro
   2. Scan with Login Macro
   3. Scan with Login Macro and Business Workflow Macro

2. Remediation/After Fixes/Code Changes Scan

   This is the scan that is performed at various checkpoints after code changes for the web application have been performed and deployed to testing area.

WebInspect Use Case – DAST **Baseline** Scans

The goals of a DAST baseline scan is to set a starting point for fixing security issues found by a web security test.

This is usually the first phase after you have defined your Application Security Program and Processes using BSIMM and OpenSAMM as a guide or reference.

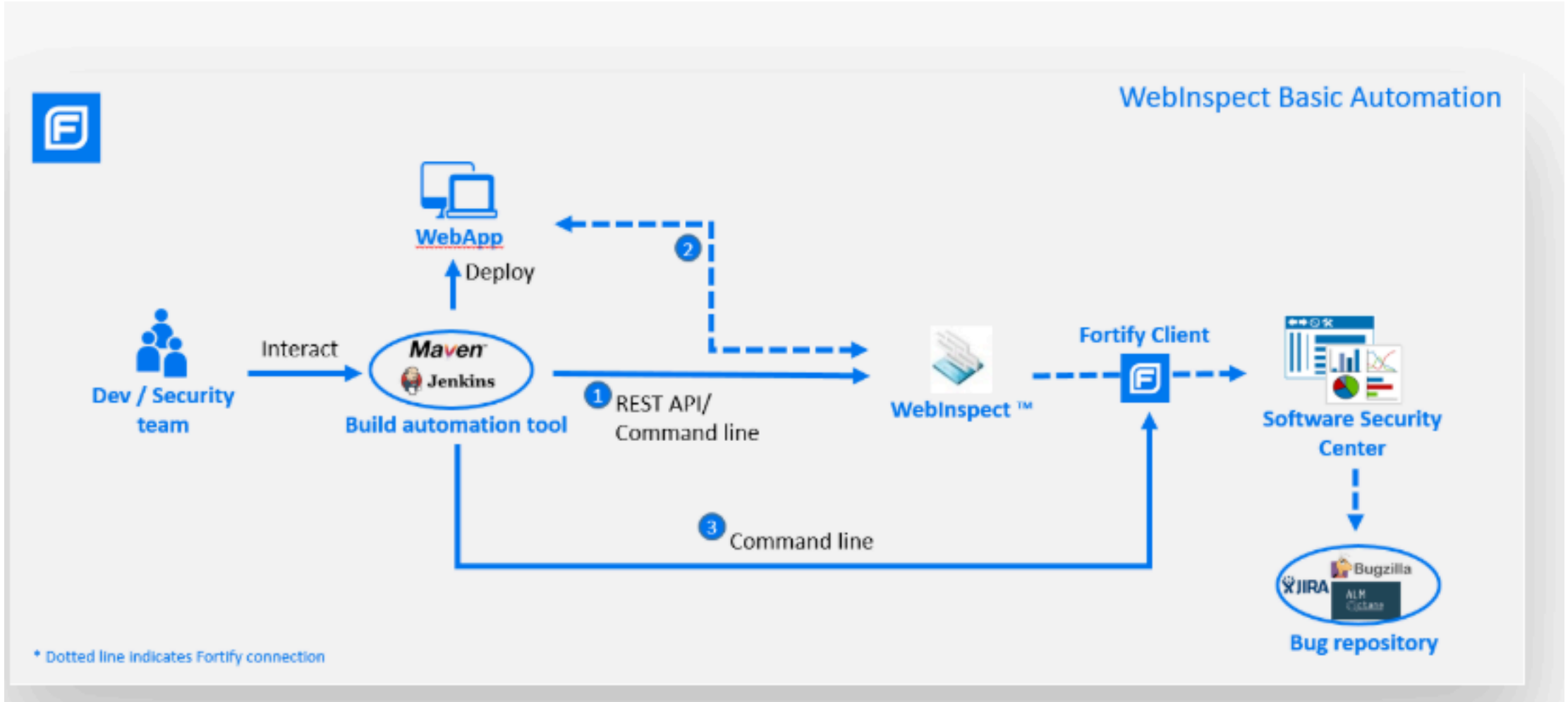
NOTE: This requires that you select the key applications and setup a schedule for the baseline scan.

WebInspect Use Case – DAST **Remediation** Scans

The goals of a DAST remediation scan is to verify that fixes have been made to an application.

This is usually the performed after a review of the baseline scan and assignment of issues via a bug tracking system. This issue should have been fixed and marked in the bug tracking system.

Fortify


WebInspect Basic Automation
WebApp
Deploy
2
Interact
Maven
Jenkins
Dev / Security team
Build automation tool
1
REST API/
Command line
WebInspect ™
Fortify Client
Software Security Center
3
Command line
Bugzilla
JIRA
Bug repository
* Dotted line indicates Fortify connection

Initial Discussion Questions.

Key Discussion Questions:

NOTE: Please provide details on the critical business applications and development processes, based on these questions.

1. Describe the software development lifecycle used and the tools that support this ? List all tools also.
2. How do you perform change management for your critical business applications applications today ?
3. Do you create threat models and data classification for your critical business applications ? How detailed and when is it done ? Do you use or plan to use OpenSAMM, BSIMM or some other security framework?
4. How do you find security issues in your applications today ? What tools or services do you use ?
5. How do you plan to use WebInspect in your current SDLC ? Are you planning to define a new process ?