SYNOPSYS®
Silicon to Software™

Application Security Testing

Alan Dewar

2022-11-22

Application Security Testing

- Introduction
- Motivating Examples
- Government Response
- Vulnerability Classification
- Vulnerability Enumeration
- Attacking the Problem
- Demo

Introduction

My Background

- MSc, Computer Science
- CUUG Board of Directors since 1998
- Synopsys since 2017

Introduction

Synopsys Background

- Founded in 1986
  - Initially silicon only
  - Electronic Design Automation (EDA)
- Acquired Coverity in 2014
  - Static Analysis
  - Software Integrity Group (SIG)

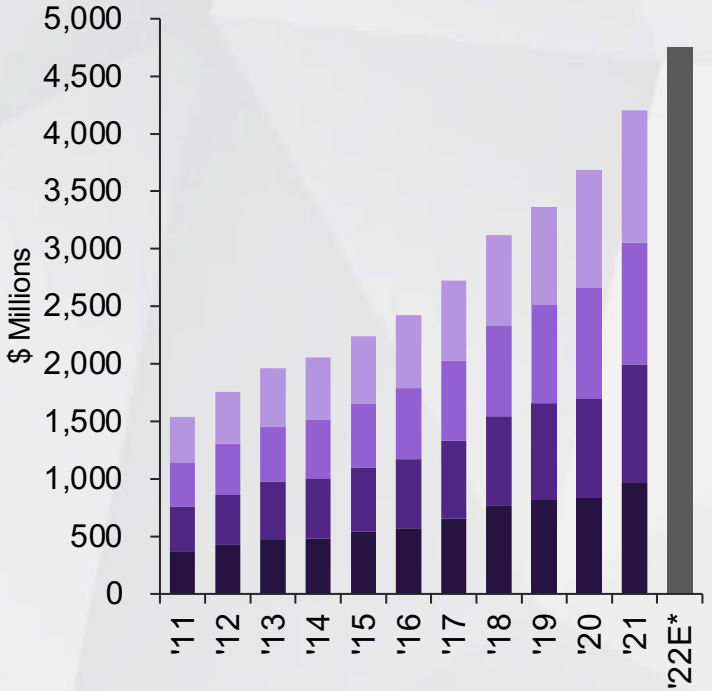
Synopsys Today: From Silicon to Software

FY21 Revenue: ~$4.2B

Employees: 16,361

Patents: 3,449

Offices: 125



Market / Region

#1 electronic design automation tools & services

Broadest IP portfolio and **#1** interface, foundation & physical IP

'Leader' in Gartner's Magic Quadrant for application security testing

Software Integrity Group (SIG) Acquisitions

Mar/2014  Coverity (San Francisco, Calgary, Seattle)

May/2014  Kalistick (France)

Apr/2015  Codenomicon (Finland) 

Jun/2015  Seeker (Israel)

Nov/2015  Protecode (Ottawa)

Dec/2015  Goanna (Australia)

Dec/2016  Cigital/Codiscope (Dulles/Boston)

Dec/2017  Black Duck Software (Boston, Global Offices)

Jan/2020  Tinfoil (Mountain View)

Jun/2021  Code Dx (New York)

Jun/2022  WhiteHat (San Jose)

Synopsys

- CUUG Sponsor since 2018

Motivating Examples

- SolarWinds: 2019-2020 supply chain attacks
  - Trojan introduced into Orion network monitoring software
  - Multiple government agencies breached
- Colonial Pipeline: 2021 ransomware attack
  - Pipeline shut down to contain attack
  - Ransom paid (but partially recovered)

Government Response

US: Executive Order on Improving the Nation’s Cybersecurity

- Remove Barriers to Threat Information Sharing Between Government and the Private Sector
- Modernize and Implement Stronger Cybersecurity Standards in the Federal Government
- Improve Software Supply Chain Security
- Establish a Cybersecurity Safety Review Board
- Create a Standard Playbook for Responding to Cyber Incidents
- Improve Investigative and Remediation Capabilities

Government Response

EU: Cyber Resilience Act

- Currently a proposal
- Bolster cybersecurity rules to ensure more secure hardware and software products
- Create conditions for the development of secure products
- Create conditions allowing users to take cybersecurity into account
- Ensure that manufacturers improve security throughout the whole life cycle
- Ensure a coherent cybersecurity framework
- Enhance the transparency of security properties
- Enable businesses and consumers to use products securely

TLA List*

- BSIMM: Building Security In Maturity Model
- CAPEC: Common Attack Pattern Enumeration and Classification
- CVE: Common Vulnerabilities and Exposures
- CVSS: Common Vulnerability Scoring System
- CWE: Common Weakness Enumeration
- DAST: Dynamic Application Security Testing
- FIRST: Forum of Incident Response and Security Teams
- IAST: Interactive Application Security Testing
- MISRA: Motor Industry Software Reliability Association
- NVD: National Vulnerability Database
- OSVDB: Open Source Vulnerability Database
- OWASP: Open Web Application Security Project
- SARIF: Static Analysis Results Interchange Format
- SAST: Static Application Security Testing
- SBOM: Software Bill Of Materials
- SCA: Software Composition Analysis
- TLA: Three-Letter Acronym

*Technically initialisms, not acronyms

Classification

CVSS: Common Vulnerability Scoring System

- Free and open industry standard for assessing the severity of computer system security vulnerabilities
- Assigns severity scores (0 - 10) to vulnerabilities
- Metrics:
  - Base Metrics for qualities intrinsic to a vulnerability
  - Temporal Metrics for characteristics that evolve over the lifetime of vulnerability
  - Environmental Metrics for vulnerabilities that depend on a particular implementation or environment
- Current version: 3.1
- Maintained by FIRST (Forum of Incident Response and Security Teams)

Classification

CWE: Common Weakness Enumeration

- Community-developed list of software and hardware weakness types that have security ramifications
- CWE Top 25
- Maintained by The MITRE Corporation
- Current version: 4.5

Classification

2022 CWE Top 25

Rank	ID	Name
1	CWE-787	Out-of-bounds Write
2	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
3	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
4	CWE-20	Improper Input Validation
5	CWE-125	Out-of-bounds Read
6	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
7	CWE-416	Use After Free
8	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
9	CWE-352	Cross-Site Request Forgery (CSRF)
10	CWE-434	Unrestricted Upload of File with Dangerous Type
11	CWE-476	NULL Pointer Dereference
12	CWE-502	Deserialization of Untrusted Data
13	CWE-190	Integer Overflow or Wraparound
14	CWE-287	Improper Authentication
15	CWE-798	Use of Hard-coded Credentials

Classification

2022 CWE Top 25 (continued)

Rank	ID	Name
16	CWE-862	Missing Authorization
17	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')
18	CWE-306	Missing Authentication for Critical Function
19	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer
20	CWE-276	Incorrect Default Permissions
21	CWE-918	Server-Side Request Forgery (SSRF)
22	CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
23	CWE-400	Uncontrolled Resource Consumption
24	CWE-611	Improper Restriction of XML External Entity Reference
25	CWE-94	Improper Control of Generation of Code ('Code Injection')

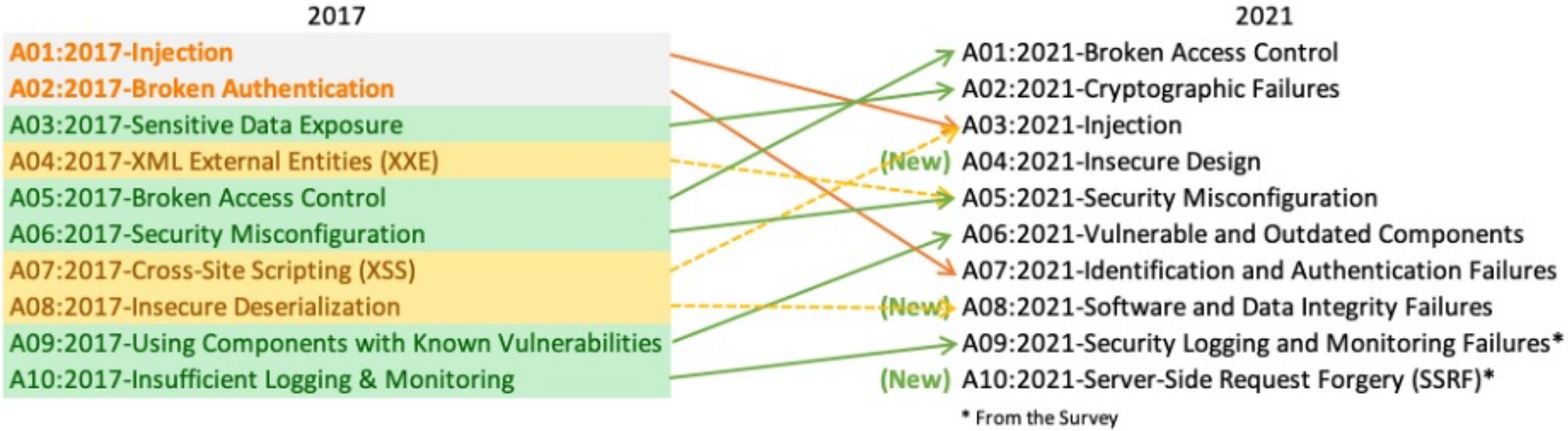
Classification

OWASP: Open Web Application Security Project

- Online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security
- OWASP Top 10
- Led by non-profit OWASP Foundation

Classification

OWASP Top 10



Classification

MISRA: Motor Industry Software Reliability Association

- Guidelines for the software developed for electronic components used in the automotive industry
- C/C++ only
- Latest revision was in 2019

Enumeration

CVE: Common Vulnerabilities and Exposures

- Mission: identify, define, and catalog publicly disclosed cybersecurity vulnerabilities
- Recent examples
  - December 2021: Apache Log4j
    - CVE-2021-44228
  - November 2022: OpenSSL
    - CVE-2022-3602
    - CVE-2022-3786

Enumeration

NVD: National Vulnerability Database

- U.S. government repository of standards based vulnerability management data
- Links to CVEs

Attacking the Problem

Contexts

- Developer
  - Command line
  - IDE
- Build Manager
  - CI/CD: Continuous Integration, Continuous Delivery/Deployment
- Security Consultant
  - One-shot overall assessment

Attacking the Problem

Tradeoffs

- Speed vs. thoroughness
  - Developers need speed
  - Security consultants need thorough results
- False positives vs. false negatives
  - High aggressiveness: false positives (noise)
  - Low aggressiveness: false negatives (missed defects)

Attacking the Problem

Security Practices

- Analyze your code
  - SAST/DAST/IAST/etc.
- Review third-party code
  - SCA: Software Composition Analysis
  - SBOM: Software Bill of Materials

Attacking the Problem

Types of Testing

- SAST: Static Analysis Security Testing
- DAST: Dynamic Analysis Security Testing
  - Fuzzing
- IAST: Interactive Analysis Security Testing
- Penetration testing

Attacking the Problem

SAST: Static Analysis Security Testing

- Examines source code
- White box approach
- No execution
- Abstract interpretation
  - Possible code paths
  - Variable value ranges (e.g., positive/negative/zero, null/non-null)

Attacking the Problem

DAST: Dynamic Application Security Testing

- Tests executing code
- Black box approach
- Fuzzing
  - Range of potentially-invalid inputs

Attacking the Problem

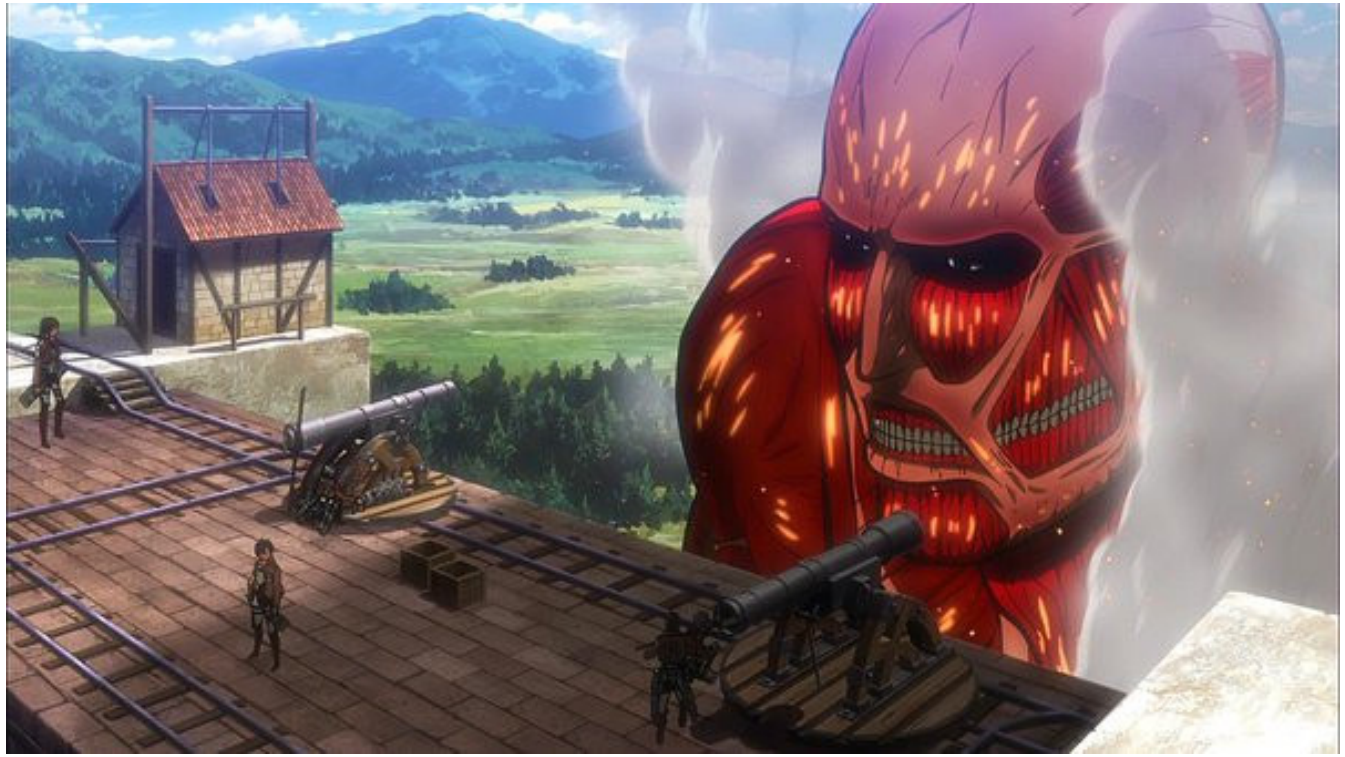
IAST: Interactive Application Security Testing

- Instruments executing code
- Manual or automated testing

Attacking the Problem

Penetration Testing

- Authorized simulated attack
- Same tools as attackers would use



Attacking the Problem

Managing It All

- Intelligent Orchestration
  - Decide what testing to apply
  - Decide aggressiveness levels
- Correlation, De-duplication
  - Combine results from various tools
  - SARIF: Static Analysis Results Interchange Format

Attacking the Problem

Best Practices

- BSIMM: Building Security In Maturity Model
- CSO/CISO: Chief (Information) Security Officer
- DevOps -> DevSecOps
  - Introduce security early in the software development life cycle
- Security Champions
  - Familiar with product under development
  - Focus on security concerns
- Security testing as release condition

Attacking the Problem

Gartner Magic Quadrant

Figure 1: Magic Quadrant for Application Security Testing

Source: Gartner (April 2022)

Demo!

Code Sight IDE Plugin

Demo!

Point and Scan

Questions?

SYNOPSYS®
Silicon to Software™

Thank You