OWASP

Open Web Application
Security Project

Software assurance with OpenSAMM

Jacco van Tuijl

jacco@owasp.org

Speaker BIO

- Hack in the Box - Core Crew NL
- Red team tester / Pen tester / security consultant for 7 year
- Software engineering background
- Software security architect @ RES Software

Why a software assurance program?

- Preventing security issues from occurring
- Finding security issues in early stage of development is much cheaper than after release
- Less vulnerabilities in software releases
- Better prepared for when security issues occur
- Keeps your product out of the “Hall of shame”
- Customer demand

Traditional security testing

- A team of developers can make more vulnerabilities in a day then a tester can find in a day

- A tester can find more vulnerabilities in a day then that a team of developers can fix in a month

- Results in a ever expanding list of known vulnerabilities

OpenSAMM

- OpenSAMM v1.0 released 2009
- OpenSAMM v1.1 (2016 = current)

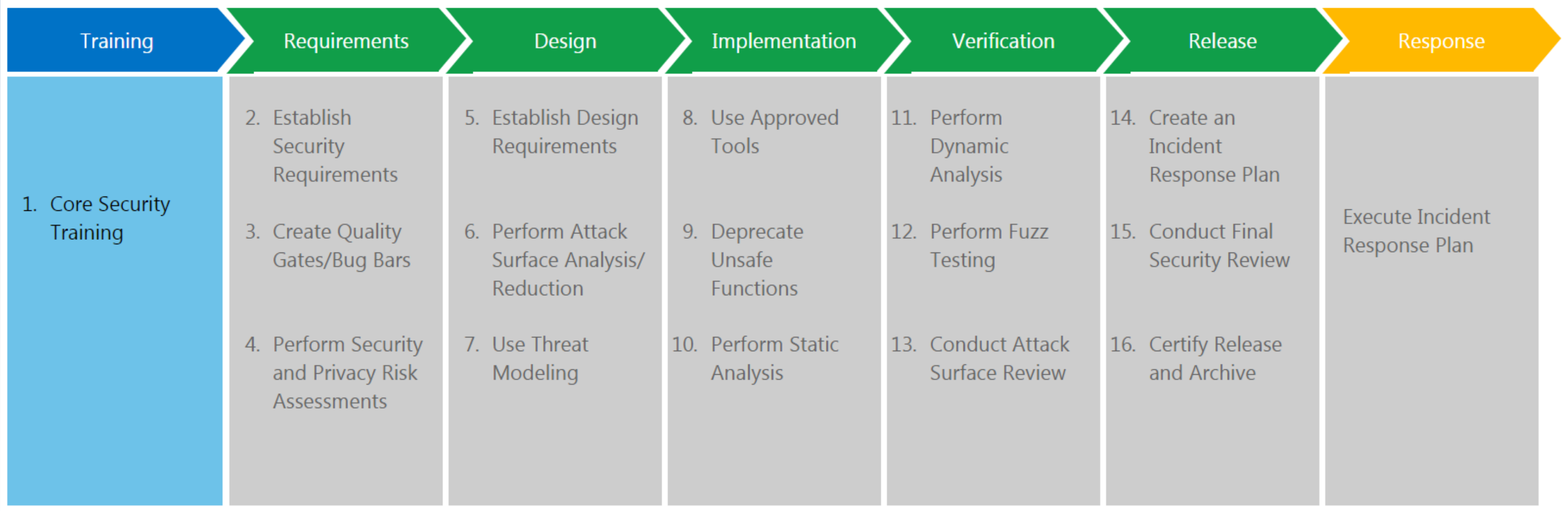
Work in progress:

- OpenSAMM v1.2 & v2
  - More tools and materials
  - Implementation guidance dev ops & agile
  - Privacy ?

Other methodologies

- OWASP CLASP - obsolete
- BSIMM- Proprietary Cigital fork of OpenSAMM alfa
- MS SDL
- SAMATE - Software Assurance Metrics And Tool Evaluation (NIST)
- SSE-CMM
- Grip op SSD – CIP (Dutch government requirement)

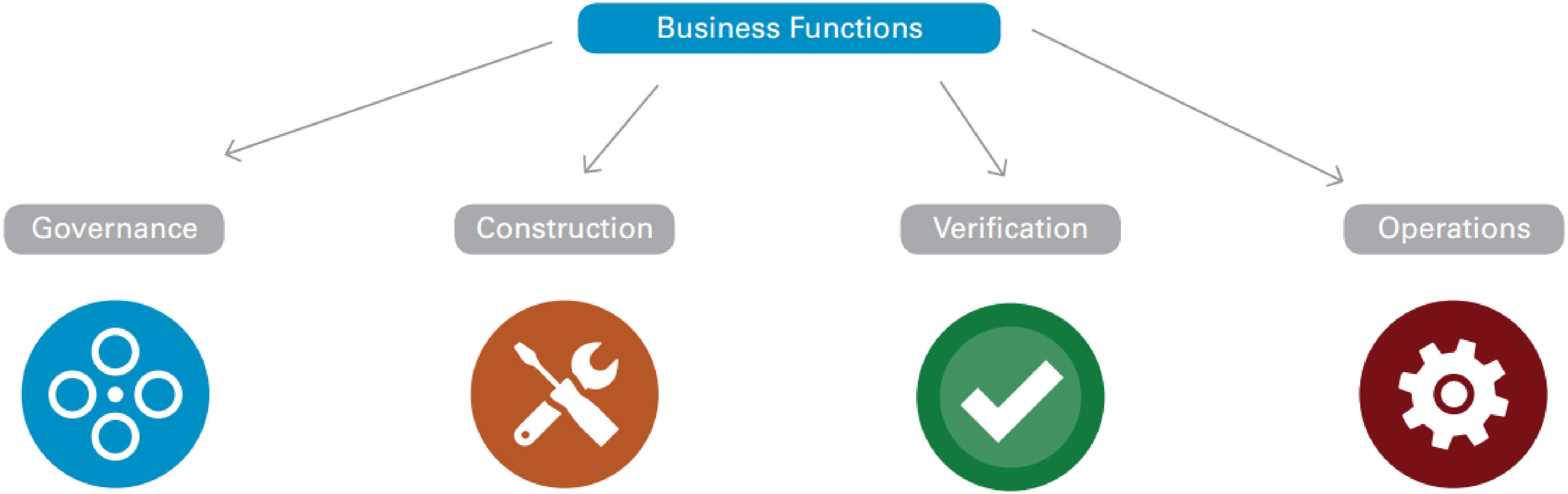
Microsoft SDL



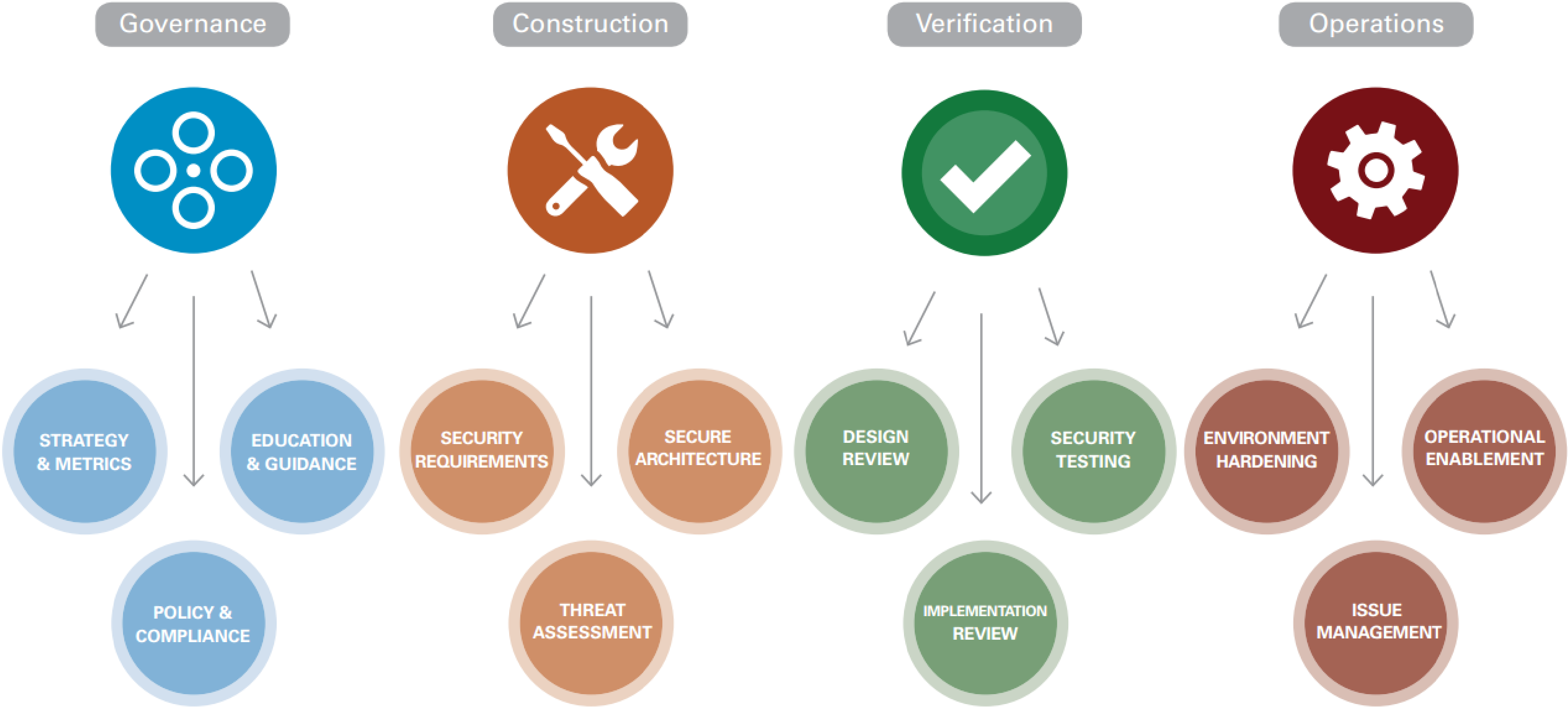
Training	Requirements	Design	Implementation	Verification	Release	Response
1. Core Security Training	2. Establish Security Requirements 3. Create Quality Gates/Bug Bars 4. Perform Security and Privacy Risk Assessments	5. Establish Design Requirements 6. Perform Attack Surface Analysis/ Reduction 7. Use Threat Modeling	8. Use Approved Tools 9. Deprecate Unsafe Functions 10. Perform Static Analysis	11. Perform Dynamic Analysis 12. Perform Fuzz Testing 13. Conduct Attack Surface Review	14. Create an Incident Response Plan 15. Conduct Final Security Review 16. Certify Release and Archive	Execute Incident Response Plan

MS SDL	OpenSAMM
1. Core Security Training	Education & guidance
2. Establish Security Requirements	Security requirements
3. Create Quality Gates/Bug Bars	Code review and security test baseline
4. Perform Security and Privacy Risk Assessments	Threat Assessment
5. Establish Design Requirements	Security Requirements
6. Perform Attack Surface Analysis/ Reduction	Threat assessment (ML1) & Design review (One of the security practices)
7. Use Threat modeling	Threat assessment (ML1)
8. Use Approved Tools	Secure architecture (ML1)
9. Deprecate Unsafe Functions	Code review
10. Perform Static Analysis	Code review
11. Perform Dynamic Analysis	Security testing
12. Perform Fuzz Testing	Security testing
13. Conduct Attack Surface Review	Design review & security testing
14. Create an Incident Response Plan	Vulnerability management
15. Conduct Final Security Review	Verification
16. Certify Release and Archive	Code signing
17. Execute Incident Response Plan	Incident response plan & team and vulnerability management

SAMM Business functions

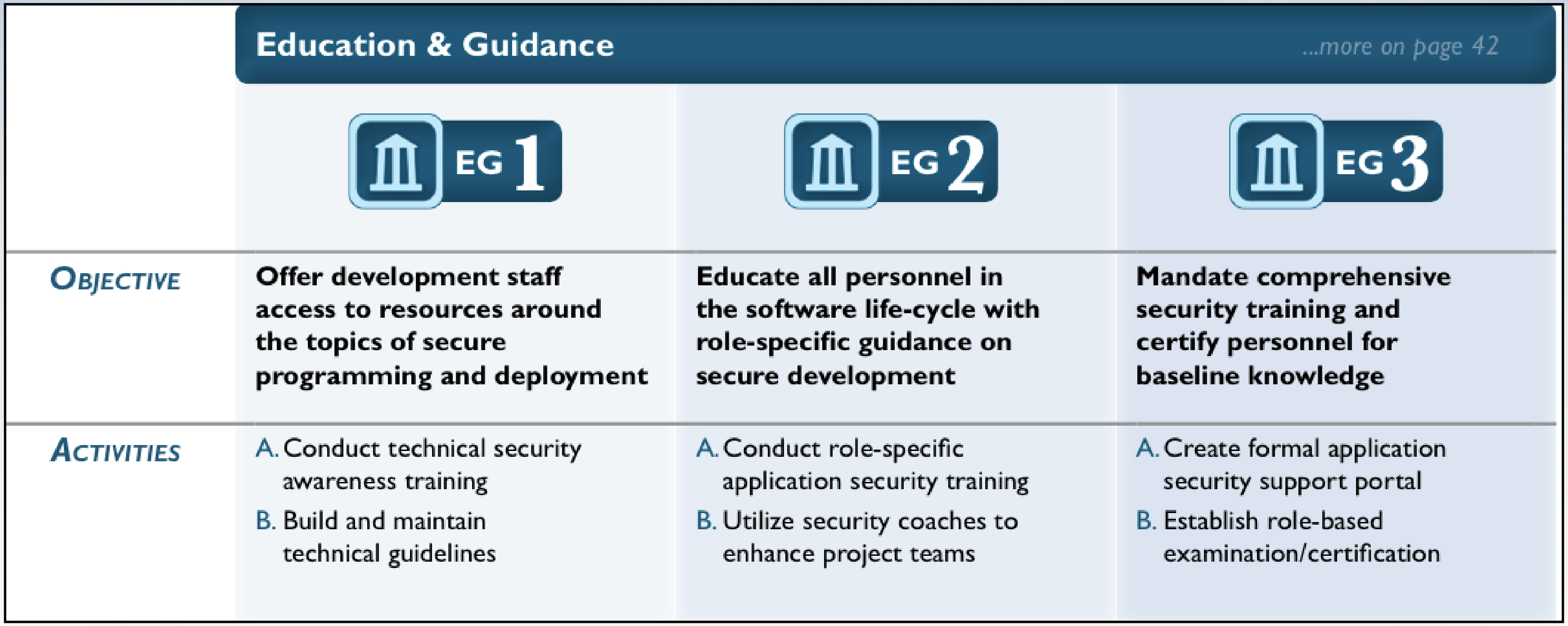


12 focus areas



OWASP
Open Web Application Security Project

3 maturity levels for each focus area



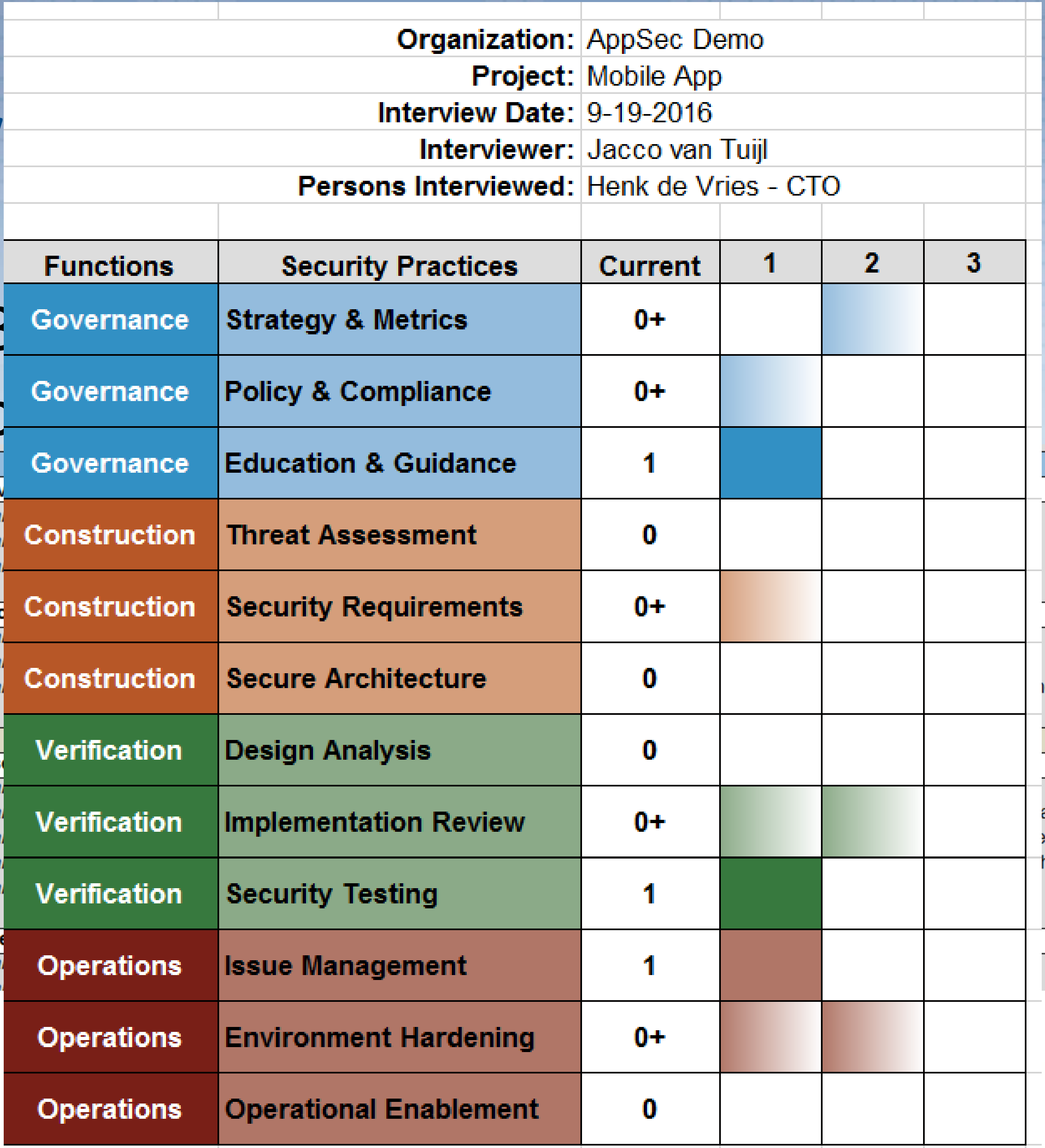
Education & Guidance *...more on page 42*

	EG 1	EG 2	EG 3
Objective	**Offer development staff access to resources around the topics of secure programming and deployment**	**Educate all personnel in the software life-cycle with role-specific guidance on secure development**	**Mandate comprehensive security training and certify personnel for baseline knowledge**
Activities	A. Conduct technical security awareness training B. Build and maintain technical guidelines	A. Conduct role-specific application security training B. Utilize security coaches to enhance project teams	A. Create formal application security support portal B. Establish role-based examination/certification

OWASP
Open Web Application
Security Project

W

- B lready b



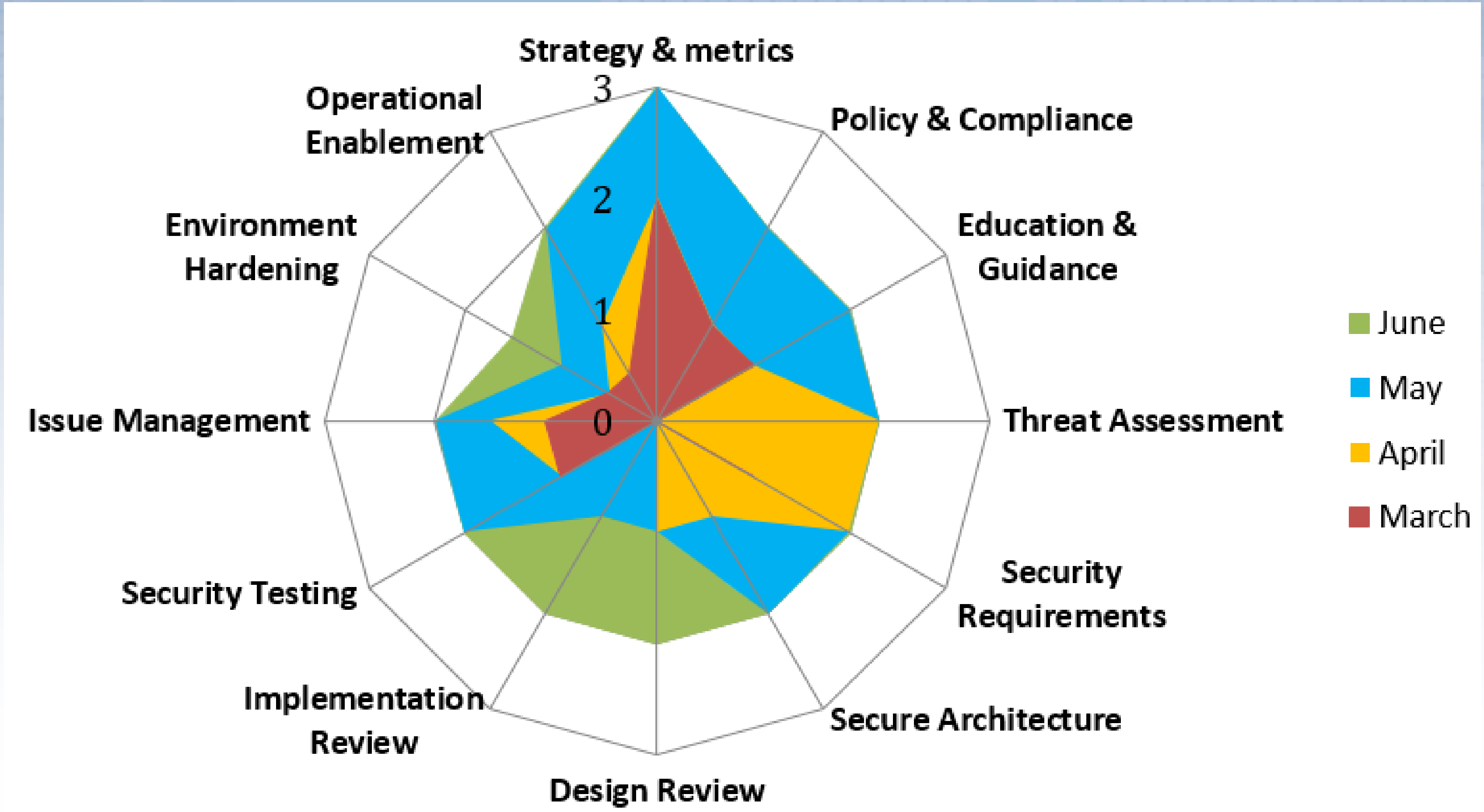
Organization:	AppSec Demo
Project:	Mobile App
Interview Date:	9-19-2016
Interviewer:	Jacco van Tuijl
Persons Interviewed:	Henk de Vries - CTO

Functions	Security Practices	Current	1	2	3
Governance	Strategy & Metrics	0+		▨	
Governance	Policy & Compliance	0+	▨		
Governance	Education & Guidance	1	■		
Construction	Threat Assessment	0			
Construction	Security Requirements	0+	▨		
Construction	Secure Architecture	0			
Verification	Design Analysis	0			
Verification	Implementation Review	0+	▨	▨	
Verification	Security Testing	1	■		
Operations	Issue Management	1	■		
Operations	Environment Hardening	0+	▨	▨	
Operations	Operational Enablement	0			

			Yes/No	Rating
EG1	Have dev		Yes	1
	Guida			
	Guida			
	Guida			
	Does eac		Yes	
	Guida			
	Guida			
	Guida	ent.		
EG2	Are thos		No	
	Guida	agement,		
	Guida	ecurity test		
	Guida	hniques.		
	Guida			
	Guida			
	Are stake		No	
	Guida			
	Guida			

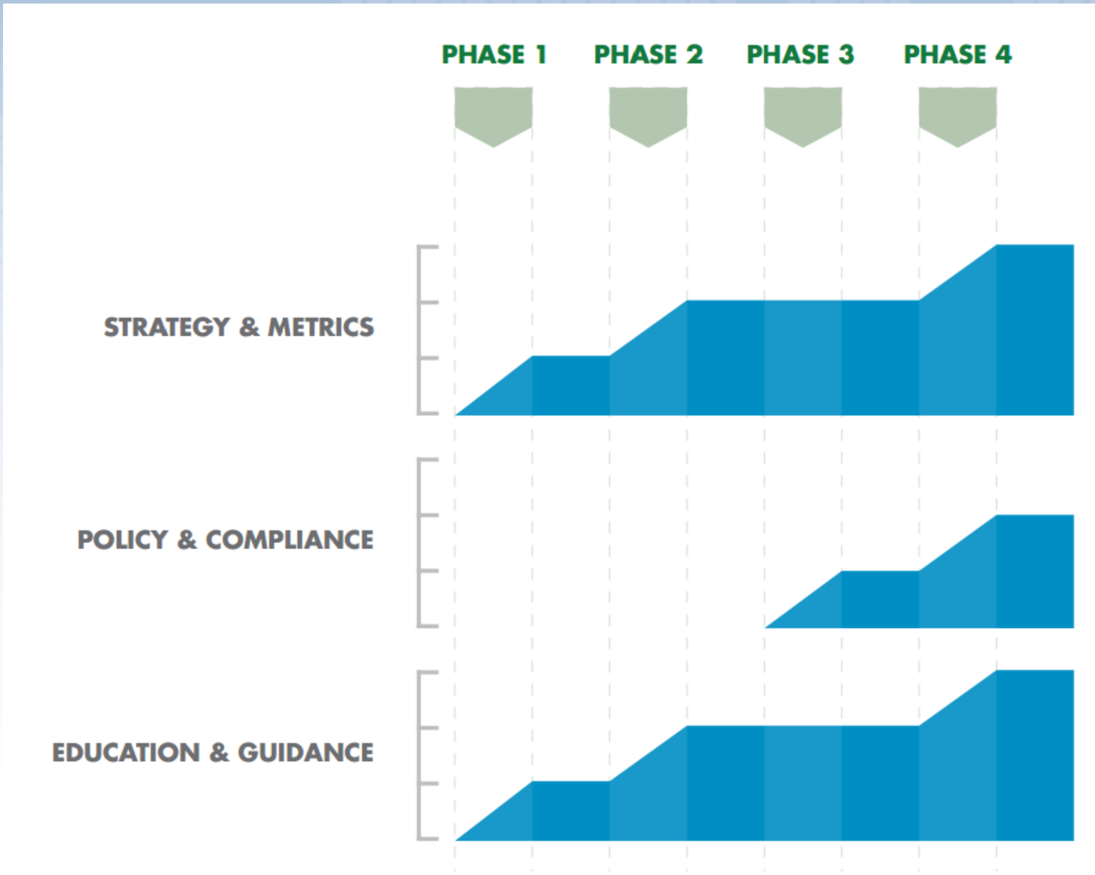
OWASP
Open Web Application Security Project

Ready made roadmaps



OWASP
Open Web Application Security Project

Roadmap


PHASE 1
PHASE 2
PHASE 3
PHASE 4
STRATEGY & METRICS
POLICY & COMPLIANCE
EDUCATION & GUIDANCE

Governance

Strategy & Metrics

- Baseline assessment
- SSDLC Roadmap
- Application risk profile
- Register security spend

Education & Guidance

Application Risk profile

- Classify each Application based on financial impact of worst-case scenario
  - Critical: the end of the organization
  - High: big losses
  - Medium: medium losses
  - Low: almost no impact

Quality Gates based on risk: education, compliance, design review, implementation review and security test

Governance

Strategy & Metrics

Policy & Compliance

- Identify external compliance drivers
- Monitor changes
- Checklist and audit
- Release gates

Identify compliancy, regulations and standards

- Law & Regulation
  - US (SOx, HIPAA, Technology Management Reform Act, Security Act)
  - EU (ECHR)
  - International
  - Canada (PIPEDA)
- Contracts & licenses
  - Customer contracts / EULA / bewerkers overeenkomst
  - Partner contracts
  - 3th party components
  - Suppliers contracts
- Company goals and values

- Industry standards
  - PCI-DSS
  - FIPS
  - ISO 27001, ISO 27035
  - OpenSAMM, MS SDL, BSIMM
  - CIP – Grip op SSD
  - Common Criteria for Information Technology Security Evaluation
  - OWASP Application Security Verification Standard
  - CMMI
  - OWASP top 10
  - SANS top 20

Governance

Strategy & Metrics

Policy & Compliance

Education & Guidance

- High-over security training
- Role-based training
- Role-based examination & certification

Maturity level 1 : High-over training

SSDLC & Security Awareness

- Microsoft Security Development Lifecycle Core Training classes
  - Introduction to Security Development Lifecycle
  - Basics of Secure Design, Development & Test
  - Introduction to Threat Modeling
  - Privacy in Software Development
- OWASP TOP 10

Maturity level 2: Role specific training

Role	Training and/or workshop
Architect	Security principles & threat modelling
Developer	Secure programming
Tester	Security testing
Requirements Engineer	Abuse-cases & Security requirements

How & where do we get security requirements ?

- Customer agreements
- Compliance / industry standards
- Access control matrix
- Misuse-cases / abuser stories
- Threat model
- Security testing
- Security practices

Access control matrix

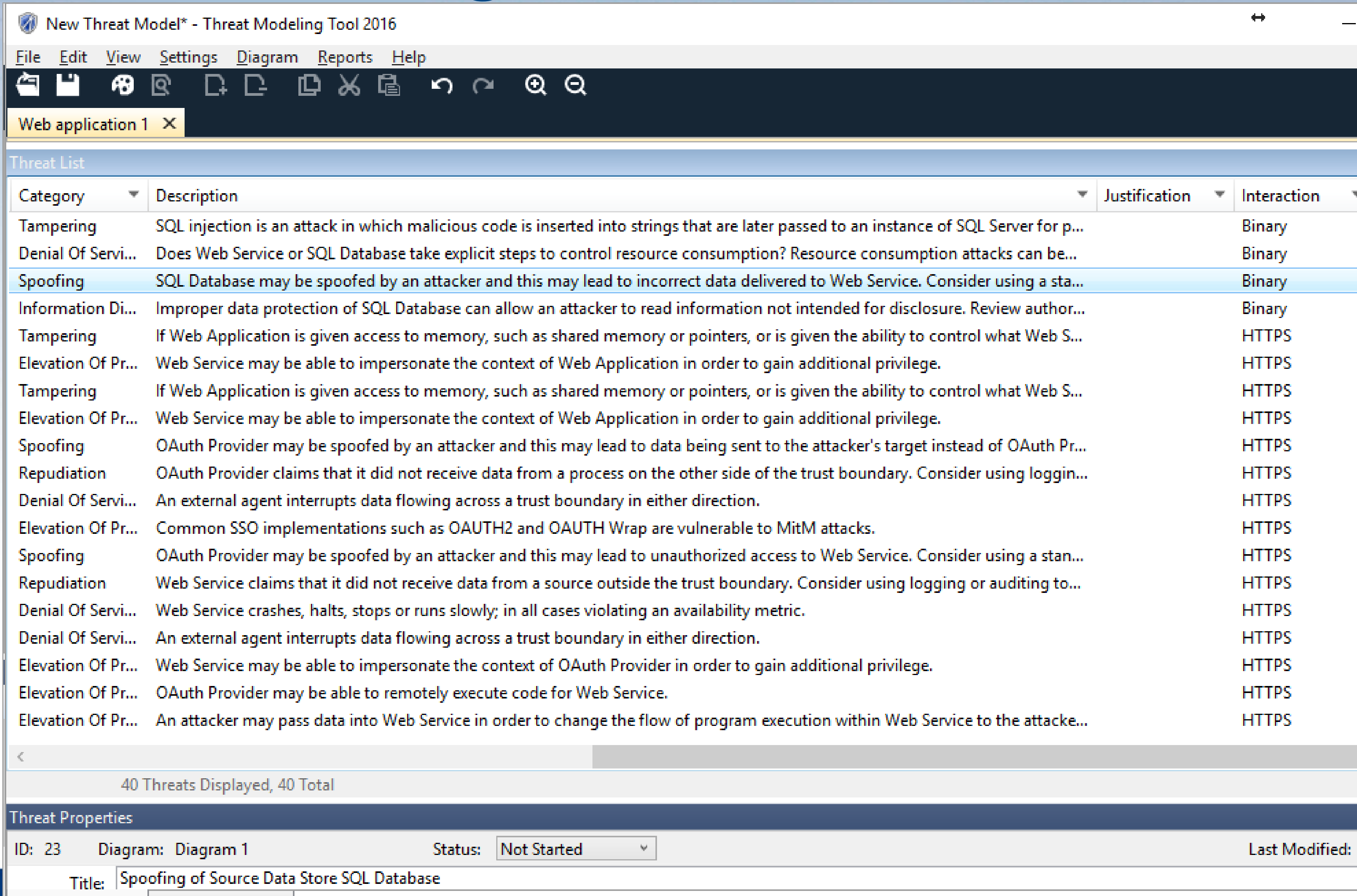
Feature: Service store runbook	Create	Modify	Execute	Read	Delete
Unauthenicated users	No	No	No	No	No
Authenticated user	No	No	Yes	Yes	No
Administrators	Yes	Dynamic	Dynamic	Dynamic	Dynamic

Threat modeling

- Microsoft Threat Modeling Tool 2016
  - **S**poofing
  - **T**ampering
  - **R**epudiation
  - **I**nformation disclosure
  - **D**enial of service
  - **E**levation of privilege

Threat Assessment

Threat modeling



Category	Description	Justification	Interaction
Tampering	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for p...		Binary
Denial Of Servi...	Does Web Service or SQL Database take explicit steps to control resource consumption? Resource consumption attacks can be...		Binary
Spoofing	SQL Database may be spoofed by an attacker and this may lead to incorrect data delivered to Web Service. Consider using a sta...		Binary
Information Di...	Improper data protection of SQL Database can allow an attacker to read information not intended for disclosure. Review author...		Binary
Tampering	If Web Application is given access to memory, such as shared memory or pointers, or is given the ability to control what Web S...		HTTPS
Elevation Of Pr...	Web Service may be able to impersonate the context of Web Application in order to gain additional privilege.		HTTPS
Tampering	If Web Application is given access to memory, such as shared memory or pointers, or is given the ability to control what Web S...		HTTPS
Elevation Of Pr...	Web Service may be able to impersonate the context of Web Application in order to gain additional privilege.		HTTPS
Spoofing	OAuth Provider may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of OAuth Pr...		HTTPS
Repudiation	OAuth Provider claims that it did not receive data from a process on the other side of the trust boundary. Consider using loggin...		HTTPS
Denial Of Servi...	An external agent interrupts data flowing across a trust boundary in either direction.		HTTPS
Elevation Of Pr...	Common SSO implementations such as OAUTH2 and OAUTH Wrap are vulnerable to MitM attacks.		HTTPS
Spoofing	OAuth Provider may be spoofed by an attacker and this may lead to unauthorized access to Web Service. Consider using a stan...		HTTPS
Repudiation	Web Service claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to...		HTTPS
Denial Of Servi...	Web Service crashes, halts, stops or runs slowly; in all cases violating an availability metric.		HTTPS
Denial Of Servi...	An external agent interrupts data flowing across a trust boundary in either direction.		HTTPS
Elevation Of Pr...	Web Service may be able to impersonate the context of OAuth Provider in order to gain additional privilege.		HTTPS
Elevation Of Pr...	OAuth Provider may be able to remotely execute code for Web Service.		HTTPS
Elevation Of Pr...	An attacker may pass data into Web Service in order to change the flow of program execution within Web Service to the attacke...		HTTPS

40 Threats Displayed, 40 Total

Threat Properties

ID: 23 Diagram: Diagram 1 Status: Not Started Last Modified:

Title: Spoofing of Source Data Store SQL Database

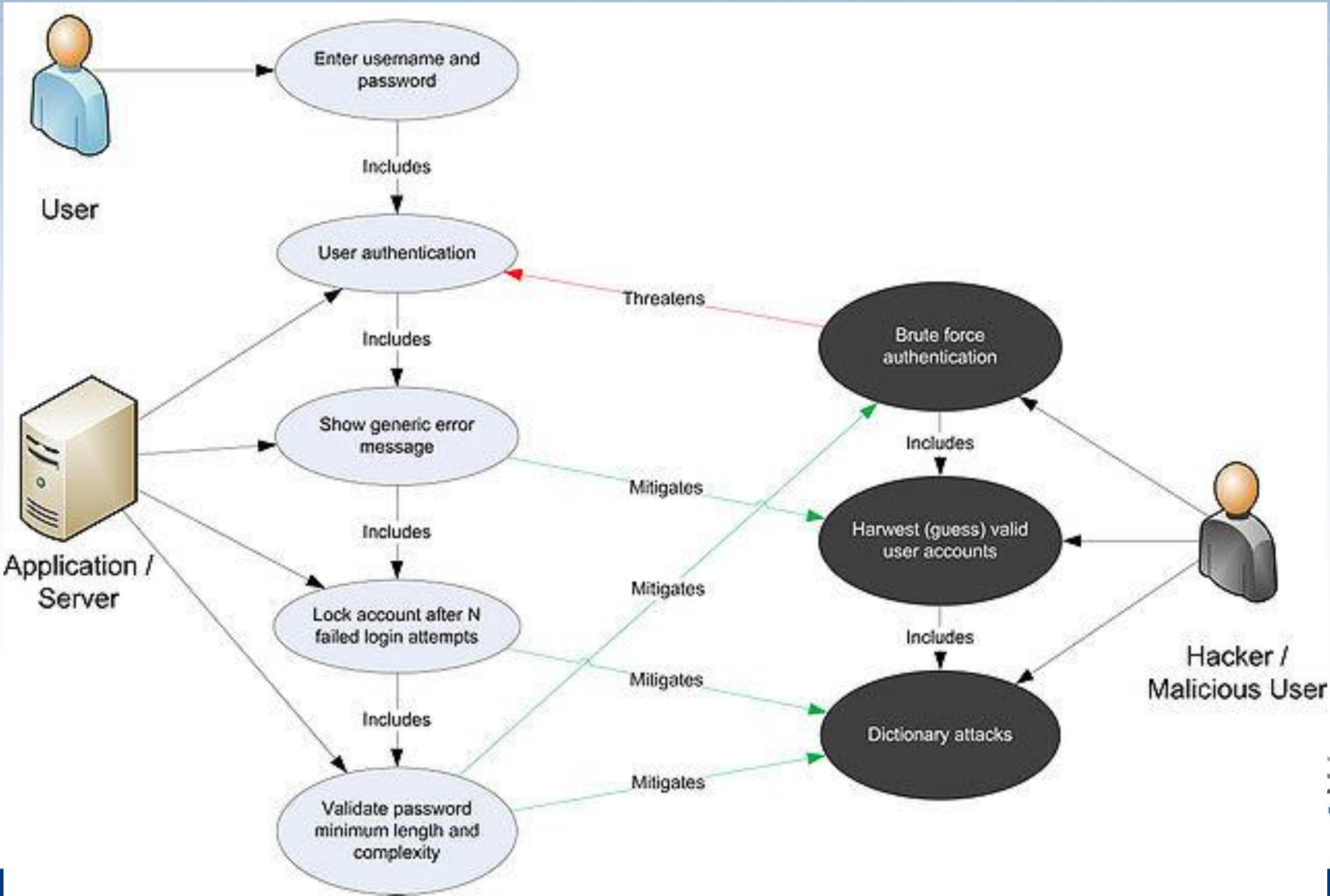
Threat modeling

- Classify control priority : High, medium or low

Quality gate example:

- All high risk controls in high or critical risk applications should be code reviewed.
- Existence of all controls in high or critical risk applications should be validated.
- The working of all medium and high risk controls should be tested.

Misuse and Abuse-cases


User
Enter username and password
Includes
User authentication
Threatens
Brute force authentication
Includes
Show generic error message
Application / Server
Mitigates
Includes
Harvest (guess) valid user accounts
Includes
Mitigates
Lock account after N failed login attempts
Includes
Hacker / Malicious User
Mitigates
Includes
Dictionary attacks
Mitigates
Validate password minimum length and complexity

Construction

Security Requirements

Threat Assessment

Secure Architecture

- Review architecture for security principles
- List of recommended technologies
- Validate usage of recommended technologies

OWASP
Open Web Application Security Project

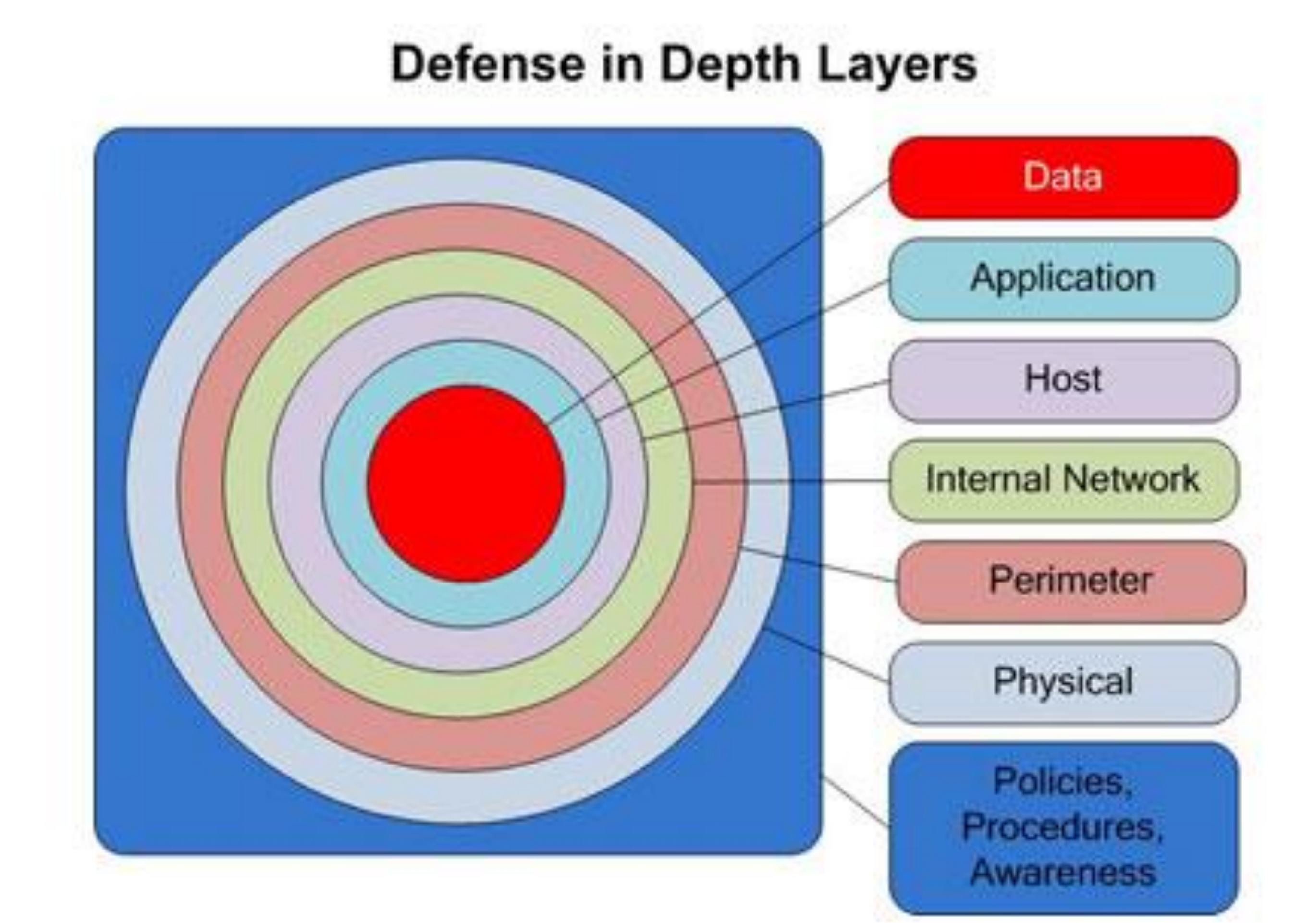
Secure Architecture

Security principles

- **Attack surface reduction**
- **Defense in depth**
- **Least privilege**
- **Secure defaults**
- Securing the weakest link
- Simplicity in design
- Fail securely
- Avoid security by obscurity
- Detect intrusions and log attacks
- Don’t trust infrastructure/services/people
- Input Validation & output encoding
- Avoid single points of failure
- Data in transit & rest protection
- Data loss prevention
- Audit trail
- Promote Privacy
- Never assume that your secrets are safe
- Complete Mediation
- Psychological acceptability (security VS usability)

Secure Architecture

Defense in depth



OWASP
Open Web Application Security Project

Defense in depth examples

- WAF + Urlscan + Input validation + Parameterized queries + data at rest encryption + output encoding
- Network firewall + IDS + Host based firewall
- Email antivirus and spam filter + strip possible harmful file formats + Host based anti-virus
- HTTPS over IPSEC over a private network

Least privilege Windows processes

1. **Local Service (best)**
2. **Network Service**
3. **Unique user account**
4. **Local System**
5. **Local administrator account**
6. **Domain administrator account (worst)**

Verification

Design Review

- Identify software attack surface
- Analyze design against security requirements
- Release gates

Security Testing

OWASP
Open Web Application
Security Project

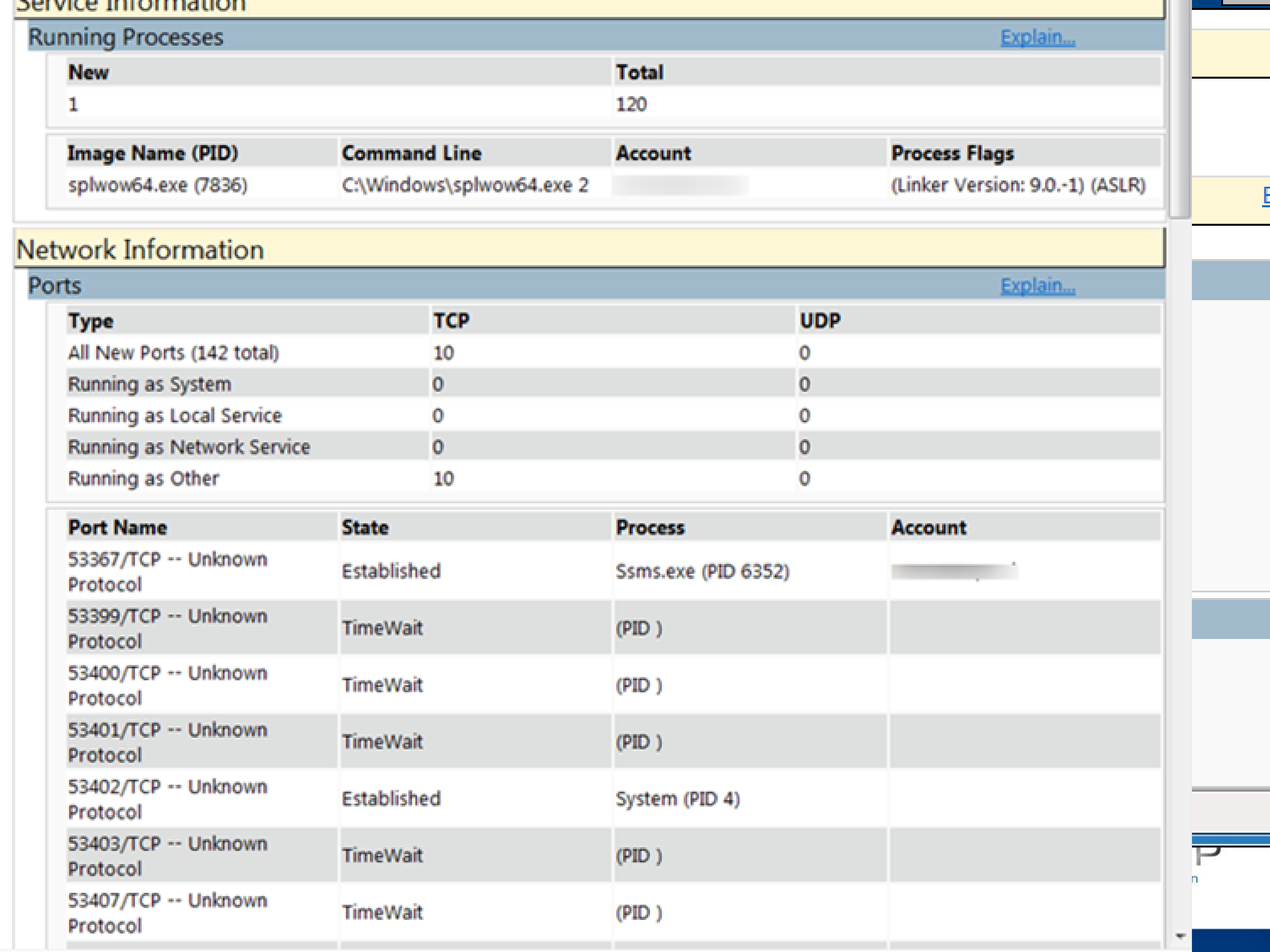
Attack surface analysis

- Look at all of your entry points: Channels, Methods and data
  - Network i/o
  - File i/o
  - Process i/o
- Rank them
  - Authenticated vs Anonymous
  - Administrator only vs regular user
  - Network vs local
  - UDP vs TCP

Design Review

Also look at sub-features

- File formats
  - Image : JPG, FLA, BMP, PNG or GIF
  - Data : csv, excel or SQL
- HTTP verbs
  - GET, POST, PUT and DELETE
- SMTP
  - Helo, EHLO, MAIL,RCPT, VRFY and EXPN
- HTTPS
  - SSLv1, SSLv2, SSLv3 , TLS1.0, TLS1.1 and TLS1.2



Service Information

Running Processes

New	Total
1	120

Image Name (PID)	Command Line	Account	Process Flags
splwow64.exe (7836)	C:\Windows\splwow64.exe 2	[redacted]	(Linker Version: 9.0.-1) (ASLR)

Network Information

Ports

Type	TCP	UDP
All New Ports (142 total)	10	0
Running as System	0	0
Running as Local Service	0	0
Running as Network Service	0	0
Running as Other	10	0

Port Name	State	Process	Account
53367/TCP -- Unknown Protocol	Established	Ssms.exe (PID 6352)	[redacted]
53399/TCP -- Unknown Protocol	TimeWait	(PID)	
53400/TCP -- Unknown Protocol	TimeWait	(PID)	
53401/TCP -- Unknown Protocol	TimeWait	(PID)	
53402/TCP -- Unknown Protocol	Established	System (PID 4)	
53403/TCP -- Unknown Protocol	TimeWait	(PID)	
53407/TCP -- Unknown Protocol	TimeWait	(PID)	

Attack surface reduction examples

- Windows
  - Authenticated RPC
  - Firewall on by default
- SQL Server
  - Xp_cmdshell off by default
  - CLR and COM off by default
- IIS
  - Off by default
  - Static files by default
- Visual Studio
  - Web service listen on localhost only
  - SQL Server Express listen on localhost only

It is not just about turning stuff off

Higher Attack Surface	Lower Attack Surface
Execute by default	Off by default
Open Socket	Closed socket
UDP	TCP
Anonymous access	Authenticated access
Admin access	User access
Internet access	Local subnet access
System	Not system
Uniform defaults	User-chosen settings
Large code	Small code
Weak/flexible ACLs	Strong/strict ACLs

Verification

Design Review

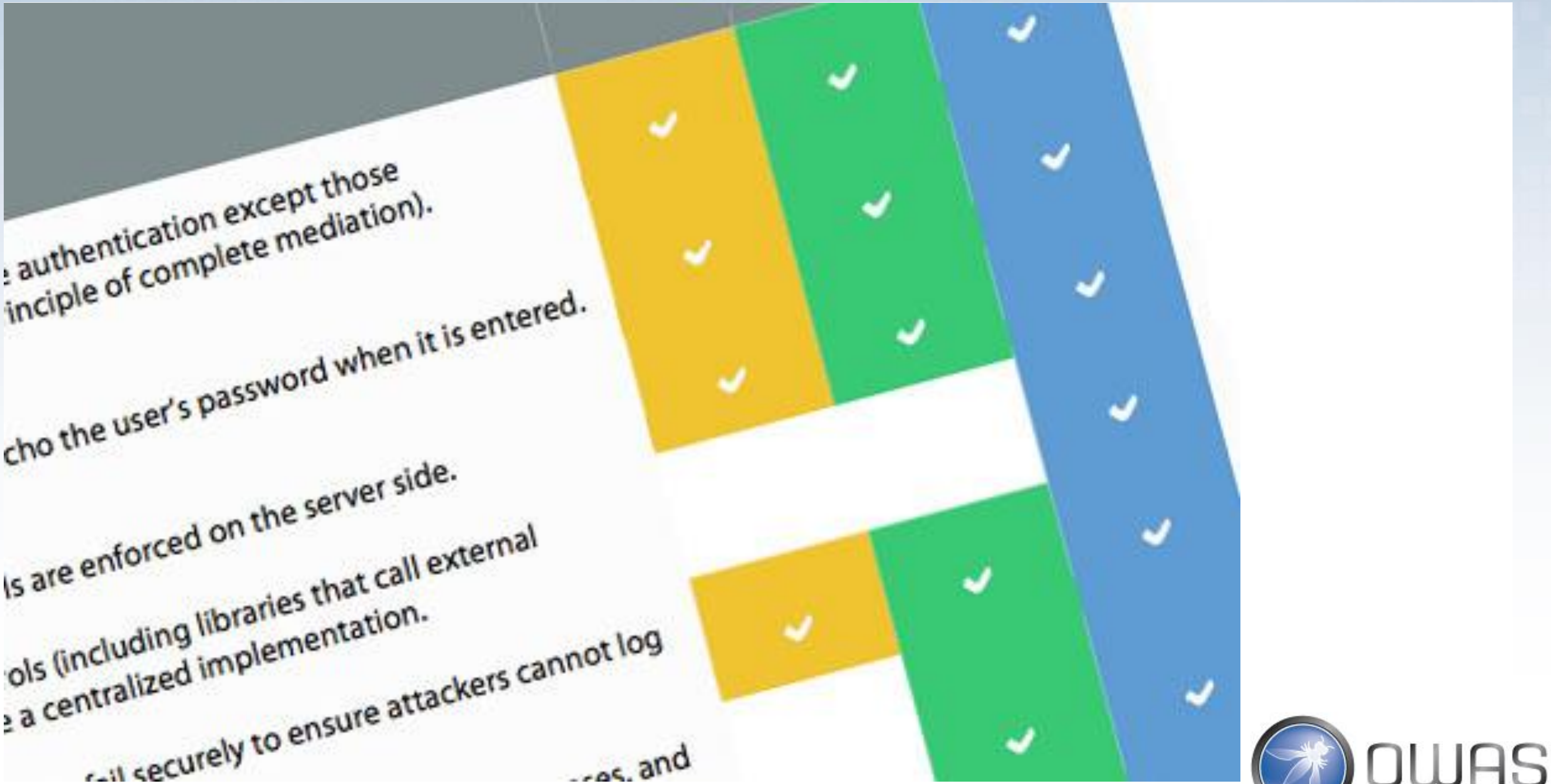
Code Review

- Checklist
- Review of high-risk code
- Automated code analysis
- Derive test cases from security requirements
- Conduct penetration testing
- Automated security testing
- Release gates for security testing

OWASP
Open Web Application Security Project

OWASP - Application Security verification Standard

- Provides 3 levels of application verification



e authentication except those
inciple of complete mediation).

cho the user's password when it is entered.

ls are enforced on the server side.

ols (including libraries that call external
e a centralized implementation.

fail securely to ensure attackers cannot log
ces, and

OWASP
Open Web Application
Security Project

Deployment

Operational Enablement

- Document procedures for typical application security alerts
- Change management
- Operational security guide
- Secure Operational environment specifications
- Install security updates
- Create security response team
- Incident response process
- Responsible disclosure
- Root cause analysis for incidents

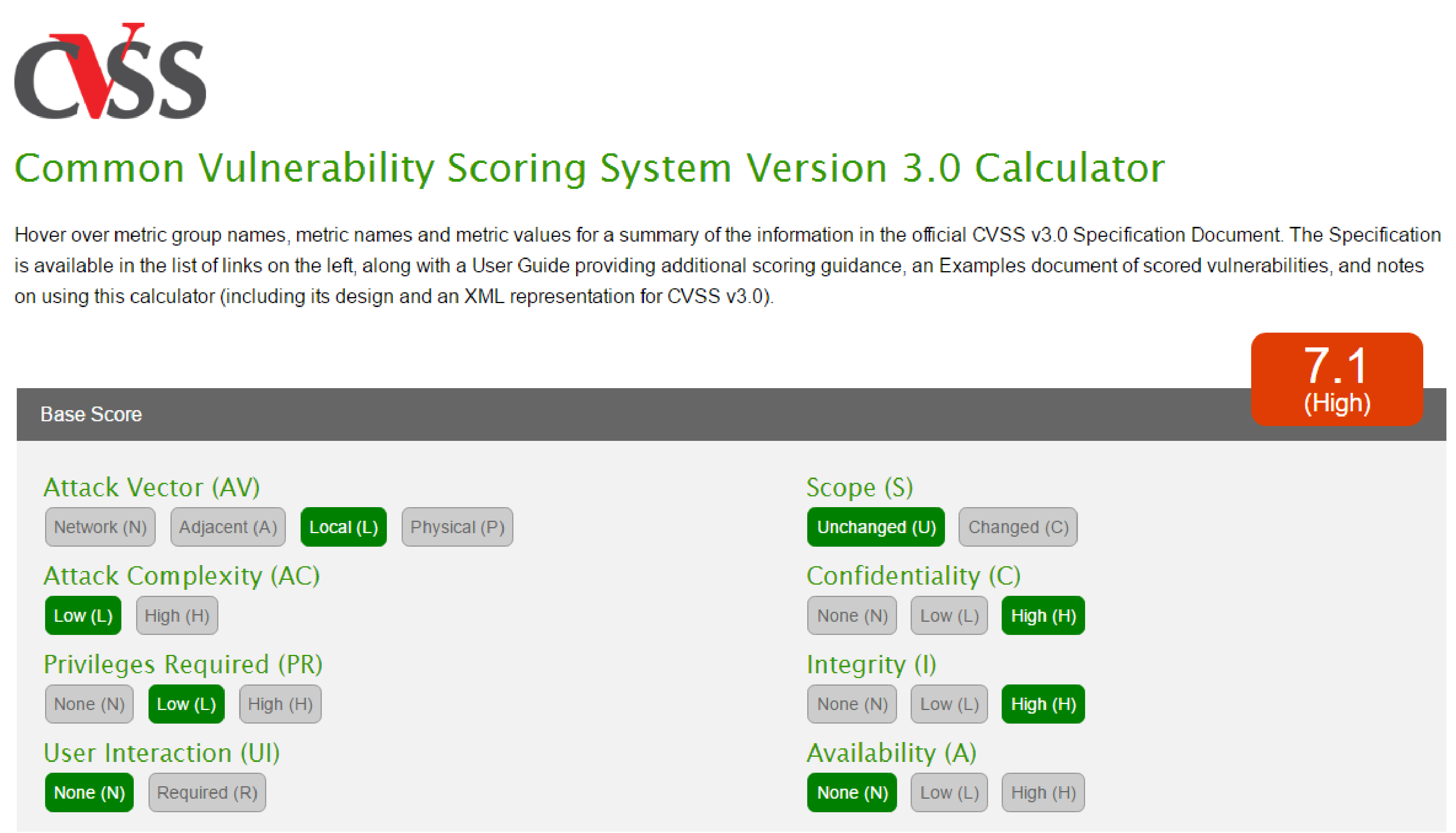
OWASP
Open Web Application Security Project

Responsible disclosure

- Responsible disclosure policy
- Facilitate security researchers that want to report security issues (without service contract or legal consequences)
- Prioritize issues
- Security bulletin - mailing list
  - application specific
  - no advertisements
- 60 day max fix time

Responsible disclosure ≠ Full disclosure

Prioritize issues



Common Vulnerability Scoring System Version 3.0 Calculator

Hover over metric group names, metric names and metric values for a summary of the information in the official CVSS v3.0 Specification Document. The Specification is available in the list of links on the left, along with a User Guide providing additional scoring guidance, an Examples document of scored vulnerabilities, and notes on using this calculator (including its design and an XML representation for CVSS v3.0).

7.1
(High)

Base Score

Attack Vector (AV)
Network (N) Adjacent (A) [✓ Local (L)] Physical (P)

Attack Complexity (AC)
[✓ Low (L)] High (H)

Privileges Required (PR)
None (N) [✓ Low (L)] High (H)

User Interaction (UI)
[✓ None (N)] Required (R)

Scope (S)
[✓ Unchanged (U)] Changed (C)

Confidentiality (C)
None (N) Low (L) [✓ High (H)]

Integrity (I)
None (N) Low (L) [✓ High (H)]

Availability (A)
[✓ None (N)] Low (L) High (H)

Responsible disclosure policy

- Clear rules
  - What is allowed and what not
  - What can be expected from the organization
- Bug Bounty program
  - Big reward will get you a lot reports: most false
  - Lot of work to analyze reports

Prioritize issues

- CVSS v3

Issues reported externally or published on the internet should get a higher priority

The higher the application risk rating the higher the priority

60 day fix time is common practice

Security bulletin

Release date	*: 10-06-2016*
Vulnerability ID	*: 16-0132*
CVSSv3 score	*: 6.2 (CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N/E:F/RL:U/RC:C)*
Severity	*: Medium*
CVE number	*: N/A*

Affected software

Application X version 1.x
Application Y version 2.3 Build 1941 and older

Summary

When using Application X or Application Y in with configuration Z a rare race condition could occur that could result in a temporary bypass of security control Q

Solution

Application X version 1.x

Upgrade Application X version 1.x to version 2.0 or newer

Application Y version 2.3 Build 1941 and older

Upgrade Application Y to version 2.3 build 2133 or newer

Workaround

Limit access to .. .. using group policy

Privacy

- Privacy impact assessment
  - NIST Privacy Impact Assessments
  - MS Application Privacy Assessment
- Avoid handling PII where possible
- Define where PII will be used for in privacy statement
- Don’t keep PII longer than required
- Data processing agreement

TIPS

- Tools available on the OpenSAMM Wiki
- Use tools & materials from MS SDL
- Join OpenSAMM Mailing list and Monthly call
- Add me on LinkedIn : Jacco van Tuijl