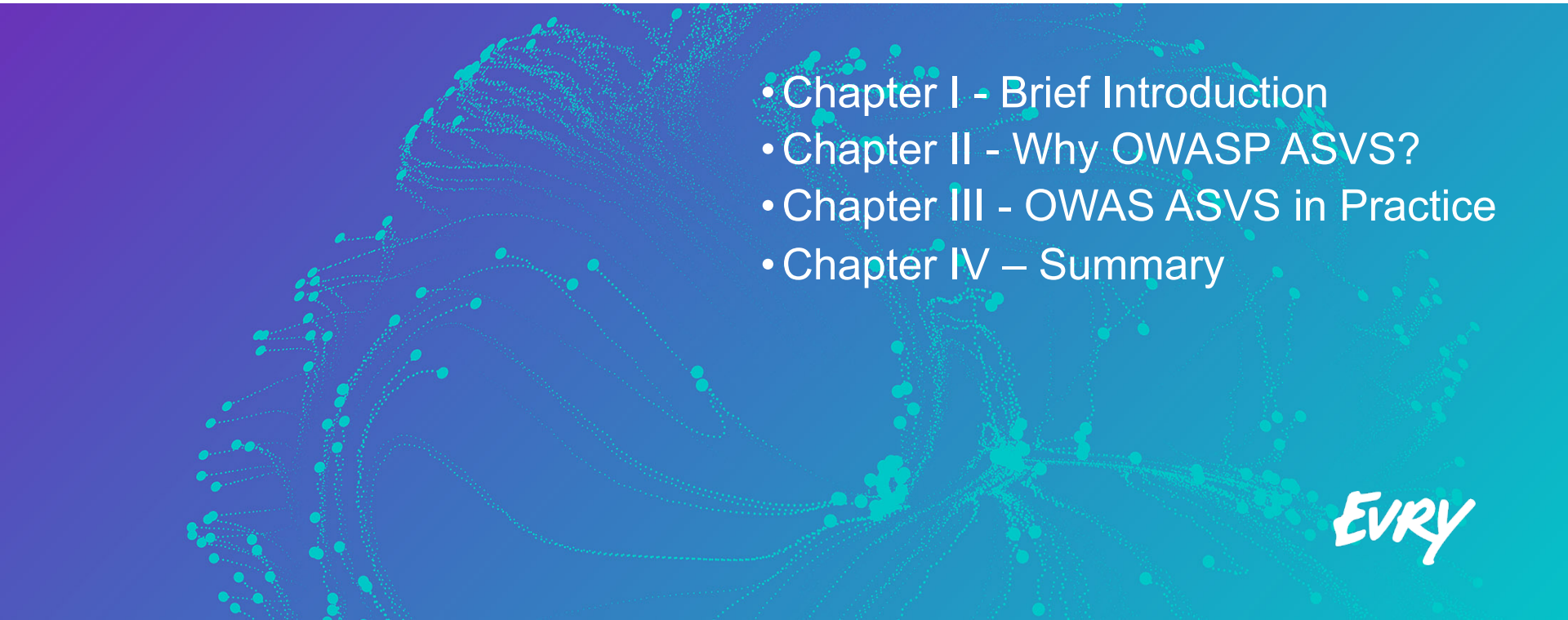
OWASP ASVS for NFTaaS in Financial Services

OLEKSANDR KAZYMYROV, TECHNICAL TEST ANALYST

EVRY

Agenda

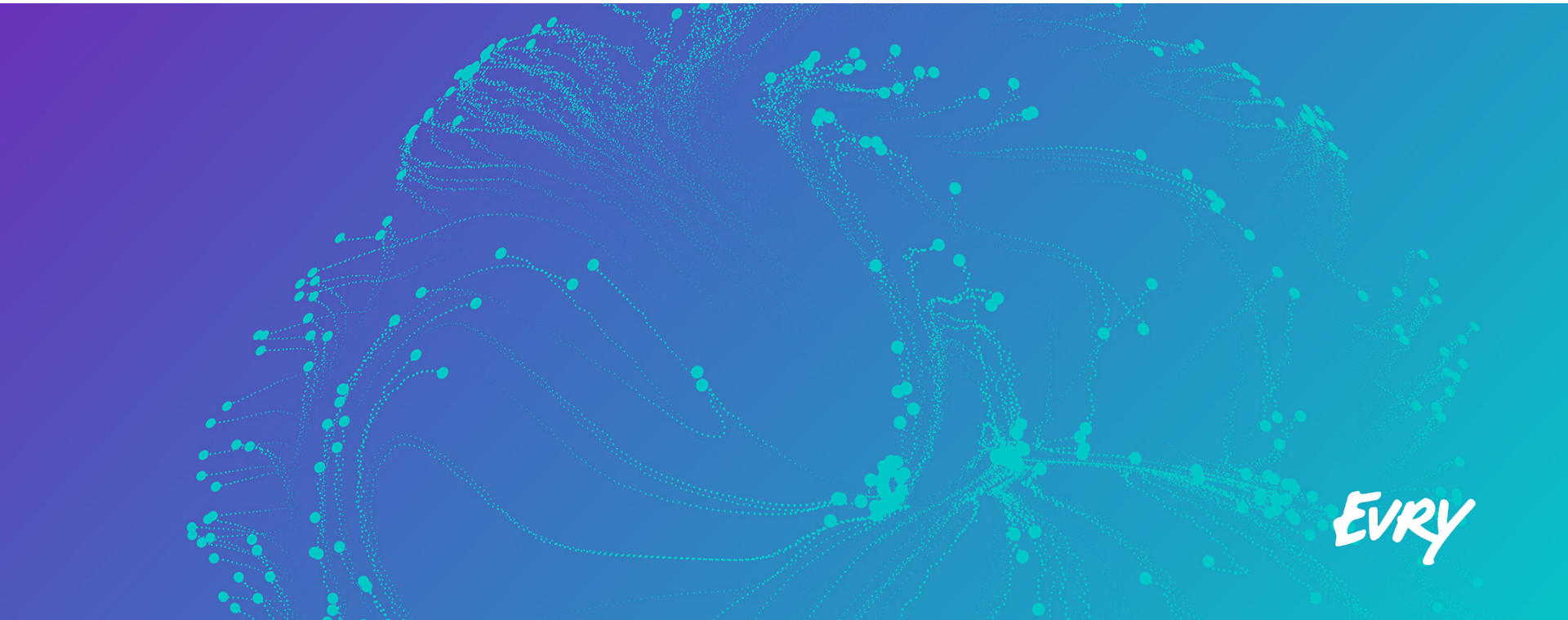


- Chapter I - Brief Introduction
- Chapter II - Why OWASP ASVS?
- Chapter III - OWAS ASVS in Practice
- Chapter IV – Summary

EVRY

CHAPTER I

Brief Introduction



Who am I?

Education

- Candidate of Engineering Sciences in Information Security
  KHNURE, Ukraine
- Ph.D. in Cryptology
  University of Bergen, Norway

Other

Certificates

- Certified Ethical Hacker
- Certified Encryption Specialist

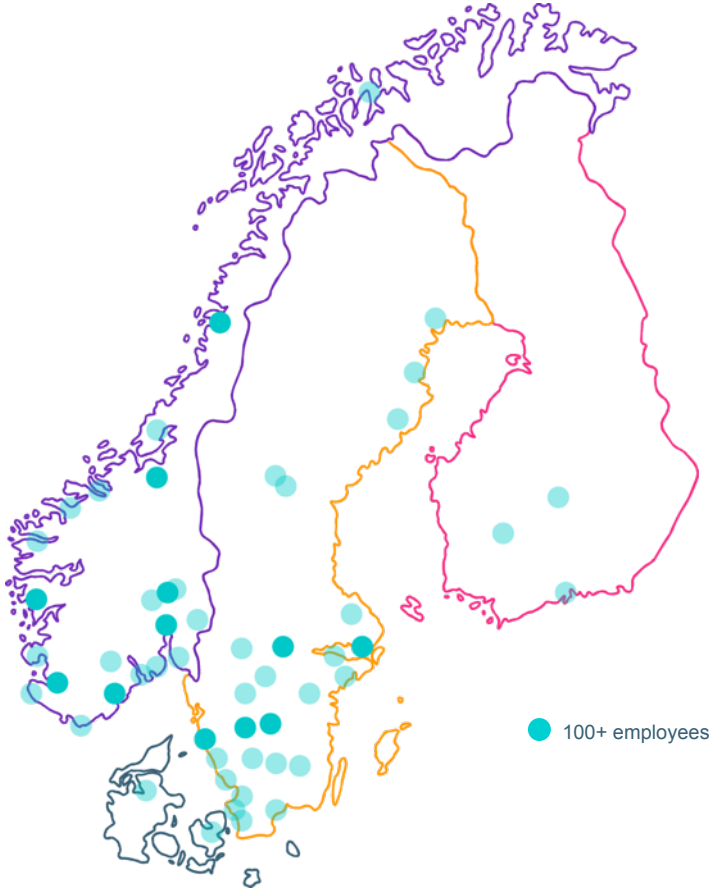
Standards

- DSTU 7624:2014
- DSTU 7564:2014

EVRY
– Nordic Champion

- 50 towns and cities with capacity to deliver
- 11 regional offices with specialist competencies
- 10.000 employees

Women	Age	Universum
26%	39yrs	#4

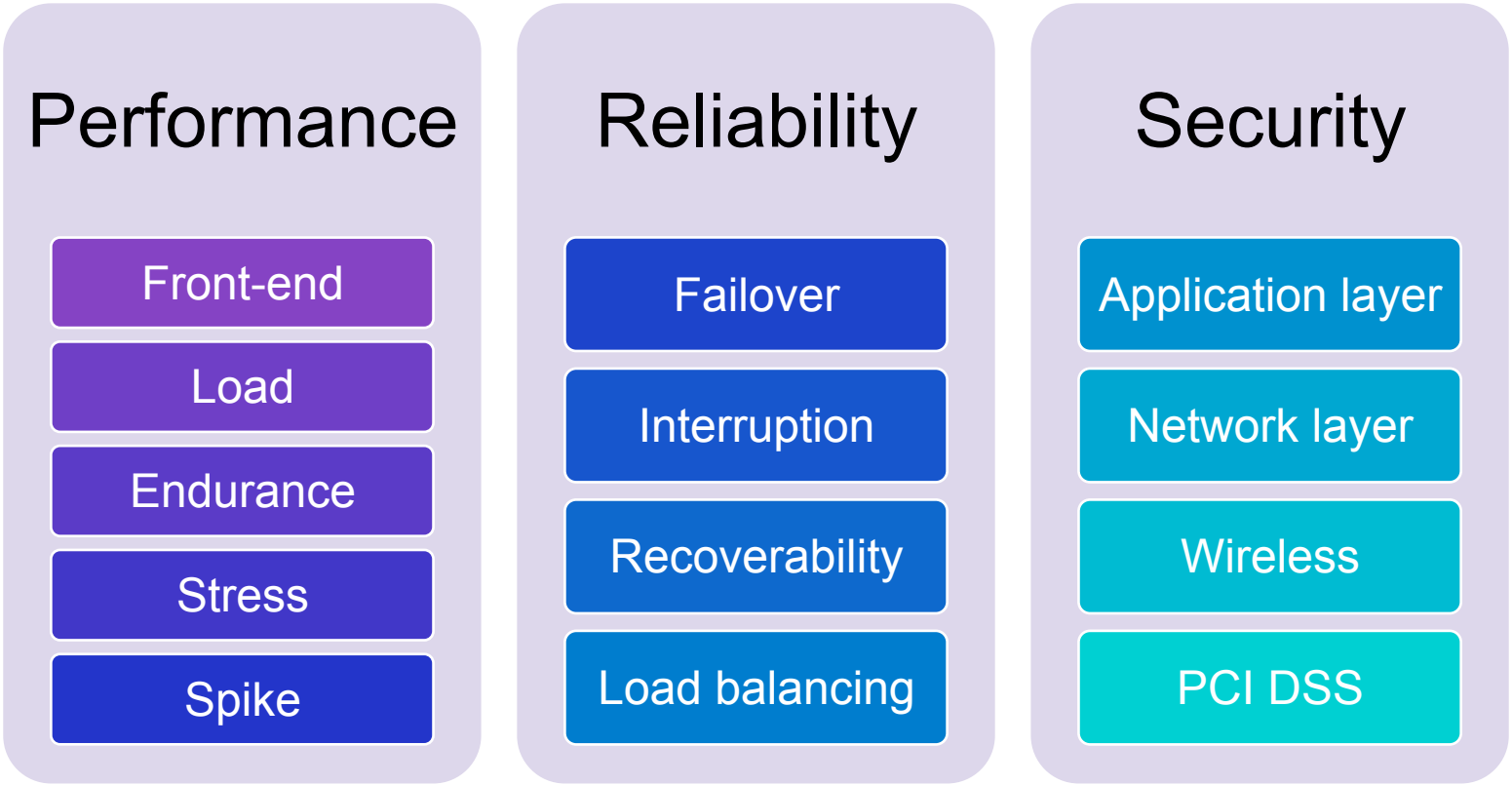


EVRY GROUP - Geographic distribution

Nordics

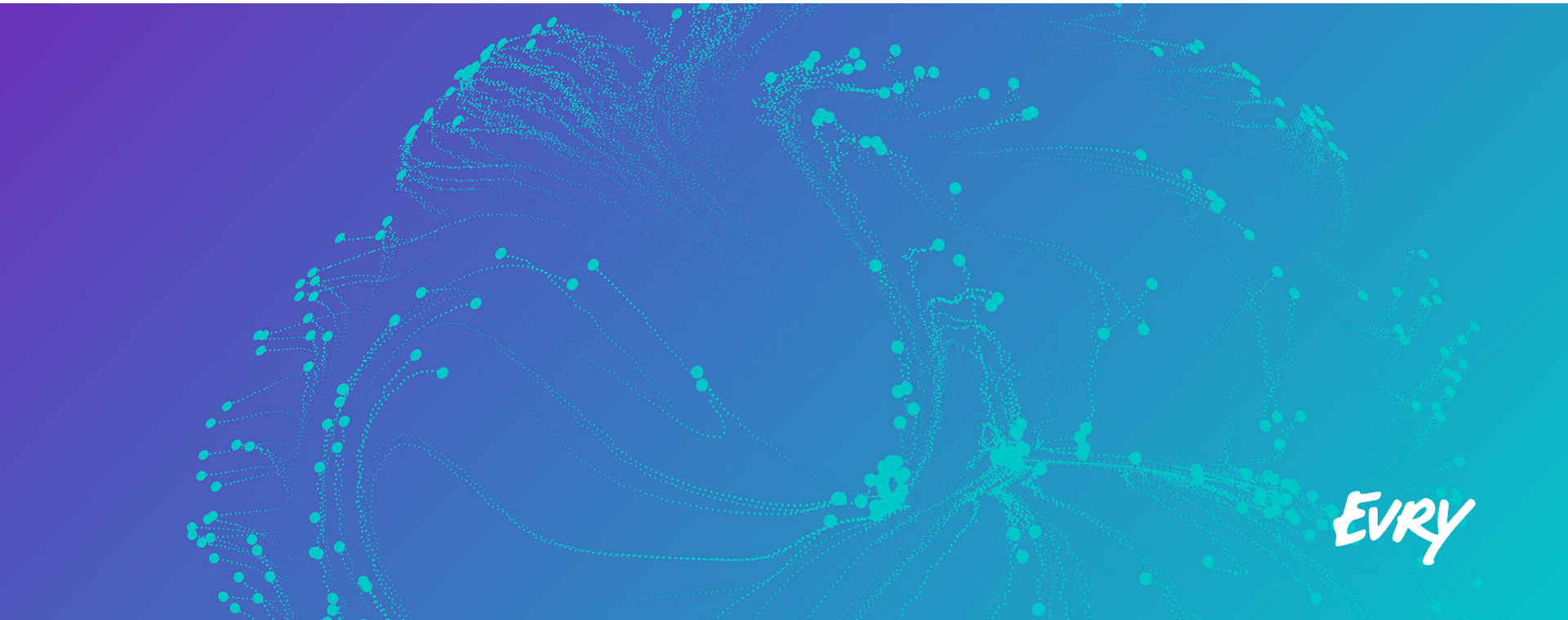
Rest of the World (Global Delivery)

NFT Department



Why OWASP ASVS?

CHAPTER II



PCI DSS Requirement 11.3

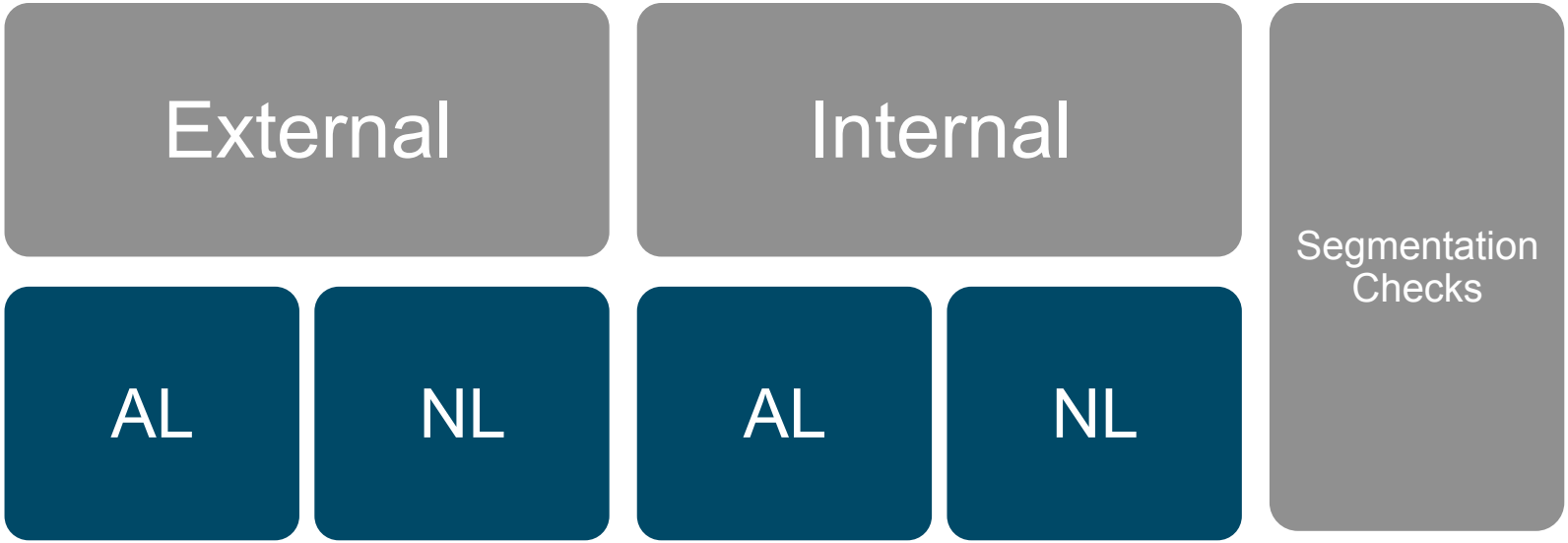
11.3 Implement a methodology for penetration testing that includes the following:

- Is based on industry-accepted penetration testing approaches (for example, NIST SP800-115)
- Includes coverage for the entire CDE perimeter and critical systems
- Includes testing from both inside and outside the network
- Includes testing to validate any segmentation and scope-reduction controls
- Defines application-layer penetration tests to include, at a minimum, the vulnerabilities listed in Requirement 6.5
- Defines network-layer penetration tests

A penetration test differs from a vulnerability scan, as a penetration test is an active process that may include exploiting identified vulnerabilities. Conducting a vulnerability scan may be one of the first steps a penetration tester will perform in order to plan the testing strategy, although it is not the only step. Even if a vulnerability scan does not detect known vulnerabilities, the penetration tester will often gain enough knowledge about the system to identify possible security gaps.

Penetration testing is generally a highly manual process. While some automated tools may be used, the tester uses their knowledge of systems to penetrate into an environment. Often the tester will chain several types of exploits together with a goal of breaking through layers of defenses. For example, if the tester finds a means to gain access to an application server, they will then use the compromised server as a point to stage a new attack based on the resources the server has access to. In this way, a tester is able to simulate

PCI DSS Penetration Testing



EVRY

NIST SP 800-115: Appendix C - Application Security Testing and Examination

Application security testing and examination help an organization determine whether its custom application software—for example, Web applications—contains vulnerabilities that can be exploited, and whether the software behaves and interacts securely with its users, other applications (such as databases), and its execution environment. Application security can be assessed in a number of ways, ranging from source code review to penetration testing of the implemented application.[43] Many application security tests subject the application to known attack patterns typical for that application's type. These patterns may directly target the application itself, or may attempt to attack indirectly by targeting the execution environment or security infrastructure. Examples of attack patterns are information leakage (e.g., reconnaissance, exposure of sensitive information), authentication exploits, session management exploits, subversion (e.g., spoofing, impersonation, command injections), and denial of service attacks.

Application security assessment should be integrated into the software development life cycle of the application to ensure that it is performed throughout the life cycle. For example, code reviews can be performed as code is being implemented, rather than waiting until the entire application is ready for testing. Tests should also be performed periodically once an application has gone into production; when significant patches, updates, or other modifications are made; or when significant changes occur in the threat environment where the application operates.

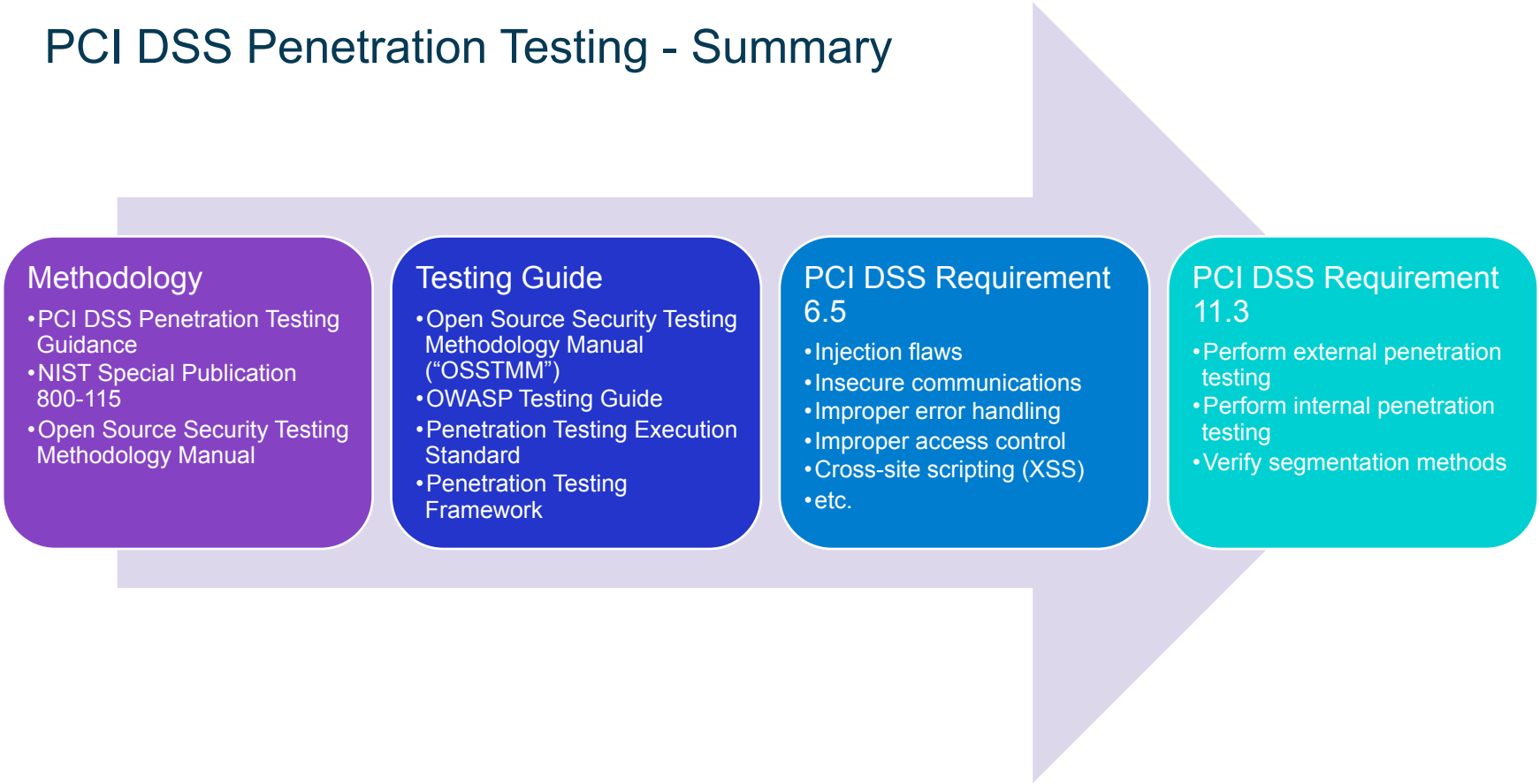
EVRY

NIST SP 800-115: Appendix E - Table E-2. Online Resources

Resource	URL
Methodologies	
Information Design Assurance Red Team (IDART)	http://www.idart.sandia.gov/
NIST SP 800-53A, *Guide for Assessing the Security Controls in Federal Information Systems*	http://csrc.nist.gov/publications/PubsSPs.html
National Security Agency (NSA) Information Assessment Methodology (IAM)	http://www.nsa.gov/ia/industry/education/iam.cfm?MenuID=10.2.4.2
Open Source Security Testing Methodology Manual (OSSTMM)	http://www.isecom.org/osstmm/
Open Web Application Security Project (OWASP) Testing Project	http://www.owasp.org/index.php/Category:OWASP_Testing_Project
Tools	
BackTrack (Linux live distribution)	http://www.remote-exploit.org/backtrack.html

EVRY

PCI DSS Penetration Testing - Summary

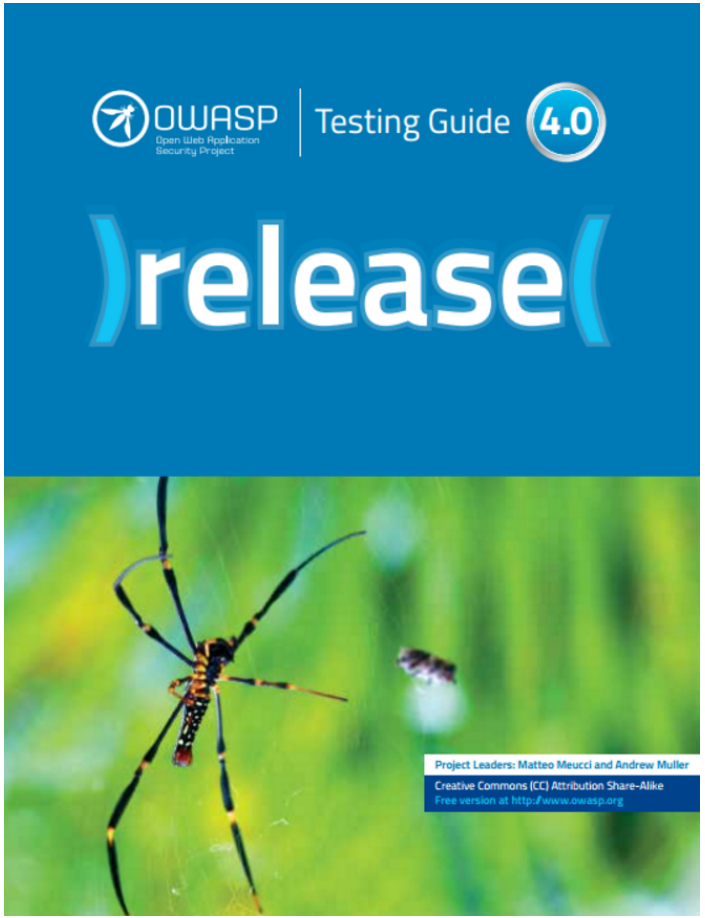


OWASP Testing Guide (from PCI Pentest Guide)

4.4 Additional Resources

There are multiple industry-accepted methodologies that may provide additional guidance on penetration testing activities, including but not limited to:

- Open Source Security Testing Methodology Manual (“OSSTMM”)
- The National Institute of Standards and Technology (“NIST”) Special Publication 800-115
- OWASP Testing Guide
- Penetration Testing Execution Standard
- Penetration Testing Framework



OWASP Top 10 2013 vs PCI DSS

OWASP Top 10 2013

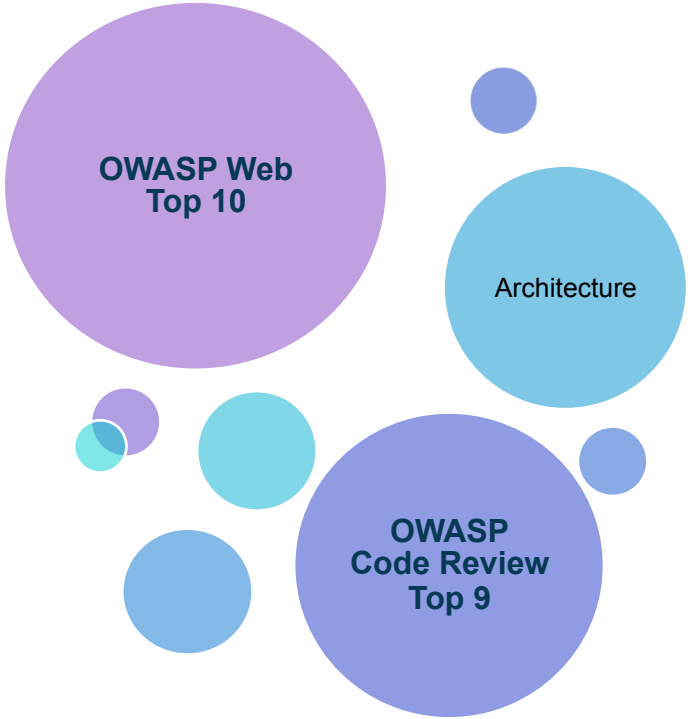
- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Components with Known Vulnerabilities
- A10 Invalidated Redirects and Forwards

PCI DSS Requirements

- 6.5.1 Injection flaws / 6.5.2 Buffer overflows
- 6.5.10 Broken authentication and session management
- 6.5.7 XSS
- ? -
- ? 6.5.6 All “high risk” vulnerabilities
- ? 6.5.5 Improper error handling
- 6.5.8 Improper access control / 6.5.3 Insec. cryptostorage
- 6.5.9 CSRF
- 6.5.6 All “high risk” vulnerabilities

- ? 6.5.4 Insecure communications

OWASP Application Security Verification Standard (ASVS)



OWASP ASVS v3.0.1

EVRY

Key parts of OWAS ASVS

Scope for the application security verification standard

Description of security verification levels

Requirements / Controls

Standards Mappings

OWAS ASVS Verification Controls (v3.0.1)

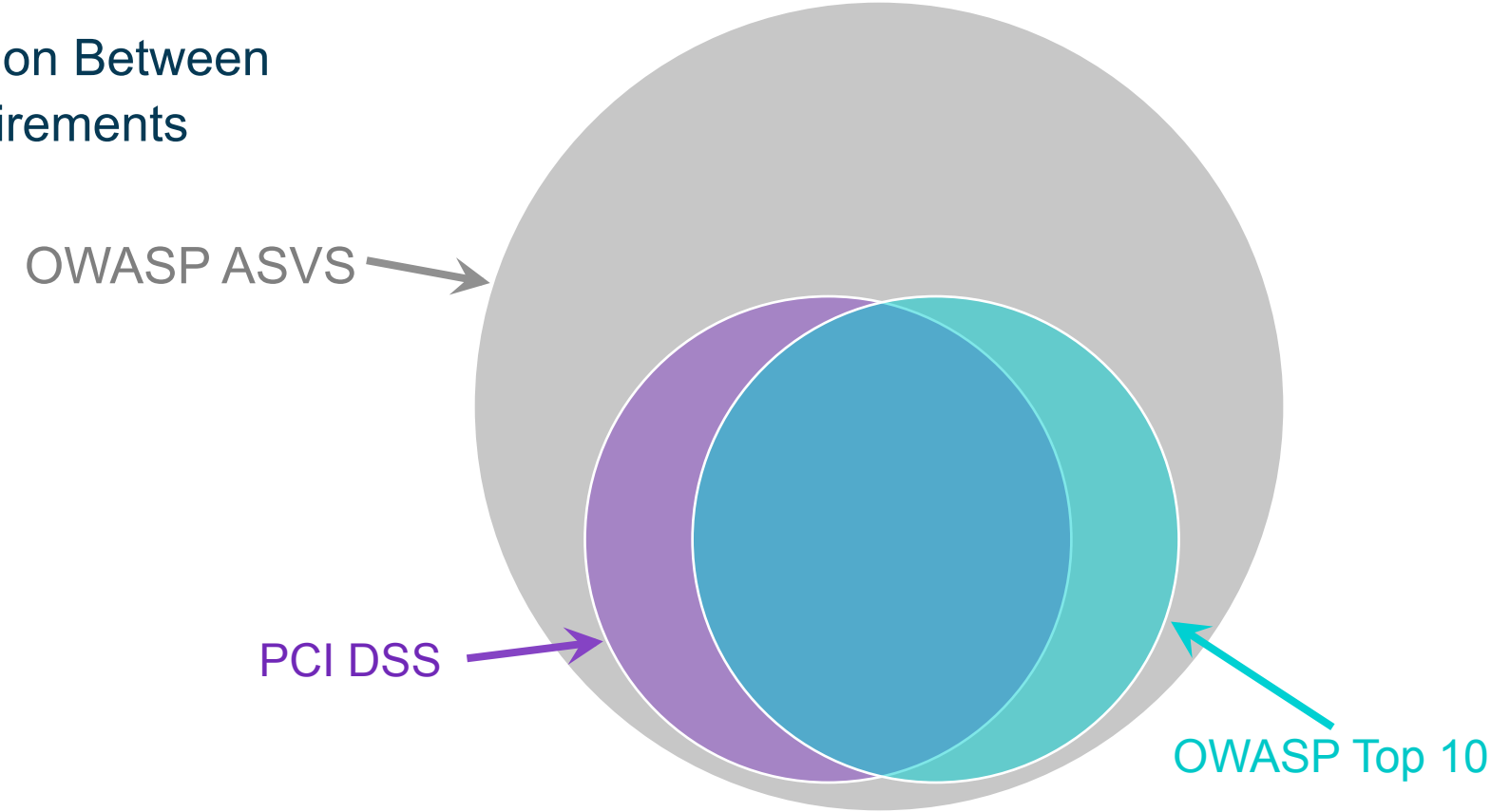
Category	Level 1	Level 2	Level 3
V1: Architecture, design and threat modelling	1	6	10
V2: Authentication Verification Requirements	17	24	26
V3: Session Management Verification Requirements	10	12	13
V4: Access Control Verification Requirements	7	11	12
V5: Malicious input handling verification requirements	10	20	21
V7: Cryptography at rest verification requirements	2	7	10
V8: Error handling and logging verification requirements	1	7	12
V9: Data protection verification requirements	4	8	11
V10: Communications security verification requirements	7	8	13
V11: HTTP security configuration verification requirements	6	8	8
V13: Malicious controls verification requirements	0	0	2
V15: Business logic verification requirements	0	2	2
V16: Files and resources verification requirements	7	9	9
V17: Mobile verification requirements	6	9	11
V18: Web services verification requirements	7	10	10
V19: Configuration	1	5	9
Total:	86	146	179

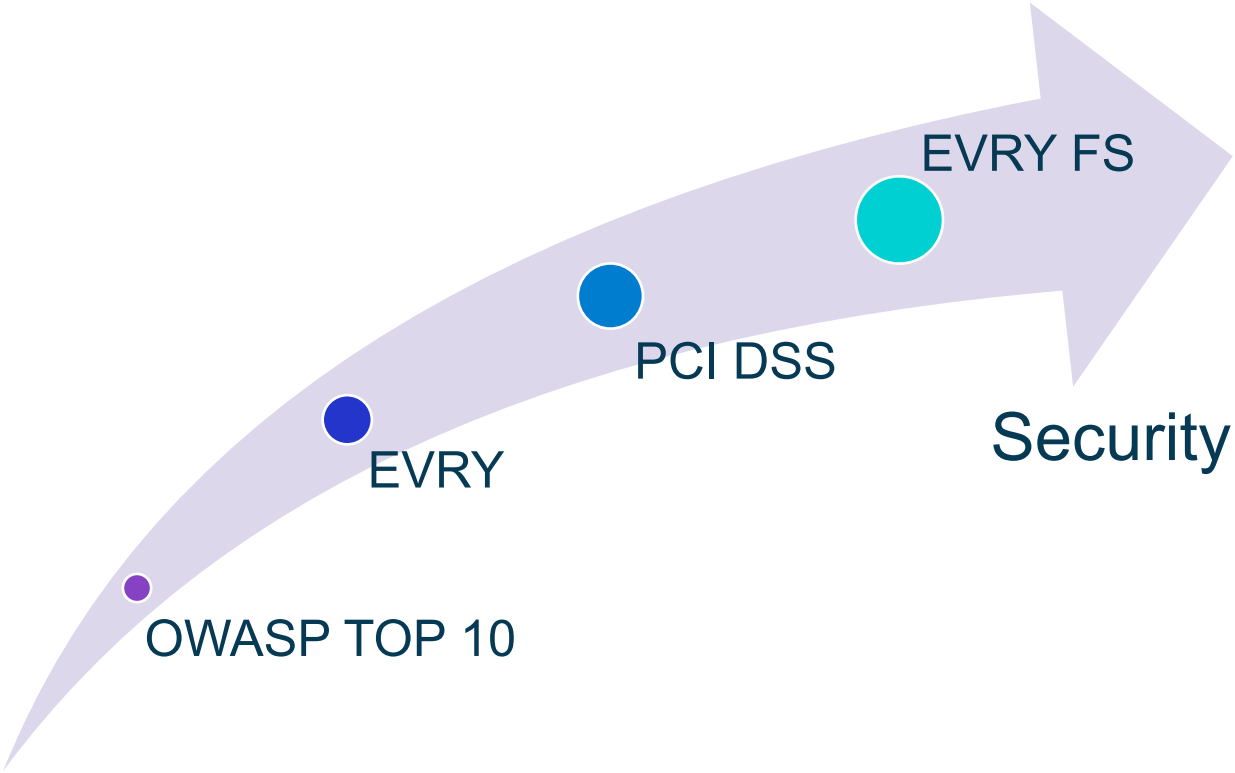
EVRY

OWASP ASVS: Standards Mappings

PCI-DSS 3.0	ASVS 3.0	Description
6.5.8 Improper Access Control (such as insecure direct object references, failure to restrict URL access, directory traversal and failure to restrict user access to functions).	v4 - all	Comprehensive mapping from Level 1 up
6.5.9 Cross-site request forgery (CSRF).	4.13	Exact mapping. ASVS considers CSRF defense to be an aspect of access control.
6.5.10 Broken authentication and session management.	v2 and v3 - all	Comprehensive mapping from Level 1 up

Relation Between Requirements

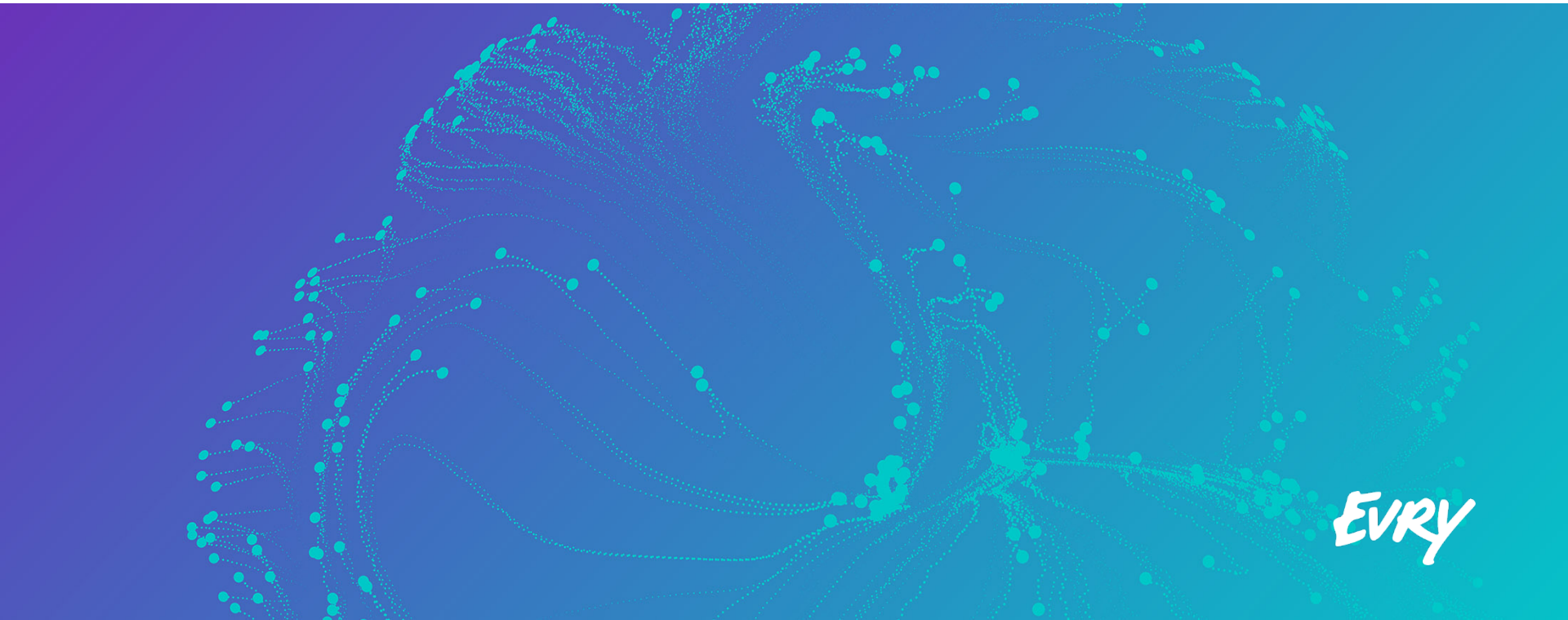




Scope for pentesting of web applications

OWAS ASVS in Practice

CHAPTER III



OWAS ASVS Verification Controls

#	Description	1	2	3
7.2	Verify that all cryptographic modules fail securely, and errors are handled in a way that does not enable oracle padding.	✓	✓	✓
7.6	Verify that all random numbers, random file names, random GUIDs, and random strings are generated using the cryptographic module's approved random number generator when these random values are intended to be not guessable by an attacker.		✓	✓
7.7	Verify that cryptographic algorithms used by the application have been validated against FIPS 140-2 or an equivalent standard.	✓	✓	✓
7.8	Verify that cryptographic modules operate in their approved mode according to their published security policies.			✓
7.9	Verify that there is an explicit policy for how cryptographic keys are managed (e.g., generated, distributed, revoked, and expired). Verify that this key lifecycle is properly enforced.		✓	✓
7.11	Verify that all consumers of cryptographic services do not have direct access to key material. Isolate cryptographic processes, including master secrets and consider the use of a virtualized or physical hardware key vault (HSM).			✓

#	Description	1	2	3

9.4	Verify that the application sets appropriate anti-caching headers as per the risk of the application, such as the following:
	`Expires: Tue, 03 Jul 2001 06:00:00 GMT`
	`Last-Modified: {now} GMT`
	`Cache-Control: no-store, no-cache, must-revalidate, max-age=0`
	`Cache-Control: post-check=0, pre-check=0`
	`Pragma: no-cache`
		✓	✓	✓

9.5	Verify that on the server, all cached or temporary copies of sensitive data stored are protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data.		✓	✓
9.6	Verify that there is a method to remove each type of sensitive data from the application at the end of the required retention policy.			✓
9.7	Verify the application minimizes the number of parameters in a request, such as hidden fields, Ajax variables, cookies and header values.		✓	✓
9.8	Verify the application has the ability to detect and alert on abnormal numbers of requests for data harvesting for an example screen scraping.			✓
9.9	Verify that data stored in client side storage (such as HTML5 local storage, session storage, IndexedDB, regular cookies or Flash cookies) does not contain sensitive data or PII.	✓	✓	✓

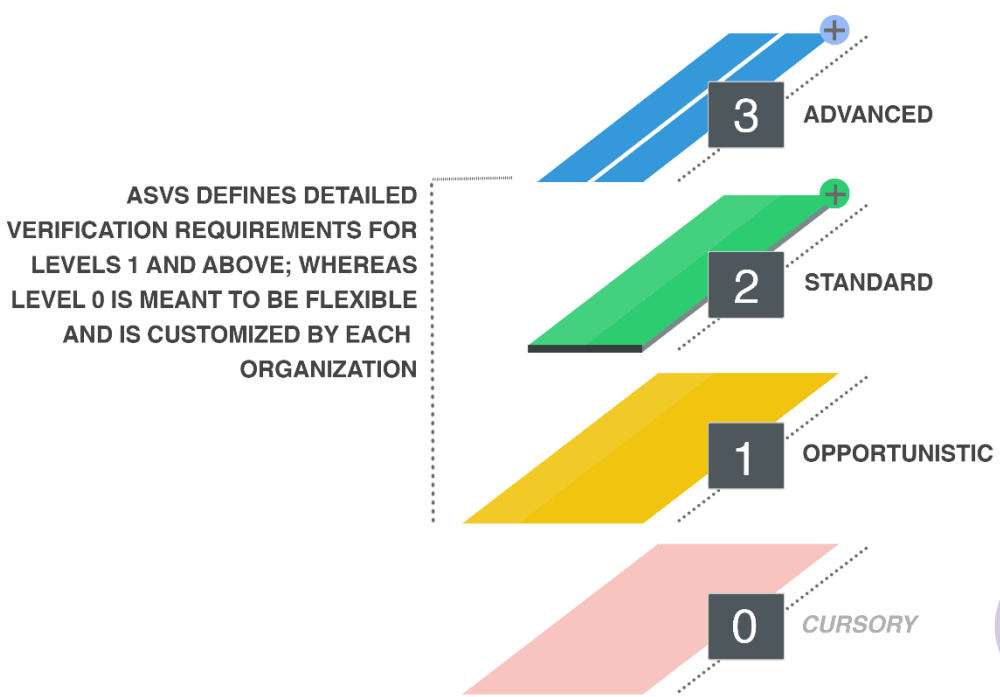
EVRY

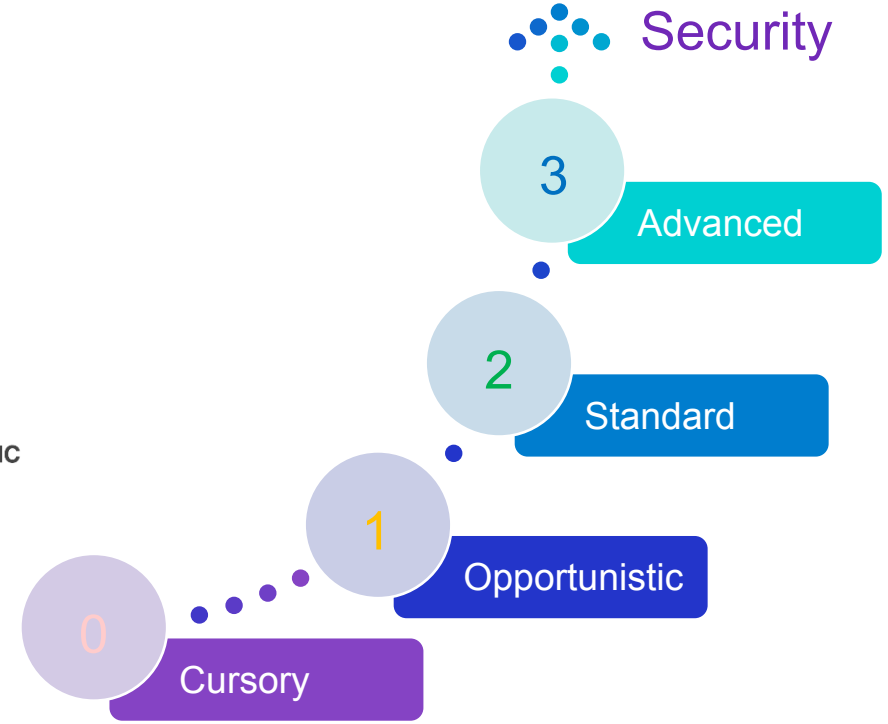
OWAS ASVS Verification Controls (v3.0.1)

Category	Level 1	Level 2	Level 3
V1: Architecture, design and threat modelling	1	6	10
V2: Authentication Verification Requirements	17	24	26
V3: Session Management Verification Requirements	10	12	13
V4: Access Control Verification Requirements	7	11	12
V5: Malicious input handling verification requirements	10	20	21
V7: Cryptography at rest verification requirements	2	7	10
V8: Error handling and logging verification requirements	1	7	12
V9: Data protection verification requirements	4	8	11
V10: Communications security verification requirements	7	8	13
V11: HTTP security configuration verification requirements	6	8	8
V13: Malicious controls verification requirements	0	0	2
V15: Business logic verification requirements	0	2	2
V16: Files and resources verification requirements	7	9	9
V17: Mobile verification requirements	6	9	11
V18: Web services verification requirements	7	10	10
V19: Configuration	1	5	9
Total:	86	146	179

EVRY

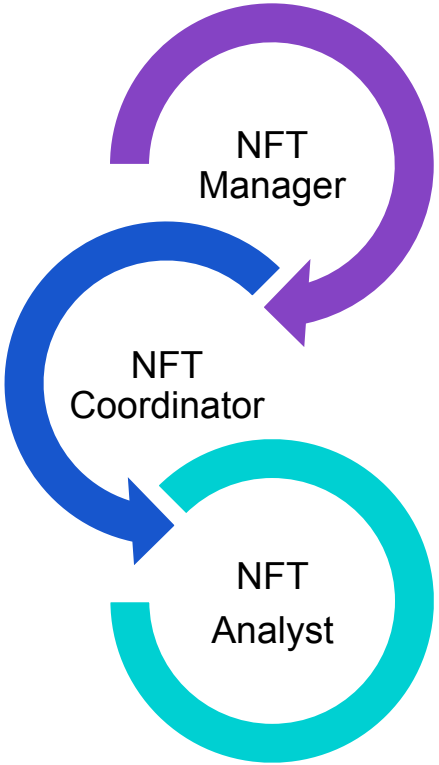
OWASP ASVS Levels

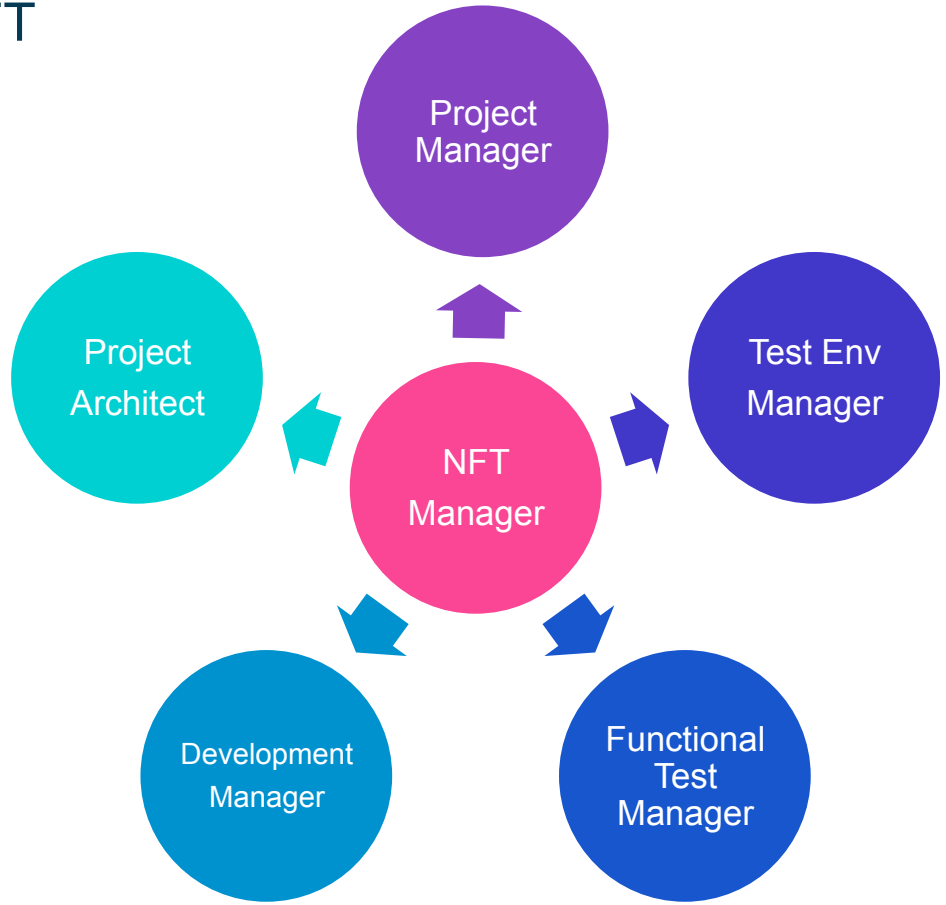




An Issue With Level Definition

Relation Between Project and NFT



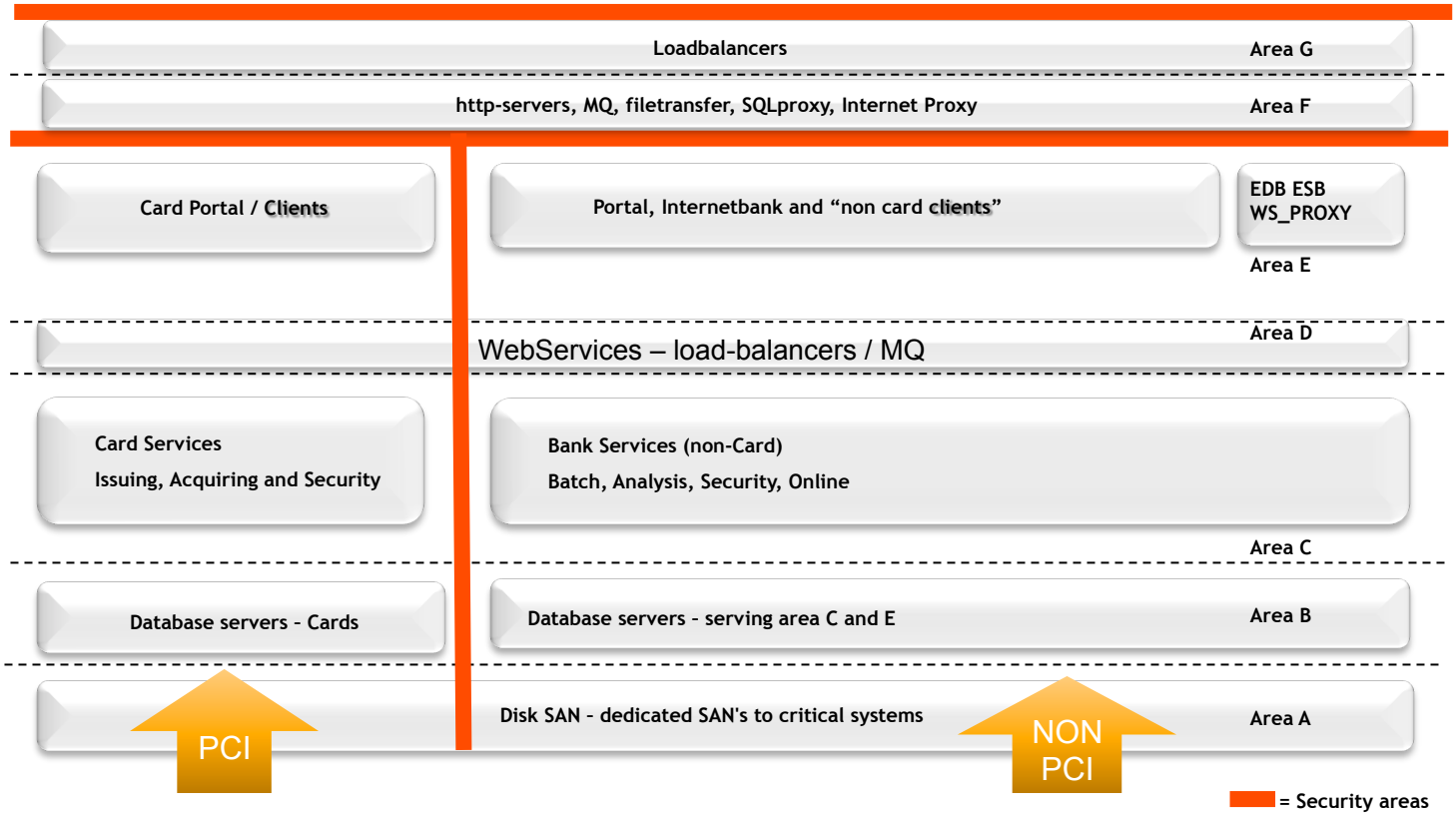


Compliance Selection at Financial Services

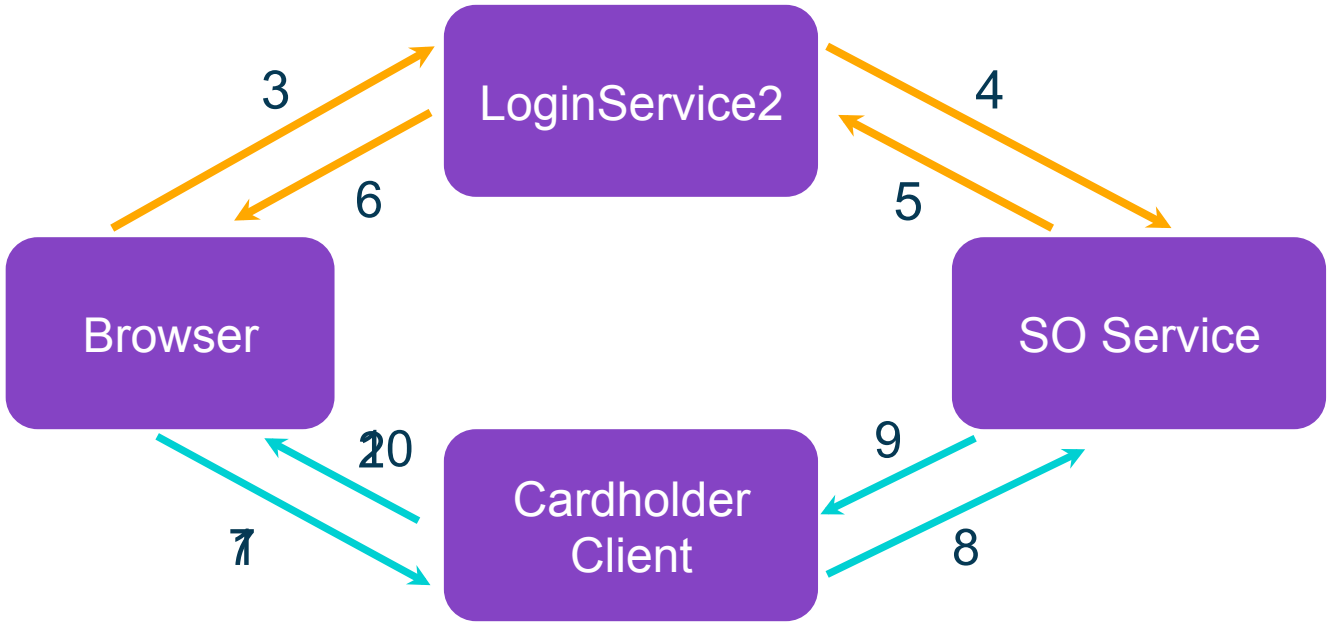
FINODS	Highly Sensitive	Moderately Sensitive	Low Sensitive
SSW - Self Service Non-Portal Applications over Internet	OWASP ASVS L3	OWASP ASVS L2	OWASP ASVS L2
SSP - Self Service Portal Applications over Internet	OWASP ASVS L3	OWASP ASVS L2	OWASP ASVS L2
CSW - Non-portal applications over dedicated Office Channel	OWASP ASVS L3	OWASP ASVS L2	OWASP ASVS L1
CSP - Portal applications over dedicated Office Channel	OWASP ASVS L3	OWASP ASVS L2	OWASP ASVS L1
ESI - Web Services Applications over Internet	OWASP ASVS L3	OWASP ASVS L2	OWASP ASVS L2
ESS - Integrated Customer solutions over Service Layer	OWASP ASVS L3	OWASP ASVS L2	OWASP ASVS L2

EVRY

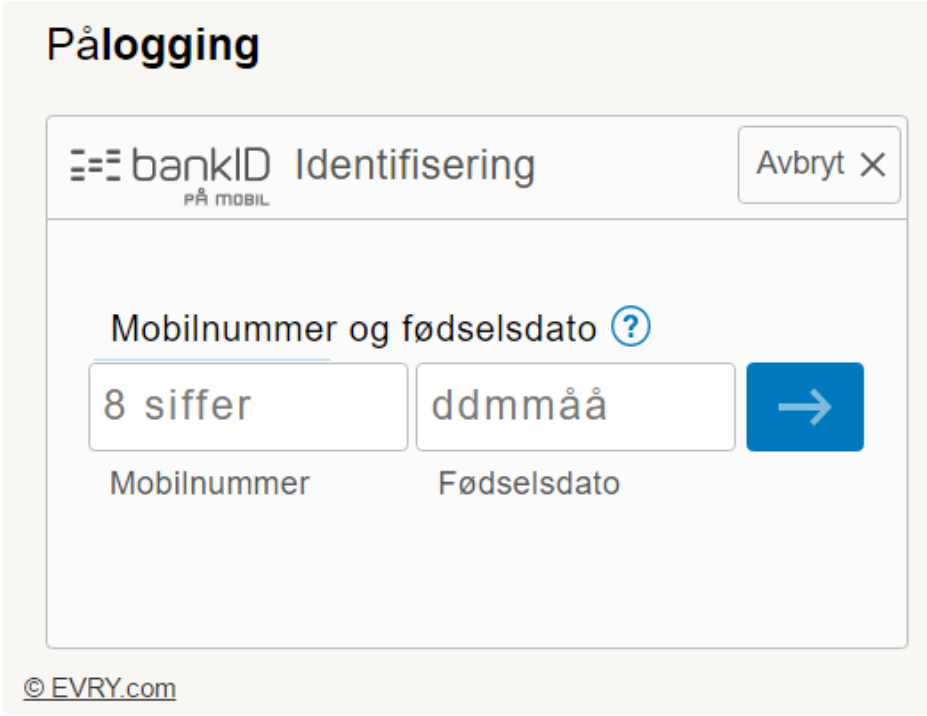
EVRY FINancial suite Operational Domains in SaaS (FINODS)



Authentication in Cardholder Client (CHC) Using LoginService2 (LS2)



LoginService2



Cardholder Client

Logg inn med BankID

Logg inn med BankID på mobil

Velge PIN

Du kan nå velge hvilken PIN det nye kortet ditt skal få.
NB: Denne PIN-koden vil også bli tildelt dine andre kort neste gang disse fornyes.

Følgende PIN-koder er ikke tillatt å velge:
- 4 identiske siffer (0000, 1111 osv).
- 4 løpende siffer (1234, 6543 osv).

Det frarådes å velge PIN-koder som kan assosieres med deg (for eksempel fødselsdato, postnummer osv).

Fyll inn ønsket PIN-kode i feltene nedenfor. Du må skrive identisk PIN i begge felter.

Ønsket PIN *

Gjenta PIN *

Velge PIN

PIN-kode oppdatert

Du har nå valgt PIN-koden til ditt nye kort.

For å motta SMS-varsling når et nytt kort blir sendt til deg med denne PIN-koden, oppgi mobiltelefonnummeret.

Mobiltelefon *

General Information on LS2 and CHC

LoginSevice2

- LS2 stays in front of almost all applications
- It is the first major security barrier
- LS2 helps to retrieve tokens (Secure Object or simply SO) and hand over it to the 3rd party applications
- Available through the Internet

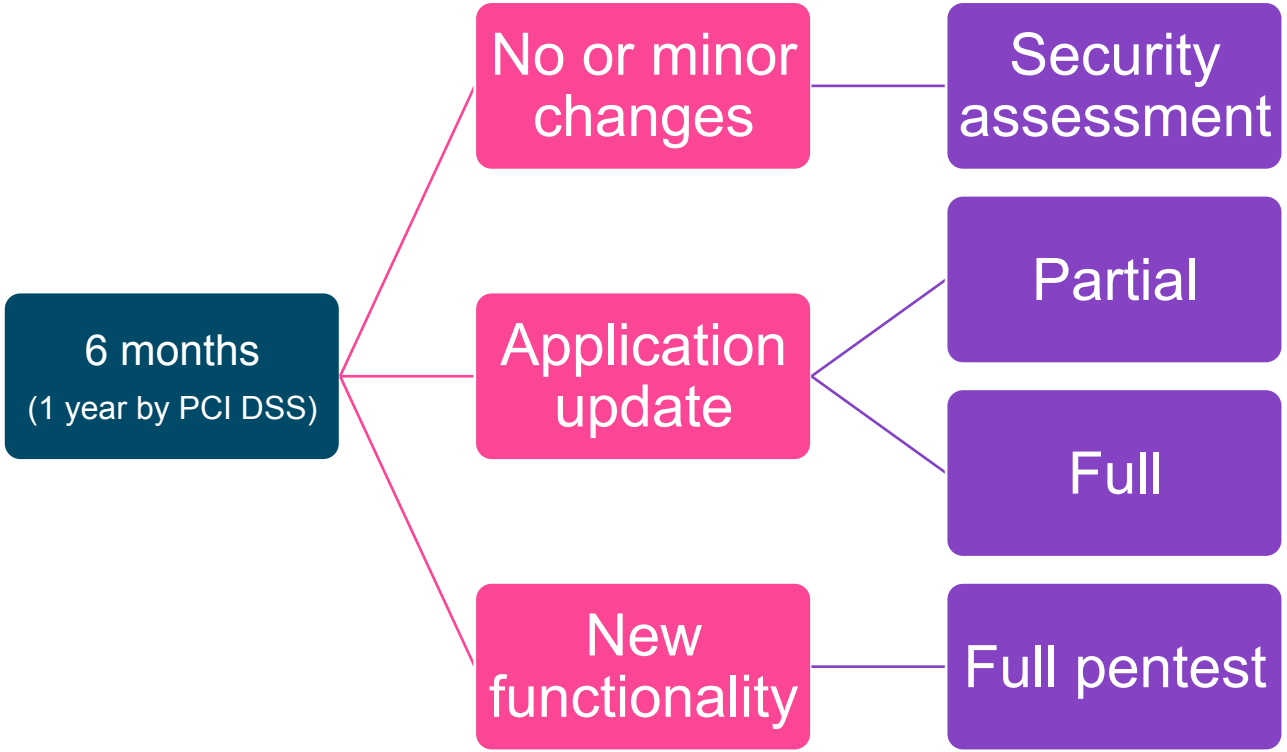
Cardholder Client

- CHC is a part of EVRY's NetBank (Online banking)
- It can be integrated with any 3rd party web application
- EVRY's NetBank is protected by LoginsService2 in front of CHC
- After logging in CHC uses SO as the main parameter in session management
- Available through the Internet

OWASP ASVS Level 3

OWASP ASVS Level 2

Security Application Life Cycle



Summary

- PCI DSS is a good starting point for any infrastructure
- OWASP ASVS is a flexible standard with minimal effort for adaptation
- For a stable security development lifecycle the following should be implemented
  - Standard operation procedures
  - Methodology for security testing
  - Security risk assessment
  - Role descriptions
  - General compliance levels

EVRY


EVRY
Digital
+\dvantage