TECNOLÓGICO SUPERIOR CORDILLERA

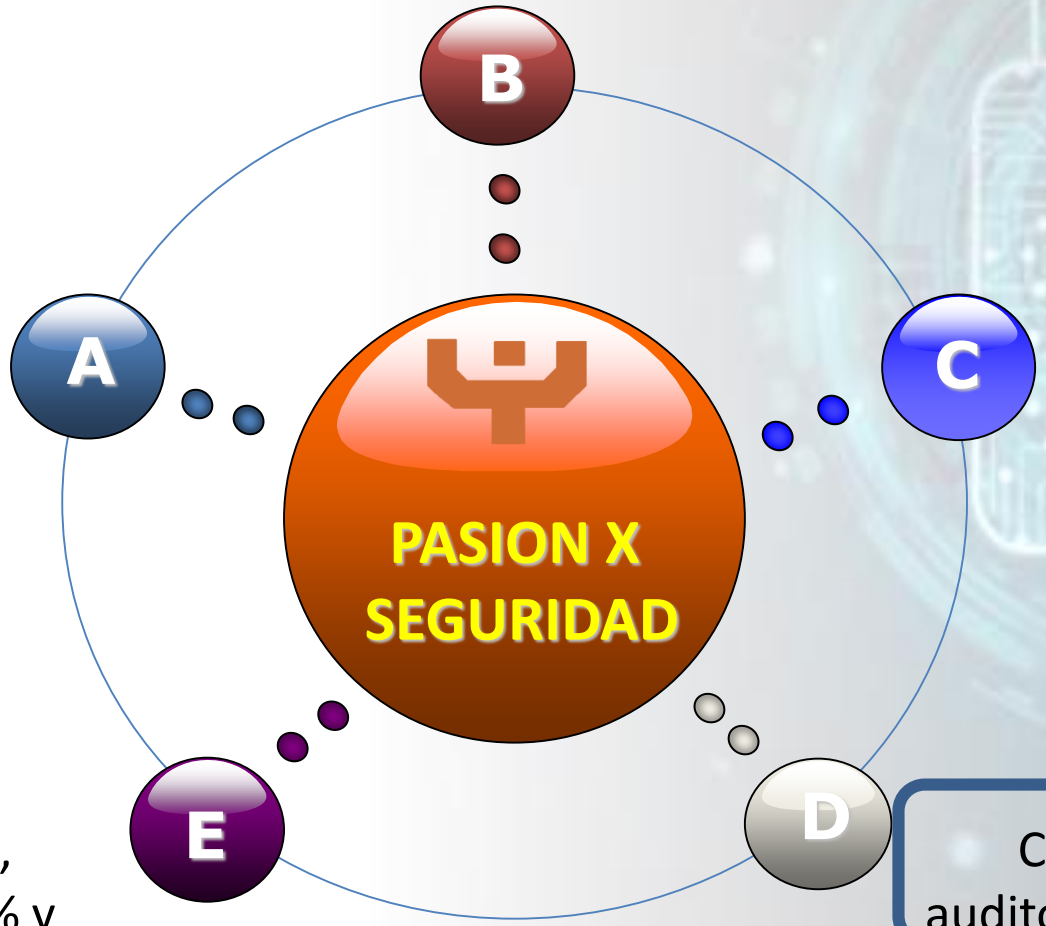
25 AÑOS

CIDE
Centro de Investigación y Desarrollo Ecuador

SEMINARIO INTERNACIONAL DE SEGURIDAD EN EL DESARROLLO DE SOFTWARE

GUIDO ROSALES URIONA

Maestrias en Redes (Aviacion Civil) y Direccion Estrategica de TI



Ing. Sistemas
xURSS

Certificaciones
CISA, CISM, LA 27001,
CBS, CISO, ISSA, CSSH

Facilitador 60%,
conferencista 30% y
Docente Universitario
10%

Consultor en Seguridad,
auditorias y FORENSE (50-20-30)

MSc. Ing. Guido Rosales

Desde el año 1999

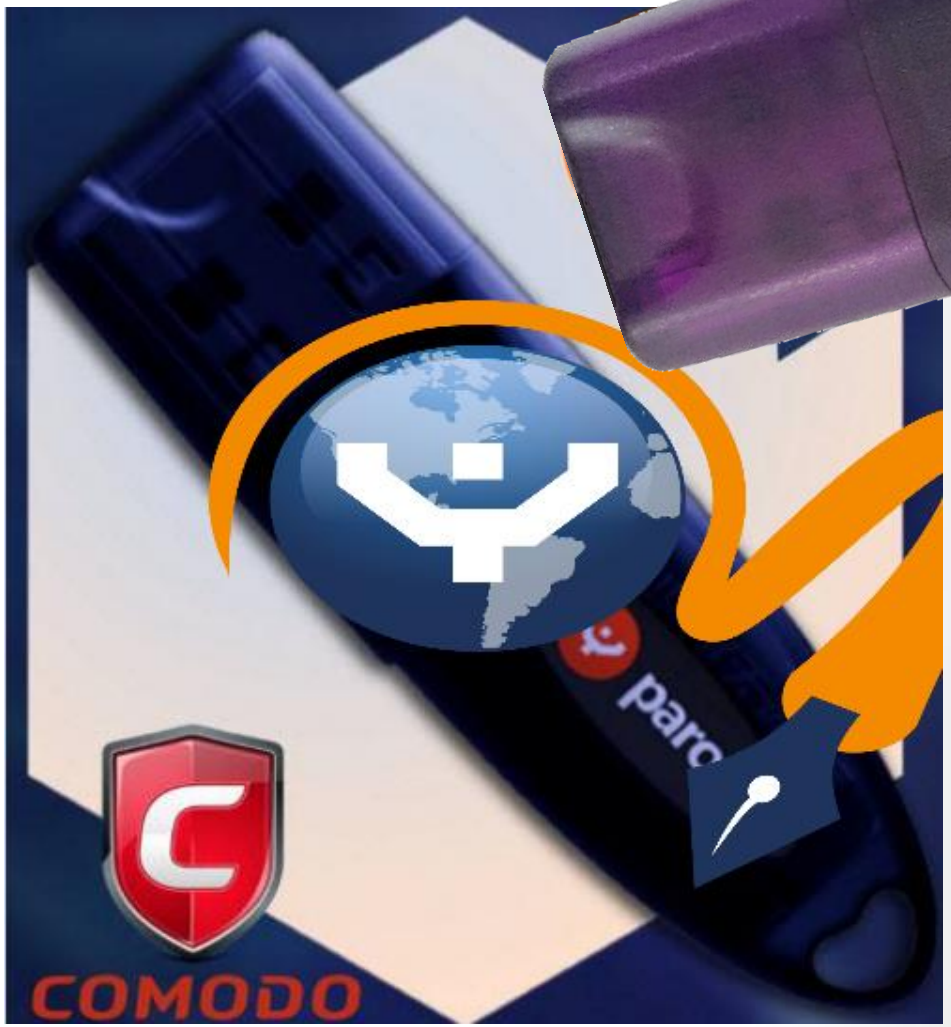
Yanapti

CYBERSECURITY

MSc. Ing. Guido Rosales

ESTANDARES PARA EL DESARROLLO SEGURO

MSc. Ing. Guido Rosales

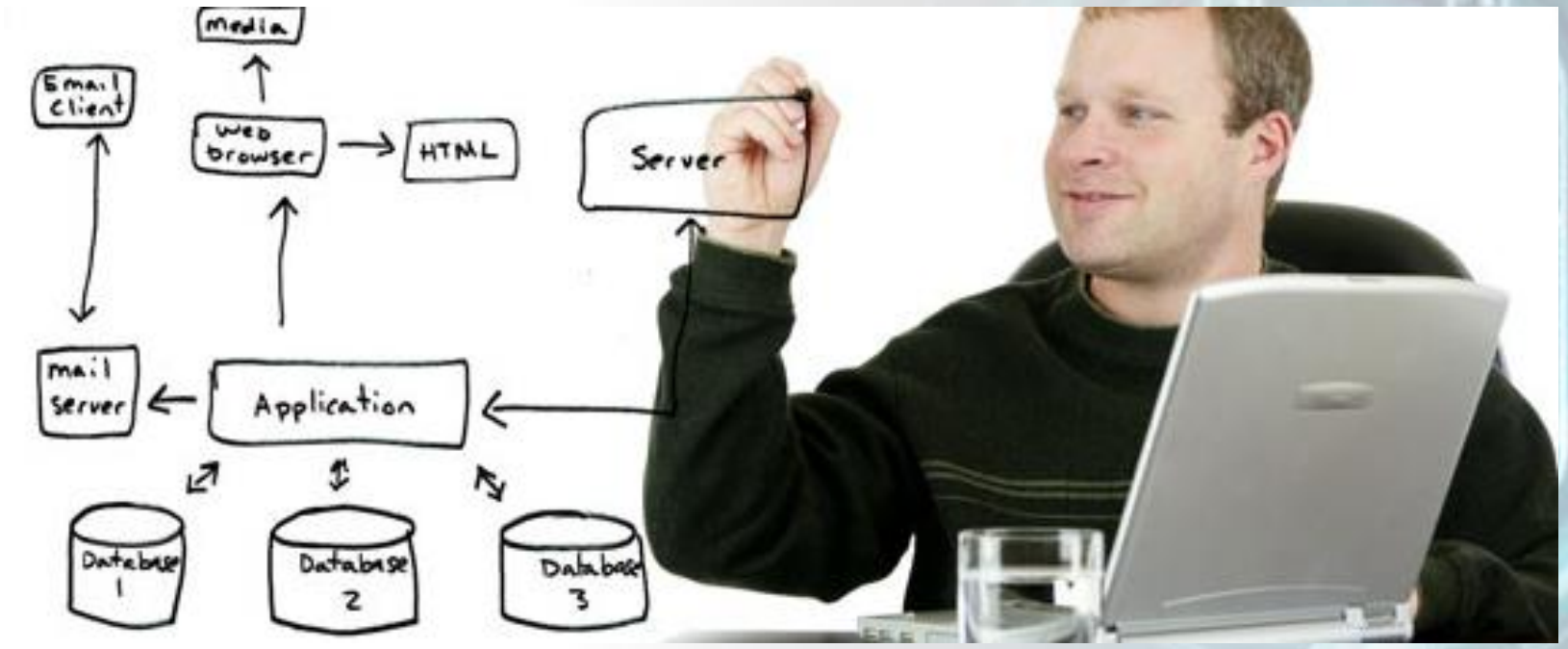

COMODO


ePass2003Auto


CORP

POR QUE SE DEBE ESTANDARIZAR?

MSc. Ing. Guido Rosales

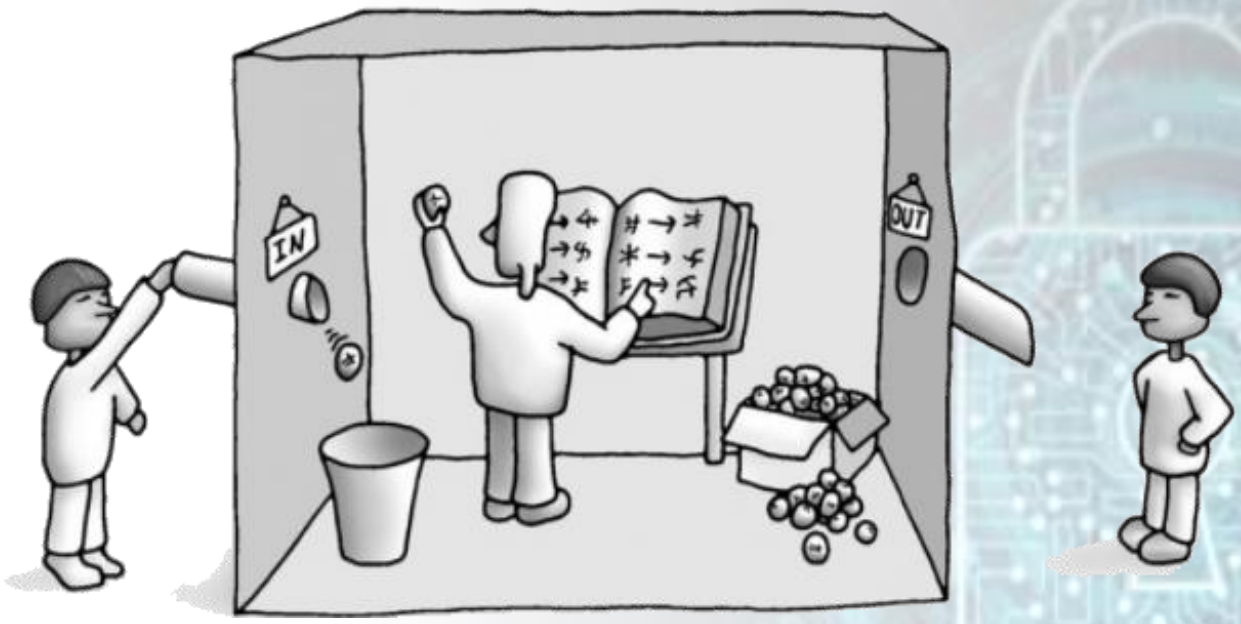


El PROGRAMADOR DE SISTEMAS

EL ARTISTA, EL CREADOR, EL JEDI, EL SCRUM MASTER

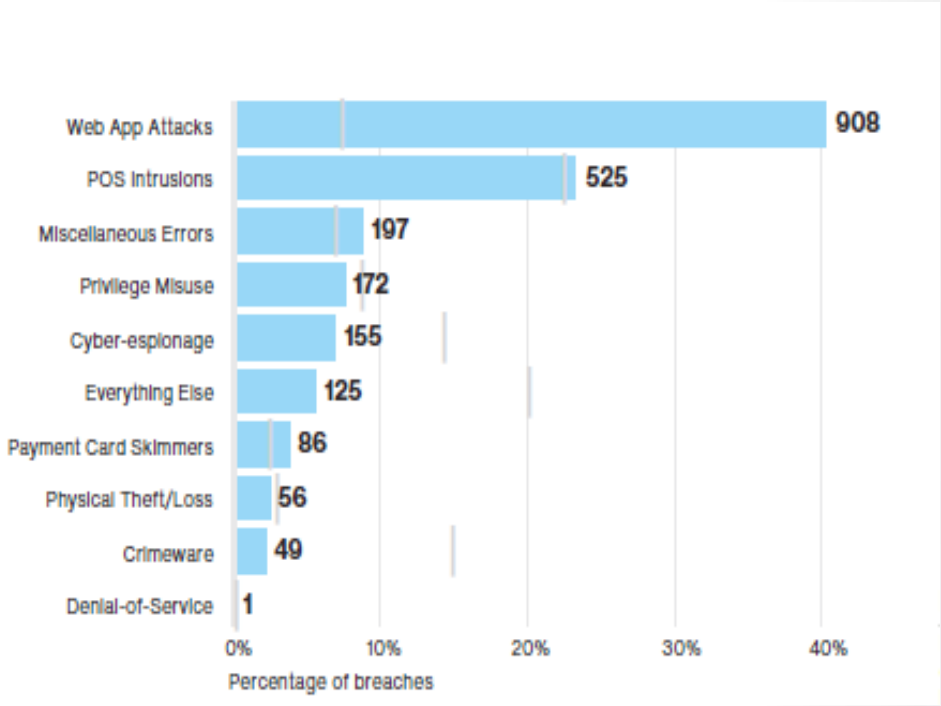
MSc. Ing. Guido Rosales

MSc. Ing. Guido Rosales


IN
OUT


CIBERCRIMINALES
6".0"
5".0"
4".0"
3".0
Hacker_
Black Hats_
Hacktivistas_
Script Kiddies_

Los ataques de aplicaciones Web representaron el 40% de todas las brechas de seguridad

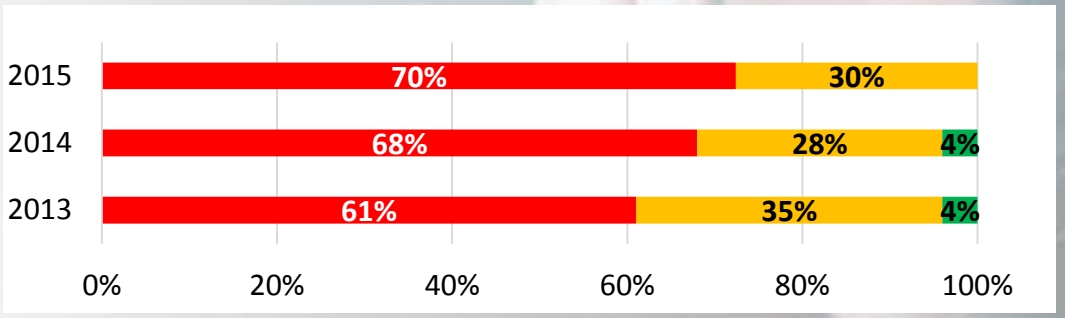


Verizon 2016 Data Breach Investigations Report

La cantidad de sitios web con vulnerabilidades de gravedad crítica está en constante crecimiento

- Severidad critica
- Media
- Baja

Positive Research 2016

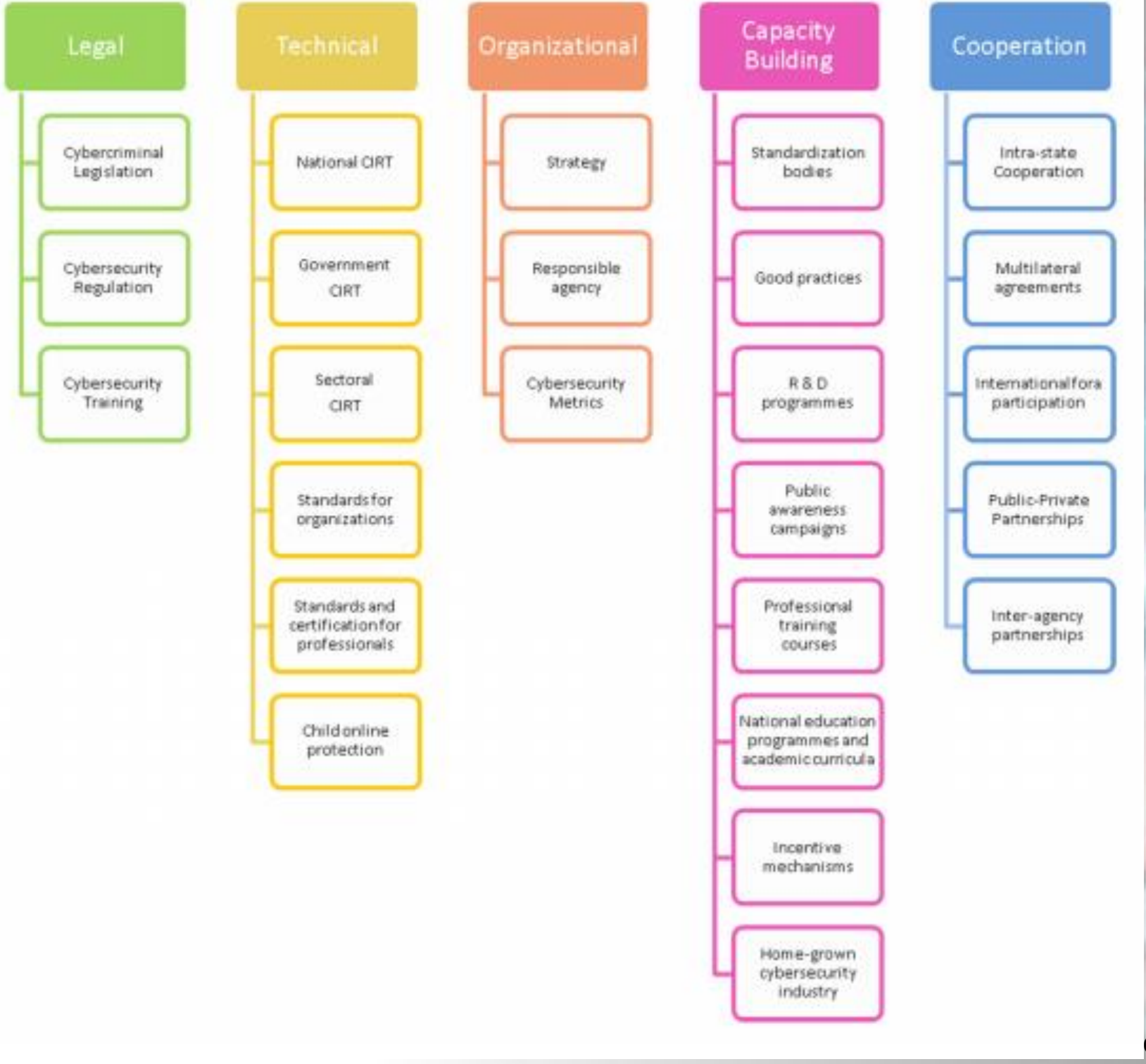


MSc. Ing. Guido Rosales

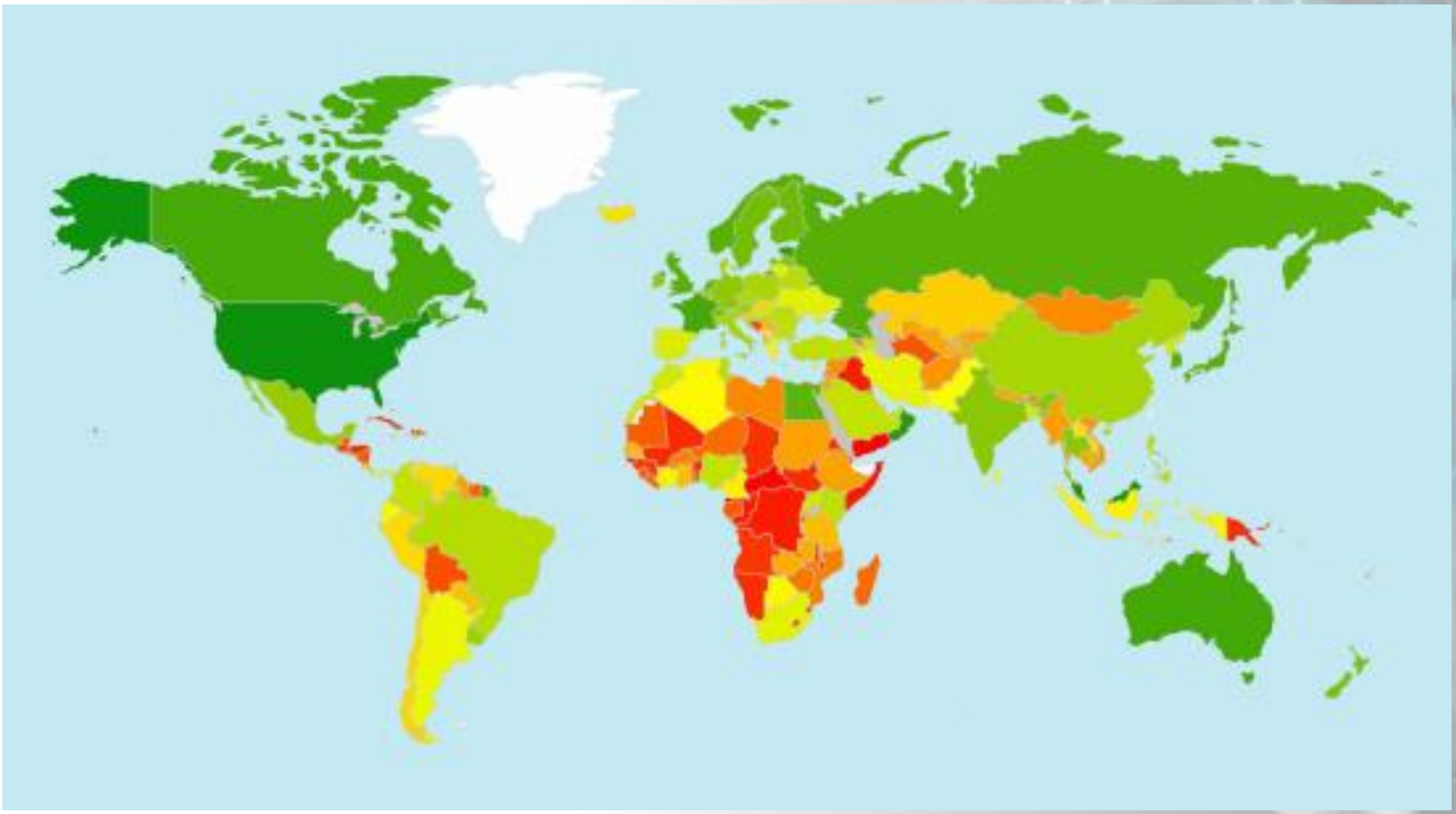
	VULNERABILIDADES - Se tienen 36 Vulnerabilidades Analizadas		
	Severidad	Debilidad	%
1	Alta	File unsafe Delete Check	76.9
2	Alta	SSL Implementation Check - SSL Certificate Verification	61.5
3	Alta	USING ACTIVITIES/IMPROPER EXPORT OF ANDROID APPLICATION ACTIVITIES	7.7
4	Alta	Certificate Pinning	0.0
5	Alta	Fragment Vulnerability Check	7.7
6	Media	WebView addJavascriptInterface Remote Code Execution	30.8
7	Media	Usage of Native codes	38.5
8	Media	Outputting Logs to logCat/ Logging Sensitive information	92.3
9	Media	Usage of Root/Superuser Permission	0.0
10	Media	Usage of Adb Backup	76.9
11	Media	SQLite Journal Information Disclosure Vulnerability	7.7
12	Media	Android PackageInfo Signature Verification / Android Fake ID Vulnerability	15.4
13	Media	Protecion of app screens by blurring when app is running in background	100.0
14	Media	Protection of capturing screenshots & sharing screens outside your app	100.0
15	Media	Protection of text fields from copying the text and paste outside your app	100.0
16	Baja	Access Mock Location	7.7
17	Baja	Usage of Installer verification code	92.3
18	Baja	Executing "root" or System Privilege Check	76.9
19	Baja	Emulator Detection Check	38.5
20	Baja	Unencrypted Credentials in Databases (sqlite db) Vulnerability check	23.1

NEGLIGENCIA vs DILIGENCIA

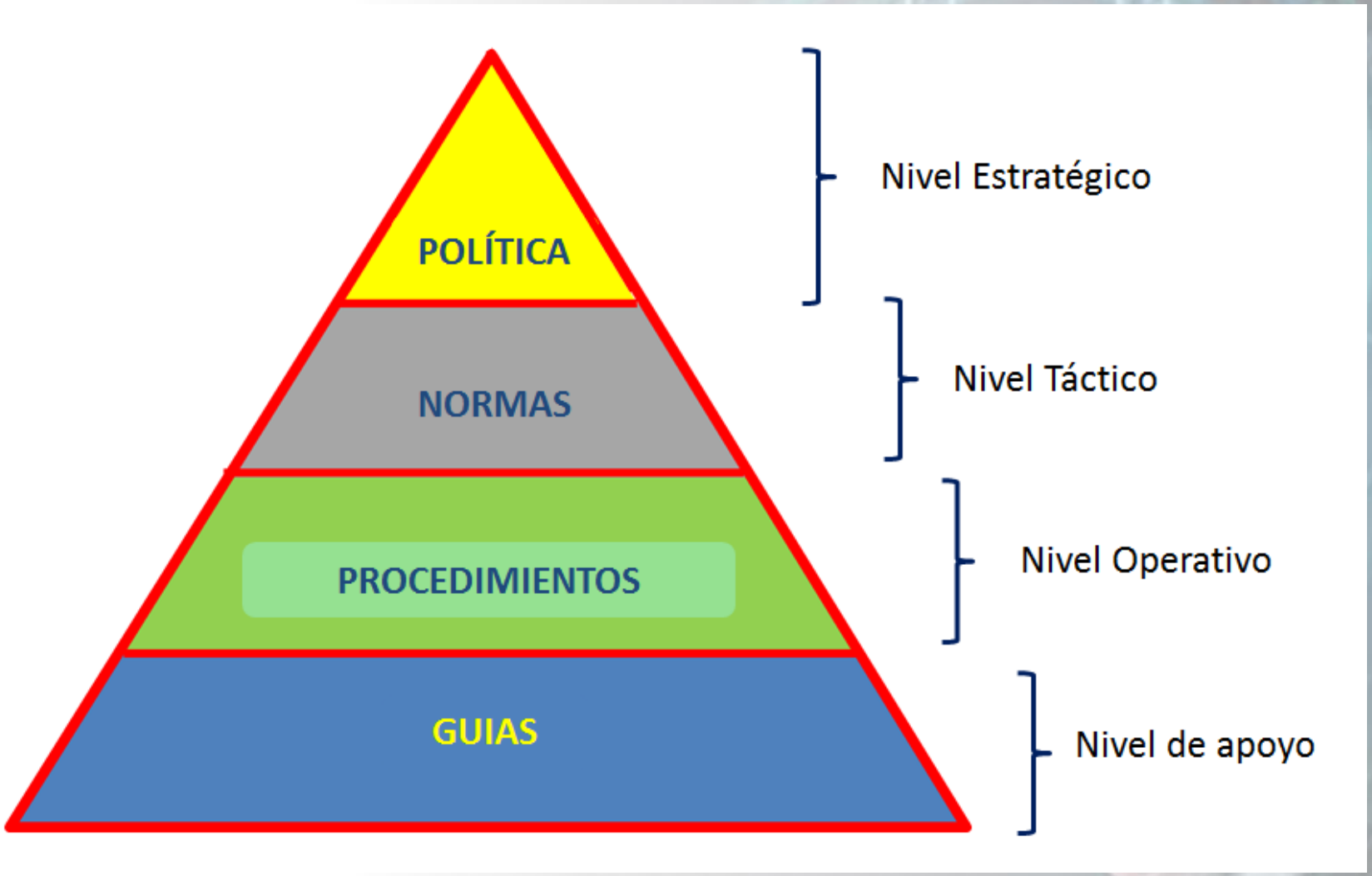
MSc. Ing. Guido Rosales


Legal
Cybercriminal Legislation
Cybersecurity Regulation
Cybersecurity Training
Technical
National CIRT
Government CIRT
Sectoral CIRT
Standards for organizations
Standards and certification for professionals
Child online protection
Organizational
Strategy
Responsible agency
Cybersecurity Metrics
Capacity Building
Standardization bodies
Good practices
R & D programmes
Public awareness campaigns
Professional training courses
National education programmes and academic curricula
Incentive mechanisms
Home-grown cybersecurity industry
Cooperation
Intra-state Cooperation
Multilateral agreements
International fora participation
Public-Private Partnerships
Inter-agency partnerships

osales



CUMPLIMIENTO



ALGUNOS EJEMPLOS

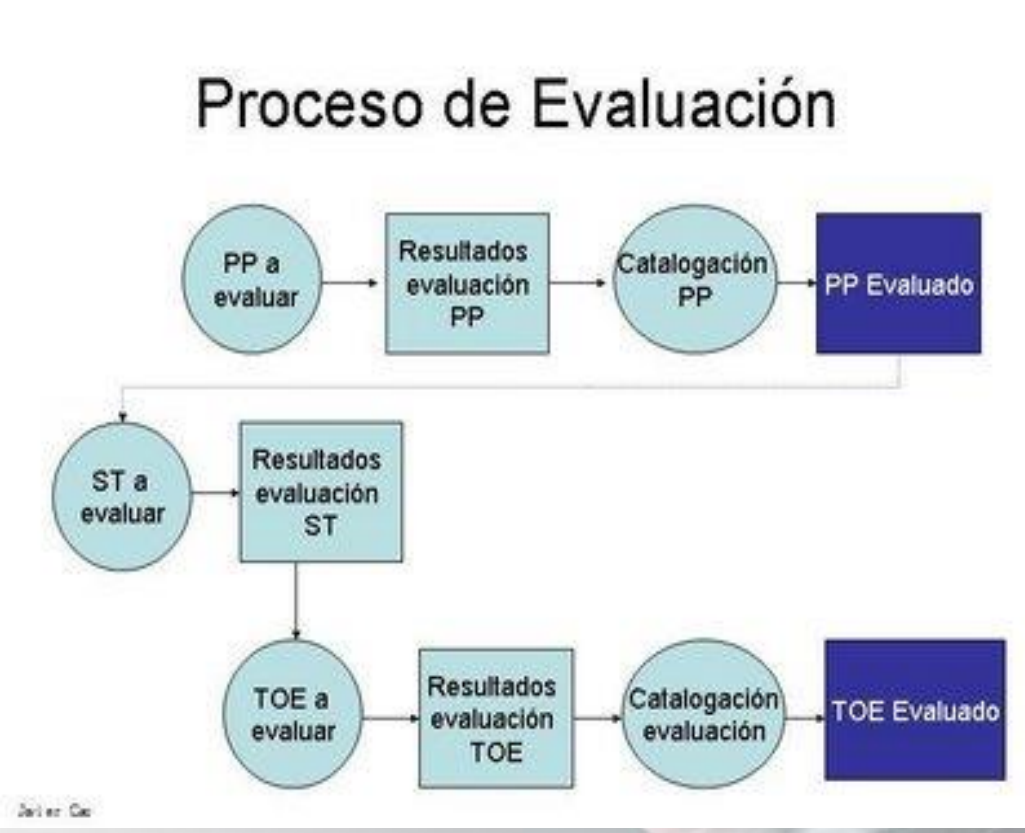
MSc. Ing. Guido Rosales

LIBRO NARANJA – SERIE ARCOÍRIS?

MSc. Ing. Guido Rosales

ISO 15408 - CC

- FAU- Auditoria
- FCO- Comunicaciones
- FCS- Soporte criptográfico
- FDP- Protección de datos de usuario
- FIA- Identificación y autenticación de usuario
- FMT- Gestión de la seguridad
- FPR- Privacidad
- FPT- Protección de las funciones de seguridad
- FRU- Utilización de recursos
- FTA- Acceso al objetivo de evaluación
  FTP- Canales seguros



1 Assurance levels
- 1.1 EAL1: Functionally Tested
- 1.2 EAL2: Structurally Tested
- 1.3 EAL3: Methodically Tested and Checked
- 1.4 EAL4: Methodically Designed, Tested and Reviewed
- 1.5 EAL5: Semiformally Designed and Tested
- 1.6 EAL6: Semiformally Verified Design and Tested
- 1.7 EAL7: Formally Verified Design and Tested

Gemalto's electronic Identity software offers governments an unprecedented level of security

SHARE THIS

Gemalto first and only company in the world to receive EAL7 certification for smart card embedded software

Amsterdam, May 14, 2013 - Gemalto (Euronext NL0000400653), the world leader in digital security, today announced that its Sealys ID Motion smart card software dedicated to national government programs has received "Common Criteria EAL7 certification". EAL7, or Evaluation Assurance Level 7, represents the highest level of security assurance within the international security evaluation scheme, also known as Common Criteria.

Windows 7 achieves Common Criteria security certification

Rate this article ★★★★★

sassoong April 28, 2011

Share 0 | 0 | 0 | 0

Microsoft endeavours to help make the computing environment as secure as possible. Part of this commitment to security involves thorough testing against widely-recognized security certification requirements.

We are pleased to advise that Windows 7, Windows Server 2008 R2 and SQL Server 2008 SP2 32 & 64 bit Enterprise Edition (English) have passed the Common Criteria (CC) certification process and achieved Evaluation Assurance Level 4 with augmentation (EAL4+). The Windows 7 and Windows Server 2008 R2 Common Criteria Evaluation and Validation Scheme Validation Report and Security Target are available for download.

Common Criteria certification is an international standard recognized by 26 member nations including New Zealand.

1 Security Target Introduction

This section presents the following information required for a Common Criteria (CC) evaluatio

- Identifies the Security Target (ST) and the Target of Evaluation (TOE)
- Specifies the security target conventions,
- Describes the organization of the security target

1.1 ST Reference

ST Title: Microsoft Windows 10 and Windows Server 2012 R2 Security Target

ST Version: version 1.0, March 17, 2016

1.2 TOE Reference

TOE Software Identification: The following Windows Operating Syst

- Microsoft Windows 10 Home Edition (32-bit and 64-bit vers
- Microsoft Windows 10 Pro Edition (32-bit and 64-bit versior
- Microsoft Windows 10 Enterprise Edition (32-bit and 64-bit
- Microsoft Windows Server 2012 R2 Standard Edition
- Microsoft Windows Server 2012 R2 Datacenter Edition

TOE Versions:

- Windows 10: build 10.0.10240
- Windows Server 2012 R2: build 6.3.9600

The following security updates and patches must be applied to the

- All critical updates as of October 31, 2015

https://technet.microsoft.com/en-us/library/dd229319.aspx

Common Criteria Security Targets

Information for Systems Integrators and Accreditors

The Security Target describes security functionality and assurance measures used to evaluate Windows.

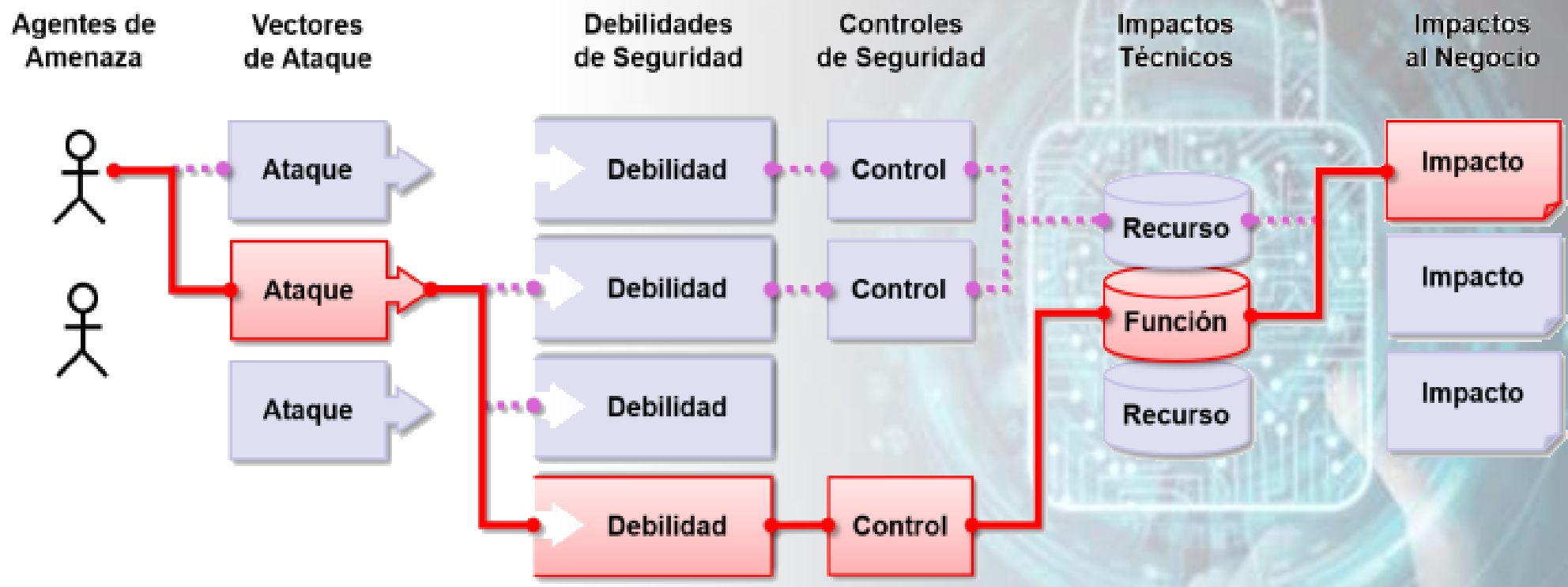
- Microsoft Windows 10 (Fall Creators Update)
- Microsoft Windows 10 (Creators Update)
- Microsoft Windows Server 2016, Microsoft Windows Server 2012 R2, and Microsoft Windows 10 Hyper-V
- Microsoft Windows 10 (Anniversary Update) and Windows 10 Mobile (Anniversary Update)
- Microsoft Windows 10 (Anniversary Update) and Windows Server 2016
- Windows 10 (Anniversary Update) and Windows Server 2016 IPsec VPN Client
- Microsoft Windows 10 IPsec VPN Client
- Microsoft Windows 10 November 2015 Update with Surface Book
- Microsoft Windows 10 Mobile with Lumia 950, 950 XL, 550, 635, and Windows 10 with Surface Pro 4
- Windows 10 and Windows Server 2012 R2
- Windows 10
- Windows 8.1 with Surface 3 and Windows Phone 8.1 with Lumia 635 and Lumia 830
- Microsoft Surface Pro 3 and Windows 8.1
- Windows 8.1 and Windows Phone 8.1
- Windows 8 and Windows Server 2012
- Windows 8 and Windows RT
- Windows 8 and Windows Server 2012 BitLocker
- Windows 8, Windows RT, and Windows Server 2012 IPsec VPN Client
- Windows 7 and Windows Server 2008 R2
- Microsoft Windows Server 2008 R2 Hyper-V Role
- Windows Vista and Windows Server 2008 at EAL4+
- Microsoft Windows Server 2008 Hyper-V Role
- Windows Vista and Windows Server 2008 at EAL1
- Windows Server 2003 SP2 including R2, x64, and IA64; Windows XP Professional SP2 and x64 SP2; and Windo

MSc. Ing. Guido Rosales

OWASP Top 10 - 2017

Los diez riesgos más críticos en Aplicaciones Web

ng. Guido Rosales



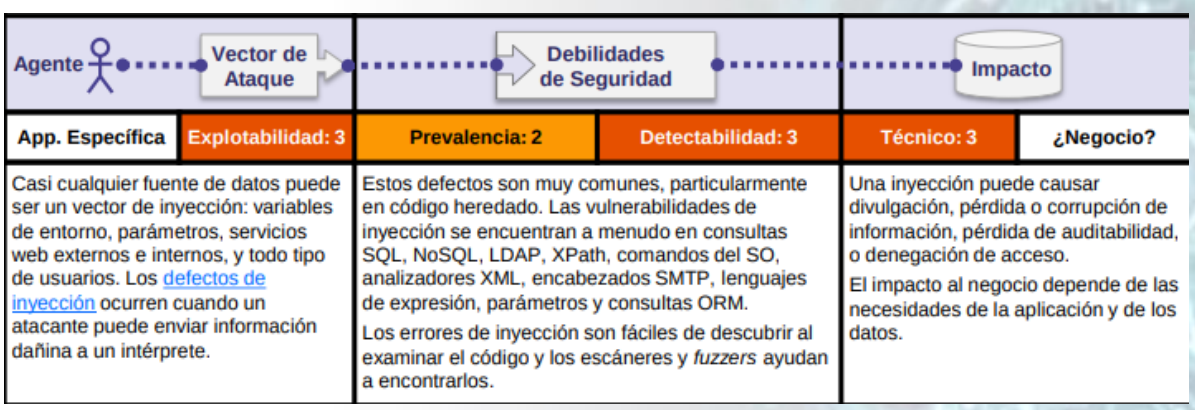
Agente de Amenaza	Explotabilidad	Prevalencia de Vulnerabilidad	Detección de Vulnerabilidad	Impacto Técnico	Impacto de Negocio
Específico de la Aplicación	Fácil 3	Difundido 3	Fácil 3	Severo 3	Específico del Negocio
	Promedio 2	Común 2	Promedio 2	Moderado 2	
	Difícil 1	Poco Común 1	Difícil 1	Mínimo 1	

OWASP Top 10 2013	±	OWASP Top 10 2017
A1 – Inyección	→	A1:2017 – Inyección
A2 – Pérdida de Autenticación y Gestión de Sesiones	→	A2:2017 – Pérdida de Autenticación y Gestión de Sesiones
A3 – Secuencia de Comandos en Sitios Cruzados (XSS)	↘	A3:2017 – Exposición de Datos Sensibles
A4 – Referencia Directa Insegura a Objetos [Unido+A7]	∪	A4:2017 – Entidad Externa de XML (XXE) [NUEVO]
A5 – Configuración de Seguridad Incorrecta	↘	A5:2017 – Pérdida de Control de Acceso [Unido]
A6 – Exposición de Datos Sensibles	↗	A6:2017 – Configuración de Seguridad Incorrecta
A7 – Ausencia de Control de Acceso a las Funciones [Unido+A4]	∪	A7:2017 – Secuencia de Comandos en Sitios Cruzados (XSS)
A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	☒	A8:2017 – Deserialización Insegura [NUEVO, Comunidad]
A9 – Uso de Componentes con Vulnerabilidades Conocidas	→	A9:2017 – Uso de Componentes con Vulnerabilidades Conocidas
A10 – Redirecciones y reenvíos no validados	☒	A10:2017 – Registro y Monitoreo Insuficientes [NUEVO, Comunidad]

A1:2017 **Inyección**	Las fallas de inyección, como SQL, NoSQL, OS o LDAP ocurren cuando se envían datos no confiables a un intérprete, como parte de un comando o consulta. Los datos dañinos del atacante pueden engañar al intérprete para que ejecute comandos involuntarios o acceda a los datos sin la debida autorización.
A2:2017 **Pérdida de Autenticación**	Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son implementadas incorrectamente, permitiendo a los atacantes comprometer usuarios y contraseñas, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios (temporal o permanentemente).
A3:201 **Exposición de datos sensibles**	Muchas aplicaciones web y APIs no protegen adecuadamente datos sensibles, tales como información financiera, de salud o Información Personalmente Identificable (PII). Los atacantes pueden robar o modificar estos datos protegidos inadecuadamente para llevar a cabo fraudes con tarjetas de crédito, robos de identidad u otros delitos. Los datos sensibles requieren métodos de protección adicionales, como el cifrado en almacenamiento y tránsito.
A4:2017 **Entidades Externas XML (XXE)**	Muchos procesadores XML antiguos o mal configurados evalúan referencias a entidades externas en documentos XML. Las entidades externas pueden utilizarse para revelar archivos internos mediante la URI o archivos internos en servidores no actualizados, escanear puertos de la LAN, ejecutar código de forma remota y realizar ataques de denegación de servicio (DoS).
A5:2017 **Pérdida de Control de Acceso**	Las restricciones sobre lo que los usuarios autenticados pueden hacer no se aplican correctamente. Los atacantes pueden explotar estos defectos para acceder, de forma no autorizada, a funcionalidades y/o datos, cuentas de otros usuarios, ver archivos sensibles, modificar datos, cambiar derechos de acceso y permisos, etc.

A1:2017
Inyección

Las fallas de inyección, como SQL, NoSQL, OS o LDAP ocurren cuando se envían datos no confiables a un intérprete, como parte de un comando o consulta. Los datos dañinos del atacante pueden engañar al intérprete para que ejecute comandos involuntarios o acceda a los datos sin la debida autorización.



¿La aplicación es vulnerable?

Una aplicación es vulnerable a ataques de este tipo cuando:

- Los datos suministrados por el usuario no son validados, filtrados o sanitizados por la aplicación.
- Se invocan consultas dinámicas o no parametrizadas, sin codificar los parámetros de forma acorde al contexto.
- Se utilizan datos dañinos dentro de los parámetros de búsqueda en consultas *Object-Relational Mapping (ORM)*, para extraer registros adicionales sensibles.
- Los datos dañinos se usan directamente o se concatenan, de modo que el SQL o comando resultante contiene datos y estructuras con consultas dinámicas, comandos o procedimientos almacenados.

Algunas de las inyecciones más comunes son SQL, NoSQL, comandos de SO, *Object-Relational Mapping (ORM)*, LDAP, expresiones de lenguaje u *Object Graph Navigation Library (OGNL)*. El concepto es idéntico entre todos los intérpretes. La revisión del código fuente es el mejor método para detectar si las aplicaciones son vulnerables a inyecciones, seguido de cerca por pruebas automatizadas de todos los parámetros, encabezados, URL, *cookies*, JSON, SOAP y entradas de datos XML.

Las organizaciones pueden incluir herramientas de análisis estático (SAST) y pruebas dinámicas (DAST) para identificar errores de inyecciones recientemente introducidas y antes del despliegue de la aplicación en producción.

Cómo se previene

Para prevenir inyecciones, se requiere separar los datos de los comandos y las consultas.

- La opción preferida es utilizar una API segura, que evite el uso de un intérprete por completo y proporcione una interfaz parametrizada. Se debe migrar y utilizar una herramientas de Mapeo Relacional de Objetos (ORMs).
  Nota: Incluso cuando se parametrizan, los procedimientos almacenados pueden introducir una inyección SQL si el procedimiento PL/SQL o T-SQL concatena consultas y datos, o se ejecutan parámetros utilizando *EXECUTE IMMEDIATE* o *exec()*.
- Realice validaciones de entradas de datos en el servidor, utilizando "listas blancas". De todos modos, esto no es una defensa completa ya que muchas aplicaciones requieren el uso de caracteres especiales, como en campos de texto, APIs o aplicaciones móviles.
- Para cualquier consulta dinámica residual, escape caracteres especiales utilizando la sintaxis de caracteres específica para el intérprete que se trate.
  Nota: La estructura de SQL como nombres de tabla, nombres de columna, etc. no se pueden escapar y, por lo tanto, los nombres de estructura suministrados por el usuario son peligrosos. Este es un problema común en el software de redacción de informes.
- Utilice LIMIT y otros controles SQL dentro de las consultas para evitar la fuga masiva de registros en caso de inyección SQL.

Ejemplos de escenarios de ataque

Escenario #1: la aplicación utiliza datos no confiables en la construcción del siguiente comando SQL vulnerable:

```
String query = "SELECT * FROM accounts WHERE custID='" + request.getParameter("id") + "'";
```

Escenario #2: la confianza total de una aplicación en su *framework* puede resultar en consultas que aún son vulnerables a inyección, por ejemplo, *Hibernate Query Language (HQL)*:

```
Query HQLQuery = session.createQuery("FROM accounts WHERE custID='" + request.getParameter("id") + "'");
```

En ambos casos, al atacante puede modificar el parámetro *"id"* en su navegador para enviar: ' or '1'='1. Por ejemplo:

```
http://example.com/app/accountView?id=' or '1'='1
```

Esto cambia el significado de ambas consultas, devolviendo todos los registros de la tabla *"accounts"*. Ataques más peligrosos podrían modificar los datos o incluso invocar procedimientos almacenados.

Referencias

OWASP

- OWASP Proactive Controls: Parameterize Queries
- OWASP ASVS: V5 Input Validation and Encoding
- OWASP Testing Guide: SQL Injection, Command Injection, ORM injection
- OWASP Cheat Sheet: Injection Prevention
- OWASP Cheat Sheet: SQL Injection Prevention
- OWASP Cheat Sheet: Injection Prevention in Java
- OWASP Cheat Sheet: Query Parameterization
- OWASP Automated Threats to Web Applications – OAT-014

Externos

- CWE-77: Command Injection
- CWE-89: SQL Injection
- CWE-564: Hibernate Injection
- CWE-917: Expression Language Injection
- PortSwigger: Server-side template injection

Category:OWASP Application Security Verification Standard Project

OWASP ASVS 3.1 (early access)

- ASVS V1 Architecture
- ASVS V2 Authentication
- ASVS V3 Session Management
- ASVS V4 Access Control
- ASVS V5 Input validation and output encoding
- ASVS V7 Cryptography
- ASVS V8 Error Handling
- ASVS V9 Data Protection
- ASVS V10 Communications
- ASVS V13 Malicious Code
- ASVS V15 Business Logic Flaws
- ASVS V16 Files and Resources
- ASVS V17 Mobile
- ASVS V18 API
- ASVS V19 Configuration
- ASVS V20 Internet of Things

OWASP Open Web Application Security Project | Testing Guide 4.0

)release(

3

OWASP
Open Web Application Security Project

Standard 0.9

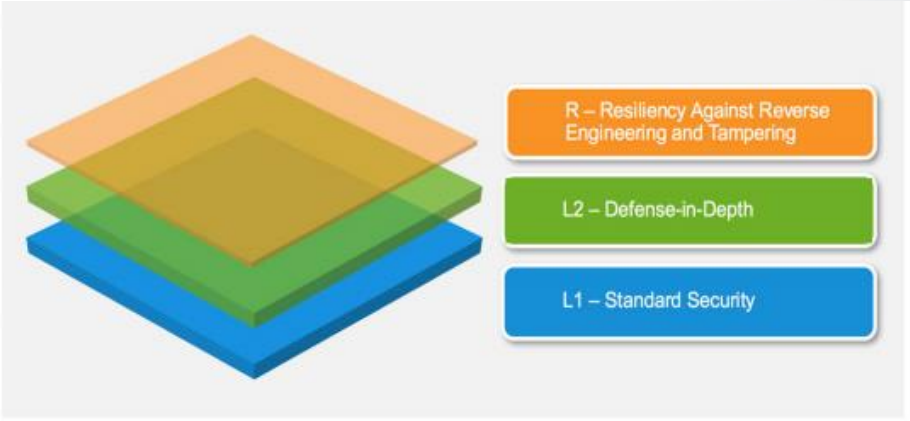
pre-release

Mobile AppSec Verification

- M1 - Improper Platform Usage
- M2 - Insecure Data Storage
- M3 - Insecure Communication
- M4 - Insecure Authentication
- M5 - Insufficient Cryptography

- M6 - Insecure Authorization
- M7 - Client Code Quality
- M8 - Code Tampering
- M9 - Reverse Engineering
- M10 - Extraneous Functionality

MSc. Ing. Guido Rosales



#	Description	L1	L2
1.1	All app components are identified and known to be needed.	✓	✓
1.2	Security controls are never enforced only on the client side, but on the respective remote endpoints.	✓	✓
1.3	A high-level architecture for the mobile app and all connected remote services has been defined and security has been addressed in that architecture.	✓	✓
1.4	Data considered sensitive in the context of the mobile app is clearly identified.	✓	✓
1.5	All app components are defined in terms of the business functions and/or security functions they provide.		✓
1.6	A threat model for the mobile app and the associated remote services has been produced that identifies potential threats and countermeasures.		✓
1.7	All security controls have a centralized implementation.		✓
1.8	There is an explicit policy for how cryptographic keys (if any) are managed, and the lifecycle of cryptographic keys is enforced. Ideally, follow a key management standard such as NIST SP 800-57.		✓
1.9	A mechanism for enforcing updates of the mobile app exists.		✓
1.10	Security is addressed within all parts of the software development lifecycle.		✓

VULNERABILIDADES - Se tienen 34 Vulnerabilidades Analizadas		
Severidad	Debilidad	%
Alta	File unsafe Delete Check	55
Alta	SSL Implementation Check - SSL Certificate Verification	55
Alta	Certificate Pinning	27
Alta	Using activities/improper export of android application activities	9
Alta	Fragment Vulnerability Check	9
Media	Usage of Adb Backup	91
Media	Usage of Native codes	82
Media	Outputting Logs to logCat/ Logging Sensitive information	82
Media	SQLite Journal Information Disclosure Vulnerability	36
Media	WebView addJavascriptInterface Remote Code Execution	18
Media	Usage of Root/Superuser Permission	9
Baja	Usage of Installer verification code	91
Baja	Executing "root" or System Privilege Check	91
Baja	Emulator Detection Check	73
Baja	Unencrypted Credentials in Databases (sqlite db) Vulnerability check	18
Baja	Access Mock Location	9

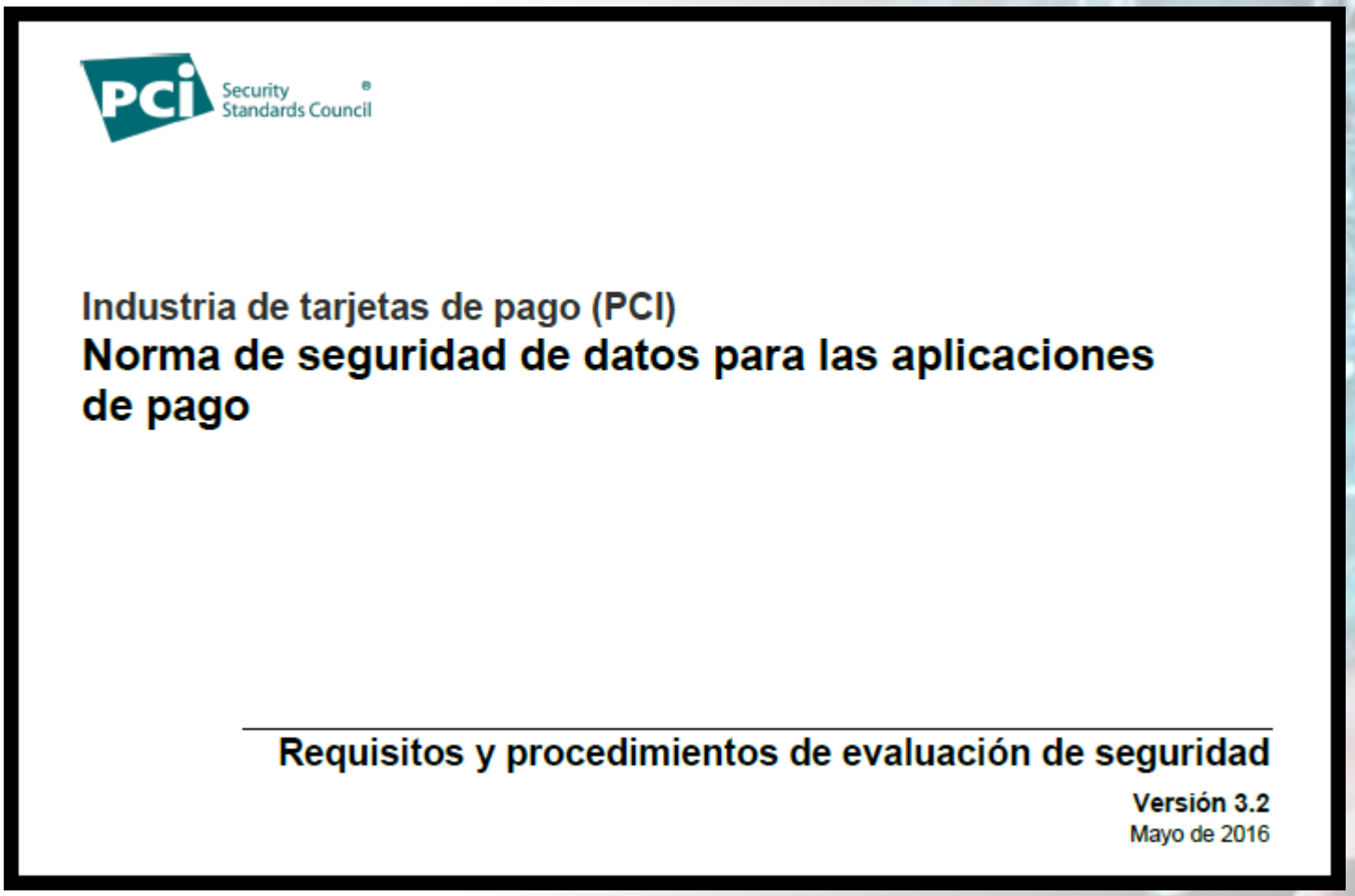
ISO/IEC 27034-1

Se basa en los siguientes principios fundamentales:

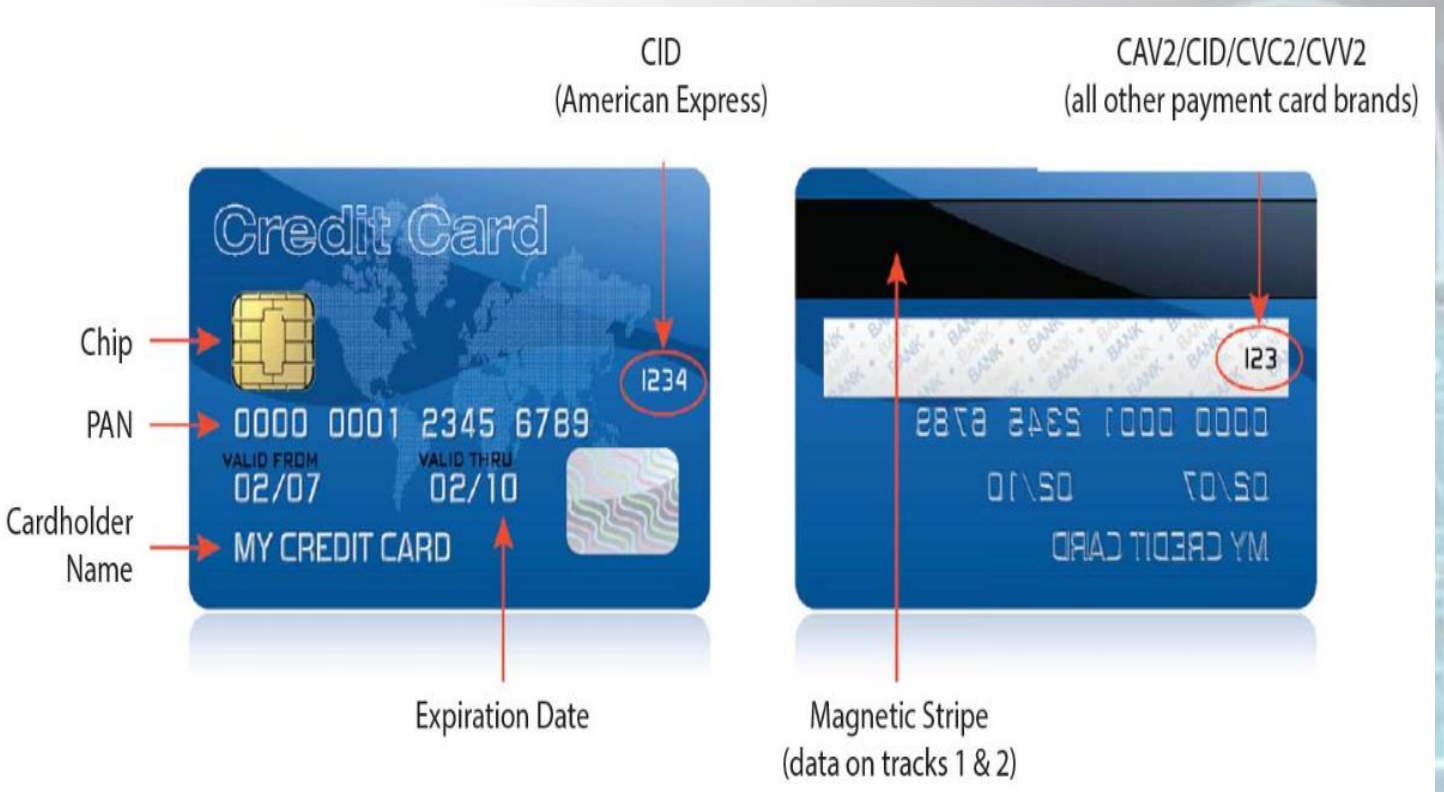
- La seguridad es un requisito
- La seguridad de las aplicaciones depende del contexto
- Inversión apropiada para aplicaciones de seguridad
- La seguridad en las aplicaciones debe ser demostrada

PCI DSS o PA DSS

MSc. Ing. Guido Rosales



MSc. Ing. Guido Rosales

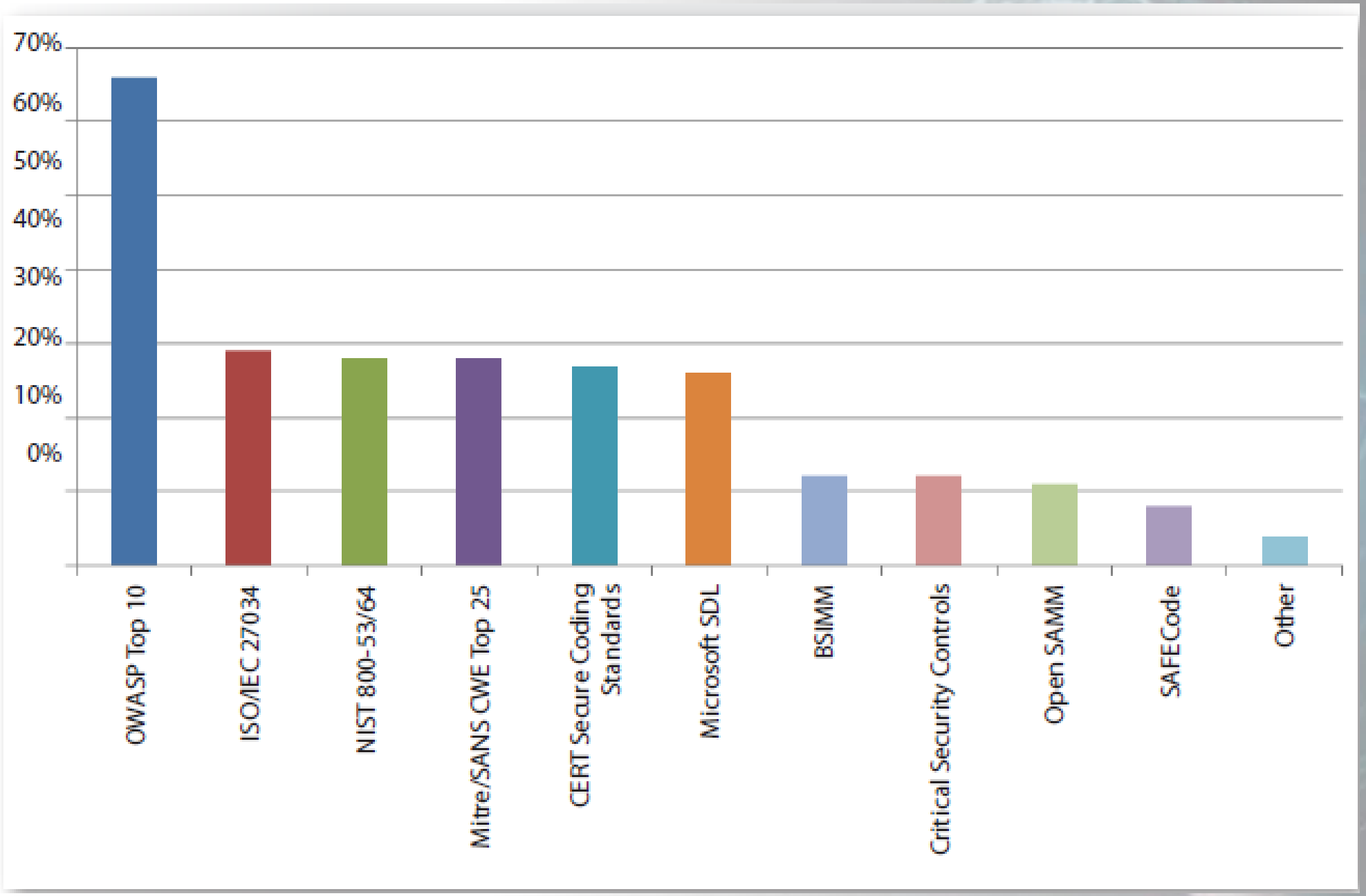


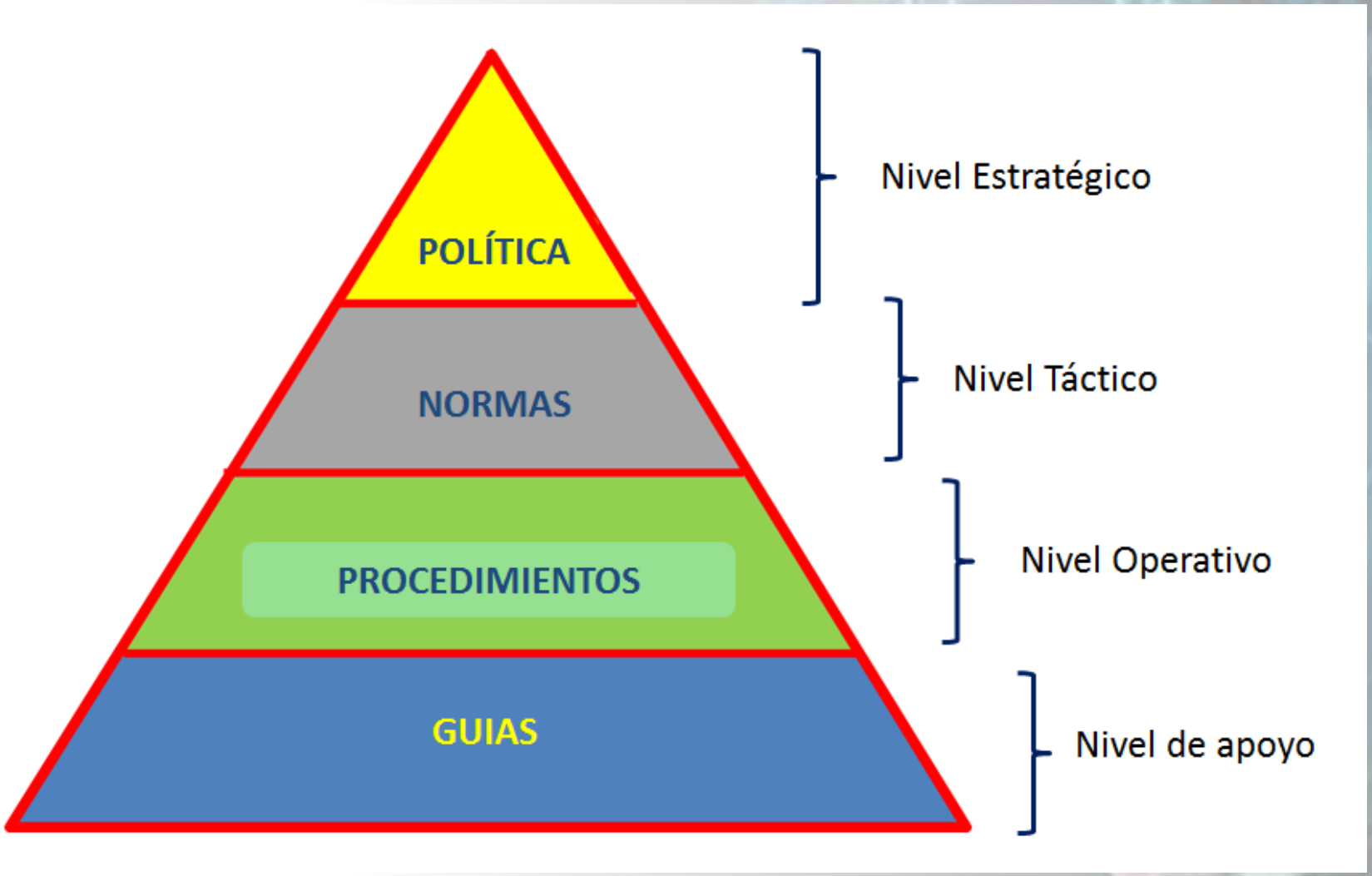
		Elemento de datos	Almacenamiento permitido	Datos almacenados ilegibles según el Requisito 2.3 de las PA-DSS
Datos de cuentas	Datos del titular de la tarjeta	Número de cuenta principal (PAN)	Sí	Sí
		Nombre del titular de la tarjeta	Sí	No
		Código de servicio	Sí	No
		Fecha de vencimiento	Sí	No
	Datos confidenciales de autenticación[1]	Contenido completo de la pista [2]	No	No se puede almacenar según el Requisito 1.1 de las PA-DSS.
		CAV2/CVC2/CVV2/CID[3]	No	No se puede almacenar según el Requisito 1.1 de las PA-DSS.
		PIN/Bloqueo de PIN[4]	No	No se puede almacenar según el Requisito 1.1 de las PA-DSS.

CONCLUSIONES

MSc. Ing. Guido Rosales

MSc. Ing. Guido Rosales


70%
60%
50%
40%
30%
20%
10%
0%
OWASP Top 10
ISO/IEC 27034
NIST 800-53/64
Mitre/SANS CWE Top 25
CERT Secure Coding Standards
Microsoft SDL
BSIMM
Critical Security Controls
Open SAMM
SAFECode
Other



MSc. Ing. Guido Rosales

MSc. Ing. Guido Rosales

TECNOLÓGICO SUPERIOR CORDILLERA

25 AÑOS

CIDE
Centro de Investigación y Desarrollo Ecuador

SEMINARIO INTERNACIONAL DE SEGURIDAD EN EL DESARROLLO DE SOFTWARE

WWW.CIDECUADOR.COM

Una vez finalizado el evento esta presentación sera publicada en su respectiva página web