Seguridad en Sistemas Abiertos

1. Seguridad en sistemas abiertos

1.1. Requisitos generales de seguridad

¿Qué requisitos de seguridad son deseables en un sistema software?

Confidencialidad

Asegurar que sólo los usuarios autorizados pueden leer los datos gestionados por el sistema.

Integridad

Asegurar que sólo los usuarios autorizados pueden modificar los recursos del sistema software.

Disponibilidad

Asegurar que sólo los usuarios autorizados pueden usar el sistema software.

1.2. La seguridad como requisito no funcional

Un **requisito no funcional** (NFR) especifica criterios que pueden usarse para valorar la **operación** de un sistema en lugar de sus comportamientos (funcionales) específicos. Los NFR se refieren a todos los requisitos que no describen qué información guardar, ni qué funciones realizar, sino características de **cómo** éstas se ejecutan en el funcionamiento del sistema.

NFR tomados de Wikipedia

- Accessibility
- Auditability and control
- **Availability (SLA)**
- **Backup**
- Capacity, current and forecast
- Certification
- Compliance
- Configuration management
- Cost
- Dependency on other parties
- Deployment
- Documentation
- **Disaster recovery**
- Efficiency
- Effectiveness
- Environmental protection
- **Escrow (trust)**
- Exploitability
- Extensibility
- **Failure management**
- **Fault tolerance**
- **Legal and IPR issues**
- Interoperability
- Maintainability
- Management
- Modifiability
- Network topology
- Open source
- **Operability**
- Performance / response time
- Platform compatibility
- **Privacy (laws)**
- Portability
- **Quality (wrt. faults)**
- Readability
- **Reliability**
- Reporting
- Resilience
- Resource constraints
- Response time
- Reusability
- **Robustness**
- **Safety**
- Scalability
- **Security (cyber and physical)**
- Compatibility (software, tools, standards, etc.)
- Stability
- Supportability
- Testability
- Throughput
- Transparency
- Usability (Human Factors)
- Integrability

NFR relacionados con la seguridad

NFR relacionados directa o indirectamente con la seguridad:

Availability (SLA)

The degree to which a system, subsystem or equipment is in a specified operable and committable state at the start of a mission, when the mission is called for at an unknown, i.e. a random, time

Backup

Copying and archiving of computer data so it may be used to restore the original after a data loss event

Disaster recovery

Policies, tools and procedures to enable the recovery or continuation of vital technology

infrastructure and systems following a natural or human-induced disaster

Escrow (trust)

Deposit of the source code of software with a third party escrow agent. Escrow is typically requested by a party licensing software (the licensee), to ensure maintenance of the software instead of abandonment or orphaning

Failure management

Fault tolerance

Property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components

Legal and IPR issues

Operability

Ability to keep an equipment, a system or a whole industrial installation in a safe and reliable functioning condition, according to pre-defined operational requirements

Privacy (laws)

Ability of an individual or group to seclude themselves, or information about themselves, and thereby express themselves selectively

Quality (wrt. faults)

Faults discovery, faults delivery, fault removal efficacy

Reliability

Dependability, or reliability, describes the ability of a system or component to function under stated conditions for a specified period of time

Robustness

Ability of a computer system to cope with errors during execution and cope with erroneous input

Safety

State of being "safe", the condition of being protected from harm or other non-desirable outcomes

1.3. Protección

¿Cómo aplicar los NFR de seguridad a un sistema?

1. Añadir características especiales de seguridad para proteger al sistema software "en bruto"
2. Aplicar técnicas de programación para diseñar y codificar sistemas software preparados para aguantar (más que prevenir) ataques.

La protección se logra mediante la adición de características como el **control de acceso** y las

técnicas de **codificación segura** que dificulten los ataques.

Ejemplo 1. Control de acceso vs codificación segura

Supongamos que un servidor web tiene un fallo por buffer overflow, y que se quiere impedir un ataque remoto por buffer overflow que envía al servidor una petición HTTP GET mayor de lo debido.

- Una forma de impedirlo es añadir una característica software al servidor, una función que vigile las peticiones HTTP que llegan por el puerto 80 y elimine las que son mayores de un tamaño predeterminado.
- Otra forma de conseguir el mismo resultado es arreglar el código fuente del servidor para eliminar la posibilidad de fallo por buffer overflow. Esta segunda forma solo es posible si se dispone de acceso al código fuente del servidor y se comprende bien el funcionamiento del mismo

1.4. Características de seguridad en sistemas abiertos

Autenticación

Identificación de entidades participantes en una comunicación y del origen de los datos.

Control de acceso

Protección contra el uso no autorizado de los recursos accesibles; aplicable a varios tipo de acceso a recursos (lectura, escritura, borrado, ejecución, etc.)

Confidencialidad de datos

Protección contra la revelación no autorizada de los recursos

Integridad de datos

Protección contra la modificación, inserción, borrado o repetición de datos

No repudio

proporcionar al receptor una prueba del origen de los datos, para Proteger contra intentos del emisor de denegar en falso haberlos enviado;
Proporcionar al emisor una prueba de entrega de los datos, para proteger contra cualquier intento del receptor de denegar en falso haberlos recibido.

2. Control de acceso

El control de acceso es la capacidad de permitir o denegar el uso de recursos. Decide qué **sujetos** están autorizados (y quiénes no) a realizar ciertas **operaciones** en un determinado **objeto**.

2.1. Factores

Para determinar si se tiene acceso a un objeto:

- algo que *tienes*: v.g. una credencial
- algo que *sabes*: v.g. un PIN, una contraseña
- algo que *eres*: v.g. una huella dactilar, ocular o rasgo biométrico

2.2. Listas de Control de Acceso

Las **Access Control Lists** (ACLs) son estructuras de datos que contienen las **autorizaciones** empleadas para implementar el control de acceso.

Una **autorización** determina qué acciones puede ejercer un sujeto sobre el sistema. La semántica de las acciones depende de la naturaleza de los objetos involucrados.

2.3. Tipos de control de acceso

- *Discretionary Access Control* (DAC)
- *Mandatory Access Control* (MAC)
- *Role-Based Access Control* (RBAC)

Control de Acceso Discrecional (DAC)

> A means of restricting access to objects based on the **identity** of subjects and/or groups to which they belong. DAC is discretionary in the sense that a subject with a certain access permission is **capable of passing on** that permission (perhaps indirectly) to any other subject (unless explicitly restrained from doing so by MAC)
>
> — TCSEC, Orange Book

Un usuario individual (o un programa que opera en su nombre) puede especificar explícitamente los tipos de acceso que otros usuarios (o programas que ejecutan en su nombre) pueden tener a los recursos bajo el control del usuarios (recursos que *posee*)

Control de Acceso Obligatorio (MAC)

> A means of restricting access to objects based on the **sensitivity** (as represented by a label) of the information contained in the objects and the formal authorization (i.e., **clearance**) of subjects to access information of such sensitivity.
>
> — TCSEC, Orange Book

Los modelos MAC fuerzan un control de acceso basado en reglas definidas por una autoridad central. Para que un sujeto sea capaz de acceder a un objeto, su **nivel de sensibilidad** debe ser al menos igual o mayor que el nivel de sensibilidad del objeto.

Modelo Bell-LaPadula (BLP)

BLP es un modelo MAC que particiona el dominio organizativo en **niveles** de seguridad y asigna una **etiqueta** de seguridad a cada nivel. Cada sujeto y cada objeto del sistema tiene que ser asociado con uno de los niveles predefinidos y con una etiqueta de seguridad.

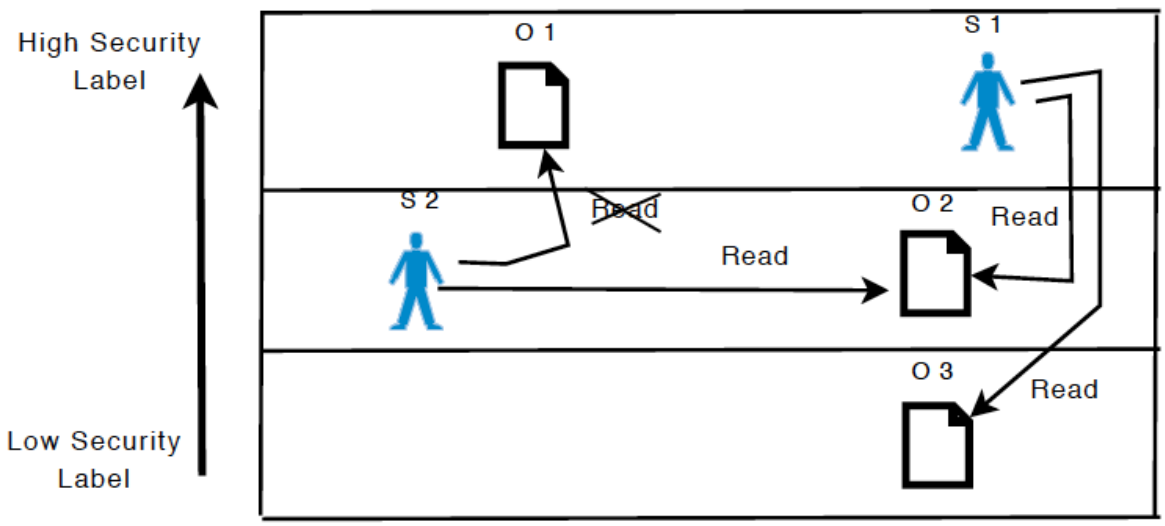
Los nombres de las etiquetas de seguridad pueden cambiar de una organización a otra. Por ejemplo:

- Militares: Top secret, Secret, Classified
- Negocios: Board Only, Managerial, etc.

Propiedades BLP

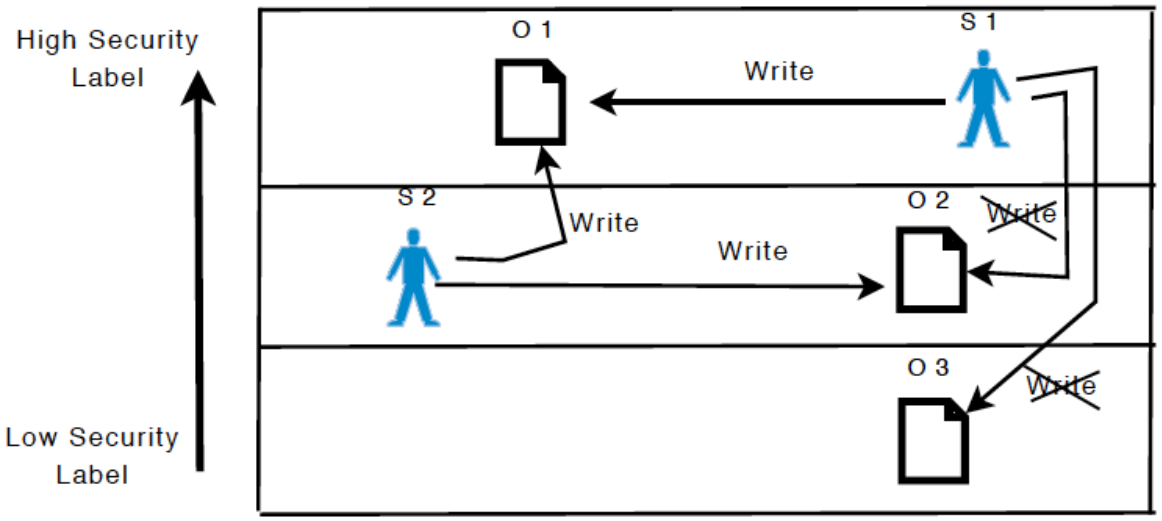
- **Seguridad Simple**: Un sujeto solo puede acceder para lectura a objetos de menor o igual nivel de seguridad (no *read-up*)
- **Seguridad-***: Un sujeto solo puede tener acceso de escritura a objetos de mayor o igual nivel de seguridad (no *write-down*)

Ejemplo 2. Seguridad simple BLP



El sujeto S1 puede leer los objetos O2 y O3 porque pertenecen a un nivel menor.
El sujeto S2 puede leer el objeto O2 porque pertenece al mismo nivel.
El sujeto S2 no está autorizado a leer el objeto O1 porque pertenece a un nivel mayor

Ejemplo 3. Seguridad- BLP*

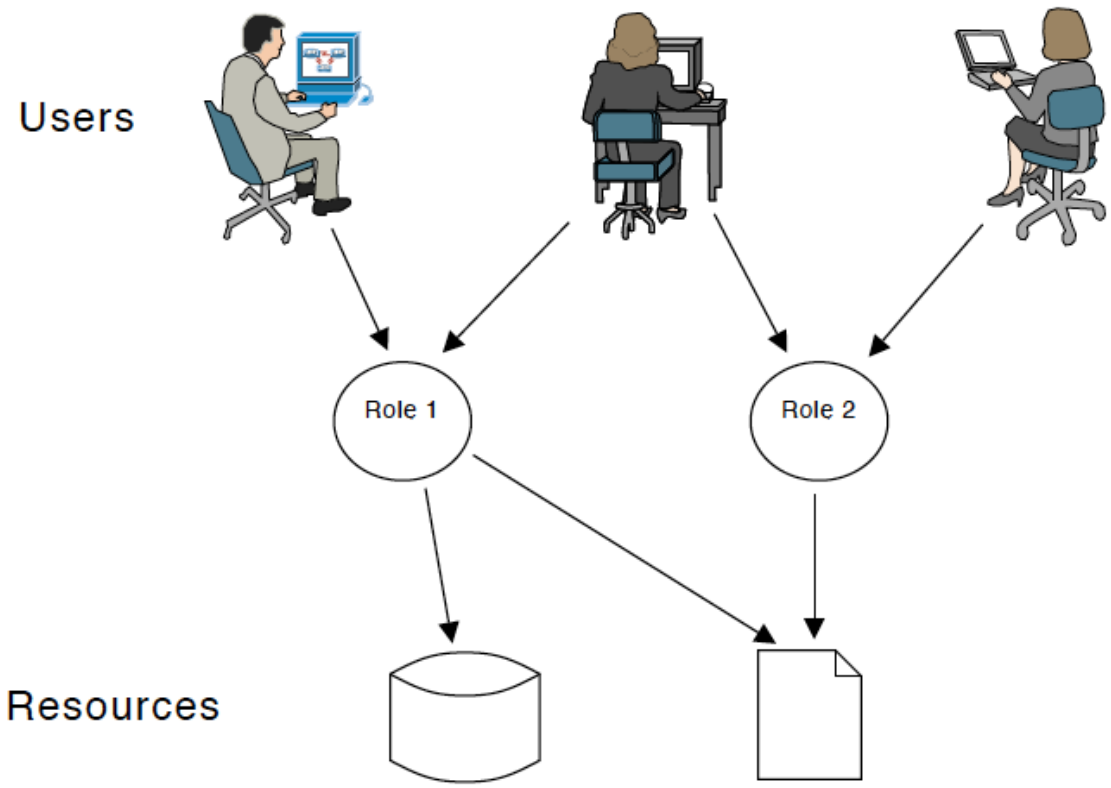


El sujeto S2 puede escribir en el objeto O1 porque pertenece a un nivel mayor de seguridad.
El sujeto S2 puede escribir en el objeto O2 porque pertenece al mismo nivel de seguridad.
El sujeto S1 solo puede escribir en el objeto O1 porque pertenece al mismo nivel de seguridad

Control de Acceso Basado en Roles (RBAC)

Un **rol** es un conjunto de acciones y responsabilidades asociadas con una actividad particular.

La relación entre roles e individuos y entre roles y recursos es *n:m*. Un rol específico puede incluir uno o más individuos y, al mismo tiempo, proporcionar acceso a uno o más recursos.



De cara a la protección de datos, la granularidad de las autorizaciones RBAC permite construir roles de forma que se aplique el **principio de separación de responsabilidades**, donde ningún individuo pueda subvertir una función.

Ejemplo 4. Ley Sarbanes-Oxley (SOX)

La ley estadounidense de "reforma de la contabilidad pública empresarial y de protección de los inversores" (conocida como *Sarbanes-Oxley Act*) fue promulgada tras los escándalos de Enron a principios de 2000.

La ley define estándares para los informes financieros (incluyendo la necesidad de preservar los registros de datos) obligatorios para las compañías mercantiles (que coticen en bolsa). Los **gestores tienen la responsabilidad** de "establecer y mantener una estructura interna de control y procedimientos de información financiera adecuados".

Esta regulación requiere a la empresa saber qué información tiene y dónde está ubicada, siguiendo objetivos de **preservación** y **disponibilidad** (protección) de datos.

RBAC es útil en situaciones de cumplimiento de regulaciones como la SOX Act, donde nadie, ya sea intencionadamente o no, debe ser capaz de alterar la información de forma inapropiada.

3. Datos abiertos

Introducción a OpenData

3.1. Open Data

Fuentes

- Repositorios OpenData: Socrata OpenData Network, etc.
- Linked Open Data: DBPedia, etc.
- Específicos de seguridad: AlienVault OSSIM es un sistema SIEM (*Security Information and Event Management*) open source que publica sus datos en abierto

La base de datos de reputación de IPs de AlienVault publica periódicamente y en varios formatos algunos datasets con información sobre varios tipos de "maldades" en internet.
Por ejemplo, en reputation.data pubica un fichero CSV con una clasificación de diversos aspectos de reputación de una lista de IPs.
Además, cada hora genera un fichero de revisión de los datos de reputación para poder trabajar con los últimos datos publicados.

Ejemplo 5. Extracto de registros de `reputation.data` de AlienVault OSSIM

```
<table border="1" class="dataframe">
  <thead>
    <tr style="text-align: right;">
      <th></th>
      <th>IP</th>
      <th>Reliability</th>
      <th>Risk</th>
      <th>Type</th>
      <th>Country</th>
      <th>Locale</th>
      <th>Coords</th>
      <th>x</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>0</th>
      <td>222.76.212.185</td>
      <td>4</td>
      <td>2</td>
      <td>Scanning Host</td>
      <td>CN</td>
      <td>Xiamen</td>
      <td>24.4797992706,118.08190155</td>
      <td>11</td>
    </tr>
    <tr>
      <th>1</th>
      <td>222.76.212.186</td>
      <td>4</td>
      <td>2</td>
      <td>Scanning Host</td>
      <td>CN</td>
      <td>Xiamen</td>
      <td>24.4797992706,118.08190155</td>
      <td>11</td>
    </tr>
    <tr>
      <th>2</th>
      <td>5.34.246.67</td>
      <td>6</td>
      <td>3</td>
      <td>Spamming</td>
      <td>US</td>
      <td>NaN</td>
      <td>38.0,-97.0</td>
      <td>12</td>
    </tr>
    <tr>
```

```
    <th>3</th>
    <td>178.94.97.176</td>
    <td>4</td>
    <td>5</td>
    <td>Scanning Host</td>
    <td>UA</td>
    <td>Merefa</td>
    <td>49.8230018616,36.0507011414</td>
    <td>11</td>
  </tr>
  <tr>
    <th>4</th>
    <td>66.2.49.232</td>
    <td>4</td>
    <td>2</td>
    <td>Scanning Host</td>
    <td>US</td>
    <td>Union City</td>
    <td>37.5962982178,-122.065696716</td>
    <td>11</td>
  </tr>
  <tr>
    <th>5</th>
    <td>222.76.212.173</td>
    <td>4</td>
    <td>2</td>
    <td>Scanning Host</td>
    <td>CN</td>
    <td>Xiamen</td>
    <td>24.4797992706,118.08190155</td>
    <td>11</td>
  </tr>
  <tr>
    <th>6</th>
    <td>222.76.212.172</td>
    <td>4</td>
    <td>2</td>
    <td>Scanning Host</td>
    <td>CN</td>
    <td>Xiamen</td>
    <td>24.4797992706,118.08190155</td>
    <td>11</td>
  </tr>
  <tr>
    <th>7</th>
    <td>222.76.212.171</td>
    <td>4</td>
    <td>2</td>
    <td>Scanning Host</td>
    <td>CN</td>
    <td>Xiamen</td>
```

```
      <td>24.4797992706,118.08190155</td>
      <td>11</td>
    </tr>
    <tr>
      <th>8</th>
      <td>174.142.46.19</td>
      <td>6</td>
      <td>3</td>
      <td>Spamming</td>
      <td>NaN</td>
      <td>NaN</td>
      <td>24.4797992706,118.08190155</td>
      <td>12</td>
    </tr>
    <tr>
      <th>9</th>
      <td>66.2.49.244</td>
      <td>4</td>
      <td>2</td>
      <td>Scanning Host</td>
      <td>US</td>
      <td>Union City</td>
      <td>37.5962982178,-122.065696716</td>
      <td>11</td>
    </tr>
  </tbody>
</table>
```

Formatos

- Abiertos (CSV, OpenDocument XML) vs. Privativos (XSLX)
- Estructurados (JSON, XML, RDF) vs. No estructurados (PDF) — Mixtos: PDF+XML
- Dataframes: R, Python pandas
- Web scraping: HTML5, RDFa, microformats
- Databases: SQL, SPARQL, DSPL
- Data protocols & APIs: Socrata SODA, Microsoft OData, Facebook OpenGraph, etc.

Esquemas

- schema.org
- Linked Open Vocabularies
- Vovabularios y ontologías: FOAF, etc.
- HTNML5 microdata
- Microformats

Data cleansing

- OpenRefine
- Python pandas
- R

3.2. Seguridad en *big data*

> Big data, macrodatos, datos masivos, inteligencia de datos o datos a gran escala es un concepto que hace referencia a un conjuntos de datos tan grandes que **aplicaciones informáticas tradicionales de procesamiento de datos no son suficientes** para tratar con ellos y los procedimientos usados para encontrar patrones repetitivos dentro de esos datos
>
> — Wikipedia

¿Qué hace que los datos sean "grandes"?

- ¿La aplicación de nuevas técnicas (AI, machine learning, data mining, etc.)?
- ¿Las enormes oportunidades de recolectar datos de muy diversas fuentes?

El big data crece con el incremento de equipos dotados con una IP.

El propósito de la liberación de esa enorme cantidad de datos puede ser:

1. para el **bien común**: evaluación de modelos económicos, identificación de tendencias sociales, etc.
2. para el **bien del negocio**: satisfacción del cliente, entregas más rápidas, mayores beneficios, etc.

Las arquitecturas empleadas para almacenar big data pueden ser también objeto de problemas de seguridad, por lo que hay que añadir:

3. para el **bien del criminal**

3.3. Ejercicio: Análisis de Datos de Seguridad desde Jupyter Notebook

Cómo usar Jupyter Notebook y R para hacer un análisis exploratorio de datos de seguridad

Introducción

¿Qué es Jupyter Notebook?

Instalación de Jupyter

Para instalar Jupyter en local, seguir las instrucciones. Se recomienda instalar el Jupyter incluido en Anaconda, en vez de mediante `pip` de python. Si se instala Anaconda, hay que instalar a continuación R para tener acceso al mismo desde Jupyter Notebook.

Por defecto, Jupyter Notebook está configurado para usar el *kernel* de Python. Para emplear otros *kernels* distintos, hay que instalarlos primero y elegirlos luego en el menú `Kernel | Change kernel...`.

Después de instalar Anaconda, debe ejecutarse una orden desde el shell del sistema para instalar el kernel de R.

Instalación del kernel de R

```
conda install -c r r-irkernel
```

Instalación del kernel de Matlab

Si se tiene instalado matlab:

```
conda install matlab_engine
```

Emulación de matlab desde R:

```
conda install -c conda-forge r-matlab
```

Caso práctico: Análisis de detección de intrusos

El caso práctico va a consistir en importar un dataset de direcciones IP maliciosas publicado por AlienVault que ayude al equipo de respuesta a incidentes a realizar un análisis exploratorio para definir alertas, registrar o bloquear actividades maliciosas.

Localizar los datos

La base de datos de reputación de IPs de AlienVault publica periódicamente y en varios formatos algunos datasets con información sobre varios tipos de *malware* en Internet.

- El fichero `reputation.data` con el que vamos a trabajar sigue el formato OSSIM.

- El fichero `reputation.data` publicado por AlienVault es un CSV con una clasificación de diversos aspectos de reputación de una lista de IPs.
- Cada hora se genera un fichero de revisión `reputation.rev` de los datos de reputación para poder trabajar con los últimos datos publicados.

Para hacer un análisis exploratorio de los datos de seguridad de `reputation.data` desde Jupyter Notebook podemos usar Python o R.

Análisis exploratorio de datos desde Python

Tareas a completar desde Jupyter Notebook

Completar el capítulo 2 - *Building your Analytics Toolbox: A Primer on Using R and Python...* de *Data Driven Security* desde Jupyter Notebook. Seguir las instrucciones para Python.

Completar el capítulo 3 - *Learning the Hello World...* de *Data Driven Security* desde Jupyter Notebook. Seguir las instrucciones para Python.

Análisis exploratorio de datos desde R

Tareas a completar desde Jupyter Notebook

Preparar el entorno

- Cambiar al directorio donde se encuentran los datos.
- Instalar el paquete `ggplot2` para pintar gráficos y probar que funciona correctamente.
- Comprobar la versión de R
- Probar los data frames con:

```
assets.df <-
data.frame(name=c("danube","gander","ganges","mekong","orinoco"),
                        os=c("W2K8","RHEL5","W2K8","RHEL5","RHEL5"),
                        highvulns=c(1,0,2,0,0))
str(assets.df)
head(assets.df)
head(assets.df$os)
```

```
assets.df$ip <- c("192.168.1.5","10.2.7.5","192.168.1.7", "10.2.7.6",
"10.2.7.7")
```

```
head(assets.df[assets.df$highvulns>1,])
```

```
assets.df$zones <- ifelse(grepl("^192",assets.df$ip),"Zone1","Zone2")
head(assets.df)
```

- Trabajar con el fichero `./data/reputation.data` en lugar de con el fichero actual

Completar el capítulo 2 - *Building your Analytics Toolbox: A Primer on Using R and Python...* de *Data Driven Security* desde Jupyter Notebook. Seguir las instrucciones para R.

Completar el capítulo 3 - *Learning the Hello World...* de *Data Driven Security* desde Jupyter Notebook. Seguir las instrucciones para R.

Completar el capítulo 4 - *Performing Exploratory Security Data Analysis* de *Data Driven Security* desde Jupyter Notebook. Seguir las instrucciones para R.

Ejercicio a completar

Entrega

Entregar el fichero del Jupyter Notebook con los ejercicios realizados.

- Será obligatorio realizar los ejercicios prácticos de los capítulos 2 y 3.
- La compleción de los ejercicios de capítulos adicionales es opcional.
- Para completar los ejercicios prácticos de algunos capítulos, es recomendable haber completado otros anteriores, por lo que se recomienda ser cuidadoso a la hora de elegir un capítulo y sus ejercicios.

3.4. Tensión apertura-seguridad-privacidad

Puntos de tensión:

- Transparencia
- Toma de decisiones
- Interoperabilidad

Oportunidades

- Los **datos abiertos** pueden mejorar la **rendición de cuentas**(*accountability*) pero…
  …pueden incrementar las responsabilidades (*liability*).
- El *tracking* de información **identificable de personas** puede mejorar la **transparencia** pero…
  …también pueden aumentar la vulnerabilidad.

Amenazas

- Muchos de los datos disponibles son difíciles de usar por su pobre **calidad** (necesidad de *data cleansing*), la falta de información sobre el método de recogida (*provenance*) o la falta de **interoperabilidad**
- **Privacidad**: ¿quién posee los datos? Especialmente relevante cuando el **propietario** no es la persona cuyos datos están contenidos en un dataset. ¿Cómo se van a usar?
- Los datos sin contexto o sin matices pueden estar **sesgados** para una **toma de decisión**
- La **anonimidad** no está garantizada. La capacidad de la tecnología *big data* para acceder, analizar y combinar datasets es cada vez mayor

A medida que el *open data* y el *big data* crecen, la preocupación por la privacidad debe ser mayor. Los gobiernos están creando regulaciones para ello.

Problemas de seguridad en big data

La implementación de plataformas big data puede ser origen de vulnerabilidades adicionales:

1. **Frameworks distribuidos**: el procesamiento está repartido entre muchos sistemas para hacer análisis más rápidos

Apache Hadoop no disponía en origen de ningún tipo de seguridad. Ahora hay soluciones como Apache Accumulo que proporciona seguridad a nivel de celda

2. **Bases de datos no relacionales**: Las bases de datos NoSQL no suelen disponer de muchas medidas de seguridad. Suelen confiar en el middleware para que las proporcione
3. **Almacenamiento**: los datos suelen estar almacenados en varias capas, dependiendo de las necesidades de coste/rendimiento (v.g. los datos prioritarios suelen guardarse en medios *flash*)
4. **Endpoints**: Las soluciones de seguridad que generan logs desde endpoints deben validar la autenticidad de dichos endpoints.
5. **Minería de datos**: son el núcleo de muchas implementaciones big data. Es importante asegurarse de que son seguras contra amenazas externas y contra *insiders* que abusan de sus privilegios para obtener información sensible

Otros problemas específicos de los propios datos del *big data*:

1. **Controles de acceso**: proporcionar un sistema que verifique con autenticación/validación

cifrada que los usuarios son quienes dicen ser, y determine quién puede acceder a qué.

2. **Auditorías granulares**: permiten averiguar las consecuencias de un ataque que pasó desapercibido, y qué debe hacerse para mejorar. Las auditorías granulares suponen una gran cantidad de datos que también deben protegerse.
3. **Procedencia** (*provenance*): afecta a los *metadatos*, que pueden ser muy útiles para determinar de dónde viene un dato, quién accede o qué se hace con el mismo.

3.5. General Data Protection Regulation

La GDPR (*General Data Protection Regulation*) es el nuevo marco legal de la UE que entra en vigor en mayo de 2018.
Está pensada para sustituir a las leyes locales de protección de datos para fortalecer la seguridad y la protección de la privacidad de los invididuos.

¿A quién se aplica la GDPR?

La GDPR define obligaciones legales específicas para todos los controladores y procesadores de datos, tanto de la UE como externos pero que proporcionan servicios a individuos en la UE. Se centra en datos de individuos en dos categorías: personales y presonales sensibles.

- Datos **personales**: datos de individuos y otra información que pueda ser usada como identificador online (v.g. direcciones IP).
- Datos **personales sensibles**: datos biométricos o genéticos.

¿Qué significa la GDPR para las empresas?

Las empresas tendrán que implementar una serie de medidas y controles de seguridad y privacidad:

- Designar un responsable de protección de datos (*Data Protection Officer*)
- Notificar infracciones de datos en 72 horas
- Inventariar todos los datos personales procesados
- Proteger los datos por diseño y por defecto
- Evaluar el impacto de la privacidad de datos
- Afrontar multas de hasta 20 M€ o el 4% de la facturación

¿Qué significa en la práctica?

Hay que poder demostrar el cumplimiento de la GDPR. Para organizaciones que no son conscientes de la relevancia de la privacidad de los datos que manejan, la nueva regulación puede causar un gran impacto.

Article 30: *Each controller and, where applicable, the controller's representative, shall maintain a record of processing activities under its responsibility*

Medidas a implementar:

- Poner en marcha una herramienta de **logging** o SEIM (*Security Information and Event Management*) que monitorice la actividad de todos los usuarios y del sistema para identificar comportamientos maliciosos o sospechosos.
- Tener en cuenta los datos almacenados o procesados en la **nube**, que también están en el alcance de los registros cuya actividad hay que mantener.

Article 32: *...the controller and the processor shall implement appropriate technical and organisational measures to ensure a level of security appropriate to the risk...*

Medidas a implementar:

- Crear un inventario de activos críticos que almacenen o procesen datos sensibles, para que se puedan aplicar controles más rigurosos.
- Emprender escrutinios de vulnerabilidades, para identificar debilidades que pudieran ser explotadas.
- Realizar evaluaciones de riesgos y aplicar modelos de amenazas relevantes para el negocio
- Comprobar regularmente los controles de seguridad para asegurar que están funcionando tal y como fueron diseñados.

Article 33: *Notification of a personal data breach to the supervisory authority;* [...]
Article 34: *Communication of a personal data breach to the data subject.*

Medidas a implementar:

- Habilitar controles de detección de amenazas para informar a tiempo y de forma fiable cuándo sucede una violación de seguridad.
- Monitorizar el comportamiento de los usuarios y la red para identificar e investigar rápidamente los incidentes de seguridad.
- Disponer de un plan de respuesta a incidentes documentado y con experiencias
- Disponer de un plan de comunicación para notificar a las partes interesadas

Medidas de implementación

- Examinar la GDPR
- Comprender qué datos personales se guardan y quién tiene acceso
- Inventariar las herramientas y competencias de seguridad existentes

- Realizar un análisis *gap* en términos de herraientas, personal, políticas y procedimientos de seguridad
- Desarrollar e implementar un plan para cerrar los gaps

4. Privacidad de datos

4.1. Privacidad

La privacidad importa a la hora de usar datos con información sensible para los usuarios, especialmente cuando la recopilación es cada vez más fácil y las técnicas de minería de datos son cada vez más sofisticadas y eficientes.

Custodios de datos

- Proveedores de servicios sanitarios
- Agencias gubernamentales
- Compañías de seguros
- Etc

Los guardianes o custodios pueden disponer de datos que podrían entregar a los analistas de datos e investigadores para el **bien común**: evaluación de modelos económicos, identificación de tendencias sociales, etc.

Pero dichos datos contienen **información personal** (v.g. registros médicos, salarios, etc.) de forma que su simple liberación puede no ser apropiada.

Una forma de resolver el problema es pedir a los usuarios que firmen **acuerdos de no divulgación** o NDA (*non-disclosure agreements*). Sin embargo, estos mecanismos

- Necesitan recursos legales y mecanismos para hacer cumplir la ley
- No pueden garatizar la protección contra la sustracción de datos aunque se tomen precauciones

Por tanto, se deben explorar soluciones tecnológicas para **anonimizar** los datos antes de liberarlos.

Leer el capítulo 11 "Privacidad" de *El Código y Otras Leyes del Ciberespacio* (Lawrence Lessig, 2001)

> Existe la creencia de que el ciberespacio no puede ser regulado; que es, en esencia, inmune al control del Estado o de cualquier otro agente. Lawrence Lessing afirma, sin embargo, que esta convicción es falsa. Nosotros podemos, y debemos, decidir el tipo de ciberespacio que deseamos tener, las libertades que éste habrá de garantizar, y quiénes se encargarán de controlarlo. En este contexto, el código -el software y el hardware- se convierte en las más significativa forma de ley, y queda en manos de los

> juristas, los legisladores y los ciudadanos la decisión sobre los principios que éste encarnará
>
> — Lawrence Lessig, El Código y Otras Leyes del Ciberespacio

4.2. Control estadístico de divulgación

Ejemplo 6. Datos censales

La liberación pública de los datos censales no debería revelar información personal sobre los individuos de la población

Existen técnicas estadísticas de control de la divulgación de datos o SDC (*statistical disclosure control*) que sirven para proteger la privacidad de los individuos en datos de uso público.

Agregación de datos

- **Tablas de contingencia**: datos tabulados del recuento de frecuencias (v.g. códigos postales, rangos de edad, fumador, etc)
- **Microdatos**: datos no agregados, donde cada fila se refiere a una persona de la población

Mecanismos de confidencialidad

- Supresión de celdas y adición de ruido (tablas de contingencia)
- Swapping (tablas de contingencia y microdatos)
- Muestreo, generalización geográfica y codificación top/bottom (microdatos): v.g. muestra del 1%, poblaciones > 100K habitantes, edad ≥ 90
- Datos sintéticos (microdatos): generar datos con características similares de distribución estadística

4.3. Ataques a la privacidad

Ataques de tipo *linking*

Combinar datos externos con datos anonimizados.

Ejemplo 7. Caso de una mutualidad de seguros

Un dataset de una mutua de seguros contenía registros médicos de empleados estatales del estado de Massachusetts. Dichos datos no contenían identificadores como nombres, números de seguridad social, direcciones o números de teléfono, por lo que se consideró seguro proporcionar dichos datos a un grupo de investigadores.

Los datos contenían información demográfica, como fecha de nacimiento, sexo y código postal. No parece común que dos individuos tengan la misma fecha de nacimiento, vivan en el mismo

distrito postal y sean del mismo sexo.

Según el censo de votantes de Massachusetts (disponible por $20), nadie tenía la misma combinación de fecha de nacimiento, sexo y código postal que el gobernador *William Weld*. De esta forma, sus registros médicos pudieron ser identificados fácilmente.

Ejemplo 8. Caso de AOL

El 6 de agosto de 2006, AOL liberó un fichero de 2 Gb con cerca de 20 millones de consultas realizadas por 650.000 usuarios, que fueron recolectadas durante un periodo de 3 meses. Además de las consultas, el dataset contenía información de la URL que fue pulsada tras los resultados de la búsqueda y cuál era su ranking.

Aunque la publicación del dataset fue retirado tras unas pocas horas, ya había sido descargado varias veces. El esquema de anonimización que emplearon para proteger los datos consistía en asignar un número aleatorio (*pseudónimo*) a cada usuario de AOL y reemplazar el ID de usuario con dicho número.

Tres días más tarde, dos periodistas del New York Times hallaron y entrevistarion al usuario número 4.417.749 del dataset. Encontraron este usuario a partir de la información semántica contenida en sus consultas: el nombre de una ciudad, varias búsquedas a partir de un apellido determinado, información relativa a la edad, etc.

En el caso de AOL, no había hecho falta ninguna tabla fidedigna (tal y como un censo de votantes) para enlazar (linking) el dataset contra dicha información. En su lugar, se usaron fuentes dispersas de información. La violación de la privacidad ocurrió porque AOL pensó en estas fuentes y en el contenido semántico de las consultas.

Ataques de tipo *record linkage*

Las técnicas de *record linkage* se usan para estimar la probabilidad de re-identificación: las probabilidades de que los individuos de un dataset puedan ser re-identificados mediante datos auxiliares.

Ejemplo 9. Caso de Netflix

Netflix anunció un premio para el desarrollo de un algoritmo preciso de recomendación de películas. Para ayudar a los participantes, liberó un dataset con 100 millones de calificaciones de 18.000 títulos de películas recopiladas de 480.000 usuarios elegidos aleatoriamente. Se eliminó la información personal, sustituyendo los ID de usuario por pseudónimos, como en el caso de AOL.

Este dataset contenía calificaciones de películas y las fechas en que se realizaron. La gran dimensión del dataset fue un atractivo para los atacantes. Un ataque a dicho dataset demostró que las calificaciones podían correlarse con los posts en los foros online públicos (como IMDB reviews), usando técnicas de ataque conocidas como **record linkage**.

Ejemplo: registro médico

Un Hospital contiene datos centralizados de sus pacientes. La información recogida incluye identificación (nombre), demográfica (edad, género, código postal, nacionalidad) y la condición médica de los individuos. Los datos se ponen en el dataset de la Tabla 1 para que los investigadores de otras instituciones (v.g. una Universidad) que estudian correlaciones entre enfermedades y los atributos demográficos de los pacientes puedan beneficiarse de su análisis. Los investigadores de la Universidad solicitan al Hospital liberar dicha tabla de datos.

Tabla 1. Dataset con registros médicos

	Name	Age	Gender	Zip Code	Nationality	Condition
1	Ann	28	F	13053	Russian	Heart disease
2	Bruce	29	M	13068	Chinese	Heart disease
3	Cary	21	F	13068	Japanese	Viral infection
4	Dick	23	M	13053	American	Viral infection
5	Eshwar	50	M	14853	Indian	Cancer
6	Fox	55	M	14750	Japanese	Flu
7	Gary	47	M	14562	Chinese	Heart disease
8	Helen	49	F	14821	Korean	Flu
9	Igor	31	M	13222	American	Cancer
10	Jean	37	F	13227	American	Cancer
11	Ken	36	M	13228	American	Cancer
12	Lewis	35	M	13221	American	Cancer

La cuestión es si liberar los datos de la Tabla 1 es seguro. El hospital dispone de una política que impide liberar la información de identificación de los pacientes. Si liberamos la Tabla 1 que contiene nombres, se violaría dicha política.

¿La eliminación del atributo *name* de la Tabla 1 hace segura la liberación de los datos?

Considérese un investigador que sabe que *Eshwar* es un hombre indio de 50 años que vive en el distito 14853. También sabe que Eshwar ha visitado el Hospital varias veces. Si el investigador ve esta tabla con los nombres eliminados, casi puede estar seguro de que Eswhar sufre cáncer, porque el 5º registro es el único que se corresponde con su conocimiento sobre Eshwar. El conjunto de edad, género, código postal y nacionalidad se conocen como atributos **cuasi-identificadores**, porque un adversario que observe estos atributos puede identificar potencialmente a un individuo del dataset.

Una forma de impedir al investigador ser capaz de inferir la condición médica de Eshwar es asegurarse que, en los datos publicados, ningún paciente puede ser distinguido de un grupo de k pacientes mediante el uso de edad, género, código postal y nacionalidad. Este criterio se conoce como **k-anonimidad** y la tabla resultante será k-anónima.

La Tabla 2 es una versión modificada del dataset con registros médicos que es 4-anónima, donde se han eliminado los nombres; los valores de edad se han generalizado a grupos de edad; los valores de género se han generalizado a *Any*; los códigos postales se han generalizado a los primeros 3 dígitos; y los valores de nacionalidad se han generalizado con diferentes ámbitos geográficos. Ahora el investigador que observe esta tabla solo sabrá que el registro de Eshwar está en el grupo segundo y no está seguro de si tuvo gripe o cáncer.

La Tabla 2 es 4-anónima, es decir, ningún registro se puede distinguir de un grupo de 4 registros en función de la edad, género, código postal y nacionalidad.
¿Ahora es seguro liberar el dataset de la Tabla 2?

Tabla 2. Dataset generalizado con registros médicos

		Age	Gender	Zip Code	Nationality	Condition
(Ann)	1	20-29	Any	130*	Any	Heart disease
(Bruce)	2					Heart disease
(Cary)	3					Viral infection
(Dick)	4					Viral infection
(Eshwar)	5	40-59	Any	14**	Asian	Cancer
(Fox)	6					Flu
(Gary)	7					Heart disease
(Helen)	8					Flu
(Igor)	9	30-39	Any	1322*	American	Cancer
(Jean)	10					Cancer
(Ken)	11					Cancer
(Lewis)	12					Cancer

Veremos que aún no es seguro liberar los datos de la Tabla 2.

Supongamos que el Hospital decide considerar las tablas 4-anónimas como seguras para liberar el dataset.

Además de la Tabla 2, hay muchas tablas 4-anónimas que pueden derivarse del dataset original de Tabla 1. Por ejemplo, la Tabla 3 es otra tabla 4-anónima derivada de la original. El 2º registro ha sido intercambiado con el 8º y el 4º registro ha sido intercambiado con el 10º.

¿Qué tabla debería publicar el Hospital? ¿la Tabla 2 o la Tabla 3?

Tabla 3. Otro dataset generalizado con registros médicos

		Age	Gender	Zip Code	Nationality	Condition
(Ann)	1	20-59	F	1[redacted]***	Any	Heart disease
(Helen)	8					Flu
(Cary)	3					Viral infection
(Jean)	10					Cancer
(Eshwar)	5	20-59	M	1[redacted]***	Asian	Cancer
(Fox)	6					Flu
(Gary)	7					Heart disease
(Bruce)	2					Heart disease
(Igor)	9	20-39	M	13[redacted]**	American	Cancer
(Dick)	4					Viral infection
(Ken)	11					Cancer
(Lewis)	12					Cancer

Intuitivamente, el hospital debería elegir la que sea **más útil** para los investigadores que solicitaron los datos. Supongamos que el objetivo principal de los investigadores es comprender cómo se correlacionan las enfermedades con el género. Dado que necesitarán que se haga las menores sustituciones que sea posible de los valores de *gender* por *Any*, la Tabla 3 parece ser mejor opción que la Tabla 2 en términos del número de reemplazos de valores de género.

4.4. Publicación preservadora de la privacidad

Hay diversas técnicas de PPDP (*Privacy-preserving data publishing*):

Resumen de técnicas de k-anonimidad

Swapping

Tabla 4. Swapping de registros

	Age	Gender	Zip Code	Nationality	Condition
(Ann)	50	F	13053	Russian	Heart disease
(Bruce)	29	M	13068	Chinese	Heart disease
(Cary)	21	F	13068	Japanese	Viral infection
(Dick)	23	M	13053	American	Viral infection
(Eshwar)	28	M	14853	Indian	Cancer
(Fox)	55	M	14750	Japanese	Flu
(Gary)	47	M	14562	Chinese	Heart disease
(Helen)	49	F	14821	Korean	Flu

	Age	Gender	Zip Code	Nationality	Condition
(Igor)	31	M	13222	American	Cancer
(Jean)	37	F	13227	American	Cancer
(Ken)	36	M	13228	American	Cancer
(Lewis)	35	M	13221	American	Cancer

Bucketization

Tabla 5. Registros en buckets

	Age	Gender	Zip Code	Nationality	BID		BID	Condition
(Ann)	28	F	13053	Russian	1		1	Heart disease
(Bruce)	29	M	13068	Chinese	1		1	Heart disease
(Cary)	21	F	13068	Japanese	1		1	Viral infection
(Dick)	23	M	13053	American	1		1	Viral infection
(Eshwar)	50	M	14853	Indian	2		2	Cancer
(Fox)	55	M	14750	Japanese	2		2	Flu
(Gary)	47	M	14562	Chinese	2		2	Heart disease
(Helen)	49	F	14821	Korean	2		2	Flu
(Igor)	31	M	13222	American	3		3	Cancer
(Jean)	37	F	13227	American	3		3	Cancer
(Ken)	36	M	13228	American	3		3	Cancer
(Lewis)	35	M	13221	American	3		3	Cancer

Tabla 6. Ruido aleatorio en registros

		Age	Gender	Zip Code	Nationality	Condition
(Ann)	1	30	F	13073	Russian	Heart disease
(Bruce)	2	28	M	13121	American	Heart disease
(Cary)	3	22	M	13024	Japanese	Cancer
(Dick)	4	20	M	13030	American	Viral infection
…	…	…	…	…	…	…

5. APIs abiertas

5.1. Open APIs

La **transformación digital** no afecta solo a las empresas individuales, sino a la **cooperación global** de todas las organizaciones e industrias para alcanzar beneficios para las empresas y los clientes.

La cooperación global entre organizaciones se basa en el concepto de **Open API** que están llevando a cabo diversas organizaciones, especialmente en el sector financiero y bancario (incluyendo las compañías innovadoras de tecnología financiera o FinTech) y en el sector de los operadores de comunicaciones móviles o MNO (*Mobile Network Operators*).

Ejemplo 10. BBVA API_Market

> API_Market es una plataforma global y abierta de APIs que permite acceder a soluciones financieras e implementarlas en una empresa. Permite gestionar, controlar y analizar pagos, verificar la identidad y notificar a los clientes, acceder a patrones de compra segmentados, etc.

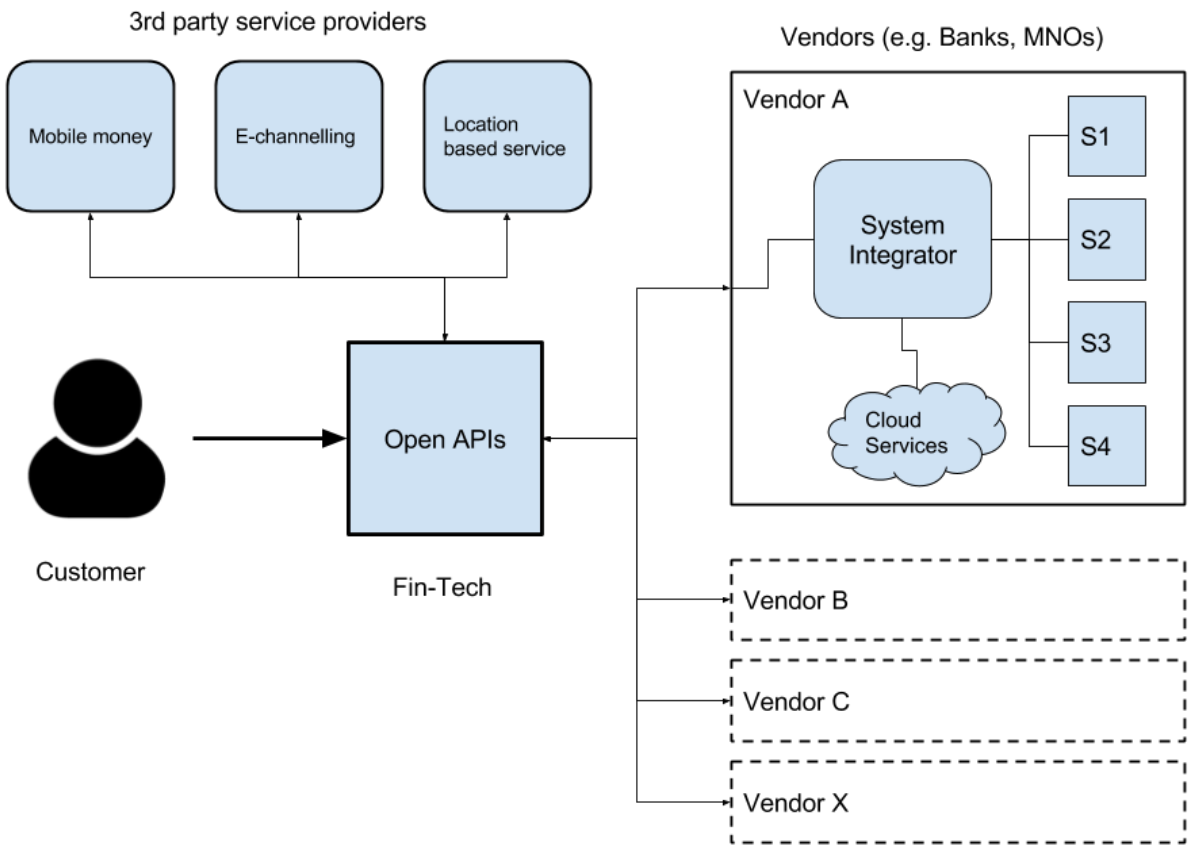
La UE ha aprobado la **Payment Services Directive 2 (PSD2)**, regulación a cumplir por todos los bancos que operen en la UE a partir del **13 de enero de 2018**. Aunque PSD2 es una regulación, ilustra claramente cuál es la relación de los clientes con sus bancos a través de una API abierta.

Ejemplo 11. GSMA API Exchange

> La **GSMA API Exchange** es un conjunto de Open APIs que permite a múltiples MNO interconectarse para disfrutar de los beneficios de una mayor base de clientes que si hicieran sus negocios con sus propias bases de clientes.

Los estándares de Open APIs proporcionan un mecanismo para interconectar empresas que ofrecen servicios similares (PSD2 para FinTech y GSMA API Exchange para MNO) y les permiten compartir bases de clientes para un mayor beneficio.

Desde la perspectiva de los clientes, podrán utilizar varias cuentas o perfiles cuando compren servicios de terceros.



Proveedores diferentes pueden exponer información sobre sus clientes (con el consentimiento explícito de éste) de manera unificada. La implementación intena de estas APIs puede ser distinta en cada caso, pero la interfaz está unificada. Mediante Open APIs, terceros proveedores (v.g. tiendas online, comerciantes, servicios basados en la ubicación, etc.) pueden captar a los clientes cuando estos están adquiriendo sus productos o servicios.

Ejemplo 12. Caso de uso en banca

Supongamos que queremos comprar un portátil de amazon.com y tenemos que hacer el pago mediante la cuenta bancaria en vez de mediante la tarjeta de crédito.

Cuando se factura el item de amazon.com, el sitio web nos proporciona la opción de seleccionar la cuenta bancaria con la que hacer el pago. Esto se consigue a través del Open API que use el banco seleccionado. Si se selecciona el BBVA, seremos redirigidos al sitio web del banco. Se confirma al API del BBVA que permitimos a amazon.com hacer un débito por la cantidad de la compra y se cierra la transacción. Sin necesidad de tarjetas de crédito o pasarelas de pago.

Pero esto funciona igual para todo tipo de industrias, no solo la banca. Actualmente se usan las plataformas móviles para identificación y compra:

- Apple Pay: usa EMV *Payment Tokenisation*
- Android Pay; usa HCE (*host-based card emulation*)
- Samsung Pay:

- Microsoft Pay: usa HCE

Personas que no tenían acceso a servicios bancarios pero sí a un teléfono móvil pueden hacer igualmente sus transacciones financieras online.

5.2. Directiva de Servicios de Pago

La versión 2 de la **Payment Services Directive** (PSD2) de la UE regula los servicios de pago y los proveedores de servicios de pago en la UE y el área economica europea (EEA).

define un conjunto de estándares RTS o *Regulatory Technical Standards* que permite a los consumidores beneficiarse de pagos electrónicos más seguros e innovadores.

Objetivos y motivación

¿Cuáles son los objetivos de la PSD2?

PSD2 alcanza a los servicios de pago y proveedores de servicios financieros innovadores, como las FinTechs, también llamados proveedores de *third party payment service* (TPPS).

Las medidas de seguridad del RTS emanan de dos objetivos principales: (1) proteger al consumidor y (2) mejorar la competencia en un mercado emergente tan veloz.

- La protección al consumidor se fomenta incrementando el nivel de seguridad de los pagos electrónicos. La RTS introduce requisitos de seguridad a los proveedores de servicios de pago (como bancos y otras entidades).
- La mejora de la competencia e innovación en el mercado de pagos minorista se logra incluyendo dos nuevos tipos de TPPS: los PIS y los AIS.
  - proveedores de *payment initiation services* (PIS): inician los pagos de parte del cliente y aseguran al minorista que el dinero está en camino.
  - proveedores de *aggregators and account information service* (AIS): proporcionan a los clientes una visión general de sus cuentas y sus balances.

Strong Customer Authentication (SCA)

¿Cómo mejorará la PSD2 la seguridad en los pagos electrónicos?

La RTS define una autenticación fuerte de clientes o SCA (*strong customer authentication*) como base para el acceso a una cuenta de pago y para hacer pagos online. Esto significa que los usuarios deben probar su identidad por al menos dos vías de tres:

- algo que saben
- algo que poseen
- algo que son

La SCA es muy usada en la UE. Por ejemplo, cuando los clientes pagan con una tarjeta en una tienda

física, se les requiere validar la transacción con un PIN. Este no es el caso de los pagos electrónicos remotos, ya sea vía tarjeta o transferencia bancaria desde un banco online. Para estas transacciones, SCA se viene aplicando en cietros países de la UE (v.g. BE, NL, SE). En otros países los proveedores de servicios de pago pueden solicitar voluntariamente aplicar SCA.

La RTS define que debe emplearse la SCA para acceder a las cuentas de pagos y para hacer los pagos. Los bancos y otros proveedores de servicios de pago tendrán que desplegar las infraestructuras necesarias para la SCA. Además tendrán que mejorar la gestión de fraudes. Los consumidores y comerciantes tendrán que equiparse y formarse para ser capaz de operar en un entorno SCA.

La RTS también permite exenciones de la SCA para no perturbar la manera en que se opera actualmente. También porque puede haber mecanismos de autenticación alternativos que sean igualmente seguros. Sin embargo, los proveedores de servicios de pago que quieran ser exonerados de la SCA deben primero aplicar mecanismos para monitorizar las transacciones y evaluar si el riesgo de fraude es bajo.

Todos los proveedores de servicios de pago tendrán que probar sus implementaciones, chequeando y auditando las medidas de seguridad. En caso de pago fraudulento, los consumidores deben recibir un reembolso pleno.

Para los pagos online, la seguridad debe ser mejorada enlazando, mediante una contraseña de un solo uso, la transacción online con su cuantía y el beneficiario del pago. Esta práctica asegura que, en caso de ataque, la información obtenida por un defraudador potencial no puede usarse para iniciar otra transacción. Este procedimiento ya está en práctica en países como Bélgica y ha conducido a una reducción significativa del fraude en pagos online.

¿Cuándo será obligatoria la SCA?

La SCA será obligatoria 18 meses tras la entrada en vigor de la RT, es decir, cuando sea publicada en el DOUE (esperado para septiembre de 2019)

Protección de datos personales

Los propietarios de cuentas podrán ejercer el control sobre la transmisión de sus datos personales bajo la PSD2 y la directiva de protección de datos de la GDPR. No puede tener lugar ningún procesamiento de datos sin el consentimiento expreso del consumidor. Además, los proveedores de servicios de pagos solo podrán acceder y procesar los datos personales necesarios para la provisión de los servicios que el consumidor acuerde.

PSD2 regula los nuevos servicios de pago que requieren acceso a otros datos de pagos de los usuarios. Por ejemplo, esto puede significar iniciar un pago desde la cuenta del cliente o agregar la información en una o varias cuentas del cliente con uno o más proveedores de servicios de pago, de cara a gestionar las finanzas personales. Cuando el consumidor quiera beneficiarse de estos nuevos servicios, tendrá que solicitarlo explícitamente al proveedor.

Los servicios de pago tendrán que informar a sus clientes sobre cómo procesarán sus datos. También tendrán que cumplir con otros derechos del cliente, como el **derecho de acceso** o el **derecho al olvido**. Todos los proveedores de servicios de pago (bancos, instituciones de pago o

nuevos proveedores fintech) deben cumplir las reglas de protección de datos al procesar los datos personales para dar sus servicios.

5.3. Screen scraping

Screen scraping

El screen scraping es acceder a los datos a través de la interfaz de usuario web (vista HTML en pantalla) con las credenciales de seguridad que el usuario emplea para acceder a la aplicación web. Los TPP podrían acceder a los datos de un cliente mediante screen scrapping sin necesitar identificación adicional vis-à-vis con el banco.

¿A qué datos pueden acceder los TPP via "screen scraping"?

La PSD2 prohibe a los TPP acceder a cualquier otro dato de la cuenta, más allá de aquellos autorizados explícitamente por el cliente. Los clientes tendrán que expresar su acuerdo para el acceso, uso y procesado de estos datos.

Con las nuevas reglas, **ya no estará permitido** acceder a los datos del cliente mediante el uso de técnicas como ***screen scraping***

Los bancos podrán desplegar un canal de comunicación (**API**) que permita a los TPP acceder a los datos que necesitan, de acuerdo con la PSD2. Este canal se usará para permitir que los bancos y TPPs se identifiquen unos a otros al acceder a dichos datos, comunicándose a través de mensajería segura en todo momento.

Los bancos pueden establecer este canal de comunicación adaptando la interfaz de las aplicaciones de banca online que proporcionan a sus clientes. También deberán crear una **API dedicada** que incluya toda la información necesaria para los proveedores de servicios de pago.

La RTS también especifica las salvaguardas de contingencia que los bancos deberán desplegar **si deciden desarrollar** una API dedicada, lo que asegurará la libre competencia y la continuidad del negocio de las TPP.

Implicaciones de la PSD2 para el scraping

Los bancos y las fintech han intentado hace *lobby* acerca del *screen scraping*. El sector bancario ha pretendido favorecer la prohibición (argumentando sus costes y la seguridad), mientras que las fintech ha pretendido favorecerlo en situaciones donde no había un API o no éste no funcionaba apropiadamente.

La PSD2 es la primera vez que una industria se ve forzada a dar acceso libre a los datos de sus clientes a un competidor potencial. En algunos casos puede ser una situación win-win, pero no necesariamente. Hay muchas razones para justificar las posturas de ambos lados. Sin embargo, en muchas otras no habrá un incentivo adecuado para proporcional un API que tenga una

solidez industrial.

La RTS ofrece a los bancos la posibilidad de elegir dos interfaces:

- implementar un API o interfaz dedicada, que sea la mejor posible si está bien construida y mantenida; o bien…
- implementar la banca electrónica con un nivel de identificación TPP por delante

Si el banco opta por la vía del API, también tendrá que fijar un nivel de identificación ante el sistema de banca electrónica (o compartirlo con el nivel de identificación de la API) que le sirva como mecanismo alternativo o *fallback*. Entonces las terceras partes tendrán que usar el API si existe y si funciona adecuadamente. En caso de que no funcione adecuadamente, los terceros podrán acceder a través de la banca electrónica, tras identificarse con un certificado válido, y emplear técnicas de screen scraping.

Para los bancos, hay una posibilidad de exención de la obligación de proporcionar un mecanismo de fallback. Las autoridades nacionales podrán definir dichas exenciones para aquellos bancos que durante un periodo de 6 meses demuestren que sus API funcionan adecuadamente a un conjunto de KPI (que aún están por definir). Esta exoneración puede ser retirada si el API deja de funcionar correctamente por un periodo de 2 semanas (según la RTS)

El screen scraping podrá seguir siendo usado para información no explícitamente regulada via PSD2, como depósitos, préstamos, planes de pensiones, acciones, fondos de inversión, etc. A menos que los bancos opten por ofrecer un API que funcione mejor que el screen scraping sobre las aplicaciones de banca electrónica y que sea gratis o más barata que el scraping.

Referencias

- Bee-Chung Chen, Daniel Kifer, Kristen LeFevre, Ashwin Machanavajjhala: Privacy-Preserving Data Publishing, *Foundations and Trends in Databases*, vol. 2, no. 1-2, pp. 1-167, 2009. DOI: 10.1561/1900000008
- Lawrence Lessig: *El Código y Otras Leyes del Ciberespacio*, Ed. Taurus, 2001
- Top 5 ReST API security guidelines
- Best practices for securely storing API keys