eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Escuela Universitaria de Ingeniería Vitoria-Gasteiz
Ingeniaritzako Unibertsitate Eskola Vitoria-Gasteiz

Desarrollo de herramientas de seguridad informática para Android

Memoria del Trabajo de Fin de Grado

presentada para optar al grado de

Ingeniería Informática de Gestión y Sistemas de Información

por

Ander Granado Masid

Director: Pablo González Nalda

Septiembre de 2017

Agradecimientos

A lauro

Índice general

Índice de figuras

Índice de tablas

Resumen y Organización de la memoria

I

Alcance del Trabajo

1 Descripción, Objetivos y Motivación

1.1. Descripción

En este trabajo se desarrolla un estudio sobre el campo de la seguridad informática, más concretamente sobre las diferentes herramientas de seguridad informática. En base a eso, se desarrolla una aplicación para dispositivos móviles que busca, de manera sencilla para el usuario, proporcionar soluciones a tareas recurrentes dentro del campo de la seguridad informática, basándose en herramientas ya existentes. Estas herramientas, usadas por pentesters, analistas forenses o hackers de sombrero blanco permiten elaborar operaciones de todo tipo, desde escanear una red inalámbrica hasta romper el cifrado de un archivo para acceder a la información que contiene.

Gran parte de esta herramientas son gratuitas [1] o incluso de software libre [2], lo que otorga la posibilidad de que dichas herramientas mejoren continuamente. El mayor problema de este tipo de herramientas suelen ser su público objetivo. Normalmente este tipo de herramientas están diseñadas para profesionales del sector, profesionales tanto con conocimientos de seguridad informática como de programación o administración de sistemas. La mayoría de estas herramientas se basan en librerías o frameworks completos, con cierta dificultad de uso, o scripts CLI. Debido a esto, cierta tarea como escanear una red, que para un experto en ciberseguridad o un administrador de sistemas se convierte en 5 segundos tecleando un comando, para un usuario medio se convierte en un auténtico quebradero de cabeza.

1.2. Objetivos

El objetivo de este este proyecto es doble. Por una parte se busca realizar un análisis del campo de la seguridad informática, un *estado del arte* del área que permita vislumbrar cuales son sus diferentes aplicaciones y a partir de ahí concretar las necesidades más importantes dentro de ese campo para, al final, acabar elaborando una aplicación para Android que nos proporcione ciertas utilidades.

Por otra parte, también se busca que la aplicación a elaborar sirva tanto para usuarios ex-

perimentados en la materia como para un público general. Para ello, un buen diseño de la interfaz gráfica (GUI) o diferentes principios de experiencia de usuario (UX) jugarán un papel fundamental. De esta manera lograremos una transición entre herramientas accesibles solo para unos pocos a una herramienta para todo el mundo.

1.3. Motivación

A día de hoy la informática es una industria fundamental dentro de la sociedad en general y de las vidas de las personas en particular. Los ordenadores personales son la herramienta fundamental de trabajo en una gran cantidad de áreas, ademas de una herramienta que se encuentra en prácticamente cualquier hogar. Los denominados Smartphones junto a Internet se han convertido en la principal herramienta de comunicación. La revolución causada por la industria llega hasta tal punto que una organización como la ONU ha declarado Internet *como un derecho humano por ser una herramienta que favorece el crecimiento y el progreso de la sociedad en su conjunto* [3].

Teniendo en cuenta toda la información que transmitimos, almacenamos y procesamos, sería lógico pensar que la seguridad de dicha información es vital. El área de la seguridad informática se encarga de ofrecer los mecanismos necesarios para que nuestra información no se vea comprometida de ninguna manera y nuestros dispositivos permanezcan seguros y con la menor cantidad de vulnerabilidades posible. Cada vez resulta mas importante, y sobre todo ahora con la llegada del Internet of Things (IoT), ya que pasamos de tener no solo nuestros ordenadores o Smartphones conectados a Internet, sino que tenemos otros dispositivos como nuestro coche o nuestra lavadora conectados, con los riesgos que ello conlleva.

Por todo esto, la seguridad gestionada por los propios usuarios

2 Viabilidad

Este capítulo tiene como objetivo realizar, en base a los objetivos marcados en el capítulo anterior, elaborar un análisis de viabilidad del proyecto, analizando las diferentes tareas que contiene, calculando los correspondientes gastos y sus inherentes riesgos, que pueden afectar, atrasando o incluso impidiendo, la realización del proyecto.

2.1. Requisitos funcionales del trabajo

Los requisitos funcionales de la aplicación (RF a partir de ahora) se elaboran en base a los objetivos descritos y son los siguientes:

- Elaborar un estado del arte que analice el campo de la seguridad informática, sus aplicaciones y analice el área del *pentesting*.
- Elaborar una aplicación que permita obtener información sobre redes y nodos de la red.
- Integrar correctamente herramientas de terceros para lograr que la aplicación resulte lo más escalable posible.
- Aplicar principios sobre experiencia de usuario (UX) y sobre el diseño de interfaces gráficas (GUI) para que la aplicación sea lo más sencilla y cómoda de usar.

2.2. Planificación del tiempo

2.2.1. Estructura de Descomposición del Trabajo

El EDT, o Estructura del Desglose del Trabajo, es un sistema jerárquico que permite organizar las diferentes tareas de un proyecto. Es una técnica ámpliamente usada para gestionar todo tipo de proyectos, especialmente proyectos de software.

A la hora de planificar el tiempo, se ha tenido en cuenta un enfoque en dos fases, a las cuales denominaremos Fase 1 y Fase 2. Esto es debido al carácter del proyecto. Por una parte, para elaborar la aplicación anteriormente mencionada, resulta fundamental realizar un estudio

sobre el campo del la seguridad informática, más concretamente sobre el área del pentesting, para poder llegar a encontrar las mejores herramientas y técnicas que permitan desarrollarla. Este estudio, bien desglosado y fundamentado, llevará a la obtención de un elaborado estado del arte, que será el principal objetivo de la Fase 1. Además, dicha fase contiene un periodo de aprendizaje y familiarización con diversos conceptos y tecnologías, que también quedarán reflejados.

En la Fase 2, en función de lo aprendido en la Fase 1, se elaborará la aplicación en base a los criterios de implementar diversas utilidades junto a un experiencia de usuario (UX) óptima, que vendrá acompañada de un buen diseño de una interfaz gráfica (GUI).

El EDT completo quedaría tal y como se muestra en la figura 2.1.

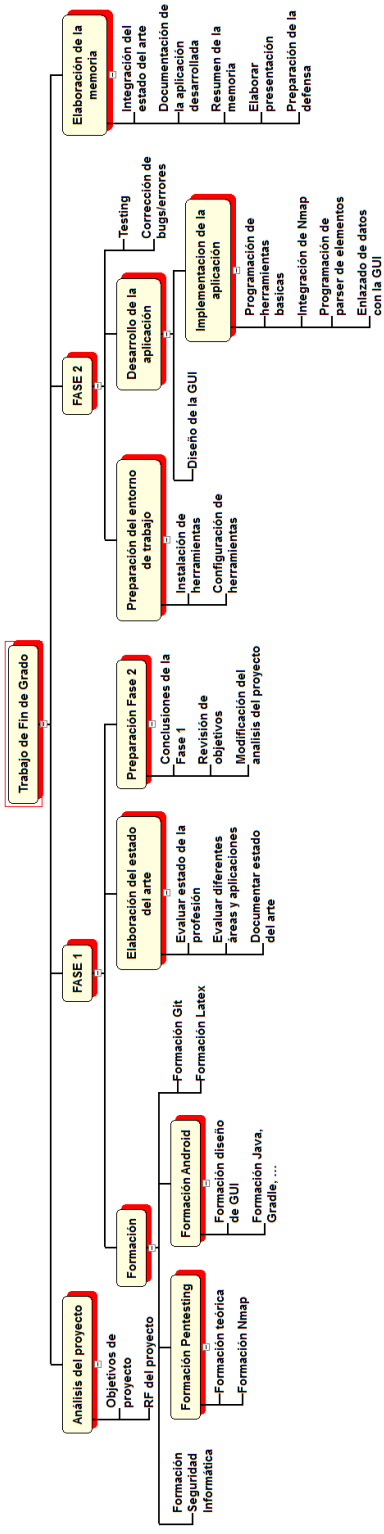

Figura 2.1.: EDT completo

2.2.1.1. Fase 1

El EDT para la Fase 1 quedaría como se muestra en la figura 2.2.

Figura 2.2.: EDT de la Fase 1

2.2.1.2. Fase 2

El EDT para la Fase 2 quedaría como se muestra en las figura 2.3.

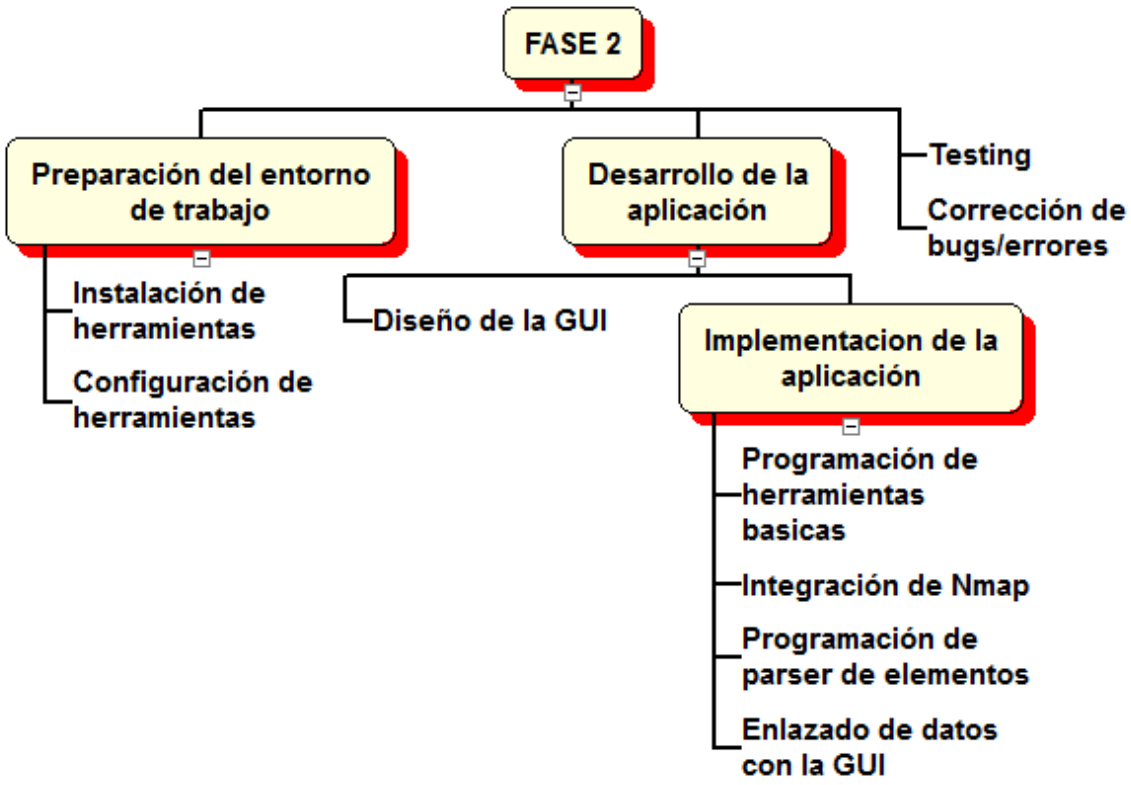

Figura 2.3.: EDT de la Fase 2

2.2.2. Agenda del proyecto

El proyecto se llevará a cabo durante varios meses, comenzando en abril. Se trabajará a media jornada (4 horas) de lunes a viernes, con las siguientes excepciones. Primero, se tendrá en cuenta el calendario de festivos oficiales para aplicar algunas jornadas festivas en cuyos días no se trabajará. Esos días quedan reflejados en la tabla 2.1. A esos festivos se les aplica unas excepciones, en las cuales, aun siendo festivo, se trabajará igualmente. Esos festivos en los que se trabaja están indicados en rojo.

Fecha	Evento
1 de Enero	Año nuevo
6 de Enero	Día de Reyes
19 de Marzo	San José
2 de Abril	Jueves Santo
3 de Abril	Viernes Santo
6 de Abril	Lunes de Pascua
28 de Abril	San Prudencio
1 de Mayo	Día del trabajo
25 de Julio	Santiago Apóstol
5 de Agosto	Virgen Blanca
15 de Agosto	Asunción de la Virgen
12 de Octubre	Fiesta Nacional de España
8 de Diciembre	Inmaculada Concepción
25 de Diciembre	Navidad

Tabla 2.1.: Calendario de días festivos oficial (de Álava)

Por otra parte, existen una serie de días en los que se aplica una reducción de la jornada laboral de 4 a 2 horas (en los días considerados laborales), debido básicamente a exámenes. Estos periodos vienen indicados en la tabla 2.2.

Nombre del periodo	Fecha de inicio	Fecha de fin
Exámenes Convocatoria Ordinaria	16 de mayo	23 de mayo
Exámenes Convocatoria Extraordinaria	27 de junio	4 de julio

Tabla 2.2.: Reducciones de la jornada laboral

En base a dicho calendario, la fecha estimada de finalización del proyecto es del 24 de Julio.

2.2.3. Tareas

El EDT completo de tareas, al que se añaden la fase inicial de objetivos del proyecto, y toda la fase final de elaboración de la memoria, presentación y la defensa quedaría de la siguiente manera.

0. Análisis del proyecto
- **0.1** Objetivos del proyecto
- **0.2** RF del proyecto

1. FASE 1
- **1.1** Formación
  - **1.1.1** Formación seguridad informática
  - **1.1.2** Formación Pentesting

- **1.1.2.1** Formación Teórica
- **1.1.2.2** Formación NMap
- **1.1.3** Formación Android
- **1.1.3.1** Formación Diseño de GUI
- **1.1.3.2** Formación Java, Gradle, ...
- **1.1.4** Formación Git
- **1.1.5** Formación LaTeX
- **1.2** Elaboración del estado del arte
- **1.2.1** Evaluar estado de la profesión
- **1.2.2** Evaluar diferentes áreas y aplicaciones
- **1.2.3** Documentar estado del arte
- **1.3** Preparación de la Fase 2
- **1.3.1** Conclusiones de la Fase 1
- **1.3.2** Revisión de objetivos
- **1.3.3** Modificación del análisis del proyecto

2. FASE 2

- **2.1** Preparación del entorno de trabajo
- **2.1.1** Instalación de herramientas
- **2.1.2** Configuración de herramientas
- **2.2** Desarrollo de la aplicación
- **2.2.1** Diseño de la GUI
- **2.2.2** Implementación de la aplicación
- **2.2.2.1** Programación de herramientas básicas
- **2.2.2.2** Integración de NMap
- **2.2.2.3** Programación de parser de elementos
- **2.2.2.4** Enlazado de datos con la GUI
- **2.2.3** Testeo
- **2.2.4** Corrección de bugs/errores

3. Elaboración de la memoria

- **3.1** Integración del estado del arte
- **3.2** Documentación de la aplicación desarrollada
- **3.3** Resumen de la memoria
- **3.4** Elaborar presentación
- **3.5** Preparación de la defensa

4. Reuniones periódicas

A continuación se explica mediante una breve definición cada tarea, además de especificar su duración en horas.

Número: 0.1.

Nombre: Objetivos del proyecto.

Descripción: Definir los objetivos que tiene que cumplir el TFG.

Trabajo estimado: 5 horas.

Número: 0.2.

Nombre: RF del proyecto.

Descripción: Definir, en base a los objetivos del proyecto, los Requisitos funcionales concretos del proyecto.

Trabajo estimado: 5 horas.

Número: 1.1.1.

Nombre: Formación seguridad informática.

Descripción: Familiarizarse con el amplio entorno de la seguridad informática y comprender las diferentes áreas, objetivos y el estado de dicho campo.

Trabajo estimado: 30 horas.

Número: 1.1.2.1.

Nombre: Formación Teórica.

Descripción: Familiarizarse con los conceptos de Pentesting, las diferentes técnicas usadas y las diferentes fases del proceso de Pentesting.

Trabajo estimado: 20 horas.

Número: 1.1.2.2.

Nombre: Formación NMap.

Descripción: Familiarizarse con el entorno de NMap, cómo implementarlo, usarlo para obtener información y de qué formas se puede obtener información estructurada y organizada para su posterior uso.

Trabajo estimado: 10 horas.

Número: 1.1.3.1.

Nombre: Formación Diseño de GUI.

Descripción: Aprender a usar herramientas de diseño de GUI, diferentes patrones de Diseño en sistemas Android, y el uso de IDEs o herramientas para desarrollar dichas GUIs.

Trabajo estimado: 10 horas.

Número: 1.1.3.2.

Nombre: Formación Java, Gradle, ...

Descripción: Aprender sobre el uso de Java para desarrollar aplicaciones Android, diferentes clases, utilidades o conceptos recurrentes en la programación para Android.

Trabajo estimado: 15 horas.

Número: 1.1.4.

Nombre: Formación Git.

Descripción: Aprender el uso de dicho sistema de control de versiones para llevar un control riguroso del desarrollo del proyecto y de la aplicación.

Trabajo estimado: 5 horas.

Número: 1.1.5.

Nombre: Formación LaTeX.

Descripción: Aprender diferentes conceptos de LaTeXpara elaborar tanto el estado del arte como el propio informe de la manera más clara y elegante posible.

Trabajo estimado: 5 horas.

Número: 1.2.1.

Nombre: Evaluar estado de la profesión.

Descripción: Analizar los diferentes campos de la profesión, las necesidades mas demandadas y los diferentes perfiles de profesionales dentro del campo.

Trabajo estimado: 5 horas.

Número: 1.2.2.

Nombre: Evaluar diferentes áreas y aplicaciones.

Descripción: Evaluar las necesidades concretas a nivel técnico, las aplicaciones más usadas y las virtudes y carencias de éstas.

Trabajo estimado: 10 horas.

Número: 1.2.3.

Nombre: Documentar estado del arte.

Descripción: Elaborar la documentación en base a toda la información recogida para obtener un elaborado estado del arte.

Trabajo estimado: 5 horas.

Número: 1.3.1.

Nombre: Conclusiones de la Fase 1.

Descripción: Elaborar una serie de conclusiones en función a todo el estudio realizado sobre el campo de la seguridad informática.

Trabajo estimado: 5 horas.

Número: 1.3.2.

Nombre: Revisión de objetivos.

Descripción: Revisión de los objetivos y los requisitos funcionales de la aplicación a desarrollar en función a todo lo investigado.

Trabajo estimado: 5 horas.

Número: 1.3.3.

Nombre: Modificación del análisis del proyecto.

Descripción: Modificar la parte de análisis del proyecto realizada anteriormente, antes de comenzar con la Fase 1.

Trabajo estimado: 5 horas.

Número: 2.1.1.

Nombre: Instalación de herramientas.

Descripción: Instalación de todo lo necesario para desarrollar la aplicación.

Trabajo estimado: 5 horas.

Número: 2.1.2.

Nombre: Configuración de herramientas.

Descripción: Configuración de todas las herramientas para que el desarrollo de la aplicación sea lo mas cómodo posible.

Trabajo estimado: 5 horas.

Número: 2.2.1.

Nombre: Diseño de la GUI.

Descripción: Diseñar una interfaz gráfica clara y sencilla de usar para interactuar con las funciones a implementar.

Trabajo estimado: 15 horas.

Número: 2.2.2.1.

Nombre: Programación de herramientas básicas.

Descripción: Programar herramientas básicas para el escaneo de redes.

Trabajo estimado: 10 horas.

Número: 2.2.2.2.

Nombre: Integración de NMap.

Descripción: Integrar el núcleo de NMap en la aplicación para poder hacer uso de toda su funcionalidad.

Trabajo estimado: 10 horas.

Número: 2.2.2.3.

Nombre: Programación de parser de elementos.

Descripción: Elaborar un puente entre NMap y la aplicación para obtener los datos de NMap y poder usarlos en la aplicación de la manera más organizada posible.

Trabajo estimado: 20 horas.

Número: 2.2.2.4.

Nombre: Enlazado de datos con la GUI.

Descripción: Enlazar los datos con las diferentes vistas a través de diversos controladores, para poder visualizar e interactuar con ellos.

Trabajo estimado: 15 horas.

Número: 2.2.3.

Nombre: Testing.

Descripción: Una vez desarrollada la aplicación, realizar un amplio testeo para comprobar que funciona correctamente.

Trabajo estimado: 10 horas.

Número: 2.2.4.

Nombre: Corrección de bugs/errores.

Descripción: En base a los errores detectados en el testeo, implementar las correcciones a dichos fallos.

Trabajo estimado: 15 horas.

Número: 3.1.

Nombre: Integración del estado del arte.

Descripción: Integrar el estado del arte desarrollado dentro de la memoria.

Trabajo estimado: 5 horas.

Número: 3.2.

Nombre: Documentación de la aplicación desarrollada.

Descripción: Elaborar en base a todo el proceso de desarrollo una documentación clara sobre la aplicación e integrarla en la memoria.

Trabajo estimado: 15 horas.

Número: 3.3.

Nombre: Resumen de la memoria.

Descripción: Terminar la elaboración de la memoria, añadiendo las diferentes secciones necesarias y el formato correspondiente.

Trabajo estimado: 5 horas.

Número: 3.4.

Nombre: Elaborar presentación.

Descripción: Elaborar la presentación en diapositivas que se usará en la defensa ante el tribunal.

Trabajo estimado: 5 horas.

Número: 3.5.

Nombre: Preparación de la defensa.

Descripción: Preparar la defensa ante el tribunal en función a la documentación elaborada.

Trabajo estimado: 10 horas.

Número: 4.

Nombre: Reuniones periódicas.

Descripción: Reuniones periódicas con el director del TFG para llevar un control del desarrollo del proyecto.

Trabajo estimado: 15 horas.

2.2.4. Entregables

2.2.4.1. Fase 1

El entregable de la Fase 1 consistirá en un estado del arte redactado sobre el campo de la seguridad informática, enfocándose en el área del pentesting.

2.2.4.2. Fase 2

El entregable de la Fase 1 consistirá en una aplicación elaborada, que sirva como herramienta para el escaneo de redes informáticas. La aplicación, desarrollada para Android, estará correctamente empaquetada, con los posible fallos corregidos y además dispondrá de una GUI sencilla de usar.

2.2.5. Cronograma

El cronograma completo, con las dos fases, como se muestra en las figuras 2.4, 2.5, 2.6 y 2.7 respectivamente.

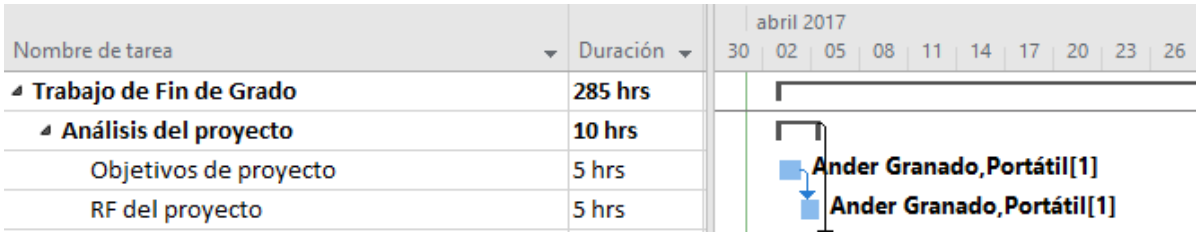

Figura 2.4.: Cronograma de la fase inicial

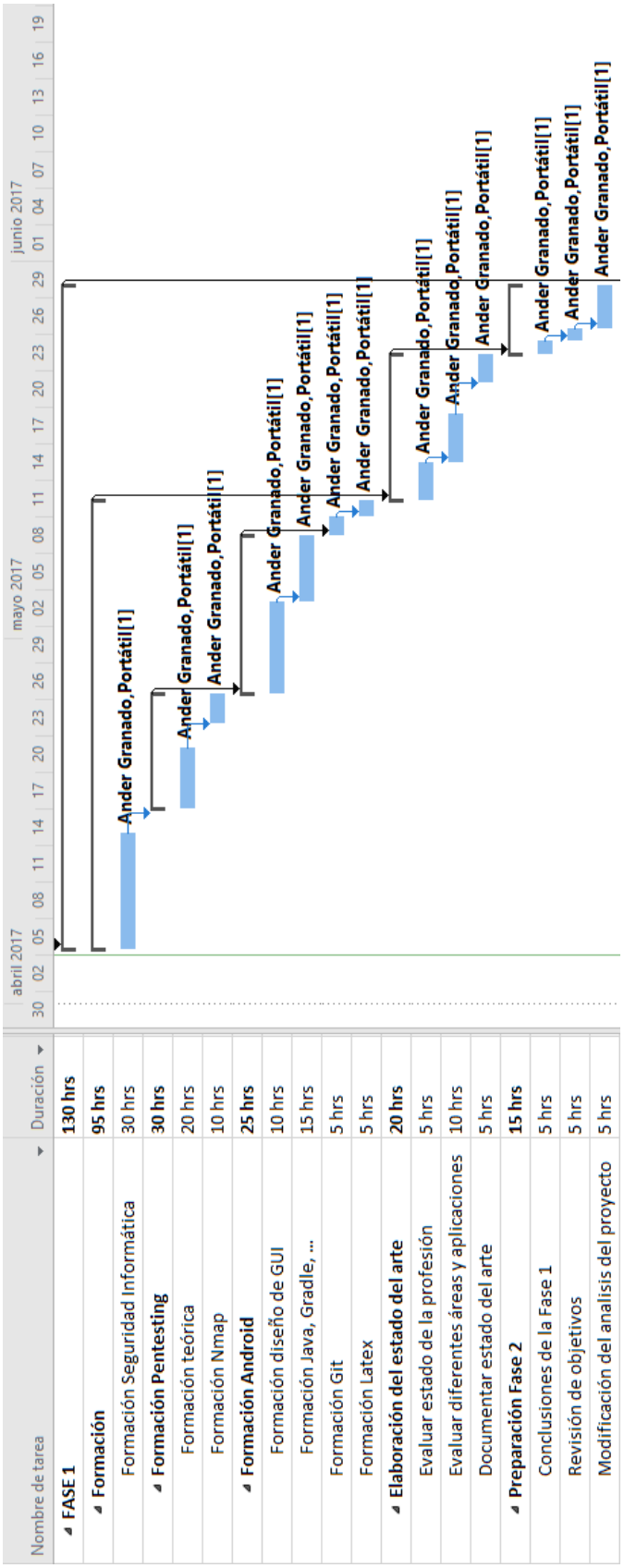

Figura 2.5.: Cronograma de la Fase 1

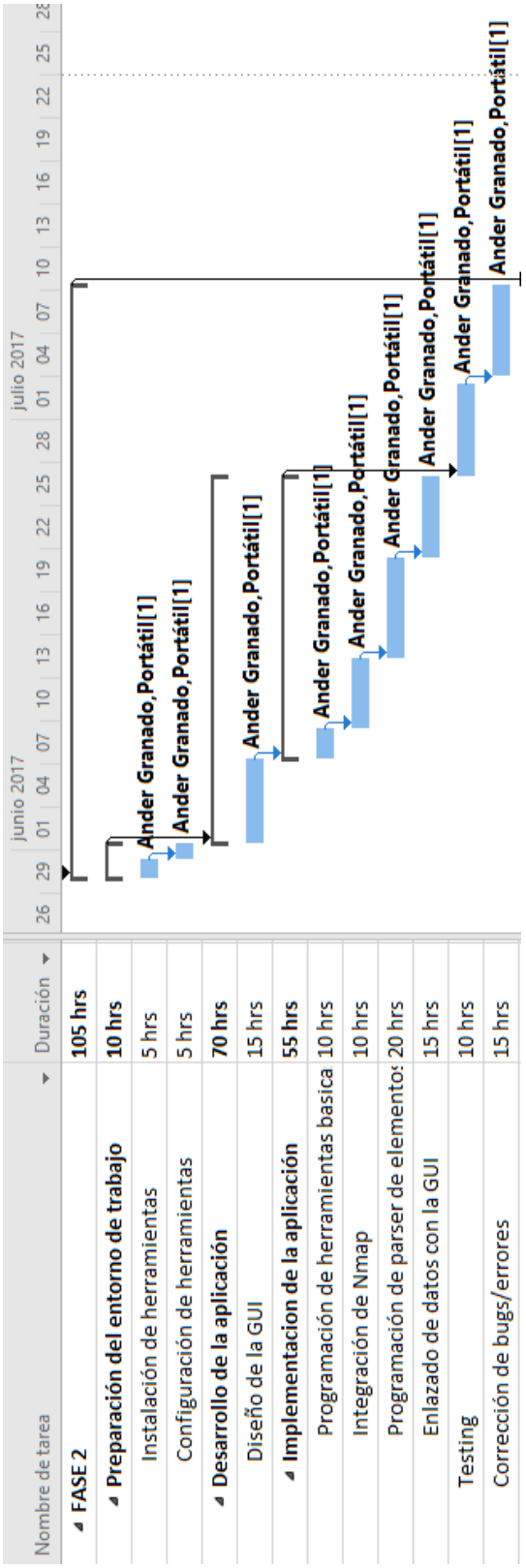

Figura 2.6.: Cronograma de la Fase 2

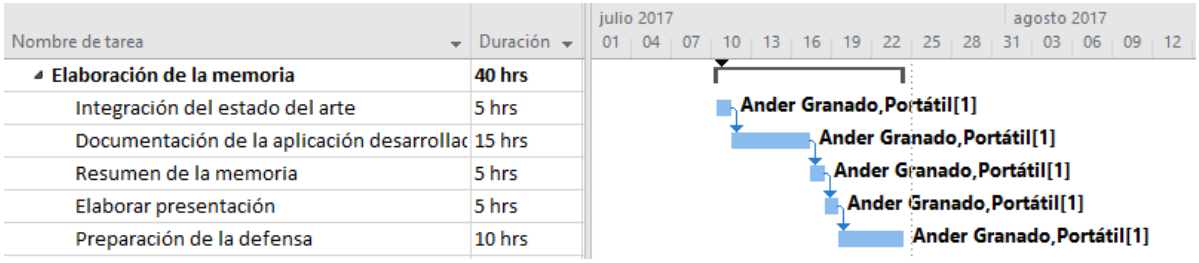

Figura 2.7.: Cronograma de la elaboración de la memoria

2.3. Gestión de costos

A la hora de elaborar un análisis de costos, con el objetivo de obtener un presupuesto, primero es necesario identificar todos los recursos que influyen en ese presupuesto. Por una parte, tendremos los recursos de trabajo, es decir empelados, y por otra parte los recursos materiales, tanto de software como de hardware. En la figura 2.8 se muestran tanto los recursos de trabajo como los recursos materiales de hardware.

	Nombre del	Tipo	Etiqueta de	Iniciales	Grupo	Capacidad	Tasa	Tasa horas	Costo/U:	Acumu	Calendario base
1	Ander Granado	Trabajo		A		100%	14.00 €/hr	0.00 €/hr	0.00 €	Prorrateo	Calendario Estudiantil
2	Portátil	Material		P			0.00 €		1.50 €	Prorrateo	
3	BQ Aquaris M5	Material		BQ			0.00 €		1.50 €	Prorrateo	

Figura 2.8.: Recursos de trabajo y materiales

Por otra parte, en la tabla 2.3 se muestra los recursos materiales de software utilizados.

Concepto	Coste
Windows 10	135,00 €[4]
Project 2016	1.369,00 €[5]
WBS Chart Pro	187,50 €[6]
Ubuntu 16.04	0,00 €
TeX Studio	0,00 €
Android Studio	0,00 €
NMap	0,00 €
Git	0,00 €
GitHub	0,00 €

Tabla 2.3.: Recursos materiales (software)

2.3.1. Presupuesto

El análisis de costes del proyecto se refleja en toda la información que aparecen entre la tabla 2.4 y la 2.10. Para ello se tienen en cuanta varios puntos. El primero, se tiene en cuenta

un salario base de 14 €/h, correspondiente un salario estándar de un analista programador. Para los recursos materiales, se tiene en cuenta un costo por uso de 1,50 €. Estos costos por uso vienen dados por el gasto que generan debido al consumo energético, internet y otro tipo de gastos derivados de su utilización. Por otra parte se realiza una separación para los recursos de software, entre los que se incluye el propio Microsoft Project, software usado para la realización de todo el análisis de viabilidad y la gestión del proyecto. Para el cálculo de las amortizaciones se considera un tiempo de amortización de 3 años, el cual traducido en horas vendría a ser 4800 horas ($3\ años\ *\ 200\ días\ laborables\ *\ 8\ horas\ =\ 4800\ horas$).

Concepto	Coste
Ander Granado	14,00 €/h

Tabla 2.4.: Recursos de trabajo

Concepto	Coste
Ordenador portátil	700,00 €
BQ Aquaris M5	250,00 €

Tabla 2.5.: Recursos materiales (hardware)

Concepto	Coste	Numero de licencias
Windows 10	135,00 €[4]	1
Project 2016	1369,00 €[5]	1
WBS Chart Pro	187,50 €[6]	1
Ubuntu 16.04	0,00 €	1
TeX Studio	0,00 €	1
Android Studio	0,00 €	1
NMap	0,00 €	1
Git	0,00 €	1
GitHub	0,00 €	1

Tabla 2.6.: Recursos materiales (software)

Concepto	Trabajo (h)	Trabajo horas extra	Coste	Coste horas extra	Importe
Ander Granado	285	0	14,00 €/h	0	4.044,00 €

Tabla 2.7.: Costo de recursos de trabajo

Concepto	Unidades	Coste	Importe
Ordenador portátil	1	1,50 €/uso	29 x 1,50 € = 43,50 €
BQ Aquaris M5	1	1,50 €/uso	7 x 1,50 € = 10,50 €
TOTAL			54,00 €

Tabla 2.8.: Costo de recursos materiales

Concepto	Coste unitario	T. de amortización	C. U. amortización	T. de uso	Importe
Ordenador portátil	700,00 €	4800 horas	0,145833 €	285 h	41,57 €
BQ Aquaris M5	250,00 €	4800 horas	0,052083 €	25 h	1,30 €
Windows 10	135,00 €	4800 horas	0,028125 €	180 h	5,06 €
Microsoft Project 2016	1.369,00 €	4800 horas	0,285208 €	25 h	7,13 €
WBS Chart Pro	187,50 €	4800 horas	0,039062 €	5 h	0,19 €
Ubuntu 16.04	0,00 €	4800 horas	0,000000 €	105 h	0,00 €
TeX Studio	0,00 €	4800 horas	0,000000 €	40 h	0,00 €
Android Studio	0,00 €	4800 horas	0,000000 €	105 h	0,00 €
NMap	0,00 €	4800 horas	0,000000 €	30 h	0,00 €
Git	0,00 €	4800 horas	0,000000 €	145 h	0,00 €
GitHub	0,00 €	4800 horas	0,000000 €	145 h	0,00 €
TOTAL					55,25 €

Tabla 2.9.: Amortizaciones de hardware y de software

Concepto	Importe
Recursos de Trabajo (R.T.)	4.044,00 €
Recursos Materiales (R.M.)	54,00 €
Costo fijo	0,00 €
Amorizaciones	55,25 €
SUMA	4.153,25 €
Gastos generales (10 %)	415,32 €
Beneficio (15 %)	622,99 €
SUBTOTAL	5.191,56 €
IVA (21 %)	1.090.23 €
TOTAL	6281.79 €

Tabla 2.10.: Total presupuesto

Con esto se concluye que, tal como figura en la tabla 2.10, el coste del proyecto asciende a la cantidad de seis mil doscientos ochenta y cinco con setenta y nueve euros (6281.79 €).

2.4. Gestión de riesgos

En este apartado se identifican y analizan las diferentes amenazas que puedan llegar a impedir el correcto desarrollo del proyecto, haciendo que éste se retrase. Para ello primero se identificarán los diferentes riesgos que puedan existir y se indicará su peligrosidad. La peligrosidad es un valor cualitativo que indica cuánto puede afectar ese riesgo al proyecto.

Se han identificado los siguientes riesgos, los cuales se muestran en la tabla 2.11.

Riesgo	Peligrosidad
Pérdida de información	Alta
Enfermedades	Alta
Dificultades en la implementación de la aplicación	Alta
Dedicación no exclusiva al trabajo	Media
Averías o problemas técnicos con los recursos materiales	Media
Cambios o ampliación de requisitos	Media
Planificación muy optimista	Media

Tabla 2.11.: Enumeración de riesgos del proyecto

2.4.1. Explicación y plan de contingencia

Tras haber identificado los riesgos inherentes al proyecto en la tabla 2.11, se hace mayor hincapié en los detalles de dicho riesgos, elaborando una descripción más detallada y analizando su probabilidad. La probabilidad de cada riesgo se muestra de manera cualitativa, ya que aportar un valor numérico concreto en este tipo de casos resulta bastante complicado. También, junto a lo mencionado, se añade para cada riesgo un plan de contingencia. El plan de contingencia consiste básicamente en aportar medidas para afrontar dicho riesgo, tanto preventivas (para antes de que ocurra) como correctoras (para en caso de ocurrir).

PERDIDA DE INFORMACIÓN

Descripción: Podría darse el caso de que parte de la información, como la memoria del proyecto, el estado del arte o el código de la aplicación se perdieran.

Probabilidad: Baja.

Peligrosidad: Alta.

Medidas preventivas: Uso de herramientas de control de versiones como Git junto a uso de herramientas cloud como GitHub o Dropbox.

Medidas correctoras: Recuperación de la información mediante herramientas de análisis de unidades.

ENFERMEDADES

Descripción: Podría suceder que el único recurso de trabajo contrajera una enfermedad o tuviera un accidente.

Probabilidad: Baja.

Peligrosidad: Alta.

Medidas preventivas: Ninguna.

Medidas correctoras: Usar horas fuera del calendario para corregir el retraso en el proyecto.

DIFICULTADES EN LA IMPLEMENTACIÓN DE LA APLICACIÓN

Descripción: Podría suceder que, durante la implementación de la aplicación, se dieran dificultades a nivel de programación a la hora de cumplir con los requisitos funcionales.

Probabilidad: Media.

Peligrosidad: Alta.

Medidas preventivas: Una formación sólida en las herramientas y tecnologías que se van a usar para el desarrollo de la aplicación.

Medidas correctoras: Replantear las tareas posteriores y dedicar un esfuerzo extra para el aprendizaje y refuerzo de las herramientas usadas.

DEDICACIÓN NO EXCLUSIVA AL TRABAJO

Descripción: Podría suceder que, debido a exámenes u otros asuntos personales, el desarrollador no pudiera aportar toda la dedicación que requiere el proyecto.

Probabilidad: Media.

Peligrosidad: Media.

Medidas preventivas: No previsible.

Medidas correctoras: Cambiar calendario de trabajo aumentando las horas para subsanar los retrasos producidos por dicho riesgo.

AVERÍAS O PROBLEMAS TÉCNICOS CON LOS RECURSOS MATERIALES

Descripción: Podría darse el caso de que alguno de los recursos materiales de hardware, como el ordenador o el teléfono móvil, sufran algún tipo de avería.

Probabilidad: Baja.

Peligrosidad: Media.

Medidas preventivas: No previsible.

Medidas correctoras: Uso de otros dispositivos para continuar con el desarrollo del proyecto, adquiridos mediante compra o préstamo.

CAMBIOS O AMPLIACIÓN DE REQUISITOS

Descripción: Podría darse el caso en el que, tras la conclusión de la Fase 1, se cambiarán requisitos funcionales de la aplicación a desarrollar o se añadirán nuevos requisitos funcionales.

Probabilidad: Media - Alta.

Peligrosidad: Media.

Medidas preventivas: Revisión constante de los requisitos funcionales durante todas las fases del proyecto para minimizar el impacto que pueda causar los cambios en ellos.

Medidas correctoras: Adaptar el calendario las tareas posteriores y el calendario de trabajo actual.

PLANIFICACIÓN MUY OPTIMISTA

Descripción: Puede darse el caso de que la planificación elaborada para el proyecto sea demasiado optimista y no haya tenido en cuenta ciertos aspectos más concretos del desarrollo del proyecto.

Probabilidad: Media.

Peligrosidad: Media.

Medidas preventivas: No previsible.

Medidas correctoras: Adaptar, y en caso de que fuera necesario retrasar, la fecha final del proyecto para dar cabida en la planificación toda esa duración extra no prevista.

II

Fase 1: Estado del Arte de la Seguridad Informática

3

Conceptos Generales

"El único sistema verdaderamente seguro es aquel que se encuentra apagado, enterrado en un bloque de hormigón en una habitación sellada con plomo y vigilada por guardias armados. Incluso entonces tendría mis dudas"

— Gene Spafford

Tal y como Gene Spafford asegura con sus palabras [7], la seguridad informática no es una ciencia infalible y el sistema verdaderamente seguro es una quimera, algo que no existe y es inalcanzable. Con esto de base, la seguridad informática tiene diversos objetivos que surgen de la necesidad de proteger la información y los sistemas informáticos cada vez más en auge. Objetivos como minimizar y gestionar los riesgos y detectar los posibles problemas y amenazas de seguridad, garantizar la utilización adecuada del sistema y la información, limitar las pérdidas y conseguir a la adecuada recuperación del sistema en caso de un incidente de seguridad, cumplir con el marco legal los requisitos impuestos por los clientes en sus contratos, etc.

Existen numerosas definiciones para el término seguridad informática. En esencia, la seguridad informática es el área de la informática que se enfoca tanto en la protección de sistemas como en la protección de la propia información.

Antes de nada, resulta necesario distinguir entre dos conceptos. Por una parte tenemos el concepto es seguridad informática, y por otra parte el concepto de seguridad de la información. Aunque al principio ambos conceptos pueden parecer sinónimos, se tratan de áreas diferentes.

3.1. Seguridad Informática

Existen un gran número de definiciones para el concepto de seguridad informática. Según INTECO [8], (Instituto Nacional de Tecnologías de la Comunicación, ahora INCIBE) la seguridad informática consiste en la protección de las infraestructuras TIC que soportan un negocio o empresa. En la norma ISO 7498 [9], en la que se recoge el modelo OSI (modelo de referencia creado en 1980 para arquitecturas de red, en contraposición a la heterogeneidad del modelo

TCP/IP), se define la seguridad informática como una serie de mecanismos que minimizan la vulnerabilidad de bienes y recursos en una organización.

Aunque existen numerosas definiciones para el concepto de seguridad informática, por lo general engloban el carácter de protección de cualquier tipo de recurso tecnológico informático, muchas veces supeditado a su uso en una organización, aunque no tiene porqué ser así.

3.2. Seguridad de la Información

A la par que el concepto de seguridad informática está el concepto de seguridad de la información. Según INTECO [8], la seguridad de la información consiste en la protección de los activos de información fundamentales para el éxito de cualquier organización.

En una definición más antigua, elaborada en la norma ISO/IEC 17799 [10] se define la seguridad de la información como la preservación de la CID, acrónimo de "Confidencialidad, Integridad y Disponibilidad" (en inglés CIA, "Confidentiality, Integrity, Availability"). El concepto de CID es uno de los pilares de la seguridad de la información, en el que se ahondará más adelante.

Mientras que la seguridad informática se encarga de proteger las infraestructuras, es decir, se basa en un plano más técnico, la seguridad de la información no tiene porque hacer referencia a ningún tipo de tecnología informática o de comunicación. Los activos de información pueden ser digitales, como correos electrónicos, páginas web, imágenes, bases de datos, etc. pero no tiene por qué ser necesariamente así. También se pueden clasificar como activos de información documentos en papel contratos, faxes, etcétera.

Al igual que ocurre con la seguridad informática, es habitual tratar el término de la seguridad de la información a nivel empresarial, cuando no está exclusivamente ligado a ese campo.

3.3. Servicios de la Seguridad de la Información

La seguridad de la información proporciona una serie de servicios que tienen como objetivo proteger todo tipo de activos de información, ya sea de una organización o de un usuario particular. La suma y complementación de estos servicios es lo que permite esa protección de la información.

3.3.1. Confidencialidad, Integridad y Disponibilidad

Dentro de estos servicios, existe el acrónimo CID ()Confidencialidad, Integridad, Disponibilidad). CID engloba los 3 servicios de la seguridad de la información existentes más importantes [11].

Figura 3.1.: Confidencialidad, Integridad y Disponibilidad

- *Confidencialidad*: garantiza que la información no se pone a disposición ni se revela a individuos, entidades o procesos no autorizados.
- *Integridad*: garantiza que la información no se modifica malintencionadamente, por parte de individuos o procesos, durante su procesado o su transmisión, y en caso de modificarse, permite detectar dichas modificaciones.
- *Disponibilidad*: garantiza el acceso y utilización de la información y los sistemas de tratamiento de la misma por parte de los individuos, entidades o procesos autorizados cuando lo requieran. Además garantiza la recuperación de la información y de los sistemas de información en caso de posibles incidentes.

3.3.2. Otros servicios

Además de CID, existen otros servicios que complementan el concepto de seguridad informática. Aunque CID consiste en tres servicios que son de obligado cumplimiento para garantizar la seguridad de la información, no resultan suficientes, ya que no cubren todo tipo de casos en los que se puede comprometer la información. Entre el diverso número que existen, se enumera y define una serie de ellos, considerados de los más relevantes [12].

- *Autenticación:* garantiza que la identidad del creador de un mensaje es legítima. Permite asegurar la autoría de la información creada o modificada.
- *No repudio:* permite, mediante diferentes mecanismos, demostrar la autoría de un mensaje e impide que el usuario niegue esa circunstancia.
- *Autorización:* permite controlar el acceso a cierto sistema o información por parte de un usuario, permitiendo dicho acceso sólo a ciertos usuarios previamente designados, una vez superado el servicio de autenticación.
- *Auditabilidad:* permite registrar y monitorizar la utilización de los distintos recursos del sistema por parte de los usuarios para garantizar el correcto uso del sistema y de su información.

- *Anonimato:* permite garantizar el anonimato de los usuarios que acceden a los recursos y consumen determinados tipos de servicios, preservando así su privacidad. Puede entrar en conflicto con otros ya mencionados, como la autenticación o la auditoría del acceso a los recursos.
- *Protección a la réplica:* impide que se haga uso de ataques de repetición que engañen al sistema provocando operaciones y modificaciones de la información no deseadas.

4

Aplicaciones de la Seguridad Informática

"Las organizaciones gastan millones de dólares en firewalls y dispositivos de seguridad, pero tiran el dinero porque ninguna de estas medidas cubre el eslabón más débil de la cadena de seguridad: la gente que usa y administra los ordenadores"

— Kevin Mitnick

Con el objetivo de garantizar todos esos servicios mencionados en el capítulo anterior, se han desarrollado una gran cantidad de áreas dentro de la propia seguridad informática, que permiten cumplir con estos objetivos. Cada área tiene sus diferentes aplicaciones y se enfoca a cumplir los objetivos de la seguridad informática.

Las áreas de investigación y los campos de trabajo dentro de la seguridad informática avanzan a un ritmo muy grande debido al propio avance de la informática. El desarrollo de áreas como los smartphones o el Internet of Things (IoT) hace necesario nuevas técnicas que permitan garantizar la seguridad de la información a todos los niveles. Áreas como el ransomware, los wearables, los automóviles o el ciberespionaje son algunas en las que más hincapié se está haciendo en los últimos años [13].

4.1. Malware

La seguridad informática siempre se ha relacionado con el concepto de virus informático. Si bien uno de los objetivos de la seguridad informática es evitar que este tipo de software consiga acceder a información o dañar sistemas, los virus informáticos no son la única manera que existe para comprometer la seguridad de la información. Aun así el campo de la seguridad informática se desarrolla después de que surgieran estas primeras piezas de malware y, debido al surgimiento de estas, se comprende la necesidad de desarrollar toda una serie de técnicas para garantizar la seguridad de la información almacenada y procesada en estos sistemas.

El concepto de malware es relativamente novedoso. El origen de programas capaces de replicarse y distribuirse se remonta hacia 1949, cuando el propio Von Neumann expuso *La*

Teoría y Organización de Autómatas Complejos [14], en la cual habla sobre pequeños autómatas capaces de replicarsepor si solos. Dos décadas más adelante, concretamente en 1971, nace el denominado el primer virus informático de la historia, llamado *Creeper* [15] (enredadera en inglés). Dicho virus no se puede considerar malware como tal ya que no causaba daño a los sistemas en los que se replicaba, simplemente mostraba un inofensivo mensaje a los usuarios (*"Soy una enredadera... ¡atrápame si puedes!"*). Para contrarrestar dicho virus surgió *Reaper* (segadora en inglés), el cual fue el primer antivirus de la historia. Sin embargo no es hasta 1984 cuando Frederick B. Cohen acuña por primera vez el término virus informático en uno de sus estudios definiéndolo como *"Programa que puede infectar a otros programas incluyendo una copia posiblemente evolucionada de sí mismo"*, haciendo una analogía con el mundo de la biología.

No es hasta la decada de los 80, en la cual los ordenadores personales comenzaban a tener popularidad, cuando se comienzan a desarrollar virus informáticos a los que si que podemos considerar maliciosos. Ejemplos famosos son el virus *Viernes 13* en 1987 o el gusano *Happy* en 1999 [16]. Desde la década de los 80 y hasta la actualidad se han ido desarrollando este tipo de virus informáticos, los cuales podemos clasificar como malware. Cabe mencionar que el malware en sí no tiene porque tener la capacidad para replicarse; se puede considerar malware cualquier pieza de software que busque un uso malintencionado o malicioso del sistema.

Dentro del malware se han ido desarrollando diferentes piezas de software que tienen sus particularidades. Dependiendo de su funcionamiento o de los usos para los que esté diseñado el malware se puede clasificar en diferentes categorías, algunas de las cuales tienen mayor auge que otras a día de hoy. En la figura 4.1 se puede observar una serie de datos del crecimiento y descenso de diferentes tipos de malware durante los últimos años.

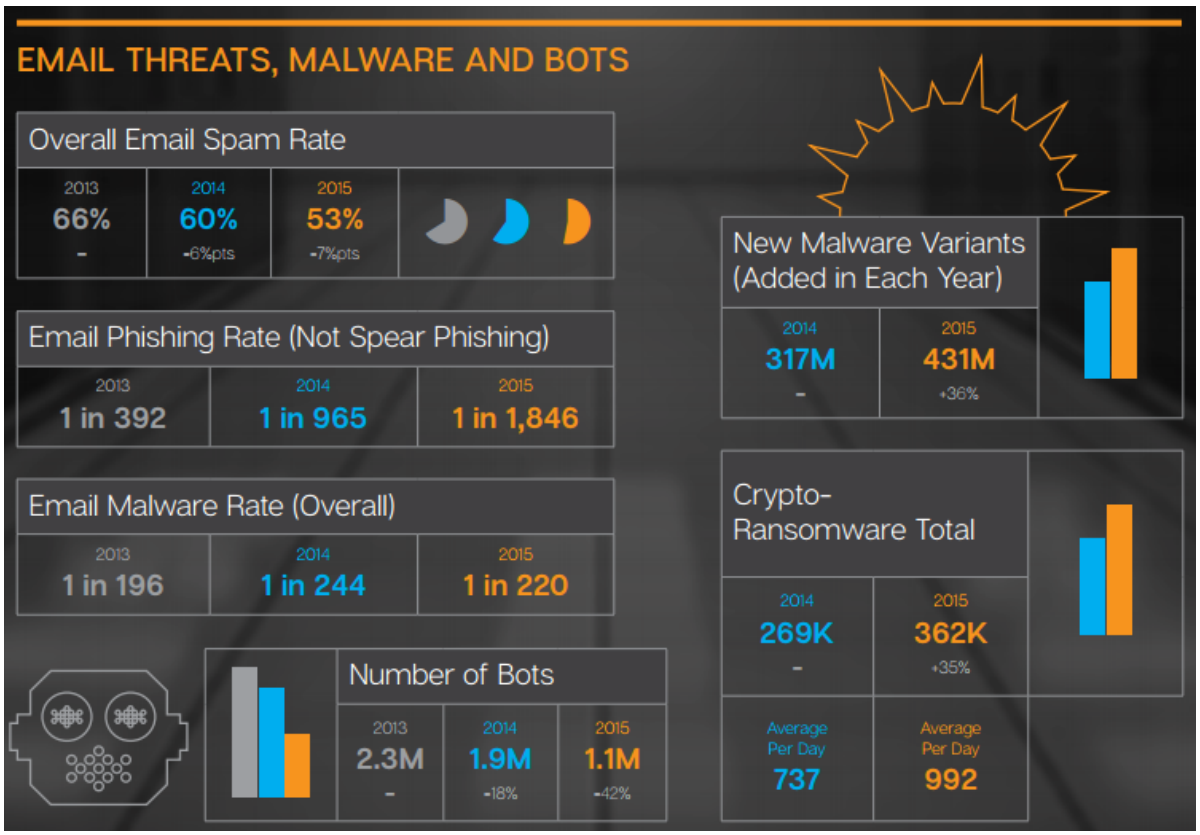

Figura 4.1.: Infografía sobre el malware en 2016 [17]

4.1.1. Gusanos

Una de las primeras categorías de malware que surge es la de los gusanos. Los gusanos (también conocidos como *worms* por su término inglés) son programas autorreplicantes que, en vez de infectar archivos concretos, se instalan directamente en los sistemas y buscan, mediante diferentes vulnerabilidades en los sistemas o técnicas como la ingeniería social, replicarse en la medida en la que les sea posible.

Fueron uno de los primeros tipos de malware y fueron una de las causas principales del desarrollo en los campos de la seguridad informática mas defensiva. Un ejemplo es el famoso gusano *ILOVEYOU* [18], el cual infectó en el año 2000 a decenas de millones de ordenadores con Windows. Otros también famosos son *Code red*, *Sasser* o *Conficker*. Son uno de los tipos de malware que mas daños han causado, causando daños estimados en miles de millones de euros en los casos mencionados.

Aun así, a día de hoy, aunque siguen considerándose una amenaza, han quedado relegados a un segundo plano, por una parte, por la capacidad de los sistemas de firewall o sistemas antivirus de detectarlos y neutralizarlos, y por otra parte, por el surgimiento de otros tipos de malware, como los que se menciona en los siguientes puntos.

4.1.2. Troyanos

Otro tipo de malware famoso son los troyanos. El nombre de troyano proviene del caballo de Troya, una gran estructura de madera que usaron los griegos en la guerra de Troya para introducir a sus soldados en la ciudad, que estaba completamente fortificada. De la misma manera, el malware denominado troyano se camufla, es decir, se oculta como un programa legítimo imitando el comportamiento de dicho programa, cuando en realidad son una forma para dar acceso a un atacante a los recursos de dicho sistema. Uno de los troyanos mas famosos fue *Zeus* [18], que llego a infectar mas de 1 millón de ordenadores. También existe otros como, por ejemplo, *CryptoLocker*, que es una mezcla entre troyano y ransomware, tipo de malware que se explica a continuación.

4.1.3. Ransomware

Según McAfee Labs [13] (actualmente parte de Intel) el ransomware es una de las mayores amenazas a día de hoy. El ransomware es un tipo de malware informático que busca beneficio económico en base a la extorsión de los usuarios. Para ello este clase de malware cifra los archivos de los usuarios para que sean inaccesibles para ellos, de tal manera que solo mediante el pago de cierta cantidad económica puedan recuperar dichos archivos.

Existen varias familias de ransomware, entre los que destacan *CryptoWall 3*, *CTB-Locker* o *CryptoLocke*. El concepto de familias proviene de que en su mayoría se trata del mismo malware con prácticamente el mismo funcionamiento, pero con ligeras variaciones es su código que les permite eludir las posibles medidas de seguridad. En la figura 4.2 se puede observar su crecimiento durante estos últimos años.

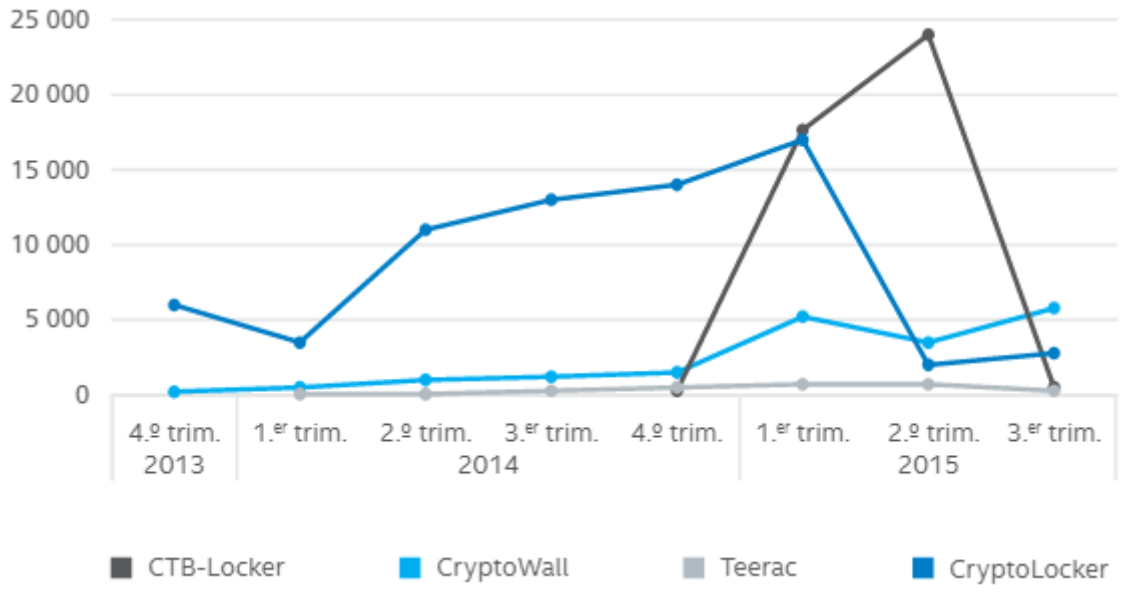

Figura 4.2.: Nuevas muestras de familias de ransomware prominente [13]

4.1.4. Rootkits

Los rootkits son una clase de malware que tiene como objetivo principal escalar privilegios en el sistema, también conocido como conseguir acceso de superusuario o *root*. No hace falta mencionar que con dichos privilegios el daño que puede causar en un sistema es inconmensurable, pudiendo dejarlo completamente inutilizado o eliminar absolutamente toda su información.

4.1.5. RATs

Los *Remote Administration Tools* también conocidos como RAT, buscan ofrecer a los posibles atacantes una puerta trasera para acceder a los sistemas para así poder introducir otros tipos de malware específicos.

4.1.6. Spyware

Un tipo de malware concreto bastante popular desde la década del 2000 es el denominado Spyware, cuyo objetivo es obtener información de las acciones de un sistema, infectando la máquina destino de manera inadvertida para el usuario, el cual desconoce que su actividad y sus datos están siendo monitorizados.

4.1.7. Keyloggers

El *Keystroke Logging*, mejor conocido como Keylogger, es otro tipo de malware usado actualmente debido a su efectividad que tiene obtener la información de las pulsaciones que se realizan en un teclado de un sistema concreto. Este mecanismo tiene como principal objetivo extraer información, especialmente contraseñas, de un usuario para después poder acceder a los servicios que usa dicho usuario. Se puede encontrar Keyloggers tanto a nivel físico, en forma de pequeños dispositivos [1][2] USB con una memoria interna, como a nivel de software, estos últimos especialmente sencillos de implementar con solo unas pocas lineas de código [19].

4.2. Dispositivos móviles

El malware afecta a los sistemas y es un riesgo para la seguridad de estos, pero no es el único. Además hay que tener en cuenta que en los últimos años el mundo de las tecnologías de la información ha sufrido cambios drásticos. ya no tenemos el modelo clásico de ordenadores personales y servidores estándar que conforman internet. Han entrado nuevas áreas que se han ido desarrollando. A medida que esta áreas han ido destacando, el campo de la seguridad

[1] https://www.keelog.com/

[2] https://store.hackaday.com/products/usb-rubber-ducky-deluxe

informática ha ido avanzando para satisfacer las necesidades de seguridad de estas áreas y adaptarse a ellas mediante nuevas técnicas y medidas de seguridad.

Una de las áreas que ha destacado estos últimos años es el área de la seguridad de dispositivos móviles. Datos como que el consumo de Internet a día de hoy es mayor en este tipo de dispositivos que en ordenadores convencionales resultan sorprendentes. En el caso concreto de España, del 78,7 % de la población que se conecta regularmente a internet, el 88,3 % de los usuarios lo hace a través de un smartphone [20]. Teniendo eso en cuenta, es lógico que la seguridad estos dispositivos sea una prioridad y también uno de los puntos de enfoque por parte de todo tipo de atacantes. Los smartphones tienen una gran cantidad de usos diferentes. Aparte de ser la principal herramienta de comunicación, también son dispositivos mediante los que se procesa una gran cantidad de información, desde información personal como fotos o mensajes a información relacionada con el ámbito empresarial como documentos o correos. Debido a que la información ya no solo se procesa y transmite desde ordenadores tradicionales, sino que en su mayoría se hace desde dispositivos móviles, resulta necesario definir e implantar técnicas y procedimientos de seguridad en este tipo de dispositivos.

4.2.1. Seguridad en smartphones

Aunque en esencia se trate de dispositivos basados en Linux, Unix o Derivados, este tipo de dispositivos tienen sus particularidades con respecto a los ordenadores personales. Lo primero, es que estos sistemas no disponen ni de la arquitectura x86/x86-64 (al menos en su gran mayoría), sino que se basan en procesadores con arquitectura ARM. Debido a esto y al enfoque de uso que tienen, usan diferentes sistemas operativos que distan de los tradicionales. Los sistemas operativos más usados para smartphones son Android e iOS, que se encuentran en un 86,22 % y 12,88 % de los dispositivos vendidos en 2016 [21], respectivamente.

Ambos son sistemas que, al estar por defecto mas limitados su uso y configuración, resultan *a priori* más seguros que los sistemas operativos de escritorio. Existen una serie de técnicas que se suelen aplicar independientemente del sistema operativo utilizado, como pueden ser la firma de aplicaciones o el uso de cifrado para ciertos casos. Aun así, este tipo de medidas son prácticamente obligatorias para dispone de sistemas mínimamente seguros. Lo que diferencia a cada uno de los sistemas son tanto el enfoque de seguridad que le dan al sistema como las técnicas o algunas medidas, técnicas o algoritmos o software concretos, que cambian de un sistema operativo a otro.

4.2.1.1. Seguridad en iOS

En dispositivos con iOS la seguridad se enfoca desde varios puntos. Por una parte desde su tienda de aplicaciones, estableciendo fuertes filtros a la hora de publicar aplicaciones para que a través de las aplicaciones no entre ningún tipo de malware.

Por otro lado, uno de los puntos más fuertes de iOS es el relacionado con en el cifrado. Debido a la ventaja que otorga el hecho de que la misma compañía elabore tanto el hardware como las capas de software de sus dispositivos, se pueden integras medidas de seguridad en el

propio hardware haciendo que funcionen con su software. Una de las más destacables es la incorporación del coprocesador llamado *Secure Enclave* [22]. Este coprocesador (disponible en el Apple S2, y en el Apple A7 y posteriores) se encarga de todas las operaciones criptográficas para la gestión de claves de cifrado de datos y mantiene la integridad de la protección de datos incluso si el kernel del sistema se ha visto comprometido. Usa memoria cifrada e incluye un generador aleatorio de números por hardware. La comunicación este este coprocesador y el procesador principal esta completamente aislada del resto del sistema.

Esto tiene dos ventajas. La primera, que el cifrado por hardware, al realizarse de manera transparente a cualquier capa de software, resulta díficilmente evitable. Por otra parte, a nivel de rendimiento, resulta mucho más eficiente que cualquier tipo de cifrado por software.

Además, todos los iOS tienen un sistema de cifrado mediante hardware y AES-256 que se sitúa entre la memoria principal y el almacenamiento flash [22], de tal manera que los datos almacenados en los dispositivos se mantienen cifrados sin que se pierda rendimiento en ello. Para cada sistema de cifrado se crea identificadores únicos tanto par cada usuario (UID) como para cada grupo distinto de dispositivos, ambos integrados en el propio hardware. Cada UID es único y no es conocido ni por Apple ni por ninguna aplicación ni usuario. Cada GID es común a una familia de dispositivos y se usa para distinguirlos entre sí. Esto hace que los datos que se almacenan sean altamente difíciles de descifrar.

Las operaciones de cifrado y descifrado concretas siguen el esquema que se muestra en la Figura 4.3 , haciendo uso también de una contraseña propia.

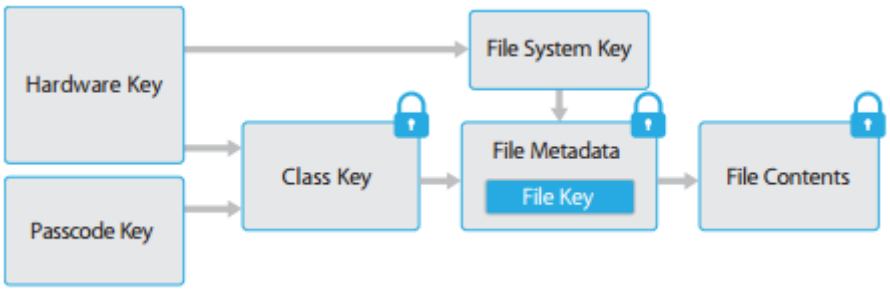

Figura 4.3.: Esquema de cifrado de un archivo en iOS

Aquí se puede profundizar mucho, de hecho en la guía de seguridad que la propia Apple publica [22] se profundiza mas, pero no se que mas se puede añadir como realmente relevante. Lo mismo, tampoco quiero dar la chapa, sino que se entienda el concepto.

4.2.1.2. Seguridad en Android

El modelo de seguridad en Android se diferencia del modelo de iOS en varios aspectos. Más que en un fuerte cifrado a todos los niveles, el cual también usa pero en menor medida, el enfoque se basa en hacer lo más segura posible cada aplicación. Esto es lógico teniendo

en cuenta que, por un lado, la tienda de aplicaciones de Android, la Play Store[3], es menos estricta que la de iOS y que, por otro lado, Google (propietaria de Android) no fabrica todos los dispositivos que integran dicho sistema. Por ello, la arquitectura interna del sistema resulta más jerárquica, y es fundamental para las diversas medidas de seguridad que se implementan. En la Figura 4.4 se muestran las diferentes capas de dicha arquitectura.

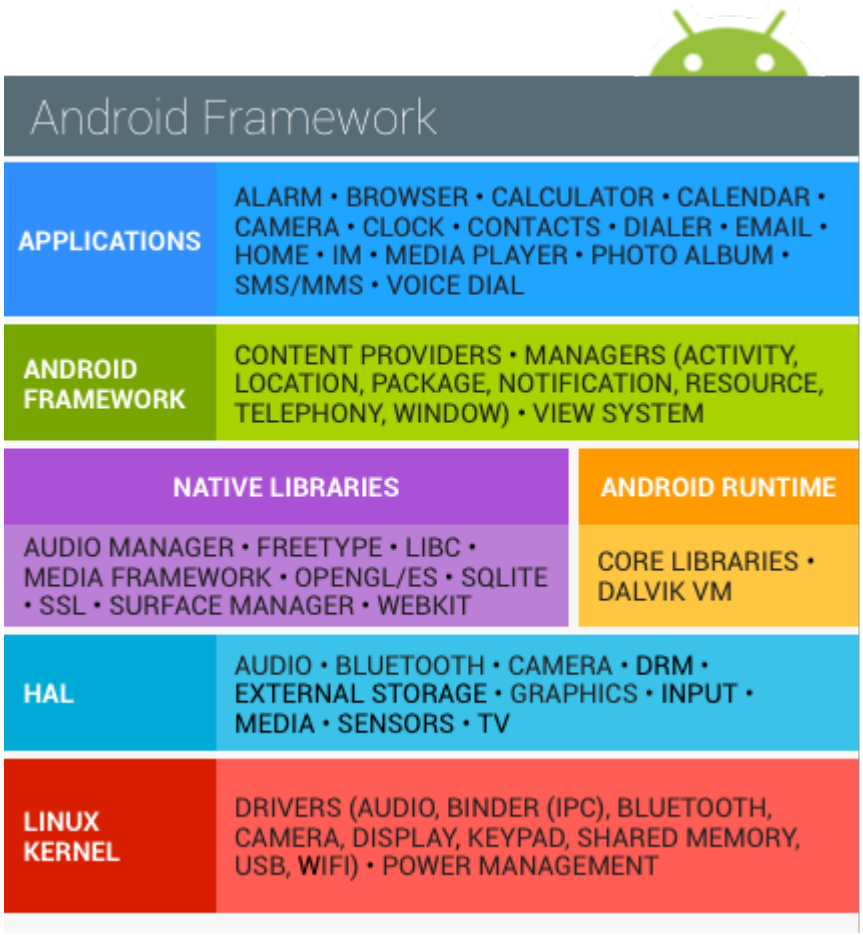

Figura 4.4.: Las diferentes capas que componen la arquitectura de Android

La esencia de la seguridad en Android reside en controlar y limitar la interacción entre las diferentes capas para evitar fallos de seguridad. Para ello se hace uso de diferentes técnicas [23] [24]:

- Técnicas como el *sandboxing*, donde el código de cada aplicación se ejecuta de manera aislada al resto de aplicaciones y, además, los datos de cada aplicación se mantienen separados [24].
- Un framework de aplicaciones que implementa ciertas técnicas comunes de seguridad como pueden ser métodos criptográficos, sistemas de permisos o una comunicación entre procesos (*IPC, Interprocess Communication*) segura.

[3] `https://play.google.com/store/apps`

- Implementar diversas tecnologías como ASLR (Address Space Layout Randomization), NX (No-eXecute), *ProPolice*, *safe_iop, OpenBSD dlmalloc*, *OpenBSD calloc*, y *Linux mmap_min_addr* para evitar errores de *buffer overflow*, para evitar que se ejecuten instrucciones concretas o mitigar otro tipo de riesgos relacionados con la gestión de la memoria.
- Un sistema de archivos cifrado mediante AES-256 para proteger los datos almacenados en los dispositivos.
- Un sistema de permisos tanto a nivel de usuario como al nivel de cada aplicación concreta para controlar las acciones de los usuarios y las aplicaciones.

Ente enfoque hace que, por una parte, el propio Android mediante su arquitectura y el uso de una VM (Virtual Machine) específicamente creada para el sistema (como se puede observar en la Figura 4.4) ofrezca un repertorio de mecanismos de seguridad que hacen de él un sistema relativamente seguro. Por otra parte, también pone empeño en que los desarrolladores tienen la necesidad de implementar medidas de seguridad en sus aplicaciones para garantizar la seguridad de las mismas. Para ello, el SDK de Android proporciona acceso a un gran numero de utilidades como, entre otras, utilidades de cifrado, de manejo de credenciales o de intercambio de mensajes.

Aquí si que vendría mejor, quizás, profundizar un poco más.

4.3. Internet of Things

También junto a la seguridad de los dispositivos móviles, que viene determinada por el auge de dichos dispositivos, el área de la seguridad del Internet of Things (conocida como IoT) es una de las áreas que más impulso está teniendo, debido al mismo motivo. Cada vez hay una mayor cantidad de dispositivos conectados a Internet, que no ha hecho más que crecer exponencialmente debido al IoT. Dispositivos que ya no se tratan de móviles, tablets u ordenadores, sino de otro tipo de dispositivos, como electrodomésticos, que han sido dotado de capacidad de computación y conexión a internet para ampliar sus capacidades y poder recopilar información de ellos. En esencia, el IoT se basa en una serie de dispositivos a los que, mediante el uso de sensores, procesadores y conexión a internet, se les da la capacidad de recoger y transmitir información a otros dispositivos, además de recibir dicha información para actuar de una u otra manera.

4.3.1. Aplicaciones

Las aplicaciones de lo que se denomina Internet of Things son amplísimas. Prácticamente cualquier sistema electrónico que disponga de una serie de sensores mediante los cuales obtiene información puede ver multiplicadas sus capacidades añadiéndole una conexión a la red. De esta manera se el dota de capacidad para ya no solo transmitir información sino incluso poder interactuar de forma mas compleja con otra serie de sistemas [25] [26].

- *Smart homes:* Mediante tecnologías IoT se puede dotar a los diferentes elementos domóticos de un hogar o edificio de capacidades para reaccionar de manera automática a diferentes cambios que se puedan dar. Desde controlar la luz o el agua hasta controlar la temperatura o las persianas, todo de tal manera que los subsistemas sean capaces de comunicarse y actuar de manera conjunta y actuar en función de la información que reciben.
- *Smart Cities:* El término Smart Cities se refiere al sistema ciberfísico que conforma una red avanzada de comunicación para optimizar el uso de los sistemas físicos de una ciudad. Puede optimizar servicios como el tendido eléctrico mediante el análisis de consumo de toda la red eléctrica o la red de carreteras mediante sistemas avanzados de control de tráfico (del cual los vehículos autónomos se pueden beneficiar). Son solo unos breves ejemplos de lo que un sistema de este tipo puede llegar a optimizar dentro de los procesos que se dan en una ciudad.
- *Monitorización medioambiental:* El IoT también puede servir para monitorizar cambios en el medio ambiente e integrar todo esa adquisición de datos para que funcione de manera conjunta con otros sistemas. Una serie de sistemas de tiempo real a los que se añade la capacidad de comunicarse e interactuar entre sí pueden ser una plataforma sólida mediante la cual detectar y monitorizar anomalías que afecten a la vida humana, animal y vegetal. Además, pueden ser una herramienta fundamental para obtener información que permita mitigar los efectos de la contaminación de cualquier tipo.
- *Sanidad:* La sanidad también es uno de esos sectores que se puede beneficiar de lo avances en el IoT. Por ejemplo, la capacidad para monitorizar diferentes parámetros, como la temperatura corporal, la presión sanguínea o la actividad cardíaca de manera constante pueden ser una herramienta fundamental a la hora de prevenir y tratar diversas enfermedades. Además del nivel patológico, se puede usar diferentes dispositivos para tomar información que permita mejorar el estilo de vida y la salud de las personas. Dispositivos como los wearables son un ejemplo de ello.
- *Smart Business:* El término Smart Business hace referencia a todas esas aplicaciones del Internet of Things que se usan para mejorar tanto la logística interna como los productos y servicios que ofrece una empresa o negocio. La gestión de inventario tiempo real o la automatización del envío de información sobre productos para ofrecer un mejor servicio post venta pueden ser buenos ejemplos de ello.
- *Seguridad y vigilancia:* Aunque pueda ser polémico, el Internet of Things también se puede aplicar en el área de la seguridad y vigilancia. Los sistemas interconectados de cámaras o los sensores para detectar químicos nocivos son dos ejemplos. El primero de ellos no se respeta la privacidad del usuario, mientras que el segundo sí. Debido a eso, aplicaciones como la primera son objeto de numerosas críticas por parte de diversas organizaciones.

4.3.2. Seguridad en Internet of Things

Una de los principales problemas del Internet of Things reside en que ya no solo se trata de el peligro de que la información se vea comprometida o revelada sino que, al tratarse de otra serie de dispositivos, los peligros son mucho mayores [27][28]. Esto es algo de lo que el mundo de la seguridad de la informática se ha dado cuenta. Estimando que el numero de dispositivos conectados a Internet pasara de 6,4 mil millones en 2016 a 21 mil millones en 2020 [29] resulta fundamental indagar en ello.

A día de hoy existen gran cantidad de casos actualidad en los que diferentes dispositivos salen al mercado con esas capacidades para conectarse Internet pero que carecen de prácticamente ningún tipo de medida de seguridad. Se han dado casos como el de Miele[30] donde a causa de la falta de seguridad de un modelo de sus lavavajillas, se puede llegar a acceder a la red local privada a donde este conectada de manera sencilla. Otro caso es el de los automóviles Tesla [31], donde más de una vez se han descubierto vulnerabilidades que permiten obtener el control de sus vehículos y conducirlos remotamente a placer del atacante. Esto es gravísimo, pudiendo llegar a poner en en riesgo la vida de personas. Por ello, los servicios de la seguridad informática son mas necesarios si cabe para este campo.

4.4. Cloud Computing

El Instituto Nacional de Estándares y Tecnologías de Estados Unidos, conocido como NIST (National Institute of Standards and Technologies) define el Cloud Computing como un modelo para hacer posible el acceso a red adecuado y bajo demanda a un conjunto de recursos de computación configurables y compartidos (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios...) cuyo aprovisionamiento y liberación puede realizarse con rapidez y con un mínimo esfuerzo de gestión e interacción por parte del proveedor de dicha nube [32].

Las estructuras cloud se pueden clasificar en tres categorías: públicas, comunitarias y privadas. En las primeras el acceso a sus servicios es abierto, mientras que en las segundas está acotado a un determinado entorno, normalmente una empresa. En la última, aunque no se tiene acceso público a sus servicios, estos se comparten entre varias entidades.

4.4.1. Tipos de servicios

El Cloud Computing puede ofrecer una serie de servicios a sus usuarios, y en función de qué servicio se ofrezca, se clasifica en una de las tres categorías diferentes existentes que se mencionan a continuación.

4.4.1.1. Software as a Service

Software as a Service consiste en un despliegue de software en el cual las aplicaciones y los recursos computacionales se han diseñado para ser ofrecidos como servicios de funcionamiento bajo demanda, con estructura de servicios llave en mano. De esta forma se reducen los

costes tanto de software como hardware, así como los gastos de mantenimiento y operación. En esta categoría las consideraciones sobre seguridad son responsabilidad del proveedor del servicio, estando limitada la configuración que puede realizar el suscriptor.

4.4.1.2. Platform as a Service

Platform as a Service es el servicio donde se ofrece una plataforma en la cual es suscriptor del servicio puede desplegar su propio software. Con ello se reducen los costes y la complejidad de la compra, el mantenimiento, el almacenamiento y el control del hardware y el software que componen la plataforma. El suscriptor del servicio tiene control parcial sobre las aplicaciones y la configuración del entorno ya que la instalación de los entornos dependerá de la infraestructura que el proveedor del servicio haya desplegado. La seguridad se comparte entre el proveedor del servicio y el suscriptor ya que el suscriptor aporta su propio software.

4.4.1.3. Infrastructure as a Service

Infrastructure as a Service es un modelo en el cual la infraestructura (servidores, software y equipamiento de red) es gestionada por el proveedor como un servicio bajo demanda, en el cual se pueden crear entornos para desarrollar, ejecutar o probar aplicaciones, El fin principal de este modelo es evitar la compra de recursos por parte de los suscriptores, ya que el proveedor ofrece estos recursos como objetos virtuales accesibles a través de un interfaz de servicio. El suscriptor mantiene generalmente la capacidad de decisión del sistema operativo y del entorno que instala. Por lo tanto, la gestión de la seguridad corre principalmente a cargo del suscriptor.

4.4.2. Seguridad en Cloud Computing

Debido al auge del Cloud Computing surge la CSA (Cloud Security Alliance), se define como una organización internacional sin ánimo de lucro para promover el uso de mejores prácticas para garantizar la seguridad en la nube. Entre otras actividades, la CSA elabora un informe periódico con las mayores amenazas de seguridad de sistemas de Cloud Computing [33]. Estas amenazas se actualizan regularmente buscando el consenso de los expertos. A continuación, se resumen las amenazas descritas en este informe. En el informe de 2016, la CSA marca como amenazas más graves, ordenadas por orden de importancia, los siguientes 12 puntos:

1. Brechas de Datos
2. Manejo inseguro de Credenciales de Acceso
3. Interfaces y APIs inseguros
4. Vulnerabilidades de Sistemas y Aplicaciones
5. Secuestro de Cuentas
6. Amenazas Internas
7. Advanced Persistent Threats (APTs)

8. Pérdida de Datos
9. Diligencia insuficiente
10. Abuso y mal uso del Cloud Computing
11. Denegación de Servicio
12. Problemas derivados de las tecnologías compartidas

Con el auge del Cloud Computing, este tipo de amenazas son cada vez más graves. Desde servicios de almacenamiento de datos hasta plataformas de despliegue de aplicaciones, cada vez más empresas y usuarios delegan almacenamiento, infraestructura, o software en servicios de Cloud Computing, por lo que, en primer lugar, ser conscientes de la ventajas, inconvenientes y, sobre todo, riesgos que acarrea el uso de estos servicios resulta fundamental para después, en segundo lugar, ser capaces de securizar lo más posible y hacer un uso seguro de esto servicios.

En esta parte si que parece que le falta algo, o al menos me da a mí la sensación, aunque tampoco quería ser redundante con lo que menciono en el capítulo de malware.

5 Pentesting

"Si piensas que la tecnología puede solucionar tus problemas de seguridad, está claro que ni entiendes los problemas ni entiendes la tecnología"

— Bruce Schneier

Dentro de en la seguridad informática un test de intrusión, o *pentesting*, como se conoce en inglés, evalúa los diferentes niveles de seguridad de un sistema informático red mediante la simulación en un entorno controlado de un ataque por parte de un usuario malicioso conocido comúnmente como hacker[34]. El propósito de una prueba de penetración es determinar la viabilidad de un ataque y la cantidad de impacto que puede causar este mismo.

5.1. Objetivos

A día de hoy el pentesting es una de las ramas más utilizadas para garantizar tanto la seguridad de la información como la propia seguridad informática de los sistemas de empresas u organizaciones. Es uno de los pilares básicos en las auditorías de seguridad informática, en las que empresas contratan a expertos otras empresas expertas en auditorías de seguridad para evaluar la fortaleza de sus sistemas informáticos.

El pentesting no es un área con unas técnicas de actuación concretas. El pentesting usa una grandísima cantidad de técnicas, a las cuales se van añadiendo técnicas que van surgiendo, para determinar la seguridad de sistema informático y, en particular, conocer las debilidades que tiene el sistema informático.

Haciendo un símil con otras áreas completamente diferentes, un pentesting podría equivaler a una serie de pruebas prácticas que se realizan, por ejemplo, en controles de calidad para todo tipo de productos, en los cuales, se ve que la mejor forma de probar su correcto funcionamiento es someterlos a una serie de pruebas. Dichas pruebas están basadas en los entornos reales en los que se utilizan esos productos. Al igual que la mejor manera de probar la efectividad de un chaleco antibalas es realizando disparos sobre él para ver hasta qué punto los materiales

como el Kevlar o de los tejidos que han sido compuestos para dicho chaleco son adecuados para parar una bala.

De la misma manera, la mejor forma de probar si un sistema informático o una red compuesta de varios sistemas es segura, o mejor dicho, hasta qué punto es segura, es elaborar ciertas pruebas e intentar explotar vulnerabilidades que puedan surgir en estos sistemas, Todo con el objetivo final de sacar conclusiones y en base a esas conclusiones mejorar y fortificar ese sistema.

Aunque, de la misma manera que jamás se harían pruebas con el chaleco antibalas comprometiendo la integridad de ninguna persona física, tampoco se elabora un pentesting comprometiendo de manera real ningún sistema. El pentesting, al igual que las pruebas del chaleco, se elaboran ambas en un entorno controlado y limitado.

Teniendo en cuenta que la seguridad de los sistemas informáticos es vital para la continuidad del negocio de una empresa u organización, y para el correcto resta desarrollo de sus actividades, y siendo el pentesting una de las mejores herramientas para garantizar dicha seguridad, hace que sea uno de los métodos más usados por todo tipo de empresas u organizaciones.

5.2. Partes

Dentro de un pentesting se diferencian diferentes etapas cada una de las cuales tiene objetivos particulares y concretos. Aunque algunas de estas partes puedan tener sentido de manera independiente, en su conjunto permiten un completo análisis de un sistema informático del cual sacar conclusiones que permitan preservar la seguridad.

A continuación, se enumeran y definen las fases de un test de intrusión o pentesting, que ya se encuentran estandarizadas en el PTES (*Penetration Testing Execution Standard*) [35]:

Figura 5.1.: Logo de Penetration Testing Execution Standard

1. *Reglas del juego, alcance y términos del test de intrusión:* en esta fase se establece una serie de protocolos entre el realizador del pentesting (normalmente una auditoría de seguridad informática) y el auditado (normalmente una empresa u organización). En esta fase se establecen los objetivos a los que llegar, los límites a los que tendrán que adherirse los auditores.

2. *Recolección de información:* fase en la que se obtiene información del sistema para enfocar posteriormente analizarla.
3. *Análisis de las vulnerabilidades:* fase en la que mediante la información obtenida, se pasa a determinar las vulnerabilidades del sistema.
4. *Explotación de las vulnerabilidades:* tras determinar esas vulnerabilidades, se pasa a intentar explotarlas para visualizar su gravedad y el daño real que pueden causar.
5. *Postexplotación del sistema:* tras haber logrado explotar una vulnerabilidad, se intenta minar lo más posible el sistema, intentando lograr un efecto en cadena para ver, hasta qué punto, se puede dañar un sistema mediante una vulnerabilidad concreta.
6. *Generación de informes:* tras todos los pasos anteriores, se condensa todo el proceso elaborado y las conclusiones a las que se ha llegado en informes, de tal manera que a través de esos informes, se pueda actuar para corregir las vulnerabilidades y fortalecer el sistema.

De todas esas fases, a continuación se profundizará en las más importantes.

5.3. Recogida de información

La información es poder. Esa frase, atribuida a Francis Bacon (aunque se desconoce si realmente llegó a pronunciarla en algún momento), no podría ser más cierta actualmente. La informática no es más que la ciencia que trata la información, y la seguridad informática es un área que se encarga de protegerla. Por ello, a la hora de realizar un pentesting, resulta esencial obtener la mayor cantidad de información posible. El éxito de muchos de los ataques e intrusiones que sufren empresas y organizaciones se debe en gran parte a la gran cantidad de información que directa e indirectamente un atacante es capaz de obtener sobre sus sistemas [36].

Por ello, la fase de recogida de información o information gathering resulta fundamental a la hora de realizar un test de intrusión. A mayor cantidad de información obtiene un atacante de un sistema mayor probabilidad de éxito tendrá al atacar.

La recogida de información, según desde qué punto se realiza, se puede separar en dos categorías: *External Footprinting* e *Internal Footprinting*. La primera hace referencia obtener información desde fuera del sistema y la segunda obtener información dentro del sistema.

5.3.1. Internal Footprinting

El Internal Footprinting engloba toda la recogida de información que se realiza una vez se tiene acceso, parcial o completo, al sistema o la red de la que se desea obtener la información. Este tipo de recogida de información no se realiza en la segunda fase de un pentesting, es decir, al comiendo del pentesting, sino que se realiza en la fase de post explotación del sistema, la quinta fase de un test de intrusión. Es lógico que, sin todavía haber explotado ninguna vulnerabilidad, y por ende no se haya accedido al sistema, no se pueda recoger información dentro de él.

5.3.2. External Footprinting

El External Footprinting engloba toda la recogida información, que al contrario que el Internal Footprinting, se realiza desde fuera del sistema. Este tipo de recogida información sí que se realiza en la segunda fase de un test de intrusión, y de hecho es el primer paso esencial a realizar. A través de diversas técnicas se busca obtener la mayor cantidad de información posible de una red o un sistema para, posteriormente, analizarla y encontrar vulnerabilidades.

Las técnicas de recogida de información englobadas dentro del External Footprinting se pueden dividir a su vez en dos subcategorías, en función del grado de agresividad de las mismas.

5.3.2.1. Active Footprinting

Por un lado, está el descubrimiento activo, denominado Active Footprinting, que destaca por interactuar directamente con la infraestructura del sistema objetivo mediante consultas al DNS, análisis de las cabeceras HTTP, enumeración de puertos y sus servicios, etcétera [34]. Seguidamente se explicarán brevemente en qué consisten algunas de estas técnicas, sin la intención de entrar en las herramientas de software concretas.

5.3.2.1.1. Escaneos DNS Una de las formas más comunes a la hora de obtener información consiste en obtener información de los servidores DNS. DNS (Domain Name Services) es un protocolo que permite la conversión entre direcciones de red numéricas, como son las IPs, y direcciones FQDN (Full Qualified Domain Name), que son direcciones del estilo miweb.es. De esta manera, DNS provee una capa de abstracción para hacer más sencillo las conexiones por parte de los usuarios y administradores a otros servicios.

La transformación de una dirección IP en un nombre de dominio se conoce como resolución DNS, y su proceso inverso (de nombre de dominio a IP) se conoce como resolución inversa. Este tipo de operaciones se da en los servidores DNS, de los cuales, mediante diferentes técnicas se puede extraer información relevante, hasta tal punto que se puede llegar a determinar ciertos aspectos de la topología de la red en la que se encuentra el sistema al que estamos intentando acceder.

5.3.2.1.2. Fingerprinting El Fingerprinting consiste en obtener información del propio sistema al que se intenta acceder. Datos como el sistema operativo y su versión o las aplicaciones que usa son esenciales a la hora de buscar vulnerabilidades que se puedan explotar en el sistema para lograr acceso a este. Aunque no se limita solo a este tipo de datos. Obtener el servidor web que usa determinado dominio o el CMS (Content Management System) es fundamental a la hora de determinar vulnerabilidades concretas para la versión de dicho software. Existen herramientas tanto para obtener esta información como para relacionarla con bases de datos, previamente elaboradas, en las que se encuentran gran cantidad de vulnerabilidades que ya han sido detectadas.

5.3.2.1.3. SMTP Otra de las áreas mediante las que se puede obtener información es la relacionada con el protocolo SMTP. Con ello se puede obtener información de los dominios de correo electrónico usados hasta direcciones de correo electrónicas concretas, lo que permite centrarse en vulnerabilidades para dichos dominios. Esto deriva en que se pueda lograr suplantar la identidad de cierto usuario.

5.3.2.2. Passive Footprinting

Por otro lado se encuentra el descubrimiento pasivo, lógicamente denominado Passive Footprinting, que recurre a la consulta de la información previamente indexada por motores de búsqueda registros públicos, foros, etcétera, por lo que no interactúa directamente con el sistema a penetrar [34].

5.3.2.2.1. Whois Whois es un protocolo TCP que permite obtener datos sobre el propietario de un nombre de dominio o una dirección IP. Entre los datos que se pueden obtener, se encuentran el correo electrónico, el nombre completo, o la ciudad, el código postal o el número de teléfono del propietario. Estos datos son de fácil acceso, existiendo incluso herramientas web que te permiten obtener dicha información mediante el mencionado protocolo.

5.3.2.2.2. Hacking con buscadores Los buscadores como Google o Bing son utilizados por la gran mayoría para encontrar sitios web. Lo que no todo el mundo conoce es que se tratan de una poderosísima herramienta para obtener información adicional sobre uno o varios sitios web. La gran mayoría de buscadores disponen de parámetros avanzados de búsqueda que permiten buscar palabras concretas en las URLs, buscar por una determinada extensión, buscar dentro un sitio web concreto. De esta forma, si se sabe qué buscar, se puede obtener información sobre qué vulnerabilidades se pueden explotar en cierta página web.

Aparte de los buscadores tradicionales, también existen una serie de buscadores especializado en la búsqueda de dispositivos, enfocados al IoT, en los que se pueden buscar desde webcam hasta electrodomésticos, con la condición de que estén conectados a Internet. Estos buscadores, como Shodan[1], permite también filtrar búsquedas y realizar búsquedas avanzadas,por lo que se permitan obtener información sobre otros sistemas que no sean directamente servidores web.

5.3.2.2.3. Social network engineering La ingeniería social, aunque no dispone de herramientas concretas, son técnicas para obtener información a partir de las redes sociales. En las redes sociales, muchas veces de manera inconsciente o si comprender las consecuencias que puede acarrear, se comparte una gran cantidad de información que, bien analizada, puede ser determinante para elaborar un ataque concreto.

[1] https://www.shodan.io/

5.4. Análisis de vulnerabilidades

Una vez obtenida toda la información posible sobre el sistema, se pasa a la siguiente fase definida en el PTES, que consiste en analizar qué vulnerabilidades concretas se pueden explotar. La información recogida se usa para obtener como resultado final un listado de vulnerabilidades que se pueden llegar a explotar en el sistema. Esta fase de análisis pasa por tres periodos.

5.4.1. Pruebas

En el primer periodo se realiza una serie de pruebas basándose en la información que disponemos del sistema, que previamente hemos obtenido. La información que nos den estas pruebas es importante, cuanto mayor número de pruebas se realicen mediante el mayor número de herramientas y técnicas posible los resultados mejorarán. Estas pruebas se engloban en pruebas pasivas o activas.

5.4.1.1. Activas

Las pruebas activas requieren interactuar directamente con el componente a auditar. Tienen una estrecha relación con el Active Footprinting, ya que se basan en la información obtenida de esa manera. Dentro de éstas se incluyen las siguientes categorías:

- *Automatizadas:* mediante diversas herramientas de software se interactúa con el sistema, enviando peticiones, escaneando los servicios, etc.
- *Conexión manual:* para evitar falsos positivos que puedan dar las pruebas automatizadas, se llevan a cabo pruebas manuales, que realizan los mismos pasos que las automáticas pero sin el uso de dichas herramientas de análisis de vulnerabilidades.
- *Ofuscadas:* con el objetivo de evitar la detección o el bloqueo por parte de sistemas IDS (Intrusión Detection System), IPS (Intrusion Prevention System) o WAF (Web Application Firewall), se realizan pruebas que se comportan de manera diferenciada con las pruebas tradicionales, alargando tiempos de espera entre peticiones, modificando ciertos aspectos de las peticiones o alternando entre objetivos.

5.4.1.2. Pasivas

La pruebas pasivas consisten en analizar la información obtenida mediante Passive Footprinting, analizar dicha información que nos permita suponer que existe cierta vulnerabilidad en el sistema. A diferencia de las pruebas activas, estas son mas subjetivas y abstractas.

5.4.2. Validación

Una vez elaboradas una serie de pruebas necesitamos correlar la información que nos ha dado esa serie de pruebas. En esta paso, el objetivo es enmarcar la vulnerabilidad encontrada

en un apartado técnico, clasificándola en una serie de categorías e indentificándolas de una manera concreta. Esta clasificación se puede realizar a varios niveles. El mas concreto consiste en identificarlas mediante, valga la redundancia, identificadores concretos como el *CVE*[2] (Common Vulnerabilities and Exposures). También se puede hacer a un nivel más global, mediante las categorias marcadas en diversas normas, como pueden ser el *NIST SP 800-53*[3] o la *Guía OWASP*[4].

5.4.3. Investigación

Tras la identificación de una (o más) vulnerabilidad, y haberla categorizado correctamente, es necesario evaluar la gravedad de dicha vulnerabilidad. Para ello se procede a realizar una investigación sobre dicha vulnerabilidad. Esta investigación puede ser privada, elaborando pruebas a nivel interno, como ataques de fuerza bruta o configurar replicas del entorno mediante el uso de máquinas virtuales (VM) para emular el entorno real y hacer pruebas con ello.

5.5. Explotación de vulnerabilidades: Ataques de penetración

Si en las dos secciones anteriores se ha explicado la recogida y el análisis de la información, respectivamente. Esas dos fases tienen como objetivo explotar una serie de vulnerabilidades. Esa es la esencia del pentesting, penetrar en un sistema mediante la explotación de dichas vulnerabilidades, realizando una serie de ataques a estos. Dependiendo del tipo de ataque, al vulnerabilidad que se explota o el objetivo del ataque, se pueden clasificar en prácticamente una infinidad de categorías. En los siguientes puntos se explican algunas de ellas, consideradas las más importantes.

5.5.1. Ataques de contraseñas

La contraseñas son un mecanismo de sobra conocido, que se remonta a mucho antes de la invención de la informática e incluso son anteriores a la propia formalización de la lógica. Si desde la antigüedad se llevan usando las contraseñas para limitar el acceso solo a ciertas personas a ciertos sitios, en la actualidad se usan para limitar el acceso por parte de cierto usuario a cierto sistema o servicio informático. A día de hoy son, aun con el auge de métodos biométricos, la herramienta de control de acceso más usada.

Partiendo de esa base, los ataques de contraseñas son cualquier técnica o mecanismo orientados a descifrar o romper las contraseñas usadas para proteger esos sistemas. Cabe destacar que todo mecanismo de autenticación por contraseñas se basa en la comparación de *hashes* y

[2]https://cve.mitre.org/
[3]https://nvd.nist.gov/800-53
[4]https://www.owasp.org/

no en la comparación de contraseñas, dando autorización solo cuando el hash de la contraseña introducida coincide con el almacenado previamente, que ha sido generado con anterioridad mediante la contraseña elegida para la protección de ese sistema.

Un *hash* no es mas que una cadena alfanumérica de longitud fija, la cual se genera en base a una *función hash*. La esencia de esas funciones radica en que generar un hash es sencillo y poco costoso, pero obtener la cadena original (es decir, la contraseña) mediante el hash es extremadamente complicado, aunque dicha complejidad depende del algoritmo usado. Ejemplos de dichos algoritmos son SHA-1 o MD5.

5.5.1.1. Fuerza bruta

Lo ataques de contraseñas se pueden elaborar de varias maneras. El método mas básico es el de fuerza bruta, que como su nombre indica, consiste en probar todas las combinaciones posibles de contraseñas. Esto puede ser especialmente costoso a nivel computacional. En las Tablas 5.1 y 5.2 se puede ver la cantidad de contraseñas posibles para diferentes casos, ademas de cuanto tiempo costaría romper cada una.

Contraseña	Posibilidades	
Numérica de 4 caracteres	10^4	10000
Numérica de 6 caracteres	10^6	1000000
Numérica de 8 caracteres	10^8	100000000
Numérica de 4 a 8 caracteres	$\sum_{n=4}^{8} 10^n$	111110000
Alfanumérica de 4 caracteres	27^4	531441
Alfanumérica de 6 caracteres	27^6	387420489
Alfanumérica de 8 caracteres	27^8	282429536481
Alfanumérica entre 4 y 8 caracteres	$\sum_{n=4}^{8} 27^n$	293292190521
Alfanumérica entre 8 y 16 caracteres	$\sum_{n=8}^{16} 27^n$	82834383195202897568241

Tabla 5.1.: Número de diferentes combinaciones de contraseñas contraseñas posibles para ciertos casos

Contraseña	Tiempo CPU[5]	Tiempo GPU[6]
Numérica de 4 caracteres	10 segundos	1,43 segundos
Numérica de 6 caracteres	16,66 minutos	2.381 minutos
Numérica de 8 caracteres	27,78 horas	3.97 horas
Numérica de 4 a 8 caracteres	30,86 horas	4,40 horas
Alfanumérica de 4 caracteres	8,85 minutos	75,92 segundos
Alfanumérica de 6 caracteres	4,48 días	15.37 horas
Alfanumérica de 8 caracteres	*107,46 meses*	*15,36 meses*
Alfanumérica entre 4 y 8 caracteres	*803540 años*	*114791 años*
Alfanumérica entre 8 y 16 caracteres	2.62 billones de años [7]	375237,29 millones de años

Tabla 5.2.: Tiempo que costaría realizar ataques de fuerza bruta para diferentes conjuntos

Se puede observar que, al menos mediante fuerza bruta y con un único ordenador estándar los ataques de fuerza bruta son prácticamente inútiles cuando la contraseña resulta ser mínimamente larga o compleja. Aun así, mediante supercomputadores y optimizaciones se pueden reducir tiempos, pero siguen siendo una solución poco eficientes.

5.5.1.2. Por diccionario

Por otra parte existen los denominado ataques de diccionario. En informática un diccionario consiste en una lista de palabras cualquiera, tengan sentido o no. Los ataques por diccionario simplemente prueban todas las opciones de un diccionario. Estos diccionarios se pueden obtener de diversas fuentes, o se pueden generar en base a palabra que guarden relación con el objetivo a atacar. De esta manera solo se prueban las combinaciones de caracteres que tengan mas posibilidades de aparecer, reduciendo drásticamente el tiempo que dura el ataque.

5.5.2. Exploits

El concepto de *exploit* es sumamente sencillo. Un exploit no es más que una pequeña aplicación escrita con el objetivo de aprovecharse de una vulnerabilidad conocida en un software. La palabra proviene del inglés, que significa literalmente aprovechar o explotar. Los exploits son la parte más importante a la hora de explotar vulnerabilidades.

Ligando íntimamente al concepto de exploit existe el concepto de *payload*. un payload no es más que una parte de código que el exploit se encarga de ejecutar en la máquina que tiene como víctima, normalmente con el objetivo de realizar algún tipo de acción maliciosa. Acciones como implementar una shell, añadir un usuario al sistema, borrar ciertos archivos,... prácticamente lo que al ataque antes se le pueda ocurrir.

Según como sea los payloads se pueden clasificar en tres tipos:

1. Los *singles*: simplemente son código que se encarga de ejecutar una tarea concreta
2. Los *stagers*: son payloads encargados de crear la conexión entre el atacante y la víctima normalmente este tipo de payloads preparan el terreno para los payloads de tipo *staged*.
3. Los *staged*: son payload con funcionalidades más complejas, que son introducidos por los stagers en las máquinas a atacar

Sin querer profundizar demasiado en un campo al que se le podría bien dedicar libros enteros, mencionar que existen una gran cantidad de exploits ya detectados, es más, existen grandes bases de datos de exploits para ser usados por parte de la comunidad de la seguridad informática. Además existen también gran cantidad herramientas que permiten descargar, actualizar, gestionar, e inyectar estos exploits.

[5] Teniendo en cuenta que se pueden probar una media de 1000 contraseñas por segundo mediante un núcleo de una CPU potente [37]

[6] Teniendo en cuenta que se pueden probar una media de 7000 contraseñas por segundo mediante un una GPU Nvidia GTX 1080 [38]

[7] 1 Billón de años = 1.000.000 de millones de años

5.5.3. Ataques a redes

Los ataques a redes son un tipo de ataques que tienen como objetivo penetrar en una red informática para poder obtener información del trafico interno de ésta, denominado capturar paquetes o *sniffing*.

Una red informática puede tener diferentes tipologías y dependiendo de ésta las características del ataque cambiaran drásticamente. El escenario varía de atacar una red PAN (*Personal Area Network*) a atacar una red LAN (*Local Area Network*) o WAN (*Wide Area Network*). También dependiendo de si la red es inalámbrica o no el ataque se podrá elaborar de una u otra manera.

Los ataques se suelen enfocar a las diferentes capas del modelo de red, y más concretamente a los protocolos específicos que lo componen. El modelo de red extendido actualmente es el modelo **TCP/IP** que toma su nombre de los dos protocolos mas importantes de éste. Los diferentes protocolos usados en cada capa del modelo TCP/IP se pueden observar en la Figura 5.2.

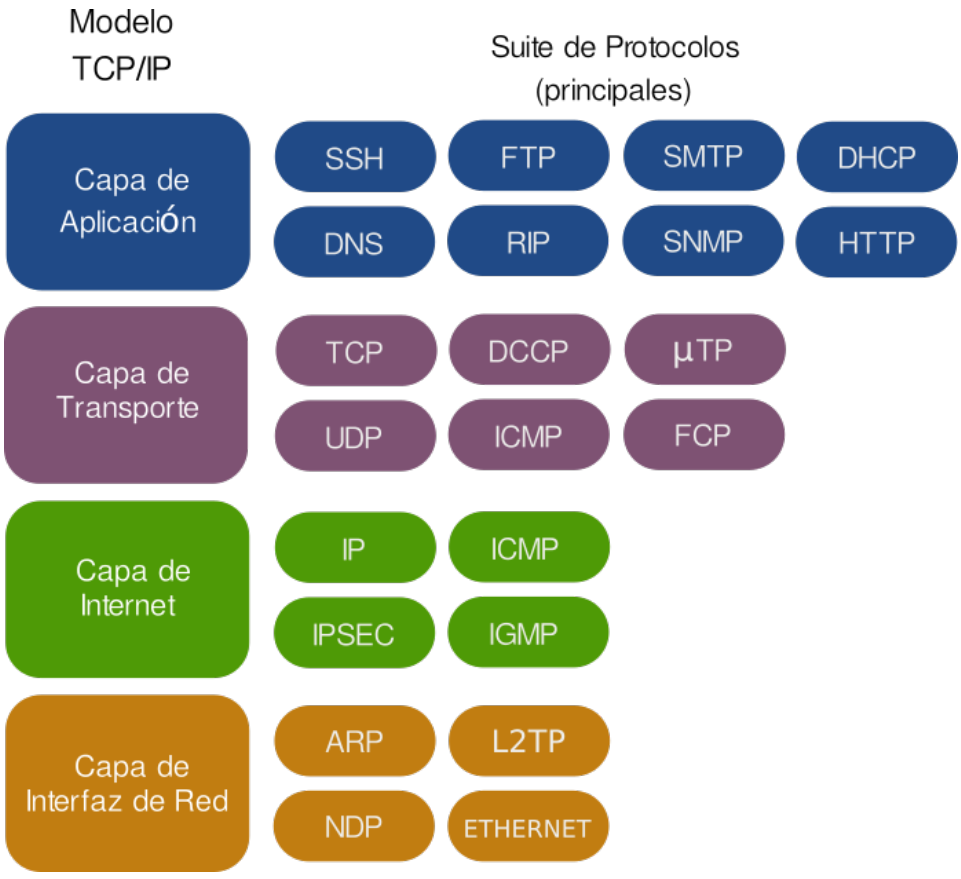

Figura 5.2.: Los principales protocolos usados en las 4 capas de TCP/IP

Existen, independientemente del modelo, diferentes términos que destacan en los ataques a redes, como son el *sniffing*, el *spoofing* o el *hijacking*.

5.5.3.1. Sniffing

El *sniffing* es una técnica que consiste en capturar los paquetes o tramas que pasan por cierta interfaz [39]. Por defecto un sistema rechaza el tráfico que no esta dirigido a én. En cambio, si un sistema actúa en modo promiscuo, aceptara todos los paquetes que le lleguen por la red. El sniffing es la técnica que obtiene todos esos paquetes. Para hacerlo, se hacen uso de herramientas llamadas sniffers, que permiten capturar estos paquetes y analizar su información capa por capa.

5.5.3.2. Spoofing

Spoofing es un concepto que consiste en suplantar la identidad [39]. Existen numerosas técnicas de suplantación como por ejemplo MAC Spoofing, donde se suplanta la dirección física de un dispositivo. También existen otras como *ARP Spoofing*, *DNS Spoofing* o *Web Spoofing*. Consisten básicamente en falsear cierta información para hacernos pasar por otro usuario, otro nodo y otra aplicación en la red.

5.5.3.3. Hijacking

Se conoce como *hijacking* a toda técnica en la que se obtiene el control de un elemento de una red, de tal manera que se pueda modificar la información que pasa por él [39]. Existen, al igual que con el spoofing, numerosas técnicas que hacen uso del concepto, como por ejemplo Session Hijacking o Browser Hijacking. De este tipo de técnicas surgen los ataques *MITM (Man In The Middle)*, que como su nombre indican, consisten en interferir en la comunicación entre dos nodos para tanto suplantar la identidad del usuario como para recibir información que se supone que solo debería recibirla directamente el usuario.

5.5.3.4. Ataques Wireless

Aquí simplemente mencionar que los ataques wireless son algo muy popular hoy en dia y enfocarlo a lo sencillo que son hacerlos, dando algunos nombres pero sin meterme en detalles de como se realizan (no quiero que sea una guía).

6

Conclusiones

A lo largo de los capítulos anteriores se ha realizado un profundo análisis sobre el campo de la seguridad informática, explicando una gran cantidad de conceptos. Primero se han definido una serie de conceptos básicos sobre seguridad informática, intrínsecos a cualquier especialidad o aplicación, como son los servicios de la seguridad de la información.

Por otra parte se ha elaborado un estado del arte de las diferentes aplicaciones de la seguridad informática, comenzando desde el surgimiento de la necesidad de proteger la información y de hacer que las transmisiones de datos fuesen seguras. Se hace un énfasis en el malware y en los tipos que existen, mostrando de que manera somos vulnerables a ataques y en que medida nuestra información esta desprotegida. Por otra parte, se desmonta el concepto tan arraigado en las mentes de los usuarios de que la seguridad informática consiste en proteger ordenadores personales de malware. Por una parte el malware no es la única amenaza existente, las intrusiones a redes o lo ataques de ingeniería social son otras formas mediante las que se puede acceder a nuestra información. Por otra parte, no solo los ordenadores personales son el único target, los avances en diferentes campos, destacando los smartphones, el IoT y el Cloud Cmputing, hace que estos sean actualmente vértices de enfoque importantes dentro de la seguridad informática.

Tras ese análisis sobre el mundo de la seguridad informática, conocer los diferentes sistemas y sus medidas de seguridad, mencionar diferentes tecnologías y enumerar y describir diferentes tipos de vulnerabilidades se puede obtener una idea clara sobre la seguridad informática. Aun así, esta idea no estará completa si no se conoce como actuar ante estas amenazas, como un hacker actúa para conseguir información de un sistema. Por ello se ha hecho especial hincapié en el pentesting. El pentesting es una técnica que mediante la imitación del comportamiento de un atacante permite detectar y explotar vulnerabilidades. Dista de un atacante malicioso, o *Black Hat Hacker*, en que el pentester usa el hacking ético y una serie de procedimientos definidos y controlados con el objetivo de detectar vulnerabilidades de todo tipo para poder posteriormente corregirlas, haciendo los sistemas más seguros. Esta técnica, que requiere un profundo conocimiento de los vectores de ataque y vulnerabilidades, además de sólidas nociones sobre redes y sistemas, permite comprender mejor, ya no solo como defenderse, sino como atacar, dando una visión mas global del conjunto.

6.1. La seguridad informática y los usuarios

Ante esto se plantea un problema serio con respecto a la seguridad de la información. El estado del arte elaborado contiene una gran cantidad de información sobre la seguridad informática, pero no es algo que un neófito en la materia pueda comprender. La seguridad informática es un campo técnico que requiere muchos conocimientos para poder aplicar sus medidas. Por lo tanto, ¿la seguridad de la información de los usuarios solo puede mantenerse mediante el trabajo de los profesionales de la seguridad? ¿Los usuarios no pueden de manera proactiva controlar sus sistemas haciendo que sean estos más seguros?

Aquí es donde entra en juego la programación. Elaborando software que pueda usar cualquier usuario implementando funcionalidad que permita hacer lo que un hacker realiza en su trabajo, pero de una manera clara y sencilla para el usuario, alejado de aspectos técnicos. Esa es la clave para hacer que los usuarios sean un elemento activo en pro de la seguridad de la información.

Pongamos un ejemplo práctico. Si uno se pone en la piel de un pentester, y quiere hacer un ataque de penetración en una red concreta, buscara vulnerabilidades para poder acceder. También puede querer encontrar maneras para defenderse o detectar elementos en una red. El pentester, un hacker con profundos conocimientos del área, sabe que uan de las mejores herramientas de escaneo de redes es NMap[1]. Mediante un par de comandos en su terminal obtendrá la información que necesita. De esa manera tan simple podrá proseguir con su trabajo.

Ahora pongamos otro caso. Si, en este caso, uno se pone en la piel de un usuario común que simplemente quiere evitar intrusos en su red local. Algo que seria trivial para un hacker o alguien con conocimientos de redes, resulta bastante complicado para un usuario normal. Dicho usuario no conoce ni NMap ni otras herramientas, no sabe ni siquiera acceder a su router para configurarlo, mucho menos sabe sobre redes o sistemas. Entonces, ¿cómo podría saber si tiene intrusos en su red? ¿La única opción que tendría sería recurrir a alguien con conocimientos sobre el tema?

6.2. Herramientas de seguridad informática para usuarios

En ese tipo de casos es donde entra a colación el software, y como este puede ayudar a los usuarios a llevar a cabo este tipo de tareas. Si queremos ofrecer una solución para el usuario, en base a lo aprendido en puntos anteriores podemos obtener varias conclusiones. Lo primero es que, debido al auge de los smartphones, nuestro software debería programarse para uno de estos sistemas. Si debe ir para uno de estos sistemas, y teniendo que elegir uno, lo mas lógico sería elegir Android, por las cuotas de mercado mencionadas anteriormente. Por otra parte, nuestra aplicación se va a basar en la recogida de información, proceso al que en un pentesting llamábamos *Information Gathering*, por lo que conocer como un pentester recoge la información es fundamental para desarrollar dicha aplicación. Después se puede entrar en

[1] https://nmap.org/

que herramientas concretas se puede usar. NMap parece a primera vista la opción mas viable dentro de ese campo. También hay que tener en cuenta herramientas de desarrollo concretas o herramientas secundarias que también pueden ser útiles.

En la siguiente parte de esta memoria se desarrollará una aplicación con lo mencionado en el párrafo anterior (y algunas herramientas más). Se explicaran en detalles las tecnologías y herramientas usadas y todo el proceso de desarrollo de la aplicación, tanto de programación como de diseño de la interfaz o de la experiencia de usuario, con el objetivo de implementar una prueba de concepto que permita vislumbrar como llevar herramientas de seguridad informática a usuarios comunes.

III

Fase 2: Desarrollo de la aplicación

7 Introducción

”La mejor forma de predecir el futuro es implementarlo”

— David Heinemeier Hansson

Explicar tras el estado del arte en que va a consistir la aplicación.

8 Tecnologías y herramientas

"Java es lo más penoso que le ha ocurrido a la informática desde MS-DOS"
— Alan Kay

Explicar las tecnologias que se van a usar para desarrollar la aplicación. He metido este punto aquí, en la FASE 2 ya que va mas ligado a ella, y ponerlo justo antes del desarrollo en sí me parece correcto.

8.1. Kali Linux

8.2. NMap

8.3. Android

8.3.1. Android SDK

8.3.2. Android Studio

8.4. Otras herramientas

8.4.1. Git

9 Desarrollo de la aplicación

”Los programas deben ser escritos para que los lean las personas, y sólo incidentalmente, para que lo ejecuten las máquinas”

— Abelson and Sussman

Aquí ya al lío. Explicar como va a ser la GUI y como se han programado los diferentes aspectos de la app.

10 Testeo y corrección de errores

"El testing de componentes puede ser muy efectivo para mostrar la presencia de errores, pero absolutamente inadecuado para demostrar su ausencia"

— Edsger Dijkstra

Tras programarlo todo pueden salir fallos, hay que hacer pruebas, e incluso habrá que cambiar cosas o añadir cosas nuevas. Todo eso va explicado aquí. NO CONFUNDIR con añadir nuevos requisitos funcionales o que se haya alargado el tiempo, eso va en la última parte, en la de conclusiones

IV

Análisis y conclusiones del Trabajo

V

Apéndices

Bibliografía

[1] Oficina de Seguridad del Internauta. 2017. URL: https://www.osi.es/es/herramientas-gratuitas.

[2] Inc GitHub. 2017. URL: https://github.com/showcases/security.

[3] El Mundo. *Naciones Unidas declara el aceso a Internet como un derecho humano*. Jun. de 2011. URL: http://www.elmundo.es/elmundo/2011/06/09/navegante/1307619252.html.

[4] Microsoft. 2017. URL: https://www.microsoftstore.com/store/mseea/es_ES/pdp/Windows-10-Home/productID.320437800.

[5] Microsoft. 2017. URL: https://www.microsoftstore.com/store/mseea/es_ES/pdp/Project-Profesional-2016/productID.324452600.

[6] Critical Tools. 2017. URL: https://store.criticaltools.com/.

[7] Gene Spafford. *Computer Recreations: Of Worms, Viruses and Core War*. Mar. de 1989. URL: http://spaf.cerias.purdue.edu/quotes.html.

[8] Insituto Nacional de Ciberseguridad INCIBE. *Implantación de un SGSI en la empresa*. URL: https://www.incibe.es/extfrontinteco/img/File/intecocert/sgsi/img/Guia_apoyo_SGSI.pdf.

[9] International Organization for Standardization ISO. *ISO 7498-2:1989. Information processing systems - Open Systems Interconnection - Basic Reference Model*. 2002. URL: https://www.iso.org/standard/14256.html.

[10] International Organization for Standardization ISO. *ISO/IEC 17799:2005. Information technology – Security techniques – Code of practice for information security management*. 2005. URL: https://www.iso.org/standard/39612.html.

[11] ISO. 2013. URL: http://www.iso27000.es/sgsi.html.

[12] Ismael Etxeberria Aguiriano. “Sistemas de Gestión de la Seguridad de Sistemas de Información”. 2014.

[13] McAfee Labs. *Informe de Predicciones sobre amenazas para 2016*. 2016. URL: https://mcafee.app.box.com/v/2016predictions.

[14] John von Neumann y Arthur Walter Burks. *Theory of self-reproducing automata*. 1966. URL: https://archive.org/details/theoryofselfrepr00vonn_0.

[15] Xataka. *La historia de Creeper, el primer virus informático jamás programado*. Abr. de 2017. URL: https://www.xataka.com/historia-tecnologica/la-historia-de-creeper-el-primer-virus-informatico-jamas-programado.

[16] Panda Security. *Classic Malware: su historia, su evolución*. URL: http://www.pandasecurity.com/mexico/homeusers/security-info/classic-malware/.

[17] Symantec. *ISTR: Internet Security Threat Report*. Abr. de 2016. URL: https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf.

[18] Hongkiat. *10 Most Destructive Computer Viruses*. URL: http://www.hongkiat.com/blog/famous-malicious-computer-viruses/.

[19] Ander Granado. *HearIiAll. A little Keylogger for Windows developed in C++*. Ene. de 2016. URL: https://github.com/ander94lakx/HearItAll.

[20] El Mundo. *El móvil supera por primera vez al ordenador para acceder a Internet*. 2016. URL: https://mcafee.app.box.com/v/2016predictions.

[21] Gartner. *Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016*. 2016. URL: http://www.gartner.com/newsroom/id/3415117.

[22] Apple Inc. *iOS Security - iOS 10 -White Paper*. 2017. URL: https://www.apple.com/business/docs/iOS_Security_Guide.pdf.

[23] Android. *Security Tips*. URL: https://developer.android.com/training/articles/security-tips.html.

[24] *VI Jornada de Seguridad y Protección de Datos de Carácter Personal*. Vitoria-Gasteiz, Spain: UPV/EHU, 2014. URL: http://lsi.vc.ehu.es/wdocs/pdd/pdd-2014/2014-Programa.pdf.

[25] Daniele Miorandi y col. "Internet of things: Vision, applications and research challenges". En: *Ad Hoc Networks* 10.7 (2012), págs. 1497-1516. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2012.02.016. URL: http://www.sciencedirect.com/science/article/pii/S1570870512000674.

[26] Jayavardhana Gubbi y col. "Internet of Things (IoT): A vision, architectural elements, and future directions". En: *Future Generation Computer Systems* 29.7 (2013). Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services; Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond, págs. 1645-1660. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2013.01.010. URL: http://www.sciencedirect.com/science/article/pii/S0167739X13000241.

[27] Quartz Media. *The easy way your smart coffee machine could get hacked and ruin your life*. Feb. de 2017. URL: https://qz.com/901823/the-easy-way-your-smart-coffee-machine-could-get-hacked-and-ruin-your-life/.

[28] Techradar. *The internet of things can be hacked – and the risks are growing every day*. Feb. de 2017. URL: http://www.techradar.com/news/the-internet-of-things-can-be-hacked-and-that-puts-your-life-at-risk.

[29] SearchDataCenter. *RSA Conference 2016: IoT se estrellará y arderá si la seguridad no está primero*. 2016. URL: http://searchdatacenter.techtarget.com/es/cronica/RSA-Conference-2016-IoT-se-estrellara-y-ardera-si-la-seguridad-no-esta-primero.

[30] Xataka. *Miele tuvo la gran idea de incluir un servidor web en un lavavajillas, pero no de aplicar la seguridad necesaria*. Mar. de 2017. URL: https://www.xataka.com/seguridad/de-dispositivo-conectado-a-hackeado-lo-que-puede-ocurrir-al-incluir-un-servidor-web-en-un-lavavajillas.

[31] El País. *La seguridad de los coches Tesla en duda... por una aplicación Android*. Nov. de 2016. URL: http://cincodias.elpais.com/cincodias/2016/11/24/motor/1479986182_473873.html.

[32] Instituto Nacional de Tecnologías de la Comunicación INTECO. *Riesgos y amenazas en Cloud Computing*. Mar. de 2011. URL: https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_riesgos_y_amenazas_en_cloud_computing.pdf.

[33] Cloud Security Alliance CSA. *The Treacherous 12: Cloud Computinf Top Threats in 2016*. Feb. de 2016. URL: https://downloads.cloudsecurityalliance.org/assets/research/top-threats/Treacherous-12_Cloud-Computing_Top-Threats.pdf.

[34] Pablo González Pérez, Gemán Sánchez Garcés y Jose Miguel Soriano de la Cámara. *Pentesting con Kali*. Juan Ramón Jimenez, 8. 28932 Madrid (España): 0xWORD, 2013. ISBN: 978-84-616-7738-2.

[35] Penetration Testing Execution Standard. 2017. URL: http://www.pentest-standard.org/index.php/Main_Page.

[36] Instituto Nacional de Tecnologías de la Comunicación INTECO. *Pentest: Recolección de Información (Data Gathering)*. URL: https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_information_gathering.pdf.

[37] Openwall Community Wiki. *John the Ripper benchmarks*. Mayo de 2016. URL: http://openwall.info/wiki/john/benchmarks.

[38] Digital Trends. *NVIDIA'S GTX 1080 CAN CRACK PASSWORDS AS EASILY AS IT CAN GAME*. Ago. de 2016. URL: https://www.digitaltrends.com/computing/nvidia-gtx-1080-crack-passwords/.

[39] Juan Luís García Rambla. *Ataques en redes de datos IPv4 e IPv6*. 2.ª ed. Juan Ramón Jimenez, 8. 28932 Madrid (España): 0xWORD, 2014. ISBN: 978-84-616-8383-3.