Título: Ingeniería social: Psicología aplicada a la seguridad informática

Volumen: 1/1

Alumno: Sergio Arcos Sebastián

Directora: M. Ribera Sancho Samsó

Depto.: Ingeniería de Servicios y Sistemas de Información

Fecha: 1 de junio de 2011

DATOS DEL PROYECTO

Título del proyecto:	Ingeniería social: Psicología aplicada a la seguridad informática
Nombre del estudiante:	Sergio Arcos Sebastián
Titulación:	Ingeniería en Informática
Créditos:	37,5
Directora:	M. Ribera Sancho Samsó
Departamento:	Ingeniería de Servicios y Sistemas de Información (ESSI)

MIEMBROS DEL TRIBUNAL *(nombre y firma)*

Presidente:	David López Álvarez
Vocal:	Mónica Sánchez Soler
Secretaria:	M. Ribera Sancho Samsó

CALIFICACIÓN

Calificación numérica:

Calificación descriptiva:

Fecha:

Resumen

Este proyecto recoge los fundamentos de los ataques de ingeniería social en los sistemas informáticos, especialmente en las grandes plataformas de Internet. El objetivo principal es lograr comprender su naturaleza y ser capaces de valorarlos como la amenaza que representan. Por tal de conseguir el objetivo se realizan pruebas de concepto para valorar el riesgo, cambiando a menudo la perspectiva a la del atacante. A raíz de este proyecto, se espera formar una base, para que en un futuro sea posible incrementar la seguridad de dichas plataformas con tecnologías de prevención, detección e intercepción de estos ataques, proponiendo la `Interacción Humano-Computador Segura´ como punto de partida.

El lector, aún no siendo responsable del mal diseño de las plataformas de Internet ni culpable de la pérdida de sus propios datos, puede concienciarse con los ejemplos mencionados y reaccionar más sabiamente en futuras situaciones delicadas.

Palabras clave

Ingeniería social, seguridad informática, Internet, red social, prueba de concepto, programación neurolingüística, *misdirection*, *rapport*, *anchoring*, interacción humano-computador segura, Teensy

Índice de contenido

1 INTRODUCCIÓN AL PROYECTO

Este proyecto trata de la manipulación a la que puede verse sometido un usuario de ordenador, a través de elementos psicológicos que le influencian a hacer una determinada tarea de la cual no es consciente. Este ataque recibe el nombre de ingeniería social, abreviado I.S. a partir de ahora.

A continuación se desarrollan los aspectos que han condicionado el proyecto.

1.1 MOTIVACIÓN DEL PROYECTO

Como entusiasta de la seguridad informática, conocía la realización de ataques a través de la red, como, inyectar código en una página web o utilizar programas maliciosos, es decir, virus, troyanos o parecidos. Estos ataques son contrarrestados arreglando y/o mejorando el producto afectado. Eso significa que invirtiendo recursos, tanto materiales como humanos, es posible establecer unos niveles de seguridad altos.

Pero hay un tipo de ataque, la I.S., que no afecta directamente a los ordenadores, sino a sus usuarios, conocidos como “el eslabón más débil”. Tal ataque es capaz de conseguir resultados similares a un ataque a través de la red, saltándose toda la infraestructura creada para combatir programas maliciosos. Además, es un ataque más eficiente, debido a que es más complejo de calcular y prever.

El interés surgió principalmente por la relación que tenía con las personas, dejando a los ordenadores en segundo plano. Aún así, este proyecto utiliza el medio de Internet como vía de

ataque, mostrando ejemplos hostiles que se han llevado a cabo.

Por todo lo anteriormente comentado, este proyecto supone un reto personal, intentando concienciar al lector sobre cómo alguien malicioso puede aprovecharse de él. Dado que cada persona tiene diferentes debilidades, esta deberá ser quien medite para gestionar sus propias medidas de protección frente a la I.S.

1.2 ESTADO DEL ARTE

Este proyecto reflexiona sobre varios trabajos de los investigadores de I.S. más importantes, como son Kevin Mitnick, Marcus Nohlberg, Markus Huber, Daniel Siegel, y otros. Recopilando el estado del arte que ha realizado cada uno de ellos es posible obtener una visión mucho más precisa, pero dada la rápida evolución tecnológica que ha habido durante los últimos años, todo lo anterior al trabajo de Kevin Mitnick en 2002 está prácticamente obsoleto.

A parte, los trabajos existentes se pueden clasificar por su enfoque. Si tiene el objetivo de enseñar cómo se ataca, se fundamenta en experiencias “vividas”, las cuales pueden ser ficticias. Si tiene el objetivo de enseñar cómo se protege, se fundamenta en un tipo de ataque el cual no es verídico, dado que tan pronto se aplique tal protección, el atacante va a modificar el método para saltársela. Es por esto que la mayoría de trabajos exponen su punto de vista frente a la I.S., nunca ofreciendo una solución definitiva. El debate se encuentra entre si la educación y la ética son suficientes para combatir la I.S.

Además, la I.S. no es exclusiva de la seguridad informática. Se utiliza en todas aquellas disciplinas que tienen un impacto sobre otras personas, consiguiendo ser más eficaces con sus propósitos, como, la religión, el marketing, la política, etc.

1.3 OBJETIVOS DE LA INVESTIGACIÓN

- Desmitificar la I.S., dejando de verla como algo sólo para gente diestra, acercándola al público general para expandir su conocimiento.
- Ofrecer una nueva visión de todo lo que rodea la seguridad informática, concienciando y alertando de los riesgos que conlleva utilizar Internet y las nuevas tecnologías.

– Validar algunas de las teorías existentes y aportar nuevas cuestiones sobre la materia.

Tales objetivos concluyen en:

– Fomentar una base de estudio de la I.S., demostrando la peligrosidad directa que representa para los usuarios, especialmente cuando la plataforma no ha tomado medidas adicionales.
– Permitir en un futuro desarrollar herramientas más eficaces para combatir este tipo de ataques, detectándolos, previniéndolos e interceptándolos.

1.4 METODOLOGÍA: SEGUIMIENTO

1) La primera fase fue leer e investigar sobre la documentación ya realizada. Todos los autores de libros, revistas, documentos e investigaciones coinciden en la carencia de información sobre el tema. Esto llevo a realizar un estudio muy amplio de los elementos relacionados. Fue tan amplio que derivó en la necesidad de resumirlo. Es el Capítulo 3, aunque no todo lo aprendido ha podido ser escrito.
2) La segunda fase fue concretar la definición de I.S., a partir de lo aprendido de la información consultada y de la meditación de la misma. Vista la dificultad que supone definir la I.S., se acompaña con pautas para identificar los ataques. Es el Capítulo 2.
3) La tercera fase fue ampliar y adentrarse con dos casos más concretos, para demostrar la amenaza que esta supone y aprender a responder preguntas subjetivas. Existen más casos concretos, como la automatización de la I.S. Es el Capítulo 4.
4) La cuarta y última fase vino a través de la anterior ya que el segundo caso concluye en una incoherencia, que muestra la entrada al mundo de la `Interacción Humano-Computador Segura´, un nuevo campo de investigación relacionado a la I.S. Se define e introduce, añadiendo una lista de ejemplos de ataques usando esta vía. Es el Capítulo 5.

2 INGENIERÍA SOCIAL

La I.S. es un proceso planificado que consiste en atacar las vulnerabilidades humanas para conseguir un beneficio, considerando que el eslabón más débil es la persona y no la máquina.

Otra acepción, cuando el enfoque no es la seguridad, es interpretar el adjetivo social de la palabra I.S. con la definición oficial de la R.A.E. [1]: “perteneciente o relativo a la sociedad”. Esto puede confundir al relacionar I.S. con diferentes obras de ingeniería en ámbitos sociales, como escuelas, hospitales y otros centros. Dado que este proyecto esta enfocado a la seguridad, no se contempla esta vertiente de ingeniería.

Existen varios sectores de profesionales que la aplican constantemente:

- Políticos: utilizan palabras, cuerpo y entorno para transmitir información a la sociedad, ayudándose de textos previamente supervisados, escenarios bien iluminados y eslóganes claros y concisos.
- *Phreakers* o estafadores telefónicos: utilizan palabras e información privilegiada para engañar y extraer más información de aquellas personas más inocentes, creando vínculos de confianza y explotando elementos rutinarios que no son considerados como un problema.
- *Scammers*, *crackers*, *phishers*: variedad de delincuentes cibernéticos que se dedican a diversos timos, como el *scam*, el *phishing*, etc.
- Todo tipo de timador tradicionalmente conocido, como los que realizan el timo de la estampita, demostrando que la I.S. se ha ido sofisticando con el paso del tiempo.

Durante el resto del proyecto, se toca únicamente la parte virtual y, dentro de esta, a tratarlo como un tipo de ataque malicioso, que es como se conoce popularmente dentro de la seguridad informática. Para simplificar y concretar, se propone una definición más simple de I.S., que es `persuadir a alguien para que haga algo´. Al final del capítulo se completa la definición con una serie de pautas que sigue este tipo de ataque.

Dentro de este campo, Marcus Nohlberg, autor de la figura 2.1, diferenció en su tesis doctoral [2] hasta seis disciplinas diferentes: sociología, psicología, economía y gestión, seguridad, derecho y educación. En el presente proyecto, se tratan las áreas psico-sociales, la seguridad informática, y en el Capítulo 5 incluso se introduce un nuevo campo.

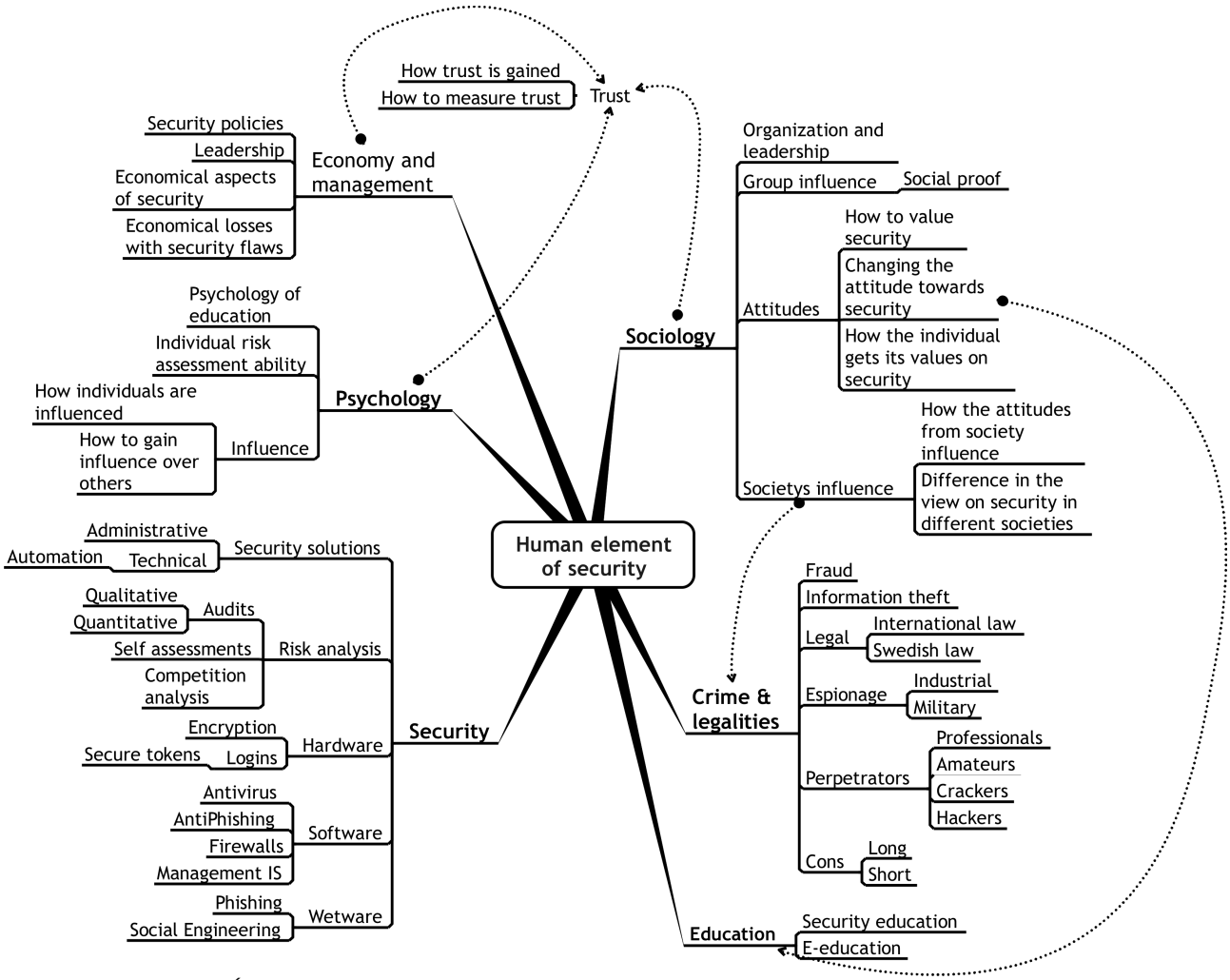

Ilustración 2.1: Áreas de investigación y tópicos conectados al elemento humano de la seguridad en I.S. [2]

Desde el punto de vista del aprendizaje de la I.S., se distinguen dos caminos diferentes. Por un lado, la interacción activa agrupa a aquellas personas que han practicado a partir de su instinto, dado que para realizar un ataque no es necesario conocer ningún concepto previo, y por el otro lado, la interacción pasiva agrupa a aquellas personas que han estudiado esos conceptos, intentándolos profundizar y mejorar.

2.1 INTERACCIÓN ACTIVA

La interacción activa es aquella que se realiza llevando la I.S. a la realidad, con el fin de causar daños y perjuicios. La habilidad y talento de la persona que lo lleva a cabo es proporcional al éxito de los futuros ataques.

Dada la malicia que esconde el ataque, la justicia se esfuerza en evitar y castigar este tipo de situaciones. Es necesario valorar si tal interacción activa es simplemente una prueba de concepto, en que medida ha afectado a las víctimas, y si el atacante sigue un código ético. En alguna ocasión, cuando el delincuente es detenido, aprovecha para continuar su carrera desde el lado de la seguridad informática, previniendo nuevos ataques de I.S.

El inicio de la I.S. se remonta a Kevin Mitnick, quien esquivó a la justicia durante casi 20 años antes de ser detenido, por realizar diferentes delitos del orden público falsificando y sustrayendo datos e información confidencial de varias universidades y departamentos del estado. Actualmente dispone de su propia auditoria de seguridad informática, tiene publicados varios libros [3][4], una película sobre él [5], ha participado en eventos como la DefCon, BlackHat y DerbyCon, y en el evento `Capture The Flag´ de la comunidad Social-Engineer.Org, conocido por ser la primera competición ética de I.S.

En su libro titulado `The Art of Deception´ [3] explica la experiencia recopilada durante años a través de historias ficticias. La mayoría de las técnicas tienen una base de influencia social, la cual se podría enmarcar dentro de lo que se llamarían técnicas de `programación neurolingüística´, abreviado P.N.L. La P.N.L es una disciplina científica inexacta que se basa en acciones que funcionan y ofrecen buenos resultados, a través de la observación del comportamiento humano. Kevin Mitnick define una metodología que consta de cuatro etapas, vista en la tabla 2.1.

CICLO DE LA INGENIERÍA SOCIAL	
ACCIÓN	DESCRIPCIÓN
Investigación	Se investiga en fuentes abiertas de información, como formularios e informes anuales, material de marketing, patentes sobre aplicaciones, entradas de prensa, revistas y magazines, contenidos en Internet, remover la basura, etc.
Desarrollar *rapport* y credibilidad	Se usa la información interna, se reemplazan identidades, se reclama a la víctima, se le pide ayuda, o se usa la autoridad.
Explotar confianza	Se pregunta o se consigue que te pregunten por tal de conseguir el objetivo marcado.
Utilizar información	Si la información es sólo una parte, se vuelve a empezar el ciclo, hasta conseguir el objetivo.

Tabla 2.1: Ciclo de ingeniería social diseñado por Kevin Mitnick [3]

El ciclo es muy simple, pero no por ello poco eficaz. Se puede resumir en buscar información, crear confianza y utilizarla. Además, siempre se produce en el mismo escenario, en el cual encontramos:

- Atacante: es una persona quien debe preparar y ejecutar la acción.
- Medio: es la vía por la que hay la comunicación. Suele ser personal, telefónico o mediante Internet, pero en alguna ocasión se hace a través del fax o el correo, pero no como medio principal.
- Víctima: es una persona manipulada que ejecuta la voluntad del atacante sin conocimiento de causa.
- Pretexto: es la historia que se crea y se argumenta por tal de convencer a la víctima, dando credibilidad a lo que se dice y como se dice. Aquí se incluye la información recopilada, su análisis y el momento en que decidimos utilizarla.

A día de hoy, dado el reconocimiento existente sobre la I.S., es necesario incrementar el uso de la interacción activa como, por ejemplo, desarrollando más competiciones éticas, para justificar los avances teóricos, tanto a nivel de ataque como defensa.

2.2 INTERACCIÓN PASIVA

La interacción pasiva es aquella que trata con la I.S. desde un segundo plano, estudiando las materias que ésta integra, pero sin contacto real. A diferencia de la activa, se mantiene en la legalidad, siendo a veces insuficiente para valorar ciertas pruebas de concepto. El objetivo también cambia, dado que mientras la pasiva puede preocuparse de medir que eficaz es un método, la activa busca sólo serlo.

Existen dos caminos de estudio destacados a la hora de buscar nuevos conocimientos que aporten valor a la I.S. El primero, formado por el conjunto de las áreas de psicología y sociología, nombrado psico-social, que estudia y aprende de los comportamientos de las personas. Es el reflejado en el elemento `pretexto´, mostrado anteriormente con Kevin Mitnick. El segundo, formado por el conjunto de las áreas más técnicas, refleja su acción sobre el atacante y el medio, buscando, entre otras, una automatización de las diferentes partes del proceso.

A continuación, se introducen dos profesionales de la I.S., Dale Pearson y Marcus Nohlberg, y una posible evolución de la I.S., la I.S. Automatizada., abreviado I.S.A., intentando englobar lo máximo posible los dos caminos mencionados en el párrafo anterior.

Dale Pearson, conferenciante y webmaster de subliminalhacking.net

Dale Pearson es una persona dedicada a la seguridad informática, al hipnotismo y al mentalismo. Promueve la I.S. a través de la combinación de las disciplinas del gráfico 2.2, que son: lenguaje corporal, mentalismo, hipnosis, P.N.L. e influencia. Esta combinación se caracteriza por materias que no son puramente científicas, pero que ofrecen una alta probabilidad de éxito.

Aún teniendo garantías basadas en probabilidades, hay que saber gestionar este conocimiento, ya que puede parecer que este conjunto de áreas psico-sociales sea más una manera de impresionar que no algo objetivo donde fundamentar una investigación. Se ha sido riguroso a la hora de analizar este conocimiento, tocando aquellas técnicas psico-sociales que más evidentes eran en el comportamiento humano. Por ejemplo, se ha valorado que analizar la confianza es adecuado, pero se ha descartado tratar el hipnotismo, que es más complejo de justificar. Él ha sido capaz de realizar algún experimento haciendo olvidar la contraseña de la cuenta de alguna persona, pero se puede dudar la veracidad.

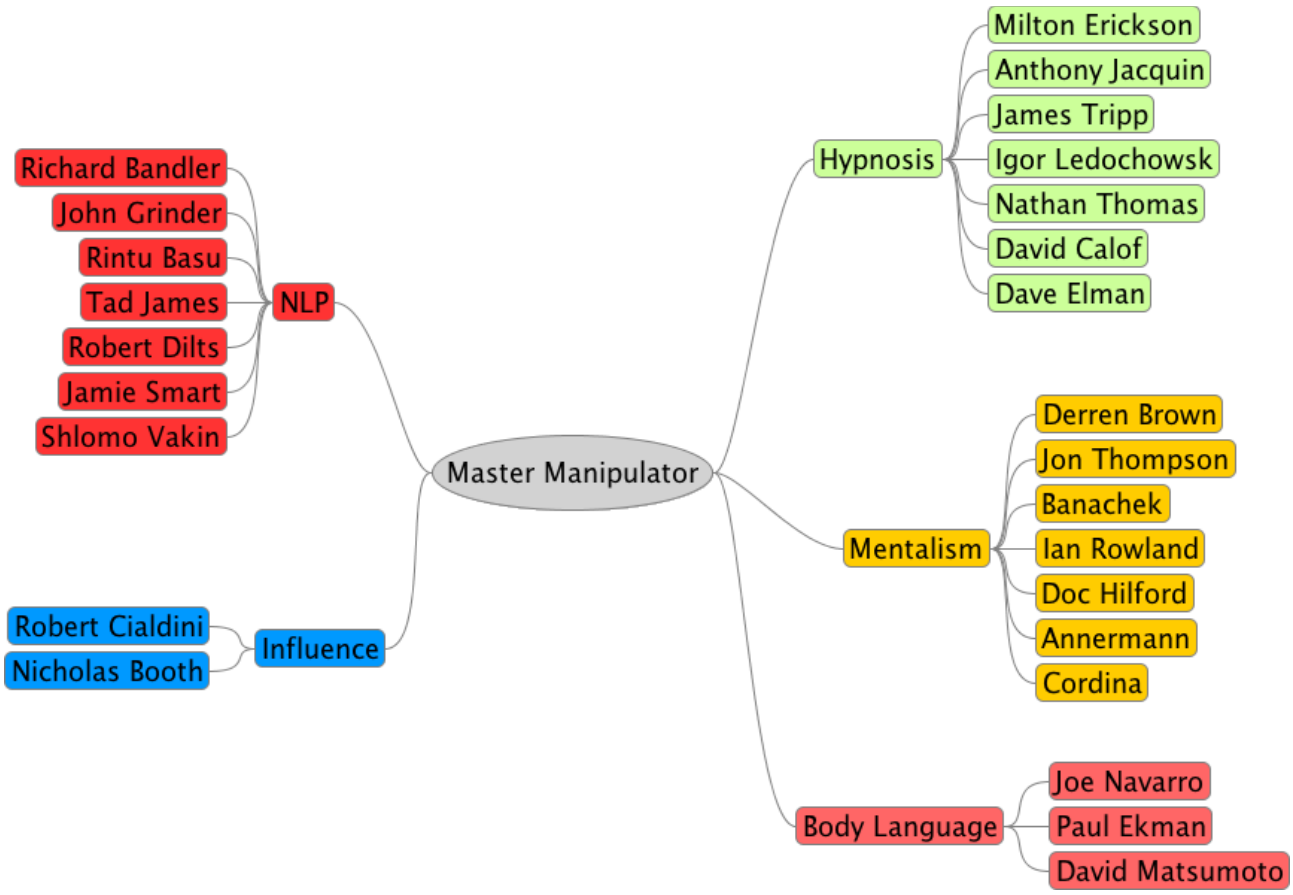

Ilustración 2.2: Esquema de las cinco materias esenciales y sus respectivos investigadores [6]

Dada la confianza que transmite Dale Pearson, se contactó con él a través de correos electrónicos para conocer de primera mano su experiencia y los puntos clave que considera de la I.S. La frase que resume su visión [6] es "*Language is very powerful, and the mind has vulnerabilities that can be abused*". Todo depende de lo lejos que uno quiere ir como ingeniero social, cuyo eje es "*the appropriate BIG BECAUSE for the subject to do as you desire*". Sabe a ciencia cierta que nunca se puede ser capaz de predecir si saldrá bien, pero anima a continuar estudiando la materia, "*and then you can really appreciate how all these components fit together*". Y como ya se ha comentado, "*hypnosis is just one form of language to do so*".

Marcus Nohlberg, Doctor en Computer Science

La tesis doctoral de Marcus Nohlberg trata la I.S. [2] "comprendiéndola, midiéndola y evaluando la protección frente a un ataque". Dispone también de una recopilación literaria muy amplia de todo lo comentado hasta el momento de su publicación. Es de los pocos que ha tratado el tema

tan directamente. Hay dos diferencias de concepto a destacar entre su trabajo y este trabajo. La primera, su trabajo se centra en el caso del estafador telefónico, que es la definición popular, mientras que este se centra en los dos conceptos raíz de la I.S. La segunda, su trabajo lo enfoca a la defensa, mientras que este se centra en el ataque.

En la figura 2.3 se muestra el ciclo definido por Marcus Nohlberg, comparable con el de Kevin Mitnick, aunque más completo y complejo. El ciclo se compone de tres agentes, que representan de fuera a dentro, el ataque, el defensor y la víctima, compuesto de cinco fases cada agente. La primera fase es la Plan/Deter/Advertise. Por lo tanto, en cada fase, se definen tres estados que definen donde se sitúa cada agente involucrado.

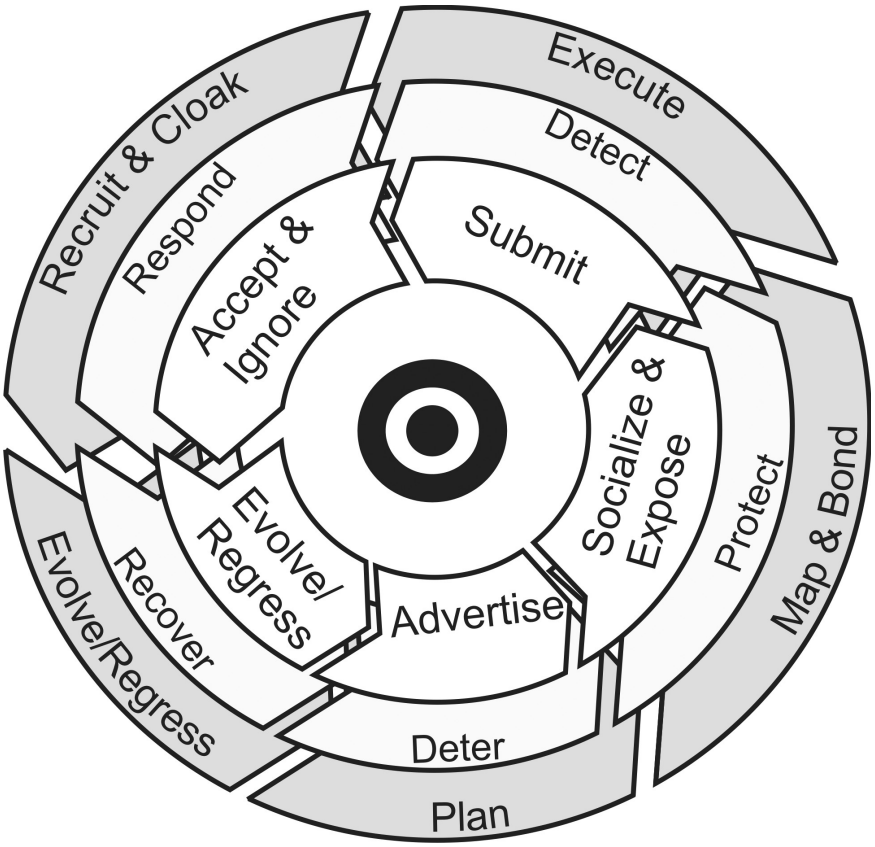

Ilustración 2.3: Ciclo de I.S. definido por Marcus Nohlberg [2]

Hay dos conceptos en la defensa ante un ataque de I.S. a destacar. El primero se basa en el trabajo que menciona de Gragg [7], "*A multi-layered defense against social engineering*", el cual expone la necesidad de diferentes capas defensivas, para incrementar el número de obstáculos al atacante y evitarlo. El segundo, la educación como la medida más significativa

para una buena protección, tratando de que no sea inadecuada y pueda usarse en contra.

Ingeniería Social Automatizada (I.S.A.)

La I.S.A. es una evolución de la I.S., basada en la automatización de los componentes presentes en la I.S. Estos componentes suelen ser el medio y el atacante. Además, conlleva un cambio de filosofía a la hora de diseñar el pretexto. La diferencia entre una herramienta que automatiza faena y una prueba de concepto de I.S.A. es la independencia que esta segunda toma a la hora de realizar el ataque, gracias a la presencia de elementos psico-sociales incorporados a su lógica de control.

Hasta 2007 no se han planteado experimentos de I.S.A, dada su complejidad técnica y la necesidad de disponer como medio redes sociales para tener un impacto relativamente visible. A día de hoy, ya ha demostrado ser una herramienta muy peligrosa, sobretodo por la dificultad en detectar que está suplantando a una persona o interfiriendo una comunicación. El concepto está muy enlazado con la inteligencia artificial, dada la necesidad de mutar con nueva información.

Como principales autores de este nuevo concepto están Markus Huber con su tesis de máster en Computer Science [8] y el equipo francés EURECOM con su trabajo titulado “HoneyBot” [9].

Tanto en el Capítulo 3 como en el Capítulo 4 se vuelve a retomar el concepto de la I.S.A., contextualizando de donde procede, sus matices, y explicando una prueba de concepto abstracta a raíz de la automatización de suplantaciones en redes sociales.

2.3 FUNDAMENTOS: ACOTANDO EL CERCO

Aparte de ofrecer la definición de `persuadir a alguien para que haga algo´, es difícil estructurar más el ataque, por ejemplo, a través de patrones. Dado que el ataque se adapta al perfil de la víctima, es muy grande el número de posibilidades. Aún así, se puede acotar con algunas características genéricas, propuestas a partir del estudio de la documentación y el conocimiento de los ejemplos vistos en el Capítulo 3.

Es un proceso simple de cara a la víctima

No importa si elaborar el proceso previo al ataque es sencillo o complejo, pero la víctima debe apreciarlo como algo normal, es decir, algo cotidiano, rutinario, sin dudar de ello ni alarmarse, a excepción de que el proceso contemple esa acción.

En casos donde se piden cosas imposibles para la víctima, esta acaba pidiendo ayuda o comprobando los datos, lo cual aumenta las posibilidades de destapar la coartada del atacante. En cambio, pedir información en un centro de atención al cliente es algo normal.

Es eficaz y peligroso por su impacto

Los ataques de I.S. se caracterizan por conseguir su fin en la mayoría de casos. No se necesitan ni grandes habilidades ni víctimas necias para conseguirlo, a veces sólo basta con tener un buen argumento o accionar el estímulo correcto.

En el examen para ser `*Certified Information Systems Security Professional*´, que es una de las certificaciones con más prestigio en el ámbito de seguridad informática, se clasifica la I.S. [10] como “el ataque más eficaz cuando los sistemas de seguridad están bien protegidos”.

Extrae nuevas funcionalidades a partir de características ya existentes

El atacante suele aprender a utilizar las herramientas que forman el sistema e intenta descubrir nuevas funcionalidades que le sirvan de punto de apoyo para realizar el ataque. A veces estas funcionalidades son las que definen el sistema, pero nadie antes había valorado que se les podía sacar provecho de otra manera. Descubrir y aprovecharse de esas deficiencias es el mayor logro de un profesional de I.S., los cuales recomiendan innovar y no reutilizar.

En las estafas bancarias, el primero en realizar un nuevo ataque siempre pasa desapercibido, en cambio, cuando el ataque se generaliza, el estafador corre peligro.

Se presenta en el momento adecuado y aparece como si fuera algo espontáneo

El atacante va generando una confianza entre él y la víctima, elaborando una relación que con el tiempo se hace más fuerte. En el momento en que la víctima este en un peor momento, el atacante debe utilizarlo a su favor.

Un mal ejemplo se ve en el Capítulo 2.1, donde se estudia un virus que, liberado en un viernes, su expansión se vio mermada por ser inicio de fin de semana. Si se hubiera liberado un lunes, su impacto hubiera sido mayor.

No quemar la fuente

La I.S. requiere bastante esfuerzo para establecer una relación de confianza con la víctima, la cual acaba haciendo un favor al atacante. Burlarse una vez conseguido el objetivo conciencia a la víctima, haciendo que sea más difícil que vuelva a caer en la trampa en un futuro.

En Internet, la transparencia en un ataque puede significar que la víctima permita utilizar su cuenta robada durante más tiempo. Si esta persona se entera que se está utilizando su cuenta para un fin ilegal, seguro que revisa e incrementa sus políticas de seguridad.

3 PSICOLOGÍA SOCIAL EN LA INFORMÁTICA

Este capítulo tiene el objetivo de abarcar elementos de la informática potencialmente relacionados con la psicología social, o área psico-social, para exprimirlos desde un punto de vista diferente. Los dos puntos clave son los factores humanos y la seguridad informática. Estos dos puntos se tratan en cada uno de los cinco apartados, dividiéndolos en pasado, presente y futuro. El pasado permite replantearse los conceptos desde otro punto de vista, el presente permite analizar los componentes más relacionados a la I.S. y el futuro permite intuir como lograr potenciar dichos ataques. Al final, se aglomeran muchas ideas diferentes, de las cuales el lector puede sacar sus propias conclusiones. Los cinco casos escogidos son los siguientes:

- Virus `*I love you*´: Se trata un caso ocurrido hace una década para entender la repercusión real de un ataque de I.S. bien elaborado.
- *Scam* y *Scambaiting*: Se tratan las estafas económicas en las que intervienen diferentes roles, con un constante aumento en la sofisticación del pretexto.
- Sabotaje: Se busca un análisis en extensión, estudiando algunos ejemplos, protocolos y herramientas relacionados con la I.S.
- Inteligencia Artificial: Se busca un análisis en profundidad, apoyado en la investigación de otras áreas para mejorar los ataques de I.S.
- Captura la bandera: Se trata de una competición que demuestra que todavía hay carencias en las medidas tomadas por las empresas.

Los cinco apartados no engloban todos la materia relacionada a la seguridad informática ni toda la materia relacionada a los factores psico-sociales, pero aporta nuevas conclusiones gracias al enfoque que se lleva a cabo (Fundamentos Capítulo 2.3).

3.1 VIRUS `I LOVE YOU´: UN PRECEDENTE

"A person who trusts no one can't be trusted"
Jerome Blattner

Con el objetivo de aprender de un caso real se ha elegido el virus `I love you´, el cual infectó millones de ordenadores enviándose él mismo a través del correo y del chat. Este virus es un ejemplo completo pero sencillo, dado que se nutre de diferentes factores psico-sociales para propagarse, pero su complejidad tecnológica es baja.

Durante los próximos puntos, se hace un seguimiento de su propagación, se comentan algunas de las repercusiones que produjo, se analiza el código fuente para ver su *modus operandi* desde dos puntos de vista diferentes, y finalmente se reflexiona sobre tres mercados actuales que podrían tener relación con un `I love you´ versión 2011.

3.1.1 HISTORIA CRONOLÓGICA DEL VIRUS

La cronología de los sucesos fue la siguiente [11][12]:

- El viernes día 4 de mayo de 2000, el virus se libera, infectando máquinas y propagándose entre usuarios de Internet con un sistema Microsoft Windows.
- El día 5 de mayo se reportan nuevas variantes, con diferentes asuntos, como "*Very Funny*", "*Joke*" y "*Mother's Day*". En pocos meses, se llegaron a reportar hasta 20, algunas con partes del código optimizadas.
- El día 8 de mayo, el N.B.I. (*National Bureau of Investigation*) filipino, colaborando con la Interpol y el F.B.I. (*Federal Bureau of Investigation*) norteamericano, arrestó a Reonel Ramones, de 27 años, como sospechoso de su propagación mundial.
- El día 11 de mayo, One de Guzmán, de 24 años, reporto que él había desarrollado un proyecto de tesis en el cual desarrollaba un virus muy similar al utilizado.

Finalmente, aún intentando arrestar a Reonel por delitos de robo de contraseñas, no pudieron hacerlo efectivo debido a las leyes Filipinas, donde la pena por escribir *malware* era

inexistente. Tres meses después, el congreso filipino entablo la ley No. 8972 [13], conocida como “la ley del comercio electrónico”, para poner fin a casos similares.

Sobre el coste total económico, los expertos lo estiman en miles de millones de dolares. Trend Micro Inc., empresa de antivirus, mostró varias cifras generales del número de ficheros infectados dividido por países el día siguiente a su lanzamiento. Datos posteriores acercaron la cifra de ordenadores infectados en todo el mundo a más de 50 millones.

Norte América	2.500.000
Europa	325.000
Argentina	150.000
Asia	129.000
Japón	40.000
Australia y Nueva Zelanda	25.500
Total de contabilizados	~3.300.000

Tabla 3.1: Cifras de Trend Micro Inc. el 6 de mayo sobre el virus [14]

En la tabla 3.1, hay tres factores culturales importantes a destacar:

1) El día de la semana. Dado que se liberó un viernes, había países que fueron menos castigados por entrar en el fin de semana, y al lunes siguiente, ya habían sido advertidos del riesgo. Por entonces el mayor número de ordenadores se concentraba en las empresas, que era la herramienta de trabajo para muchos empleados.
2) El idioma. Se suponía que iba a paliar la propagación del virus. Es lógico pensar que si se recibe un correo en un lenguaje que no se comprende, al menos se tomen medidas extras de protección, como preguntar directamente al remitente. Hay casos, como Argentina, que demuestran que la gente abrió el correo sin ninguna comprobación.
3) La religión. En India no hubo a penas contagio porque nadie se atrevía a abrir un archivo que contenía palabras clasificadas como prohibidas en su religión. Proveedores de servicios indios aseguraban que sus clientes no reportaban actividades sospechosas.

En el año 2003, el FORSCOM (*U.S. Army Forces Command*), hizo público un informe de 67 páginas [15] con las conclusiones de este virus. Reportaron 2258 servidores infectados, 12.010 horas de personal pérdidas y un coste total aproximado de 79.200 dolares. Los antivirus Norton y McAffee, presentes en todos sus departamentos, fueron ineficaces para parar la propagación.

3.1.2 ANÁLISIS DEL VIRUS

Después de conocer la historia, un análisis más extenso demuestra que participaron una variedad de factores bastante relevantes dentro del flujo de eventos. Son divididos entre la parte técnica, donde se analizan las funcionalidades del virus, y la parte psicológica/sociológica, donde se buscan respuestas a las preguntas de las relaciones causa-efecto detectadas.

3.1.2.1 TÉCNICO: CÓDIGO FUENTE

El lenguaje del virus es `Visual Basic Script´ (VBS) de `Microsoft Windows´, un lenguaje interpretado. Todos los usuarios infectados disponían del código. También se puede descargar de http://www.therockgarden.ca/security/love_letter.txt [16/11/2010].

El virus explota los siguientes elementos:

El sistema operativo era vulnerable por defecto

El sistema de interpretación de scripts, el cual ejecuta el lenguaje VBS, estaba activado por defecto en todas las ediciones `Microsoft Windows´ del mercado. Esta era una herramienta desconocida por la comunidad. Además, también contribuyeron las inexistentes políticas de seguridad en la separación de permisos, para modificar tanto ficheros de terceras aplicaciones, como ficheros y registros del sistema.

Camuflaje de la extensión original

El virus aprovechaba que el sistema operativo ocultaba la última extensión del fichero. Esto era debido a que el sistema leía de derecha a izquierda hasta el primer punto y lo ocultaba, dando la sensación que el nombre fuera un fichero de texto. El icono que se utilizaba para las extensiones ".vbs" era similar al del fichero de texto de extensión TXT.

037	`c.Copy(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")`
262	`male.Attachments.Add(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")`

Tabla 3.2: Código: Partes donde se le asigna el nombre real

Propagación por correo a través de Microsoft Outlook

Uno de los vectores de propagación era el correo electrónico. La función `spreadtoemail()´ era la encargada de leer la agenda de contactos del `Microsoft Outlook´ y enviar un correo con el virus a cada contacto. Esta acción sólo se realizaba una vez, a través de marcar una entrada en el registro del sistema. En el código se extraen los contactos a través de la variable `out´.

```
258 set male=out.CreateItem(0)
259 male.Recipients.Add(malead)
260 male.Subject = "ILOVEYOU"
261 male.Body = vbcrlf&"kindly check the attached LOVELETTER coming from me."
262 male.Attachments.Add(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
263 male.Send
```

Tabla 3.3: Código: Creación del nuevo correo

Infección de ficheros, persistencia y propagación por chat

El último paso del virus, era recorrer todo el disco duro y modificar ciertos ficheros dependiendo de su extensión. Las extensiones afectadas eran: VBS, JS, JSE, CSS, WSH, SCT, HTA, JPG, JPEG, MP3 y MP2. Como caso especial, trataba el ejecutable de la aplicación mIRC, un cliente de chat. El objetivo era reemplazar el fichero `script.ini´, el cual contenía las acciones predeterminadas del cliente, y añadir la funcionalidad de que cuando alguien entrará a la misma sala de chat que la víctima, esta le enviase el virus en versión `HTM´, como si fuera una web.

```
176 scriptini.WriteLine "n0=on 1:JOIN:#:{"
177 scriptini.WriteLine "n1=  /if ( $nick == $me ) { halt }"
178 scriptini.WriteLine "n2=  /.dcc send $nick "&dirsystem&"\LOVE-LETTER-FOR-
    YOU.HTM"
179 scriptini.WriteLine "n3=}"
```

Tabla 3.4: Código: Propagación por DDC

Otros vectores de ataque

Al iniciarse el virus, una de sus funciones era apoderarse del registro de sistema y iniciarse cada vez que se reiniciará el ordenador. Otro de los objetivos del autor era poder ampliar las funcionalidades en un futuro. Diseñó el virus para que al abrir `Internet Explorer´, descargará de una de las cuatro rutas preparadas, todas con dominio http://www.skyinet.net/, un programa ejecutable llamado `WIN-BUGSFIX.exe´. Este programa no se ha podido localizar, pero información externa apunta a que es un recolector de contraseñas, es decir, un *keylogger*. El hecho de acceder al exterior a buscar un programa estático ayudó a las autoridades para detectar y disminuir el daño del virus, mediante solicitar a los ISP que bloquearan dicho dominio,

deteniendo al menos la propagación de este ejecutable.

Ataque a través de scripts ActiveX

El virus, en su proceso, creaba un fichero con extensión `HTM´ dentro del sistema, que era el que se enviaba por mIRC. Este virus se aprovechaba de la activación de ActiveX en el navegador para poder ejecutar código VBScript, el cual auto-contenía al virus. Esta parte es la más compleja a nivel de código.

3.1.2.2 PSICOLÓGICO: INTERFAZ, CONTENIDO Y PROCEDENCIA

Además de los factores técnicos y sociológicos, hay tres más que contribuyeron a que el usuario ejecutara inconscientemente el virus:

Falta de advertencias visuales

Cuando se realiza una tarea rutinaria, se acaba realizando de forma inconsciente. Si un usuario recibe un correo y este tiene un adjunto, es normal que quiera saber de que se trata. El sistema operativo debe actuar en consecuencia, avisando y evitando el riesgo lo máximo posible.

1) Faltó alguna advertencia del peligro que podía significar ejecutar ese tipo de fichero, dado que se estaba realizando una operación con permisos de administrador, poniendo en peligro el sistema. Actualmente, la versión de Windows 7 pone solución a este problema: avisa al ejecutar un fichero y avisa cuando el sistema necesita permisos de administrador. Esta medida no elimina ni virus ni troyanos, pero dificulta su propagación.

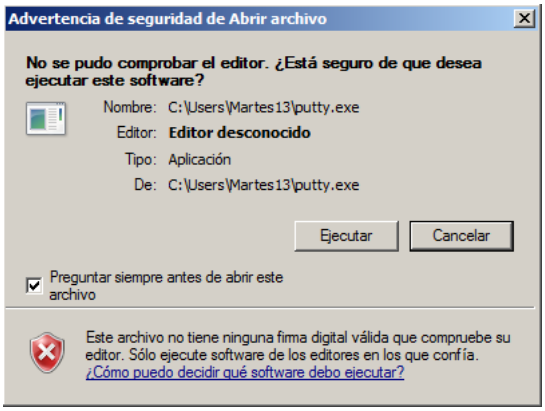

Ilustración 3.1: Windows 7: Mensaje de advertencia

2) El icono que se utilizaba era muy similar al del fichero de texto que podía tener el `TXT´ original. A día de hoy sigue ocurriendo; aunque no sean idénticos, cuesta diferenciar los diferentes iconos. En la figura 3.2 hay nueves iconos de Windows 7, apreciando la poca diferencia entre ficheros potencialmente peligrosos como el `BAT´, `WSF´, ´VBS´, `EXE´, `JS´ y `JSE´, en comparación con los `TXT´, `ODT´ y `CONTACT´.

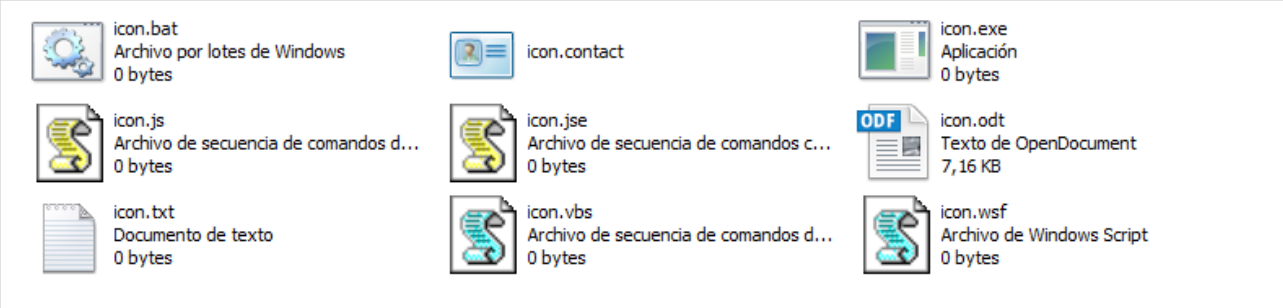

Ilustración 3.2: Windows 7: Iconos por defecto

Palabras escogidas

Se han buscado todas las palabras que pertenecían al texto que leía el usuario 3.5. La primera fila indica donde se encontraba ese texto, la segunda linea se compone de las palabras del texto y la tercera es el número de resultados que Google detecta en cada una de ellas, devuelto en millones de unidades.

Título del mensaje		
I	*LOVE*	*YOU*
8.720M	1.620M	8.390M

Texto del mensaje								
kindly	*check*	*the*	*attached*	*LOVE*	*LETTER*	*coming*	*from*	*me*
30.5M	868M	12.300M	124M	1.620M	252M	357M	8.470M	2.590M

Nombre del fichero			
LOVE	*LETTER*	*FOR*	*YOU*
1.620M	252M	11.210M	8.390M

Tabla 3.5: Conteo de las palabras utilizadas

Habiendo casi 7.000M de personas en todo el mundo, donde no todos tienen acceso a un

ordenador, aquel que dispone de uno generalmente acaba leyendo cosas en inglés. En este caso es probable que sepa el significado de la palabra `*love*´, y lo relacione con algo agradable. Esta relación, causada por un estímulo, recibe el nombre de *anchoring*, y se trata en el Capítulo 3.5.3.1. Así que de haber utilizado una palabra compleja, hubiera pasado más desapercibido.

Procedencia de confianza

El medio podía ser chat o correo, pero ambos procedían de un remitente conocido con el cual se compartía algo. Normalmente ningún amigo envía algo perjudicial, por lo que se procedía a saltarse la duda de comprobarlo. Esta falsa sensación se crea también en otros contextos, como cuando hay mucha gente utilizando una determinada aplicación. Se piensa que es segura ya que alguno la habrá revisado y se habrá preocupado de examinar si llevaba algún tipo de *malware*. Robert Cialdini, en su tesis doctoral [16], comenta seis patrones comunes de influencia. El patrón que se ejecuta en estos casos se llama `*Social Proof*´ o consciencia social.

Actualmente, los clientes de IRC suelen bloquear por defecto el envío DCC (*Direct Client-to-Client*), y el correo dispone de aplicaciones capaces de aplicar las siguientes medidas:

1) Filtros complejos de detección de *spam*, comprobando las palabras que componen el título y el cuerpo del mensaje, la dirección hacia donde apuntan los enlaces si existen, los archivos adjuntos, etc. Muchos tienen antivirus incorporado.
2) El sistema SPF (Convenio de Remitentes, del inglés *Sender Policy Framework*) es una protección contra la falsificación de las direcciones de envío. El servidor que envía los correos debe firmarlos para poder alegar la procedencia real.
3) Nunca ejecutar por defecto ni mostrar las imágenes o elementos multimedia de los ficheros con contenido HTML, impidiendo que el emisor conozca detalles técnicos del receptor, como su versión de navegador, su dirección IP, etc.

Aparte de lo visual, el contenido y la procedencia, se pasa a valorar las dos partes implicadas. Estas son el atacante, que es quien se supone que envió el virus a la red, y las víctimas, que principalmente fueron trabajadores de grandes y medianas empresas.

Respecto al atacante

Cometió dos errores graves. El primer error, utilizar código de otra persona. Esta otra persona salió en su propia defensa, lo cual restringió el cerco de búsqueda de culpables por el origen de

su publicación. El enlace que se detectó fue que tanto el culpable como esta persona habían compartido facultad. El segundo error, fue el afán de protagonismo. El culpable había añadido en dos partes del código datos personales que podían identificarle.

001 002	REM barok -loveletter(vbe) <i hate go to school> REM by: spyder / ispyder@mail.com / @GRAMMERSoft Group / Manila,Philippines
HTM	<META NAME="Generator" CONTENT="BAROK VBS - LOVELETTER"> <META NAME="Author" CONTENT="spyder / ispyder@mail.com / @GRAMMERSoft Group / Manila, Philippines / March 2000"> <META NAME="Description" CONTENT="simple but i think this is good...">

Tabla 3.6: Código: datos personales del culpable

Respecto a las víctimas

Las víctimas habían sido dotadas de una herramienta que les jugo una mala pasada. No fueron las culpables de que el sistema no pusiera medidas de seguridad, pero si que fueron el elemento más significativo para que se propagará el virus. A partir de esta experiencia, se debe concienciar a los usuarios que pueden aparecer nuevos peligros sin documentar en un futuro, lo cual requiere de sus propias habilidades para evitarlo en la mayor medida posible.

3.1.3 EVOLUCIÓN: EXPANSIÓN INFORMÁTICA

A partir del virus estudiado y otros ataques, la informática ha ido aprendiendo de sus debilidades. Mientras que se han creado nuevos mercados con nuevas amenazas, se han ido proponiendo y configurando nuevas medidas de protección.

A continuación, dejando de lado el virus `I love you´, se estudian tres elementos relacionados con los flujos de información de recientes tecnologías.

3.1.3.1 MONITORIZACIÓN Y SUPERVISIÓN

Con la intención de visualizar el *e-crime* o crimen electrónico, se presentan un par de gráficos ofrecidos por el grupo Cymru, que se encarga de seguir el proceso a gran escala de la evolución y propagación de los nuevos ataques. En ambos gráficos se mantiene la siguiente escala de

colores de la figura 3.3.

Ilustración 3.3: Escalera de colores (marcan cantidad) [17]

Las zonas rojas y blancas (derecha), son zonas con gran porcentaje de ordenadores infectados con algún tipo de *malware*, justo lo contrario que las zonas negras y azules (izquierda). El gráfico 3.4 trata de un mapa mundi con las zonas infectadas marcadas.

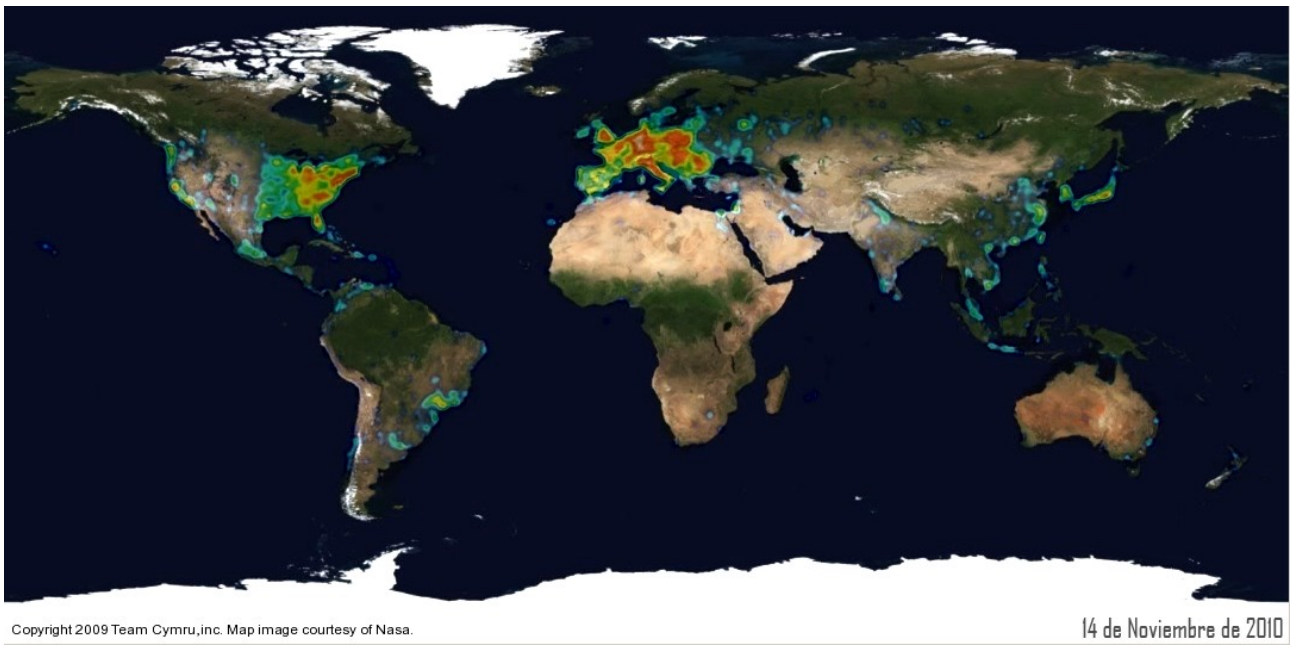

Ilustración 3.4: Mapa mundi de actividad maliciosa [17]

Norte América, Europa, Japón, China y Brasil son los puntos de mayor concentración. Hay una correlación entre los países del primer mundo y el número de ordenadores infectados. Si se pudiera hacer un análisis de donde proceden los ataques, seguramente el gráfico seria diferente, aunque se apunta a que los puntos clave son Rusia, Brasil y China. Aún así, otros países, como los africanos, son los más utilizados para usarse de puente/*proxy*. Estos países usados de puente son aquellos que no disponen ni de legislaciones ni de controles, viviendo una situación semejante a la de Manila antes del virus “I love you”, donde la pena es muy baja.

El gráfico 3.5 se llama mapa de Hilbert, por el algoritmo que utiliza, llamado Curva de Hilbert.

Este algoritmo permite distribuir en bloques de cuatro todas las agrupaciones de IPs. Cada píxel que vemos representa 4.096 direcciones IP.

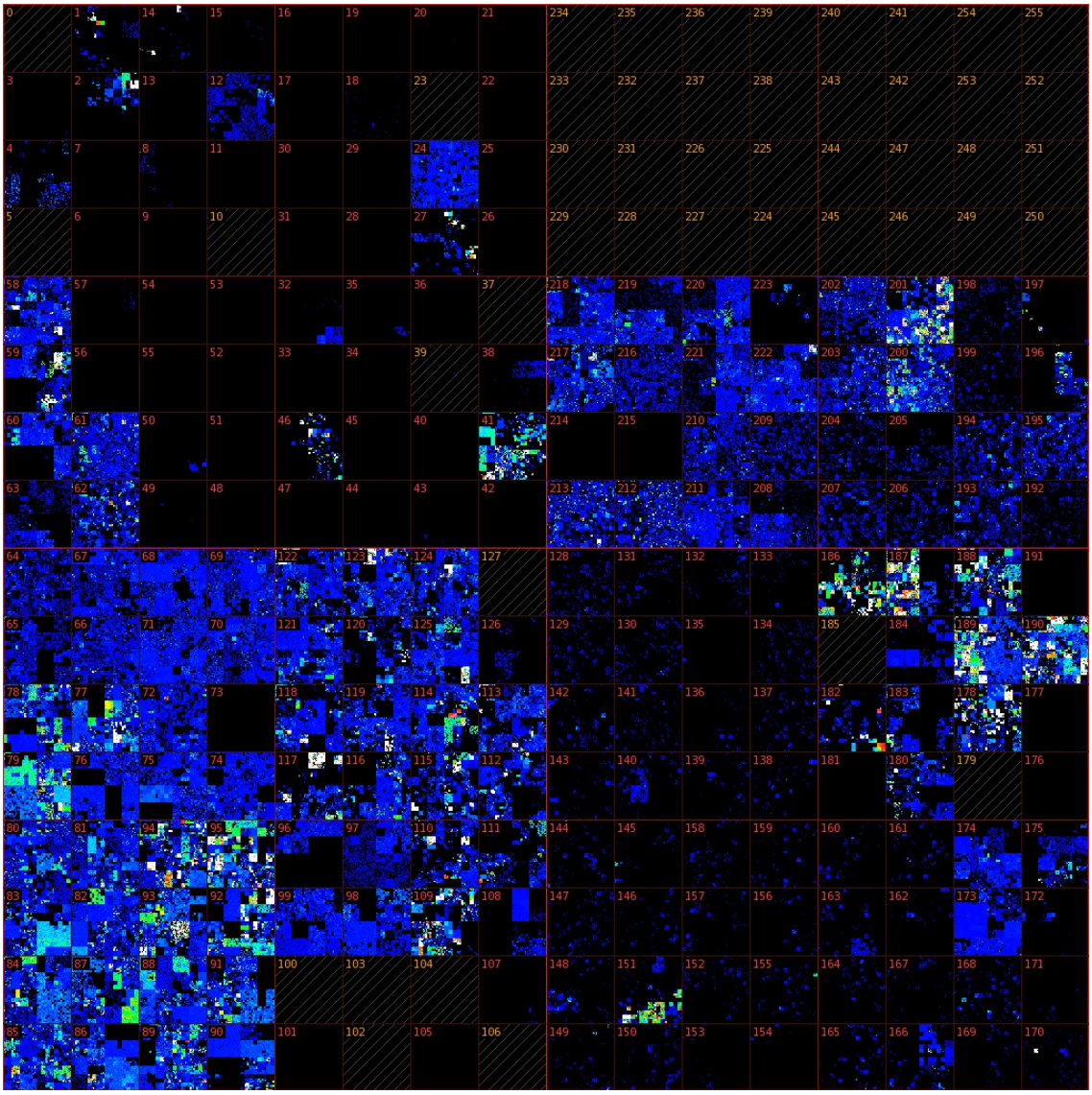

Ilustración 3.5: Mapa Hilbert de actividad maliciosa [18]

El número que aparece es el inicio de la IP. Las zonas naranjas, que también están ralladas, son niveles *bogons,* las zonas negras, no tienen *malware*, las zonas azules tienen algo, y así se incrementa hasta el color blanco. En general, hay zonas realmente castigadas, que se corresponden a las zonas marcadas en el mapa mundi como infectadas.

3.1.3.2 INFORMACIÓN Y REDES SOCIALES

Al recoger información, llamado *harvesting*, es clasificada en tres tipos según su disponibilidad:

- Información pública: Es aquella que el usuario sube a la red alguna vez con conocimiento. Normalmente se puede encontrar a través de redes sociales, en foros, en listas de correo, etc. Uno de los problemas que hay es que esta información se mantiene durante años y años en Internet, quedando una huella de esa persona casi imborrable . Se pueden consultar repositorios de caches para acceder a versiones antiguas de páginas, o incluso sitios especializados de rastreo, como Google o ShodanHQ.com. Aplicando las llamadas técnicas de Google Hacking, que no son más que consultas muy definidas con palabras clave, se puede extraer también información sensible de personas o empresas.

- Información privada: Es aquella que el usuario introduce en una página o aplicación web, pero no quiere compartirla con nadie. Pueden ser datos personales, cuentas bancarias, fotografías, etc. y la pérdida de esta información suele ser por imprudencia y mala configuración de las opciones de privacidad del usuario, siendo indexadas un buscador, o por un ataque. Un ejemplo de ataque es el que se vivió con la `Lista Robinson´ [19], caso donde podría haber quedado expuesta por una mala protección de su web, hecho que hubiera tenido repercusiones a nivel de Leyes de Protección de Datos (LOPD), vista en el Capítulo 3.5.1.1.

- Información oculta: Es aquella que guardan aplicaciones en sus ficheros pero que normalmente no conocemos su existencia. Se les llama metadatos, y suelen estar presentes en ficheros de documentos de texto, en imágenes, e incluso en la comunicación de algún protocolo. En alguna ocasión, también se han encontrado usuarios y contraseñas de servicios FTP en el código fuente HTML de alguna página web. Existen herramientas tanto para extraer información de estos ficheros, por ejemplo, FOCA, como para borrarla, por ejemplo, OOMetaExtractor.

Referente a la información que se ha mencionado, esta se puede encontrar mediante diversos métodos: encontrar archivos con datos sensibles a través de redes *peer-to-peer*, poder localizar personas y saber que hacen estudiando su geo-localización, ofrecida por servicios que usan, analizar los metadatos EXIF que se añaden por defecto las cámaras digitales, encontrar registros perdidos con búsquedas en Google, etc. Johnny Long tiene un libro [20] donde

menciona y pone ejemplos de muchas de estas técnicas, titulado "*No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing*". El nombre proviene de la sencillez de obtener información por Internet.

Entrando a las redes sociales, con unos 550 millones de usuarios, Facebook se sitúa como la más grande y popular. Según el informe [21] hay más de 1 millón de cuentas comprometidas, utilizadas por organizaciones criminales para dos fines: hacer publicidad, en muchos casos portando *malware*, y obtener los perfiles completos de los contactos próximos, dado que las opciones de privacidad suelen denegar ciertos accesos a los desconocidos.

La disciplina de la `Interacción Humano-Computador Segura´, tratada en el Capítulo 5, se plantea la necesidad de disponer en todas las redes sociales de un panel de privacidad donde se controle que se permite ver y a quien. Es uno de los temas más trascendentes en la actualidad, donde hay un complejo debate sobre como gestionar, por un lado, el comportamiento de las masas, *Social Proof*, y por el otro, el diseño técnico de las redes sociales.

Sobre la red social Twitter, Daniel G.Siegel expone en su tesis [22] un nuevo planteamiento de como realizar un ataque. Propone que en vez de adaptar el ataque al perfil de una víctima concreta, mejor buscar el perfil adecuado para el ataque específico que se lleva a cabo. Esto consigue afectar a más personas dado que permite una mayor eficiencia en el rastreo de nuevas víctimas vulnerables. Este tipo de ataque también recibe el nombre de *whisphing*, de `*whale phishing*´, o ataque a gran escala.

Además, Twitter también esta generando algunos malos hábitos. Dado que sólo se permiten 140 caracteres, los reductores de direcciones como `bit.ly´ se han hecho muy famosos. `bit.ly´ proporciona poder crear un enlace exclusivo, que al acceder, automáticamente redirige a otro enlace. Esto crea un problema de visibilidad, porque antes de ir no se ve la dirección a la que se va, lo cual puede llevarte a una página maliciosa, donde reside un *exploit,* o una página dedicada al *phishing*.

3.1.3.3 HISHING Y SMARTPHONES

El *Hishing* es la mezcla de hardware y *phishing*. Este ataque se basa en la venta de periféricos nuevos o de segunda mano con *spyware* o *malware* instalado. En las modalidades de ataque, lo

normal es modificar sólo el software para que haga funciones maliciosas, pero también se han dado casos [23] donde se ha visto modificado el propio hardware.

Los *smartphones* son el último nuevo gran mercado que se ha creado, absorbiendo muchos nuevos usuarios día a día. En el sistema operativo Android, no fue hasta la existencia del virus DroidDream que se puso especial atención al tema de la seguridad.

El sistema Android, basado en el *kernel* de Linux, requiere permisos únicos otorgados al instalar cada aplicación, utiliza un sistema de firmas único para identificarlas, división de permisos entre las aplicaciones para que no compartan recursos, y la posibilidad de borrar aplicaciones remotamente, si estas fueron adquiridas a través del programa Market de Google. El virus DroidDream [24] fue capaz de sorprender este sistema de protección, ya que utilizaba un *exploit* para conseguir privilegios de administrador, descargando e instalado una aplicación sin necesidad del Market. Esta aplicación robaba los datos de usuarios, comunicándose con un servicio externo. Este virus demostró que la única vía para asegurarse era no padecer de la vulnerabilidad para conseguir permisos de administrador, presentes en todos los dispositivos con un *firmware* inferior al 2.3.3 (Gingerbread).

Pero la I.S. no precisa de métodos tecnológicos para afectar al usuario de un *smartphone*. Es posible crear una aplicación que requiera permisos para obtener los contactos o enviar mensajes de texto automáticamente, y hacer pensar que es legítima la aplicación. Sólo es necesario hacer que la aplicación haga alguna cosa real con los permisos otorgados para que no haya sospechas de porque se necesitan esos permisos. Por ejemplo, si la aplicación se supone que es un juego, levantará dudas si pide permisos de enviar mensajes de texto, pero si la aplicación sirve para configurar domóticamente el hogar, no. Este engaño se puede potenciar con las mismas herramientas que ofrece el SDK de Android, con ProGuard, ofuscando el código fuente de la aplicación para que sea muy difícil descubrir que se está haciendo.

Un caso también de reciente aparición [25], relacionado con el *hishing* y los *smartphones*, es que los desarrolladores de Android que ven que sus aplicaciones se comercializan gratuitamente en programas que no son el Market de Google, decidiendo infectar maliciosamente estas aplicaciones para que la gente no las consiga gratuitamente.

3.1.4 REFLEXIÓN

Primero se ha tratado el virus `I love you´, analizando sus valores sociales, culturales, psicológicos y técnicos. Después se ha tratado la evolución para controlar y visualizar plagas de *malware.* Finalmente se han tratado también los diferentes tipos de información, haciendo especial mención a las redes sociales y a los dispositivos móviles.

El lector debe empezar buscando su nombre en Google. Muchas veces no se es consciente de la información de uno mismo que hay por Internet. Es muy peligroso que el atacante sea capaz de sorprender con información publica, porque será cuando, como víctimas, pensemos que se está delante de alguien autorizado.

Es recomendable revisar la configuración de privacidad de todas las redes sociales en las que se tenga una cuenta, verificar que información se ha publicado por medios externos y, en caso de haber algo que no queramos que esté, ejercer el derecho de protección de datos para que se retire.

3.2 SCAM Y SCAM BAITING

"The important thing is not to stop questioning; curiosity has its own reason for existing"
Albert Einstein

En este apartado, se analizan algunas de las relaciones con lo lucrativo que es el *e-crime* o crimen electrónico, donde la I.S. tiene alguna relevancia. Se ve con un ejemplo simple y uno complejo, para más adelante definir roles y mostrar una pequeña introducción a la organización, de tanto la parte que obtiene beneficios, como de quien debe impedirlo. Finalmente, se ve el concepto más importante de la I.S. y la técnica más sofisticada del *phishing*.

3.2.1 HISTORIA: UN PAR DE EJEMPLOS

Un *scam* es una estafa a través de un medio electrónico. La diferencia con el *hoax* o bulo es que el objetivo del *scam* es conseguir una suma monetaria de la víctima, convirtiéndolo en algo delictivo. Se van a tratar un par de ejemplos cotidianos, aún llevando más de 20 años en práctica.

3.2.1.1 LA ESTAFA NIGERIANA

El nombre de la estafa nigeriana, también conocido como `timo 419´, proviene del número del artículo del código penal que se vulnera en Nigeria, sitio de donde proceden la mayoría de estas estafas. Esta práctica consistía en enviar miles de cartas físicas a los países del primer mundo pidiendo sumas de dinero a partir de una historia ficticia. Requería infraestructura, recursos económicos y contactos. Se estimaron [26] ganancias de hasta 5.000 millones de dolares, habiéndose considerado la tercera industria más grande del país.

Con la expansión de Internet, aparecieron y se organizaron en otros países: Sierra Leona, Costa de Marfil, Ghana, Togo, Benin y Sudáfrica. Además, para dar más confianza a las víctimas, han creado oficinas en sitios como Ámsterdam, Londres, Madrid y Dubai.

El símil con la I.S. es el pretexto que se crea. Cuanto más novedoso y creativo es, más

posibilidades de éxito de engañar a las víctimas. Es difícil sacar un patrón común para todos sus ataques, pero a rasgos generales, funcionan de la siguiente manera:

1) Se compra a un vendedor externo direcciones de correos de millones de personas.
2) Se envía un correo a cada una de estas personas.
3) A los incautos que contestan, se les sigue la conversación.
4) Si la víctima sigue mostrando interés, se le pide algo, que normalmente es dinero.

Estas etapas se complementan con directrices generales usadas para identificar al atacante, variables en cada caso:

- Aseguran promesas de compartir o transferir mucho dinero como recompensa
- Esperan una actuación urgente
- Usan cuentas de correo gratuitas
- Dicen ser personas importantes
- Dicen contactar por búsqueda personalizada
- Realizan propuestas con diferentes asuntos elaborados
- Buscan culpabilidad al gobierno o otras personas
- Buscan extranjeros
- Requieren la identificación de la víctima
- Piden dinero anticipado

Como víctima, el mejor consejo que se ofrece es usar el sentido común. Esto también es vigente en Internet, donde se pueden encontrar los siguientes casos:

- Contactan *webmasters* poseedores de *blogs*, a los cuales les ofrecen poner publicidad. El engaño se produce cuando se pacta publicidad por una cantidad X, y los estafadores envían un cheque falso de X+Y, donde Y no es una cantidad demasiado elevada. Después de enviar el cheque, piden que se les devuelva la diferencia. La víctima normalmente no comprueba el cheque falso dado que tardan en el banco muchos días, por lo que decide reenviar esa cantidad.

- Se hacen pasar por chicas rusas con poco dinero, las cuales se enamoran rápidamente de la víctima. Esta víctima tiene el perfil de hombre de 40 años, no demasiado agraciado en la vida, y que desconoce tanto Internet como la situación rusa. Estas chicas piden una cantidad económica para viajar hasta la víctima, alegando que es muy peligroso que vengan ellas dada su situación de peligro. El escenario inicial suelen ser páginas de contactos, donde se pueden empezar intercambiando fotografías o videoconferencia pre-grabadas por la webcam.
- Utilizan ebay y otras plataformas de compra-venta para producir diferentes tipos de fraude. Lo típico es que siempre se tenga que pagar a una cuenta bancaria, la cual no suele ser de un banco del país. Los escenarios más conocidos son dos. El primero, es robar la cuenta a un usuario ya veterano y utilizar su buena reputación para vender gangas en un breve periodo. El segundo, es crearse una nueva cuenta, comprar artículos instantáneos, como libros electrónicos, conseguir ellos mismos la buena reputación, y a continuación, hacer el mismo proceso de engaño. Nunca usan PayPal alegando que les cuesta mucho la comisión, cuando es el único método que garantiza la transacción.

Aparte de todo lo anteriormente expuesto, hay otra vertiente que se está explotando, que es que en vez de atacar a muy poca gente con grandes sumas de dinero, se tiende al método de pequeñas cantidades entre mucha gente. Este método es muy beneficioso para ellos, dado que los bancos no pueden auditar todo. Reciben el nombre de micropagos, y también se utilizan para blanquear dinero.

3.2.1.2 EL MÉTODO ESTADÍSTICO

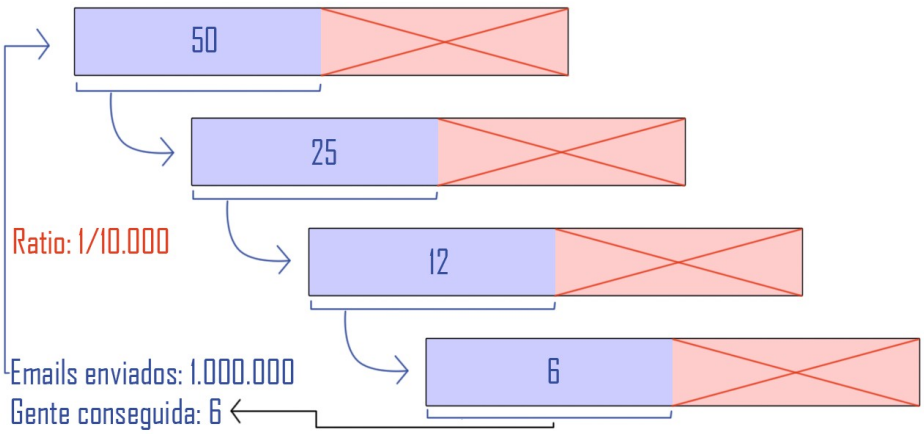

Ilustración 3.6: Esquema del scam estadístico

Este *scam* empieza enviando una cantidad muy grande de *spam,* invitando a la gente a realizar una apuesta en algo binario, es decir, se gana o se pierde, como, por ejemplo "¿quien ganará la *champions*?". A una mitad, se les dice apostar por una opción, y a la otra, apostar lo contrario. El porcentaje de gente que lo intentará puede ser de 1 cada 10.000. Se supone que 100 personas de 1 millón apostarán. Una mitad acertará y otra fallará. La que falle, desconfiará y se irá, pero los 50 que han acertado volverían a apostar. Este proceso se debe repetir al menos un par de veces más. Una vez se ha hecho cuatro veces, el número de personas que habrán acertado será de 6 personas aproximadamente, como se ve en la figura 3.6, disponiendo de su confianza para otros motivos.

Esta técnica es más compleja que la anterior, dado que aquí se ha demostrado algo que ha producido un beneficio. De los seis patrones identificados por Robert Cialdini, todos ellos aparecen en mayor o menor medida. Se enumeran y se relacionan con este ejemplo:

- *Reciprocity*: La consciencia de la víctima pide que se devuelva el favor.
- *Scarcity*: La víctima esta viviendo una oportunidad única.
- *Authority*: El atacante ha demostrado ser un experto.
- *Commitment and Consistency*: Todo lo que el atacante dijo se ha cumplido.
- *Social Proof*: No es sólo una víctima, sino que hay un grupo.
- *Liking*: Todas las víctimas comparten la pasión por el dinero fácil.

3.2.2 ANÁLISIS: ROLES

En el 2008, se reportaron 105.000 millones de dolares moviéndose en el mercado del *e-crime*. Si fuera un país, sería el 58 de 197, entre EAU y Nueva Zelanda. Si fuera una empresa, seria la número 15 del Fortune500, doblando a Microsoft [27]. Pero también se movió dinero en contratación de servicios para su protección. A continuación, analizamos los lados opuestos de los roles en el mundo de la seguridad informática.

3.2.2.1 VISIÓN DEL ATACANTE

Como muchas organizaciones, hay una cúpula que se encarga de dirigirla sin interactuar

directamente. Dado que estas actividades reportan mucho dinero, pueden subcontratar a expertos *freelance* a realizar estas acciones con dinero en negro. Estos expertos, se encargan de gestionar todos los servicios, muchas veces alquilando externalizando sus propios servicios. Dada la complejidad, se han creado diferentes áreas de especialistas muy concretas:

- Creación de *malware*. *Rootkits*, troyanos, virus y otros ejecutables que capturan el control de los ordenadores y los recopilan en servidores para su uso masivo.
- Configuración de servidores con diferentes servicios (IRC, HTTP, SMTP, etc.). Especialistas en administración de sistemas, quienes dirigen los paneles de control de los *bots* o máquinas *zombies*. Algunos sólo se dedican a alquilarlos, creando grandes redes de *bots* llamadas *botnets*.
- Buscadores de vulnerabilidades. El mundo de los *0days* o vulnerabilidades sin reportar necesita de gente especializada en la ingeniería inversa, dedicándose a vender por el mercado negro tales vulnerabilidades a un elevado precio.
- Blanqueo de dinero. Hay una serie de gente dedicada a hacer viable el negocio. Actualmente se utilizan los micropagos desde paraísos fiscales, pero también hay casos donde se utilizan personas inocentes para que ayuden en la tarea.

Son pocos aquellos que se siguen dedicando a contestar correos desde algún cibercafé de Nigeria. Uno de los últimos usos para esta gente es saltarse los *captchas*, que son las imágenes distorsionadas para verificar que quien envía la petición es una persona física y no un programa automático.

Finalmente, los datos personales, como tarjetas de crédito, perfiles, cuentas de correo, etc. son vendidos a clientes interesados.

3.2.2.2 VISIÓN DEL DEFENSOR

Hay tres grupos de defensores de la seguridad profesionales, a parte de los aficionados.

Auditorias Informáticas

En el 2009 nació el Consejo Nacional Consultor sobre Cyber-Seguridad (CNCCS), formado por las empresas más punteras españolas en esta materia [28]. Ya existían organismos así en Europa

y Estados Unidos, pero se decidió gracias al nivel demostrado y la experiencia aportada de las auditoras españolas. Los servicios que se proponen son, entre otros, la formación de profesionales con conocimientos del sector, investigación y desarrollo para la prevención de nuevas técnicas, vigilancia digital de imágenes corporativas o gubernamentales y un servicio anti-fraude.

CERT CSIRT / CVE

El `*Computer Emergency Response Team´* (CERT) se encarga de estudiar vulnerabilidades, el comportamiento de la red y ayudar a mitigar todos los problemas derivados. Es un equipo que se ha ido replicando por todo el mundo, hasta llegar a España. Uno de los servicios más conocidos es la organización de vulnerabilidades por los números de serie, `*Common Vulnerabilities and Exposures´* (CVE).

Ley

Para que puedan realizar los diferentes equipos sus tareas, deben ampararse en las leyes que existen para poder hacer eficaz su trabajo. Un ejemplo de esto es el que se ha visto del virus `I love you´, donde no se pudo procesar al delincuente. Es difícil marcar la delgada linea que diferencia un delito de otro y de no serlo. En el Capítulo 3.5.1.1 se abarca más este tema.

Dentro de cada apartado, se puede desglosar más, pero ya se aparta demasiado de la temática de la I.S. Lo que es importante destacar, es que dentro de todo el desglose realizado, actualmente, muy poca gente se está dedicando a temas de I.S. Esta reconocido que esta sea un problema muy serio, pero al ser algo poco tangible, es difícil justificar la necesidad de aportar más medios para su defensa.

3.2.3 EVOLUCIÓN: MÁXIMOS EXPONENTES

Un *scam* es algo simple, pero lo que lo hace efectivo es el número de complementos con los que se envuelve. Estos complementos pueden ser mejoras en la parte técnica o en la parte de los factores psico-sociales, tal como se describe la I.S.

A continuación se analiza uno de cada, el *rapport* por la parte de factor psico-social y el `*spear phishing´* por la parte técnica.

3.2.3.1 RAPPORT

En la comunicación cara a cara entre personas, los porcentajes del medio de transmisión de información están repartidos de la siguiente manera [29]: el 7% para las palabras utilizadas, del 20% al 30% para el tono de voz, y el resto, que va del 60% al 80%, para el lenguaje corporal.

Estos elementos son los que van creando la confianza entre las personas, pero en Internet, todo lo que no es multimedia, que es la mayor parte, es lenguaje escrito. Significa que el total del peso que antes se podía repartir está en un sólo elemento, que es el contenido.

En términos de I.S., existe la expresión de `crear *rapport*´ (acompasamiento en castellano). Se define como llegar a un estado mental de sincronización con otra persona, que produce ganas de compartir más momentos con ella. Es como la sensación de estar con una persona y estar convencido de que las palabras que le dices le están llegando y las está entendiendo tal como se quiere que las entienda.

Como se mencionó, el *rapport* se elabora a base de un sofisticado uso de las palabras. Hay que ser elocuentes y seguir una metodología, que consiste en saber qué quiere la persona y ofrecerle la solución adecuada. A partir de la repetición de dicho proceso, la persona otorga su confianza. El principio de que alguien confíe se basa en empezar confiando en él. Si el vínculo que se crea no es o no parece algo natural, falla.

En el ejemplo de la estadística se reflejan los elementos comentados, pero se puede llevar a otro contexto o acompañarlo con otros recursos. Por ejemplo, en las redes sociales, tener un perfil más completo da la sensación de ser más transparente, mientras que tener más amigos, da la sensación de ser una persona real y localizable. Markus Huber, en una de sus conclusiones de su proyecto de I.S.A., observa que su programa es descubierto por no realizar ninguna falta de ortografía, lo cual era un síntoma de duda.

Pero igual que se habla de contextos, se puede hablar de los otros recursos. Añadir videos, canciones, etc. esta considerado un gesto de cariño hacia la otra persona, dando a ver que se quiere compartir ese algo que resulta especial. También se pueden añadir juegos, encuestas, enlaces a otras páginas, etc. Cualquier elemento que es una excusa para seguir manteniendo el contacto con una víctima es suficiente motivo para ser valorada por un atacante. Cuanta más fe y menos crítica tenga la víctima, la situación es más adversa para esta.

Ilustración 3.7: Sentimientos: ¿No te sientes tierno al ver la imagen?

En la metodología de Kevin Mitnick, la primera etapa es extraer información, después crear un perfil, y finalmente aprovecharse de ello, pero a veces, si uno se dirige a un público demasiado general o se busca automatizar el ataque con una lista cualquiera de correos, se necesita otro tipo de técnicas. No ofrecen resultados tan exitosos como con un público concreto, pero es un inicio. Suelen ser técnicas de `programación neurolingüística´ o técnicas de mentalismo, utilizadas en otros ámbitos sociales.

Una técnica es el *cold reading*, basada en decir frases, llamadas *Barnum Statements,* de carácter general con cabida en cualquier situación. Un ejemplo serian las estafas nigerianas, donde en los primeros párrafos incluyen frases como “No aceptas que los otros te digan que creer” o “Te sientes culpable y preocupado sobre cosas que están fuera de tu control” para luego animarte y proponerte un buen negocio a partir de que reconocen tu valía. En Internet se puede realizar a través de otros contenidos, como en el ejemplo de la figura 3.7, que al ver un corazón rojo y un oso de peluche uno se siente más tierno.

A diferencia del *rapport* masivo, en el cual no hay más voluntad que conseguir el objetivo y olvidarnos, cuando se está elaborando *rapport* selectivo, se puede incluso poner a prueba que esté funcionando, creando nuevas situaciones que parezcan de rechazo a esa persona, para ver el comportamiento de la víctima y saber si se está logrando la sincronización.

Finalmente, una de las características de los ataques de I.S., es no quemar la fuente, es decir, mantener en secreto que algo haya pasado. Si se ha conseguido establecer *rapport* con alguien, y no es imprescindible romper esa relación, siempre será beneficioso para futuros usos, sino la víctima aprenderá y costará más engañarla la siguiente vez.

3.2.3.2 SPEAR PHISHING

El `*spear phishing*´ es una variante de *phishing*, pero más eficiente y compleja. Se puede equiparar con todo lo que se ha hablado de I.S., pero especializada sólo en el envío de correos. La idea es analizar *offline* las vulnerabilidades del sitio web, y aprovechar el momento adecuado para enviar un correo malicioso a algún usuario del sitio, imitando a uno auténtico.

Una opción, de las más utilizadas, es el cambio de remitente. Si se está utilizando el medio del correo, se puede aparentar como si el correo se hubiera enviado desde otra cuenta. Se puede lograr lo que el virus "I love you" lograba infectando el ordenador, pero sin hacerlo. Otra opción adicional, es plagiar la apariencia de un mensaje original para hacerse pasar como tal.

Por ejemplo, si se recibe un correo de PayPal diciendo que hay que cambiar la contraseña en una dirección web que se facilita, aunque el remitente sea del servicio técnico de PayPal y aunque tenga la apariencia, da desconfianza. Se puede preguntar uno mismo porqué él y porque entonces.

Ahora con la situación un poco más compleja. En ese correo, se dice que se debe cambiar la contraseña en una dirección web que se facilita, pero se argumenta que es debido a un error del sistema. Hasta aquí es lo mismo que el caso anterior, pero esta vez, se recibe el correo justo en el momento en que se acaba uno de registrar la cuenta. En la pila de correos hay uno oficial diciendo "Bienvenido a PayPal", pero justo encima de este, hay otro diciendo "Bienvenido a PayPal – Error en activación".

En este caso, sería lógico pensar que el único que conoce ese momento del registro a PayPal es él mismo, cosa que en muchos servicios no es así. Se debe a la información que desprende el sitio web, la cual se puede ir extrayendo y procesando con programas automatizados, llamados *spiders*.

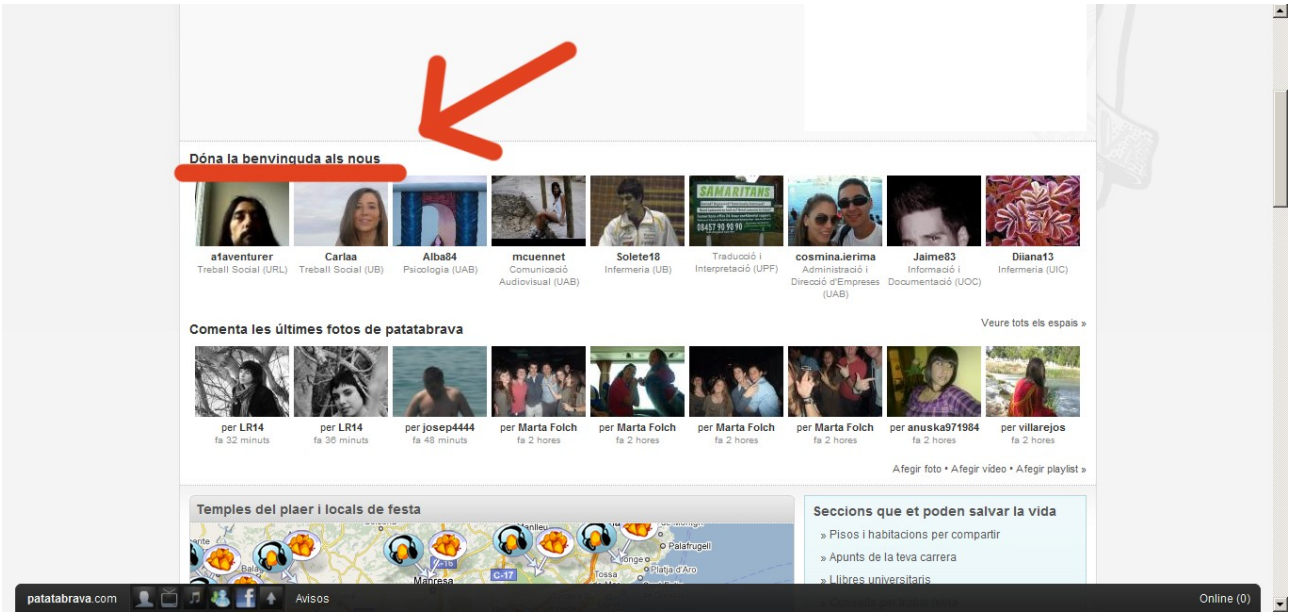

Ilustración 3.8: `Spear Phishing´: Ejemplo de web vulnerable

En la imagen 3.8 se ve una web que muestra la foto de perfil de los nuevos miembros, saliendo su nombre y cara por la página principal, dándole la bienvenida. Ese momento se puede aprovechar para enviar el correo diciéndole algo relacionado a su foto y validación. No es igual el conocimiento de una persona que se acaba de registrar en la página web que el de una persona que lleva tiempo, lo cual provoca que el inicio consista en seguir instrucciones donde las veas, siempre que estén entre unos límites del sentido común.

Finalmente, la complejidad de tal ataque irá vinculada al éxito de éste, y por suerte, como víctimas, no siempre es trivial extraer la información suficiente para realizar un `*spear phishing*´ de calidad. El problema que existe es que muchas webs son vulnerables porque su modelo de negocio es vulnerable de por sí, como la web de la imagen 3.8, dado que es una red social que esta utilizando a los nuevos miembros para conseguir más. Si quita esta característica de su web, ésta perdería valor.

3.2.4 REFLEXIÓN

Primero se han tratado un par de ejemplos de timos destacando sus factores psico-sociales. Después se han desglosado roles por atacante y por defensor en el área de la seguridad informática. Finalmente se han abarcado el *rapport* y el `*spear phishing*´, conceptos avanzados de la I.S.

El lector debe mentalizarse que la informática se ha vuelto bastante fácil, permitiendo preparar un ataque muy sofisticado en poco tiempo. Hay que hacer el esfuerzo de quitar el mito de que el atacante "no se va a tomar tantas molestias", porque elaborar un pretexto es económicamente muy barato con relación a lo que puede obtener. El nivel de conocimientos de un atacante puede llegar a superar incluso a un defensor, por lo que el ataque no será siempre cutre y sencillo. Normalmente dependerá de la información que se haya compartido de manera más o menos involuntaria en la red.

La mejor recomendación para evitar este tipo de ataques, tanto fuera como dentro de la red, es siempre dudar al menor síntoma de incoherencia, utilizando el sentido común.

3.3 SABOTAJE

"Cualquier tonto puede decir la verdad,
pero se requiere un hombre de cierto sentido para saber como mentir bien"
Samuel Butler

En este apartado se abarcan muchos temas diversos que todos convergen por su fin. En el primer punto hay un acercamiento al lado más personal y humano, en el segundo se habla de los protocolos y sistemas tecnológicos en general, y en el tercero se tratan las herramientas más actuales de I.S.

3.3.1 HISTORIA: LA OSCURA MENTALIDAD HUMANA

La I.S. se utiliza de manera inconsciente por todos en muchas situaciones. Una muy popular entre adolescentes es quitar la cuenta de correo de algún amigo o amiga. Aún sin poseer conocimiento técnico, a través de triquiñuelas se es capaz de conseguirla. A veces utilizan `*shoulder surfing*´, otras `*dumpster diving*´ y en otras llegan a preparar algún pequeño engaño de *phishing*. La repercusión es limitada y normalmente se hace por pequeñas disputas, pero no siempre las intenciones son tan inocentes.

Primero, dos ejemplos de ataques técnicos que llegaron a atentar contra vida humanas. Después, se ve un caso de la inocencia humana, donde se aprovecha el desconocimiento y las ansias de poder para convertir al atacante en víctima. Por último, una historia real de un grupo llamado ANONYMOUS contra una empresa de seguridad informática, utilizando la I.S. como herramienta para conseguirlo.

3.3.1.1 RIESGO HUMANO: EJEMPLOS DE FALTA DE ÉTICA

En marzo del 2008, una serie de personas no identificadas vulneró la seguridad de la página web epilepsyfoundation.org y modificó el código fuente, redirigiendo con *javascript* a una animación. La web es un foro de gente con problemas de epilepsia, y resulta que la imagen cambiaba varias veces de color por segundo, produciendo que reaccionaran los usuarios con ciertos problemas motores, incluso llegando a sufrir *shocks*.

Una acción típica de una broma podría haber causado algún fallecimiento, pero analizando otros casos de los que se dan constantemente, se observan peores intenciones.

Stuxnet, un gusano con *rootkit* incluido que afecta a sistemas Windows, se dedicaba a apoderarse de sistemas S.C.A.D.A. y manipularlos de forma oculta. Un sistema S.C.A.D.A. es un software de administración para el control, supervisión y adquisición de datos, es decir, es el software que utiliza, por ejemplo, una central nuclear para regular los cambios de temperatura, la presión, el estado del proceso, etc.

Los expertos de ingeniería inversa del Symantec Security Response determinaron que el objetivo de Stuxnet era un central nuclear de Irán. Lo peor es que los atacantes tenían conocimientos avanzados de sistemas S.C.A.D.A. en centrales nucleares, estando el código muy preparado contra cualquier intento de averiguar que quería hacer.

En la red ahora lo tachan de [30] “la pieza de *malware* más refinada jamás creada”. También podría haber recibido otros nombres más macabros si hubiera hecho que se escaparan gases o produjera una reacción en cadena, como se argumenta [31] que fue posible.

3.3.1.2 SPYWARE: OFUSCACIÓN Y DESCONOCIMIENTO

El *spyware* es un programa residente en un ordenador que va capturando datos sensibles del ordenador infectado y los va enviando al atacante. Un ejemplo conocido es el *keylogger,* programa que recoge las pulsaciones del teclado. También hay otras variantes como el *adware*, que sirve para insertar publicidad en los ordenadores infectados, y además, también tiene funciones de analizar que se visita para crear un perfil concreto de lo que le interese a la víctima. Pero a diferencia del *spyware*, el *adware* se suele instalar con consentimiento, así que no es ilegal si es autorizado.

Hace tiempo, lo frecuente era que el programador de *malware* fuera el mismo que lo utilizaba. Esto significa un conocimiento completo del código que se ejecuta. A día de hoy, se pueden comprar los servicios de un programador especializado que realice ese programa, o simplemente, buscar alguno liberado, por foros, listas de correo, etc.

Cuando se compra uno personalizado, el programador es recompensado, pero cuando se libera, normalmente no lo es. Esto ha llevado a que el programador se espabile e ingenie nuevas maneras de sacar provecho a sus programas.

Dada la sencillez que supone configurar un programa *malware* a través de una interfaz gráfica, ha disminuido la curiosidad de saber que hay detrás, anteponiendo las ganas de probar el programa al hecho de que sea lo que dice ser. Entra el factor `*social proof*´, donde ya se piensa que otra gente lo habrá revisado, y el factor de la ignorancia, donde aún queriendo, uno no es capaz de conseguir averiguar como funciona, simplemente por la complejidad que supone. Además, el programador puede utilizar `empaquetadores´ que ofuscan el programa para ponerlo todavía más difícil de entender.

Al final, la cadena se compone por el programador obteniendo una serie de personas que utilizan inconscientemente el *malware,* el cual tiene un *spyware,* que no deja de reportarle resultados sin él tener que hacer nada.

Uno de los intentos más importantes de la historia, es el que sucedió en el repositorio CVS oficial del Kernel de Linux [32], donde un atacante consiguió acceso privilegiado y modificó lineas del código por tal de obtener acceso root en cualquier máquina. Dada esta maniobra podría haber conseguido implementar una vulnerabilidad en millones de sistemas sin apenás esfuerzo. Esto no ocurrió dada la rápida intervención de los demás programadores de la comunidad, los cuales, al seguir una estricta política de *commits*, consiguieron ver la irregularidad en el sistema. Estas herramientas no son propiamente I.S., pero para aplicar I.S. hay que empezar conociendo el entorno donde uno se mueve, en este caso, la seguridad informática.

3.3.1.3 ANONYMOUS & HBGARY

Anonymous es un grupo creado a partir de varias redes sociales, sin objetivo definido, rebelándose a favor de la sociedad. Se les reconoce por llevar la mascara de V de Vendetta.

Hay un caso donde este grupo utilizo la I.S. para conseguir desmantelar una firma de seguridad informática, llamada HBGary, contratada por el gobierno de Estados Unidos para que investigará a Wikileaks y a sus miembros. La conversación, mantenida entre Greg, miembro de

Anonymous, y Jussi, administrador de sistemas de HBGary, fue la siguiente [33]:

De Greg a Jussi. Asunto: Necesito ssh en `rootkit´ (nombre del servidor)

G→J: Estoy en Europa y necesito acceso `ssh´ al servidor. ¿Me puedes abrir el cortafuegos y permitirme entrar a través del `ssh´ por el puerto `59022´, o otro cualquiera?. ¿La contraseña de `root´ sigue siendo `88j4bb3rw0cky88´ o ya se ha cambiado a `88Scr3am3r88´? Gracias.

J→G: Hola, ¿tienes la ip pública o necesito aceptar todo? La contraseña es `w0cky´, y no se permite acceder como `root´ directamente.

G→J: No, no tengo la ip pública ahora porque estoy preparado para una pequeña reunión y tengo prisa. Si eso ponme de contraseña `changeme123´, permite todo y yo ya conectaré por `ssh´ para configurarmelo.

J→G: Ok, ahora se acepta cualquier conexión en el puerto `47152´ por `ssh´. Lo estoy probando para asegurarme que funciona. Tu nueva contraseña es `changeme123´. Estaré conectado si me necesitas. En Europa... pero ¿no pasarás por Finlandia? :-) _jussi

G→J: Si puedo sacar más tiempo puede que nos veamos. Estaré por Alemania un poco más. Por cierto, no puedo conectar por `ssh´ en el servidor `rootkit´. ¿Seguro que la ip era todavía la `65.74.181.141´? Gracias.

J→G: ¿Funciona ya?

G→J: Si jussi, gracias. ¿Has reseteado el usuario `greg´?

J→G: No, tu cuenta tiene de nombre `hoglund´.

G→J: Cierto. Ya estoy dentro. En gratitud te enviaré pronto un correo. Estamos en contacto. Gracias.

[Tiempo después, Jussi sospechó]

J→G: ¿Abriste algo en un puerto superior?

(El texto esta traducido del original, cambiando algunas palabras para entenderse mejor.)

En este ejemplo real se ve lo sencillo que fue manipular a una persona sólo por hacerle pensar que, con quien hablaba, era una persona autorizada. Utilizando buenas palabras, preguntando poco a poco, dejando entrever las palabras y argumentando que tenia prisa, pasó de no tener nada, a obtener todos los datos necesarios para robar una base de datos de 50.000 usuarios.

3.3.2 ANÁLISIS: TECNOLOGÍAS

A día de hoy, se dispone de un escenario repleto de diferentes tecnologías. Esto significa que lo primero que hay que hacer es conocerlas, por lo menos algunas de las más populares y más relacionadas como posibles fuentes de extracción de información.

Primero, hay que mentalizarse que gracias a Internet ya es viable operar con grandes cifras. No es difícil lograr manipular a más de un individuo, sobretodo cuando todos cumplen el mismo patrón. Después, se analizan dos protocolos de comunicación, porque aunque la manera más común sea navegando, también hay vías alternativas a tener en cuenta. Todo vale cuando el premio lo merece. Y finalmente, a través de la duda, se intenta ser un poco más críticos con la información que se recibe.

3.3.2.1 BOTNETS

Una *botnet* es una red de ordenadores infectados, llamados *zombies,* controlados normalmente todos juntos desde un único sitio con el fin de hacer una tarea a la vez. Estos ordenadores están infectados por *malware* y se entienden con un mismo protocolo. Antes se usaba el IRC, pero actualmente la tendencia cambia a ser el HTTP [34]. Incluso se han utilizado redes sociales como Twitter para controlarlos a la vez [35].

Se pueden utilizar las *botnets* para recolectar información de forma rápida, anexándolos a algún programa *spider*, y hacer acciones personalizadas, como crear y usar cuentas de una red social. Con una *botnet*, a la hora de preparar una nueva táctica de I.S., se puede plantear a gran escala. Sólo es necesario crear un ambiente ficticio de `*social proof´* para dar la sensación que existe un trabajo en equipo.

Uno de los elementos para evitar el uso de *botnets* y la automatización de peticiones es el *captcha:* a través de una imagen con letras un poco distorsionadas se consigue que la máquina no reconozca el número a introducir. Se han desarrollado herramientas capaces de saltárselos, pero hay veces que la variabilidad hace más eficientes otros métodos.

Gracias a la I.S. es posible utilizar a otras personas a introducir el sistema de *captcha*. El proceso es el siguiente:

1) Se crea una página web que sea capaz de tener muchas visitas. Se suelen utilizar páginas web con contenidos eróticos o sitios de *hoaxes*.
2) Se crea un sistema gestor medianamente complejo, que recoge el *captcha* que se requiere saltar, y se almacena junto a la fecha y el identificador del *bot* que ha hecho la petición.
3) Se pasa en la página web el primer *captcha* almacenado del gestor. Si ha transcurrido mucho tiempo entre la fecha de ese *captcha* a la actual, se pasa al siguiente, y se refresca el anterior dado tener el *bot* que lo produce.

Si funciona bien, habrá gente intentando entrar en la página y al mismo tiempo resolviendo el problema sin darse cuenta.

3.3.2.2 DNS Y VOIP

En un ataque de I.S., para poder explotar la ventaja del conocimiento, hay que saber como funcionan los diferentes protocolos de Internet. Se analizan dos de muy populares.

En el caso del Domain Name System (DNS), existen denominaciones propias para los ataques, como es el *pharming*. Se basa en cambiar la resolución de los nombres de dominios para que apunten a servidores que están bajo el control del atacante. Lo que se explota es la apariencia de la web consultada, dando al usuario de pensar que está en el servicio original. Es una técnica realmente efectiva, pero no es fácil conseguir que la víctima este en el escenario adecuado. Las maneras más comunes son, modificar el archivo de configuración de DNS del sistema operativo, o atacar a los servidores DNS del servidor víctima. También se ha popularizado el atacar a los *routers* de los hogares, ya que también son parte de la cadena, transfiriendo los DNS pre-configurados a todos los ordenadores de la red interna.

Otro uso es extraer información. Si está mal configurado y permite hacer llamadas desde el exterior se pueden analizar una serie de valores que determinan si se ha estado visitando una web concreta anteriormente. Así se puede elaborar un perfil del usuario, es decir, se puede conocer qué servicios de correo utiliza, qué portales de noticias, etc.

Para protegerse uno del *pharming*, se han creado empresas como OpenDNS. Dispone de mecanismos para acelerar las búsquedas pero al mismo tiempo de filtros para la web, protección de redes contra *phishing*, contra *malware* y *botnets,* y hasta creación dinámica de *whitelists* y *blacklists*, que son listas de palabras clave que se permiten o se deniegan respectivamente.

En el caso del VoIP (*Voice over IP*), también hay denominaciones propias, como el *vishing* y el *SMiShing*. El termino *vishing* es una mezcla entre *voice* y *phishing*, y su objetivo es engañar a víctimas telefónicamente de forma automática. Los pasos son los siguientes:

1) Se hace un rastreo de números de teléfonos pertenecientes a una zona. Así se tiene en cuenta el idioma a utilizar.
2) La víctima que descuelga, escucha una grabación que advierte de un peligro con su tarjeta de crédito y pide que se llame a un número falso para cambiarla.
3) Al llamar, se le pide confirmación de los datos, momento en que son guardados.

Este ataque funciona bastante porque la víctima no sospecha un ataque tan sofisticado.

Para defenderse de esta técnica, además de tener que ir denunciando los números utilizados para que las autoridades los bloqueen, hay que tener el conocimiento de que existen. Los bancos y organismos competentes deben dejar claro cuales son sus maneras de contactar al usuario, y que nunca pedirán sus propios datos.

3.3.2.3 CONFIANZA ONLINE

Un fallo es pensar que se está haciendo algo para protegerse cuando realmente expone a peores consecuencias. Un ejemplo es cuando por utilizar una contraseña demasiado larga, se apunta en un *post-it* y se pega al lado del ordenador. Al querer aumentar la seguridad de la contraseña, se crea un peligro mayor.

En temas de apariencia, no hay que guiarse por elementos seductores, como dibujos, iconos, *banners*, sellos, palabras positivas o cualquier otro tipo de contenido. Absolutamente todo lo que se ve dentro del navegador es suplantable. Un ejemplo muy ilustrativo de mala repercusión es el que se está extendiendo desde Confianza Online. Es un proyecto que abalan

sitios oficiales como red.es, autocontrol.es y aecem.org, con la voluntad de recaudar dinero a través de la supuesta confianza que inspiran. Que ninguno de los tres sitios lo use correctamente en su página web es un símbolo para dudar de tal marca.

Ilustración 3.9: Confianza Online: sello y lemas

Según ellos, todas las webs que tengan ese sello, son webs éticas y con garantías. La realidad es que cualquier web criminal puede ponérselo, dado que sólo necesita pagar la cuota, sin ningún tipo de certificación extra. No existe panacea para comprobar si se navega en una web segura o no, y menos por una imagen. No utilizan ningún mecanismo tecnológico adicional, como pueden ser los certificado SSL, y aunque la web ConfianzaOnline.es si tenga un certificado SSL de primer nivel, no avala que sus clientes lo tengan.

Además, este sello abre nuevas oportunidades para ser explotado mediante la I.S. como, por ejemplo, incrustando el sello en un foro y reportándolo en el sistema de reclamaciones. Pondría a ambas partes en confrontación, pudiéndose aprovechar la información para sacar partido por alguno de los dos lados.

Hasta con la capa más técnica, aún siendo claramente útil, hay que ir con cuidado. Una de las medidas más eficaces, el certificado SSL (*Secure Sockets Layer*), puede verse comprometido si la red esta siendo atacada por un ataque *Man-In-The-Middle*, es decir, escuchando la comunicación entre ordenadores. Hay que utilizar un servidor externo que te certifique ese certificado para darle total validez, y aún así, si la versión del paquete SSL no esta actualizada, también se puede ver vulnerado. Existen herramientas como StripSSL o FireSheep que facilitan la tarea de espiar en redes ajenas.

3.3.3 EVOLUCIÓN: POTENCIANDO LA PLANIFICACIÓN

Durante este caso se han comentado bastantes elementos tecnológicos relacionadas con Internet. Por eso se centra este último apartado en programas que sirven para hacer pruebas de auditoria de I.S. También se ha añadido un punto de factores psico-sociales, la *misdirection* o distracción, para potenciar las planificaciones de los ataques.

Primero, se trata una técnica psico-social relacionada al proceso. Se basa en aprovecharse del desconocimiento y la opacidad que generan las nuevas tecnologías. Después, se tratan herramientas relacionadas con la I.S. Se comenta una *suite* creada especialmente para elaborar tácticas y escenarios de I.S. y otra independiente, para controlar y monitorizar a las víctimas que entren en una pagina web en concreto.

3.3.3.1 MISDIRECTION

La técnica de *misdirection* consiste en distraer a la víctima, normalmente por sus limitaciones perceptivas, mientras se busca un objetivo de forma desapercibida. La P.N.L. y el mentalismo son dos áreas de las que experimentan sus efectos. En el caso de tratarlo directamente con personas, se aprovechan las limitaciones de la memoria y la visualización. Por ejemplo, los magos la usan en sus espectáculos para sorprender a la gente. Les requiere mucha habilidad con las manos y una pequeña trama que seguir. En la I.S. se plantea similar.

Hay dos puntos claves al diseñar la trama. El primero, es entender cual es la salida de información (*output*), y el segundo, aprovechar el desconocimiento de las personas sobre las posibilidades que ofrecen las herramientas que utiliza.

Sobre el primer punto clave, la salida de información, en los ordenadores principalmente es la pantalla, aunque pueda también ser sonora o por un periférico. Por lo tanto, hay que poner especial énfasis a la apariencia de la interfaz gráfica.

Algunos ejemplos usados en webs son: ocultar ventanas, abrir nuevas, plagiar imágenes, plagiar distribuciones de objetos, mostrar advertencias, redireccionar varias veces, etc. A través de jugar con todas estas características, aparecen técnicas como el *tabnapping*, que consiste en una página maliciosa que se congela cuando se abre, pero al detectar que el usuario cambia de

pestaña en el navegador, se autorefresca con el diseño de otra página, por ejemplo, la del correo. Se puede incluso optimizar y abrirle pestañas adicionales a la víctima, o hacerlo más inteligente y sólo utilizarlo si abre una pestaña de ese mismo servicio, para aumentar la confusión.

Otro vector de ejemplos en aplicaciones ejecutables, son los programas gratuitos que luego resultan contener *spyware* o *adware*, descritos ya en el apartado 3.3.1.2.

Sobre el segundo punto clave, el desconocimiento, por lo general viene dado. Un usuario básico de Internet tiene una visión limitada más allá de lo que necesita, por lo que es muy fácil persuadir a esa persona con buenas palabras y buena educación hasta llevarla al escenario preparado. Contra un usuario avanzado, es cuestión de cogerlo desprevenido, pidiéndole ayuda y mostrando de forma indirecta el enlace hacia el escenario preparado. Pedir ayuda a un usuario avanzado para aprovecharse de él recibe el nombre de ingeniería social inversa. Un ejemplo de esta situación sucede con las aplicaciones de Facebook, donde el atacante guía a las víctimas a registrarse en un sitio externo. Esto no es necesario porque Facebook ya ofrece las herramientas suficientes para no necesitar salir de él.

Se ha popularizado bastante el combinar esta técnica descrita con otra llamada *clickjacking*, que consiste en sobreponer capas ocultas encima de la página enfocada, y hacer que la víctima haga click sobre lugares concretos, que estos a su vez accionan funciones de esas capas ocultas. Estas funciones hacen llamadas a webs externas sin una debida protección, pudiendo hacer acciones indebidamente. Este ataque el nombre de *Cross-Site Request Forgery*.

Hay más recursos sobre *misdirection*. Por ejemplo, la creación de falsos *logs* en el sistema y la creación de otros posibles elementos irracionales harían que, en caso de llegar a un análisis forense por parte de la brigada de delitos telemáticos, se dieran unos resultados tan absurdos que no se pudiera dictar sentencia por falta de coherencia.

Volviendo a un contexto humano, se destaca la presencia del engaño en la mayoría de ocasiones. Si se quiere mentir a la víctima, existen tres recomendaciones [36] a seguir:

1) Escoger la mentira correcta

   La historia que se explica debe ser coherente. Mejor no intentar explicar a un experto algo que hace cada día o explicar que algo sobre un lugar cuando pueda haber alguien que

también haya estado, sabiendo que lo que se dice no tiene sentido.

2) Estar en relax

Hay que auto-convencerse de la mentira con técnicas como *mind scripting*, que es la repetición consecutiva de una frase a si mismos muchas veces, y estar preparado para ponerse en ese papel. La experiencia es un buen aliado.

3) Mantenerlo simple

No hace falta justificar cosas cuando no se piden. No se debe dar la oportunidad de encontrar incoherencias en la historia narrada por hablar más de lo que se necesita. También puede ocurrir que al atacante se le olvide de alguna parte de lo que había pensado y tenga que improvisar, lo cual es más sencillo si parte de una base simple.

La segunda en Internet no tiene mucho sentido. El atacante se puede tomar el tiempo que quiera para elaborar la trama y ejecutarla en el momento adecuado. En cambio, la primera y la tercera son importantes porque definen un contexto e imponen unos límites. Igualmente, un atacante debe tener alternativas y no abusar de las mentiras, dado que cada una incrementa su posibilidad a ser descubierto.

3.3.3.2 SOCIAL ENGINEERING TOOLKIT

El Social Engineering Toolkit (S.E.T.), tal como se ve en la figura 3.10, es una herramienta creada por David Kennedy (ReL1K) para agilizar patrones de ataques relacionados con la I.S. En 2009 apareció la primera versión, pero ya ha logrado un reconocimiento por los profesionales de la seguridad informática. Se complementa con el *framework* Metasploit, otra herramienta para auditar servidores, el cual permite enviar *payloads*, es decir, permite enviar ciertas acciones especificadas por el programador.

Ilustración 3.10: SET: consola, versión 1.0

A continuación, una pequeña explicación de lo que soporta en su versión 1.0.

1. Spear-Phishing Attack Vectors

Se trata de un sistema avanzado de *phishing* con capacidad de ser personalizado a partir de plantillas, incrustar ficheros adjuntos, etc. Permite manipular la dirección del remitente si cuentas con un servidor de correos.

2. Website Attack Vectors

Es un conjunto de varias técnicas que atacan al navegador desde una web vulnerable:

- Infección a través de un *applet* de Java
- *Exploits* | *payloads*
- Recolección de credenciales por BeEF
- *Tabnapping*
- *Man Left In The Middle*

- *Webbugs*
- Clonación de páginas

3. Infectious Media Generator

Permite crear los ficheros necesarios para hacer los CD-ROMS, USB y DVD autoinfectables. Este método suele ya ser un poco obsoleto, debido a que únicamente afectaban a Windows, y a partir del Windows Vista, esta desactivada la opción por defecto.

4. Create a Payload and Listener

Si se instala el *framework* Metasploit en su ruta predeterminada, este es capaz de crear un ejecutable con las opciones que se le indican. Normalmente se usa ese ejecutable para enviarse a través de un correo o subirlo a una página web maliciosa.

5. Mass Mailer Attack

La misma opción que el `*spear phishing*´, pero sólo la parte de enviar los correos, sin plantillas ni adjuntos. Permite enviar en formato HTML o plano.

6. Teensy USB HID Attack Vector

S.E.T. ayuda a programar la placa Teensy, un circuito programable con forma visual similar a la de un *pendrive*. Este permite transformarse en un teclado, un ratón y/o una memoria interna, escribiendo código malicioso y ejecutándolo como si fuera el usuario quien lo escribiera. S.E.T. ofrece una suite de plantillas programables en los lenguajes PowerShell y WSCRIPT. También ofrece la posibilidad de uso con BeEF. Afecta a cualquier sistema operativo que permite conectar un periférico USB sin confirmación.

7. SMS Spoofing Attack Vector

Permite usar un servicio gratuito o de pago para enviar mensajes a móviles cambiando el emisor.

3.3.3.3 BROWSER EXPLOTATION FRAMEWORK (BEEF)

El Browser Exploitation Framework (BeEF) es una herramienta web usada por profesionales de la seguridad informática para extraer y analizar información del navegador que lo carga. Esto es realmente interesante si se plantea seleccionar un grupo concreto entre muchas víctimas.

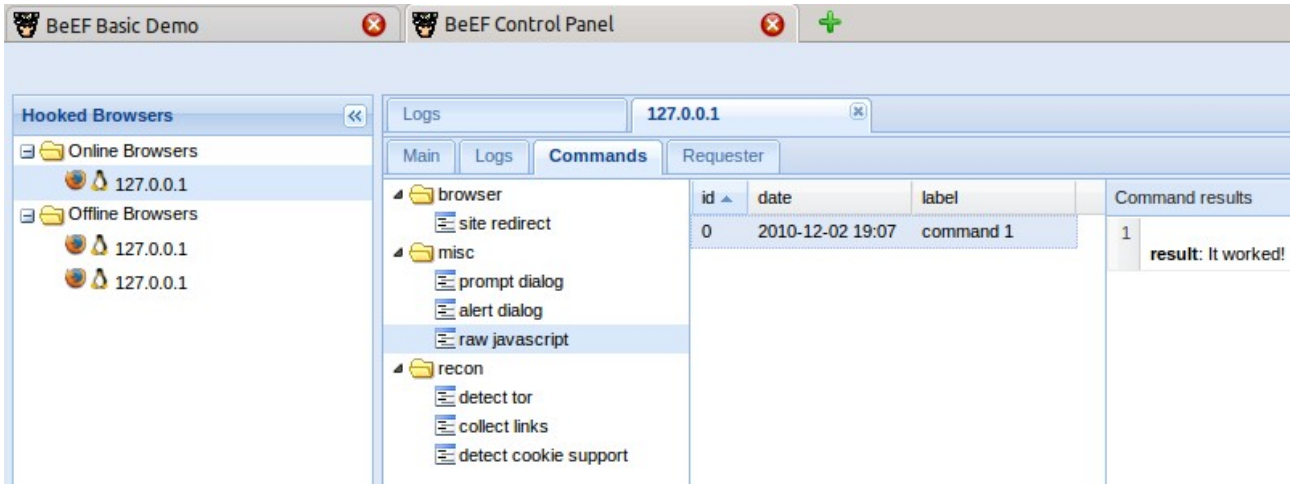

Ilustración 3.11: BeEF: zombie y panel de control

Dispone de las siguientes opciones:

- Configuración de módulos extra personalizados por el atacante
- Registro de las teclas pulsadas de las víctimas
- Escaneo de puertos de otras redes con varias víctimas a la vez
- Integración con Metasploit por XML-RPC
- Detección de la red de *proxies* Tor
- Detección de módulos del navegador

3.3.4 REFLEXIÓN

Primero se han tratado algunas historias relacionadas con las personas y la tecnología, llegando a ver un ejemplo real de ataque mediante correos. Después se han tratado diferentes técnicas, protocolos y herramientas, intentando mencionar cosas específicas y genéricas de la I.S. Finalmente se ha hecho hincapié en el factor social de la distracción, innovando el proceso del ataque.

En el apartado anterior ya se aconsejo al lector que utilizará el sentido común. Habiendo ampliado el número de casos con malas intenciones, el lector debe valorar hasta que punto debe fiarse de lo que le rodea. Hay que ir con cuidado con las promociones o gangas que se encuentran sin más, a las aplicaciones que piden demasiados permisos, a los sitios que dicen tener la solución definitiva, etc. Se mueven muchos millones a través del *e-crime*, y cuanto más afectado se vea el usuario, la tendencia será más alcista. Depende de todos hacer la red más segura, y por esto también hay que exigir *software* de mejor calidad, teniendo que invertir más las empresas que realizan las plataformas de distribución, como Facebook o Android, para tener unas garantías mínimas.

3.4 INTELIGENCIA ARTIFICIAL (I.A.)

"Artificial intelligence is no match for natural stupidity."
Anonymous

En todas las áreas la investigación es muy importante, pero un ataque de I.S. que quiera asegurar su éxito tiene que innovar constantemente nuevos escenarios para lograr sorprender a cualquier víctima. Por tal de informarse de aquellos conceptos que se salen de un vector lineal de ataque, se estudia el mundo de la I.A., el cual abre las puertas a crear ataques más complejos, más precisos y más complicados de detectar, al mismo tiempo que afectan a más individuos. Ello conlleva nuevos retos que superar, con nuevas dificultades y problemas, y son lo que puede ser el principio de un gran cambio, hacia una I.S. más agresiva y peligrosa.

3.4.1 HISTORIA: DE LA I.A. A LA I.S.A.

Se observan algunos de los fundamentos de la I.A., para después conocer alguna técnica presente entre profesionales de I.S.

3.4.1.1 EL TEST DE TURING Y LA SALA CHINA

En el año 1950, Alan Turing, a través de un artículo para la revista Mind [37], vuelve a formular la pregunta "*Can machines think?*" con una prueba de concepto conocida como Test de Turing. Esta consiste en, sin poder interactuar físicamente, averiguar si se está comunicando con un humano o una máquina.

En el año 1980, John Searle, publica un artículo [38] con el fin de rebatir la eficacia del Test de Turing. Él propone el experimento de la sala china, que consiste en agrupar una serie de elementos que usados en cadena devuelven el resultado esperado, pero individualmente no, alegando que sólo lo simula.

En la I.S., el máximo logro a alcanzar es simular el comportamiento y el patrón de acciones que una persona es capaz de realizar, todo de una manera transparente. Dado que el medio de

comunicación es escrito, se puede reducir el ámbito a sólo tener que clonar la capacidad de lectura y escritura de una persona real, que es lo único visible en un Test de Turing. Por lo tanto, si se es capaz de conseguir aquellos elementos usados en la sala china, se es capaz de hacer un sistema suficientemente bueno para pasarla. La dificultad reside en identificar tales elementos y conseguir tenerlos a disposición de una manera inmediata.

El ejemplo más popular de sala china a día de hoy son los chatbots, aplicaciones programadas a partir de reglas y con una base de conocimiento específica, los cuales devuelven una salida acorde a la entrada que reciben. Entre ellos, se destaca Eliza, programado en 1966 por Joseph Weizenbaum.

3.4.1.2 INGENIERÍA SOCIAL AUTOMATIZADA (I.S.A.)

Como se ha visto en apartados anteriores, el *phishing* se nutre de conceptos de I.S., pero no es propiamente un ataque de este tipo. Esto es porque no existe consciencia que haga que el ataque se moldee a las circunstancias e imprevistos. Una vez se envía un correo malicioso, aunque se trate del tipo `*spear phishing*´ con un diseño muy bien elaborado, no evoluciona dependiendo de la reacción del usuario.

En muchas ocasiones, por ejemplo, la que se produce con los *scams*, es una persona quien tiene que escribir y enviar los correo de contestación personalizados. Esta tarea es muy rudimentaria, y, igual que pasa con los estafadores telefónicos, *phreakers*, les requiere una cantidad de tiempo elevada.

En un intento de resolver ambos problemas, Markus Huber escribió su tesis [8] creando una nueva prueba de concepto, abreviada A.S.E., de `*Automated Social Engineering*´. Esta prueba de concepto es un chatbot que automáticamente aplica I.S. sobre las víctimas escogidas. Es un proceso más laborioso de programar que un *phishing*, pero mucho más maleable, con los beneficios de poderse hacer de forma totalmente automática.

El motor de chatbot que utiliza es ALICE, creado por Richard S. Wallace, publicado bajo licencia GNU/GPL. Markus añadió algunas modificaciones, introduciendo a la base de conocimiento las seis reglas de la influencia explicadas por Robert Cialdini [16]. Eligió Facebook como la red social para explotar, y finalmente hizo un Test de Turing con un grupo de veinte

personas, donde diez chatearían con una persona y las otras diez con el chatbot.

El resultado tuvo poco éxito. Todas las veinte personas descubrieron con quien se comunicaban, con bajos margenes de error. Las causas del fracaso fueron las siguientes:

- Los participantes conocían que estaban en un experimento
- Los participantes se comunicaron entre ellos quien era quien
- La persona podía contestar dos preguntas encadenadas, A.S.E. no
- La persona hacia errores gramaticales, A.S.E. no
- La persona sabia qué información tenia en su perfil, A.S.E. no

Las dos primeras no existirían en un caso real de I.S La tercera esta enlazada a la complejidad del motor del chatbot. Y las dos últimas, son las claves para disparar el éxito de un ataque de I.S., las cuales se trataran otra vez en el Capítulo 4.

3.4.2 ANÁLISIS: AUTOMATIZANDO EL APRENDIZAJE

El titulo del apartado hace referencia a la I.A., queriendo hacer énfasis a la precisión y velocidad superior que es capaz de tener una máquina en comparación con la Inteligencia humana. La dificultad reside en el conocimiento y las reglas que se pueden introducir en la máquina, por lo que siempre se acaba creando un autómata inteligente para una tarea muy específica y acotada. De no ser así, la afirmación de la superioridad no es tan clara.

Se ha seleccionado una herramienta para concienciar a los usuarios usando el ataque como método para mostrar los riesgos de uno real. Se utiliza la I.A. para complementar y potenciar la Inteligencia humana.

3.4.2.1 CONCIENCIANDO CON PHISHING AUTOMÁTICO

La educación de la seguridad informática es un tema muy complejo [39], por lo que hay que entender este apartado como una herramienta más y no como una solución. Resulta interesante

por la idea que hay detrás, pero requiere de una excelente ejecución en usabilidad y seguridad.

A partir de un documento [40] del centro de investigación `*CyLab Usable Privacy and Security*´ se creó la herramienta PhishGuru, utilizando el concepto `*Automated Phishing Education*´. Consiste en automatizar una herramienta para concienciar a los usuarios de que pasaría si hubiera sido real. El funcionamiento se basa en enviar *phishing* a los empleados internos de la empresa que haya contratado los servicios de este software, y si el usuario cae en la trampa, se le explica que ha pasado. Es un proceso que se basa en el castigo para concienciar a la gente, evitando la desmotivación que suele generar la directa explicación.

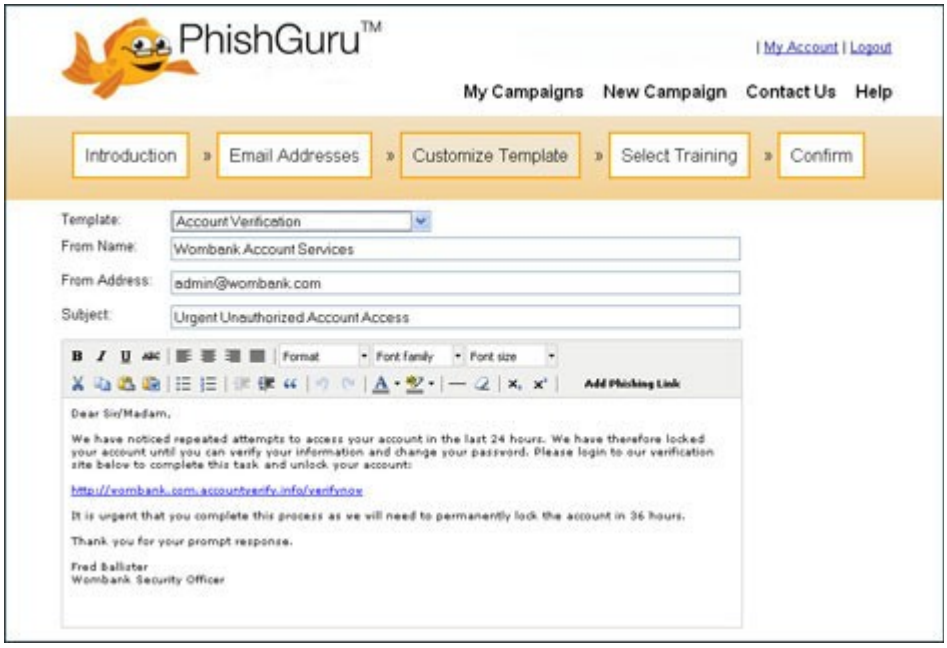

Ilustración 3.12: Diseño y proceso de PhishGuru [41]

La idea original e implementación del proyecto son buenas, pero se quedan realizando una plantilla estándar para todos los casos, como se ve en la figura 3.12, por lo que es mejorable. Se puede conseguir si en vez de dejar al administrador generar el patrón se utiliza la Inteligencia artificial. La herramienta mejorada puede detectar el *phishing* real, modificar su funcionamiento, y reenviarlo a los usuarios. La mejora de esta propuesta es obtener en todo momento una metodología de *phishing* real, dado que está tan actualizada como el que se envía por Internet.

La peligrosidad reside en asegurarse de que el propio *phishing* no comporte un problema de seguridad añadido a este servicio, pero a nivel económico supone menos tiempo invertido que tener que generar patrones manualmente, y una mayor eficacia por su semejanza a la realidad.

Igual que la propuesta de mejora en este campo, se puede extrapolar a cualquier otro, a nivel de educación o, visto desde el punto de vista del atacante, de sondeo. Económicamente, depende del grado de personalización que se busca.

3.4.3 EVOLUCIÓN: ¿QUÉ HAY DE NUEVO, VIEJO?

Disminuir el trabajo del atacante e incrementar la eficiencia de los ataques es posible gracias a la I.A., pero también existen otras maneras de conseguirlo.

Este apartado se va a dedicar a dos mejoras diferentes. En el primera se introduce a la jerga, enfocado a las palabras, y a la P.N.L, enfocado a la manera de comunicarse. En la segunda se estudia otra estrategia de la I.S.A. sobre el protocolo de IRC, el cual se vuelve a tratar con los casos de validación del Capítulo 4.

3.4.3.1 COMUNICACIÓN: JERGA Y P.N.L.

Cuando se realiza un ataque de I.S. presencialmente, uno de los primeros consejos es vestir la ropa adecuada. El objetivo es pasar desapercibido integrándose en el grupo como uno más. Hay que pensar en los complementos del oficio, como el maletín, la tarjeta de identificación y el uniforme de trabajo. La primera impresión es fundamental para complementar otros factores como el *rapport*.

Una vez se establece el contacto, para seguir pasando desapercibido, hay que controlar las palabras que se utilizan. Los estafadores telefónicos han desarrollado bastante su comunicación a través de las palabras, con tecnicismos y jerga específica del oficio, para elaborarse una mejor coartada. La mejor manera de conseguir este tipo de vocabulario es a través del prueba y error, contactando con gente relacionada del circulo social de la víctima, aunque también hay otras maneras peculiares, como rebuscar en la basura para conseguir plantillas o documentos oficiales de las que extraer las palabras clave.

El subconsciente recibe 11 millones de datos de información por segundo, pero sólo pasan de 16 a 40 clasificados como significantes para representar la realidad [6]. Esto significa que se

debe ser breve y conciso en las conversaciones, es decir, hay que crear frases lo más cortas posibles. También hay que dotar las frases de positivismo, haciéndolas en su forma afirmativa, evitando palabras como `no´, `nunca´ o `a veces´.

Las partes que una persona recuerda son el principio y el final de una conversación, así que el centro de la conversación es el mejor momento para tratar el tema más polémico, y terminar desviando la conversación. Sobre la entrada, esta no debe ser brusca, o se produce un rechazo por falta de confianza, creando un mal precedente. Hay que graduar las intenciones durante las conversaciones, por tal de suavizar el impacto y el recuerdo.

Existe una rama que estudia todo lo mencionado, llamada `programación neurolingüística´ (P.N.L.), aunque algunos `ingenieros sociales´ ya la denominan `hacking neurolingüístico´ [36]. Se basa en utilizar técnicas que resultan exitosas con un alto número de personas. Aunque no sea una ciencia exacta, en la seguridad informática es muy importante, dado que si el sistema es vulnerado, no importa de la manera en que lo haya sido, sino del daño que causa.

La P.N.L. se creó para ayudar ante traumas, fobias y malestares mentales, originariamente estudiada por psicólogos terapeutas. Justamente por su gran experiencia en campos del comportamiento humano, se está utilizando como conjunto de técnicas que permiten desviar, reconducir y dirigir cualquier conversación. A veces una ráfaga de palabras sin sentido aparente es suficiente para bloquear al individuo. Esto lo realizan con éxito individuos tan peculiares como los que se dedican al *pickpocking*, haciendo preguntas del estilo `¿hoy es el cumpleaños de tu madre?´ mientras son capaces de robar carteras.

Otra táctica que utilizan [6] es hablar al oído derecho de las víctimas, obedeciendo con más frecuencia que en el izquierdo. Esta recomendación, aplicada en Internet, seria sustituible por elegir el momento más adecuado en que la persona lea lo que se le ha escrito. Una persona cualquiera no suele tener el mismo carácter por la mañana que por la noche, cuando esta más cansada.

3.4.3.2 BOT IN THE MIDDLE

Se va a inspeccionar una herramienta que ya se mencionó anteriormente. Se trata de Honeybot, un proyecto dirigido por EURECOM [9], basado en el concepto de la I.S.A. En este caso, la sala

china es una persona, por lo que es imposible descubrir el Honeybot con el test de Turing.

Honeybot es un programa diseñado para conectar a dos personas en un canal de chat (IRC) y, con un saludo inicial y un conjunto de técnicas adicionales, prolongar la conversación que mantienen.

El bot es invisible para ambas, como un ataque `*Man In The Middle´*. A parte, tiene otra funcionalidad: la inserción y reemplazo automático de direcciones maliciosas dentro de esta conversación. Esta dotado de sofisticados elementos de influencia social, que a partir de las palabras adecuadas, convencen al usuario de lo que más interesa en el momento. También dispone una buena cantidad de filtros para detectar patrones de palabras comunes. Para protegerse intenta evadir bots de spam y usuarios con derechos de administración.

Para hacer las pruebas utilizaron tres canales de chat (*Dating 1, Dating 2* y *Chat*), tres tipos de enlaces (*IP Address, TinyURL* y *Myspace*), y tres vectores de ataque (*Keyword, Random* y *Replacement*). Se añade a continuación el dibujo de la tabla que realizan 3.13, producido el mejor resultado en el canal *Dating 1*, usando *TinyURL* como sistema de enlaces y reemplazando un enlace existente con un 87.5% de éxito.

Channel	Link Type	Keyword	Random	Replacement	Total
Dating 1	IP Address	146/289 = 50.5%	126/211 = 59.7%	7/12 = 58.3%	**279/512 = 54.5%**
	TinyURL	163/266 = 61.3%	127/197 = 64.5%	14/16 = 87.5%	**304/479 = 63.5%**
	Myspace	154/273 = 56.4%	124/174 = 71.3%	14/18 = 77.8%	**292/465 = 62.8%**
	Total	**463/828 = 55.9%**	**377/582 = 64.8%**	**35/46 = 76.1%**	**875/1456 = 60.1%**
Dating 2	IP Address	56/126 = 44.4%	29/76 = 38.2%	0/1 = 00.0%	**85/203 = 41.9%**
	TinyURL	82/139 = 59.0%	38/68 = 55.9%	0/0 = 00.0%	**120/207 = 58.0%**
	Myspace	70/121 = 57.9%	31/61 = 50.8%	1/2 = 50.0%	**102/184 = 55.4%**
	Total	**208/386 = 53.9%**	**98/205 = 47.8%**	**1/3 = 33.3%**	**307/594 = 51.7%**
Chat	IP Address	0/0 = 00.0%	17/86 = 19.8%	2/4 = 50.0%	**19/90 = 21.1%**
	TinyURL	0/0 = 00.0%	25/91 = 27.5%	0/1 = 00.0%	**25/92 = 27.2%**
	Myspace	0/0 = 00.0%	3/78 = 03.9%	0/1 = 00.0%	**3/79 = 03.8%**
	Total	**0/0 = 00.0%**	**45/255 = 17.7%**	**2/6 = 33.3%**	**47/261 = 18.0%**

Ilustración 3.13: Tabla de porcentajes de clicks en cada nivel [42]

La manera más sencilla de detectar que se está hablando a través del bot es preguntarle su nombre de usuario. En caso de ser diferente, se sabría que se está delante del bot, pero es muy fácil subsanar esto técnicamente, filtrando el nombre y reemplazándolo. De ser así, se tendría

que diseñar un programa que a partir de repeticiones y patrones únicos del bot, lo clasificará en una lista negra. Esto es algo realmente complejo, al menos en un sitio donde hay tan pocos parámetros. Si en vez del chat fuera otro sitio como, por ejemplo, el `*msn messenger´*, habría un perfil externo que poder consultar una foto de perfil, un nombre, y una frase personal, aumentando en 4 el número de elementos para verificar que esa persona es quien es. En otro ejemplo, el Facebook, el elemento se incrementa bastante más.

Pero a partir de esta idea, y simplemente cambiando el entorno del IRC, se puede sorprender a mucha gente con un concepto casi indetectable.

3.4.4 REFLEXIÓN

Primero se ha tratado la historia de la I.A., desde el test de Turing hasta la I.S.A. Después se ha tratado la educación automática con el programa PhishGuru. Finalmente se ha tratado la P.N.L. y el proyecto de `*bot in the middle´*, ampliando las técnicas a manejar en la I.S.

En este caso, el lector puede plantearse buscar herramientas como PhishGuru para adoptar o proponer en su ambiente de trabajo. Utilizar la I.A. para mejorar la Inteligencia Humana es un concepto a explotar mucho más, dada la carencia que existe en la seguridad de tantos sistemas. Proyectos como `*bot in the middle´* dejan en evidencia que un ataque tan complejo como ese es capaz de lograr unas cuotas de infección sorprendentes, por lo que la I.S. es, y tiene las cualidades para seguir siendo, una manera de ataque que va a ir en crecimiento. Al final, lo que esta sucediendo es que, mientras se aumentan las comunicaciones directamente entre usuarios por redes sociales, *chats*, correos, etc., aumenta paralelamente el fraude en esos medios, dado que se pasa a tener menos filtros tecnológicos intermedios, facilitando la tarea del atacante con la suplantación de identidades. A diferencia de las máquinas, las personas somos más difíciles de prever, por lo que si un atacante se hace pasar por otra persona, este no necesita imitarlo a la perfección, ya que como víctimas no sabemos por las palabras usadas si es, quien dice ser.

3.5 CAPTURA LA BANDERA (C.T.F.)

“A human being is not attaining his full heights until he is educated”
Horace Mann

El nombre `Captura La Bandera´ (C.T.F.) en el contexto de seguridad informática se utiliza para realizar campeonatos de hacking ético. En la Defcon 18, se realizó el primero orientado a la I.S., el cual ha aportado nuevas preguntas y respuestas. Este campeonato trato de atacar con I.S. a 25 empresas del Fortune500, todas ellas de carácter tecnológico, como Google, Facebook o Twitter.

La importancia de estudiar el C.T.F. reside en averiguar cuál es la mejor manera de protegerse de la I.S. Un sector apunta a la educación, pero otro la ve insuficiente. Desde este trabajo se afirma que se necesita algo más que educación, como, por ejemplo, herramientas bien diseñadas. Los problemas vienen cuando la educación es inadecuada, por eso, durante todo el apartado se ven diferentes enfoques de la protección complementando la educación.

Se aprovecha el contexto de esta competición para abordar la parte más defensiva, donde entra la educación, las normativas de seguridad y sus procedimientos. Se finaliza con una técnica de P.N.L. que se aprovecha de influenciar a la gente a través de estímulos.

3.5.1 HISTORIA Y CONTEXTO

Antes de comentar las aportaciones de la competición, hay ciertos elementos próximos interesantes de analizar. Como decía Mei Yao Ch'en en el libro `El arte de la guerra´ [42], “Lo que depende de mí puedo hacerlo; lo que depende del enemigo nunca está garantizado”.

Se hace hincapié en la legislación, las normativas que existen, una leve introducción a la ética y finalmente se exponen los resultados del concurso.

3.5.1.1 LEGISLACIÓN Y NORMATIVAS ACTUALES

La competición, realizada en Estados Unidos, fue supeditada por el F.B.I., para que no se produjera ninguna vulneración de privacidad o daño lógico a las empresas que participaban. La gente organizadora también lo preparo para no sacar datos sensibles. La intención no era provocar daño, sino avisar del posible daño, pero había que pasar por el procedimiento legal.

En España, extraído del O.S.S.T.M.M. Version 3 (página 186), hay que adoptar las siguientes regulaciones:

- Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal (LOPD): "La ley reconoce a toda persona el derecho a saber por qué, para qué y cómo van a ser tratados sus datos personales y a decidir acerca de su uso." [43]. Esta se aprueba con el Real Decreto 1720/2007.

- Ley 31/2002 de Servicios de la Sociedad de la Información y el Correo Electrónico (LSSICE): "se aplica al comercio electrónico y a otros servicios de Internet cuando sean parte de una actividad económica [..]. Y si además realiza contratos on-line deberá añadir la siguiente información con carácter previo al proceso de contratación [..]. Y si hacen publicidad por vía electrónica." [44]

- Real Decreto-Ley 14/1999 de Regulación de la Firma Electrónica: "fue aprobado con el objetivo de fomentar la rápida incorporación de las nuevas tecnologías de seguridad de las comunicaciones electrónicas en la actividad de las empresas, los ciudadanos y las Administraciones públicas" [45]

Aparte, hay una serie de estándares ISO/IEC a los cuales atenerse y poder seguir un protocolo definido y acotado, enfocados a sistemas de control de seguridad de la información. Cada país adopta las medidas como quiere y establece metodologías con nombres específicos para sus administraciones. En la siguiente figura 3.14 se hace una relación general de varios países, donde se presenta la metodología, si disponen de una herramienta oficial, las reguladores que engloba y su estado de madurez.

	País de Origen	Organización	Idiomas disponible	Herramienta	Conformidad Regulatoria	Modelo de Madurez ISS
CRAMM	UK	British CCTA (Central Communication and Telecommunication Agency	Inglés Holandés Checo	Si (CRAMM expert)	ISO/IEC 17799 GLBA HIPPA	No
OCTAVE	USA	Carnegie Mellon University (USA), CERT (Computer Emergency Response Team)	Inglés	Si	N/A	No
MEHARI	Francia	CLUSIF - Club de la Sécurité Informatique Français	Francés Inglés Alemán Italiano	Si (RISICARE)	ISO/IEC 27002 ISO/IEC 13335 Basilea II SOX	Indicadores de Madurez
MIGRA	Italia	N/A	Italiano Inglés	Si (MIGRA tool)	Series ISO/IEC 27000 Regulación de Privacidad en Italia	No
Au IT Security Handbook	Austria	Austrian federal chancellery	Alemán	No	ISO/IEC 13335 ISO/IEC 17799	No
A&K Analysis	Holanda	Dutch ministry of internal affairs	Holandés	No	ISO/IEC 17799 VIR (Dutch Government Information Security Act)	No
SP800-30	USA	National Institute for Standards and Technology NIST	Inglés	N/A	N/A	No
IT Baseline Protection Manual	Alemania	Federal Office for Information Security (BSI)	Alemán Inglés	Si (GSTOOL)	ISO/IEC 27001 ISO/IEC 17799 KonTraG (Ley Federal de Control y Transparencia del mercado) Basilea II TKG (Ley Federal de Telecomunicaciones) BDSG (Ley Federal de Protección de Datos)	Si (tres niveles)
MAGERIT II	España	Consejo Superior de Administración Electrónica	Castellano Frances Inglés Italiano	Si (EAR/PILAR)	ISO/IEC 27001 / 2005 ISO/IEC 15408 / 2005" ISO/IEC 17799 / 2005" ISO/IEC 13335 / 2004 LOPD 15/1999	No

Ilustración 3.14: Metodologías de análisis de riesgos [46]

Las certificaciones como la serie ISO/IEC 27000 son opcionales, dado su coste económico, pero aseguran un marco mínimo de seguridad, a través de un proceso asociado estandarizado. En la siguiente figura 3.15 hay hasta cuatro definiciones:

- Evaluación: Análisis mediante un proceso metodológico de la capacidad de un producto o sistema TIC para proteger la propiedad de seguridad de la información.
- Certificación: Determinación de una evaluación satisfactoria, con unos criterios preestablecidos.
- Valoración: Análisis de las características de un sistema para manejar información, con arreglo a una finalidad, un determinado nivel de seguridad y señalando unas condiciones de confidencialidad, integridad y disponibilidad.
- Acreditación: Resultado de la valoración para manejar información, señalando unas condiciones de confidencialidad, integridad y disponibilidad. .

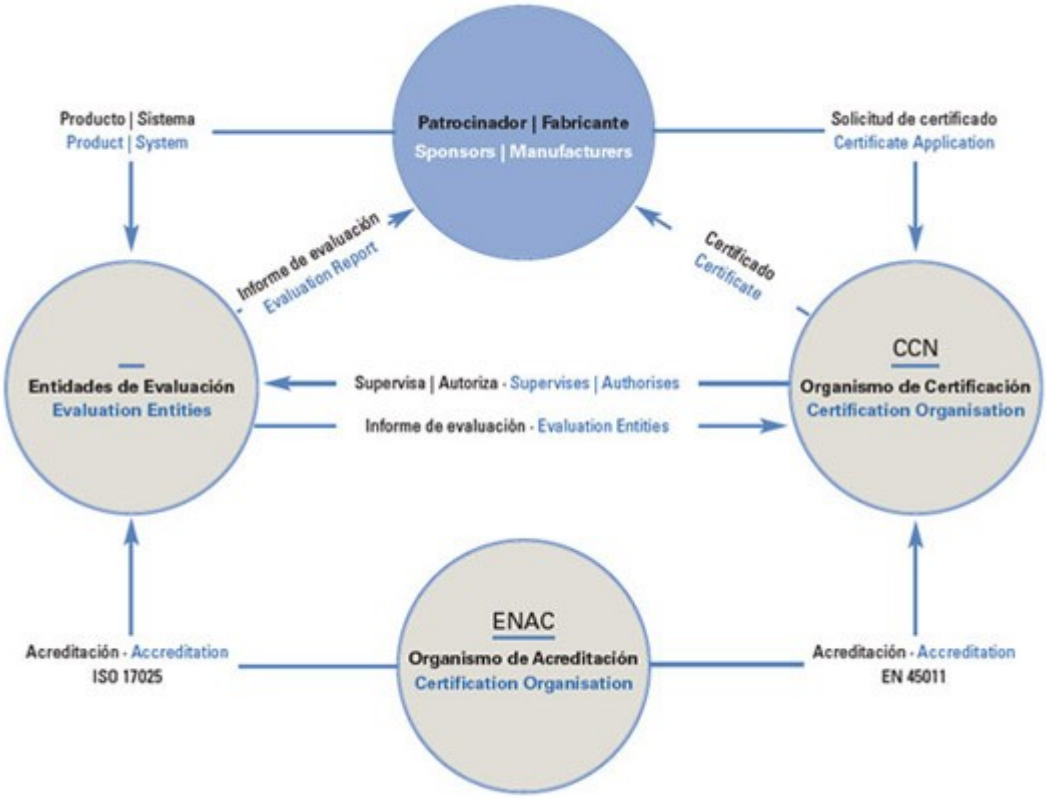

Ilustración 3.15: Esquema de certificación [47]

3.5.1.2 ÉTICA EN LA I.S.

Se ha visto que legalmente como investigador se está acotado, pero aparte de disponer de un marco bastante amplio para hacer cosas, es posible lanzar un ataque y esconderse. Dado que siempre hay un peligro, a día de hoy, la seguridad informática se refugia mucho sobre el área del análisis de riesgo, que más adelante se explica. Pero como última barrera, siempre queda la ética personal, la cual puede hacer que se reporte una vulnerabilidad en vez de aprovecharla.

El elemento más importante en la seguridad informática es el individuo que tiene un conocimiento suficiente para realizar el ataque. En caso de un auditor profesional, cualquier metodología, como la O.S.S.T.M.M., tiene una sección donde se menciona su uso ético. Un ejemplo es la prohibición de utilizar los conocimientos para aprovecharse en la negociación con un cliente, exagerando la gravedad del problema. En caso de que se trate de un buen samaritano, cualquier empresa debería agradecer que se le reporten sus problemas y no tratar este individuo como un delincuente, siempre que no haya utilizado la información para fines ilegales, animando a esta persona para que siga ayudando. Y por último, en caso de que se trate

de un individuo malintencionado, el motivo de su mal comportamiento puede deberse a demasiados factores. El factor más común es la ganancia económica, la cual debe ser penada.

El problema que tienen los profesionales del área es cómo realizar una prueba de concepto ética correctamente. Hay que tener claros los datos que se van a observar, como se recogen y cuales se acaban mostrando, pero además, hay que valorar hasta que punto se va a afectar a la persona poniéndola en una situación comprometida, aunque sólo sea de prueba. Es posible, que el sujeto usado para la prueba, amparado por la ley, pueda denunciar al investigador si este le causa algún perjuicio, aún sin intención. Es por esto, que se han llegado a plantear si se han de hacer experimentos en este campo, lo cual choca contra la realidad, dado que el delincuente no tiene tal barrera.

Aparte, en el dilema sobre la ética, influencia la edad de la persona que lo juzga. Cada una ve con diferentes ojos la tecnología, y cada uno le da un valor a unas cosas u otras, habitualmente confundiendo conceptos. Dicen [48] que falta un problema de cohesión, de mensaje único y uniforme, y finalmente, tener claros lo que se puede y no hacer, pero a su mismo tiempo, saber el porqué.

Un buen ejemplo de competición ética es la Captura La Bandera de este apartado, porque ayuda a identificar los agujeros en la seguridad humana o *wetware*, y hasta identificar a gente con potencial que realizan I.S. sin tenerlo presente. Todo ello siempre que se hagan de forma responsable, con buena intención y en un entorno controlado, como este caso.

3.5.1.3 RESULTADOS DEL CAPTURA LA BANDERA

La competición de Captura la bandera realizada en la 18a edición del evento DefCon, popular por traer a los mejores hackers del mundo, se basó en atacar un total de 15 empresas del Fortune500 a través de ataques de I.S. telefónicos. Las empresas participantes fueron BP, Shell, Apple, Google, Microsoft, Cisco Systems, Proctor and Gamble, Pepsi, Coca-Cola, Symantec, Phillip Morris, Walmart, Mcafee y Ford. El grupo responsable de organizarlo fue Social-Engineer.org, contando con participantes como Kevin Mitnick. La competición fue un éxito, con más de 130 llamadas y 14 de las 15 empresas perdieron algún tipo de dato supuestamente sensible. Sólo once empleados opusieron resistencia.

La competición perseguía dos objetivos. Uno era estudiar el tipo de pretexto que se utilizaba a la hora de atacar, y el otro era concienciar sobre la peligrosidad de la I.S. La consciencia se consigue solamente por conocer las estadísticas obtenidas del número de empresas vulnerables, pero estudiar el pretexto es un poco más complejo.

Antes de la competición, los concursantes tenían dos semanas para obtener toda la información que quisieran de la empresa que les había tocado. Tenían que elaborar un informe y entregarlo a la organización. Estos informes rebelaron que las fuentes consultadas más veces eran las redes sociales, con Google, LinkedIn y Facebook en la cabecera. Google View Street fue uno de los servicio más curiosos de los que usaron. Una vez esta empezó, tenían 25 minutos para usarlos en tantas llamadas como querían a su empresa objetivo. De estas llamadas, se extrajeron las siguientes conclusiones [49]:

- La mitad de llamadas iban directamente a un Call-Center. De la otra mitad, un cuarto, a empleados específicos.
- Los 3 perfiles más usados para suplantar eran el de empleado interno, entrevistador y de cliente.
- Más de la mitad de los que contestaban el teléfono no ponían resistencia alguna.
- Poca consciencia de la importancia de la información que se revela. No tenían claros hasta que punto los datos que daban eran información sensible o no.
- El 90% de los casos en los que se preguntó por abrir una dirección maliciosa se hizo. En un 50% de los casos en que se preguntó por otra información también se consiguió.
- La mitad de los empleados usaban sistemas Windows XP con el navegador Internet Explorer 6. Ningún caso reportado utilizaba GNU/Linux o Firefox.
- Hasta un 20% de respuestas exitosas cuando se preguntaba por el sueldo o otra información personal.
- El empleado utilizaba la frase “estoy ocupado” para evadir las preguntas.
- El hecho de ganarse a una persona de dentro con técnicas de *rapport* acababa reportando muchos de los objetivos a conseguir de forma más sencilla.
- Algunos participantes obtenían objetivos simplemente dejándose corregir.
- El número de teléfono desde donde se llamaba no era comprobado apenas por los empleados, dejando una herramienta de verificación sin utilizar.

3.5.2 ANÁLISIS: PREVER Y GESTIONAR

Cuando se dispone en un sitio información sensible, más vale prevenir que curar, dado que las multas por la ley L.O.P.D. u otras pueden ser muy fuertes. Pero la seguridad no es algo que se pueda medir ni proteger al 100%. Es cierto que cuanto más dinero se invierta en un sistema, estará más protegido a ataques y errores, pero el crecimiento de la inversión es exponencial. Por eso, existen los análisis de riesgos o las metodologías, que garantizan un mínimo de protección a un precio razonable a las circunstancias.

A continuación se trata una introducción a este campo, principalmente a los elementos que lo identifican, y luego se trata una metodología llamada O.S.S.T.M.M., abierta a todo el mundo.

3.5.2.1 ANÁLISIS DE RIESGOS

Ilustración 3.16: Esquema de relación entre elementos clave con MAGERIT [46]

Hay diferentes análisis de riesgos, pero se pueden explicar a partir del que se usa en España, de siglas MAGERIT (figura 3.16), que proviene de "Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información de las Administraciones Públicas". Se catalogan en dos etapas: evaluación de riesgo y gestión de riesgo.

En la etapa de la evaluación de riesgos, se empiezan listando todos los activos que tiene una empresa. Los activos son aquellos elementos tangibles de los que la empresa dispone, desde una mesa hasta un programa de contabilidad. Esta lista hay que valorarla, dándole un valor, que depende de lo importante que es ese elemento para la empresa. Por ejemplo, la mesa podría ser fácilmente reemplazable pero no el programa de contabilidad, por lo que hay que puntuar con mayor nota el segundo que el primero. Una vez se tiene la lista ponderada, se ponen sobre el programa MAGERIT, que tiene una base de datos con posibles amenazas, que son factores que pueden provocar pérdidas a la empresa, como que se incendien las oficinas o que se caiga el sistema. A partir de la frecuencia de las amenazas, la degradación que se produce en los activos y el impacto que estos suponen, se obtiene un riesgo, que es el que, de forma normalizada, muestra aquellos activos que se deben subsanar con más urgencia.

Se supone que en la lista de amenazas de MAGERIT se valoran los ataques de I.S., pero esto no es cierto. Los activos de los análisis de riesgos deben ponderarse manualmente, por lo que es una persona contratada quien debe poner una valoración subjetiva del interés para esa empresa. Por ejemplo, si la empresa tiene su mercado en Internet, la puntuación que puede recibir perder la conexión será muy alta, pero es muy difícil valorar si esa empresa puede ser objeto de un ataque de I.S., dado que todas las empresas lo pueden ser. Es por esto que los auditores no tienen una consciencia de la amenaza y posterior riesgo que puede producir un ataque I.S., anteponiendo lo más evidente al resto de cosas. Por lo tanto, si en el análisis MAGERIT se contemplan elementos de seguridad humana, *wetware*, pero luego no se da la nota correcta, sólo se obtiene una falsa sensación de seguridad.

En la etapa de gestión de riesgos, valorando la ponderación normalizada obtenida, contando la frecuencia y, sobretodo el impacto, se ponen las salvaguardas, encargadas de mitigar el efecto de las amenazas. Esta protección también desprende un riesgo residual, el cual nunca se va a poder subsanar a un coste razonable. Por ejemplo, guardar las copias de seguridad externamente protege a posibles asaltos al centro donde se opera, pero lleva a un problema secundario si se ataca al centro donde éstas estén.

3.5.2.2 METODOLOGÍA O.S.S.T.M.M.

Ilustración 3.17: Emblema [50]

O.S.S.T.M.M. significa `*Open Source Security Testing Methodology Manual*´, una metodología abierta para conocer la seguridad de cualquier empresa o institución. Es gratuito bajarse el documento y aplicarlo, pero hay que pagar por el sello oficial. Especialmente la O.S.S.T.M.M. se enfoca a los detalles técnicos de los elementos que requieren ser probados, que hacer antes, durante y después de la prueba de seguridad, y como medir los resultados.

Este trabajo se empezó con la versión 2.2, de diciembre del 2006, pero a finales del 2010 salio la 3.0, por lo que se tratan ambas. Entre ellas son incompatibles, dado que esta última esta totalmente renovada. Ya están trabajando en la 4.0.

Los fundamentos que persigue la O.S.S.T.M.M. son:

- Qué testear. Localizar los objetivos adecuados.
- Cómo testearlos. Localizar la entrada y salida de información.
- Identificar los diferentes tipos de control existentes.
- Qué se queda sin testear.

Una vez se han examinado los 4 puntos anteriores, se elabora el llamado `*Security Test Audit Report*´ o STAR, que documenta y permite comparar diferentes redes con valores normalizados, como se ve en la figura 3.18.

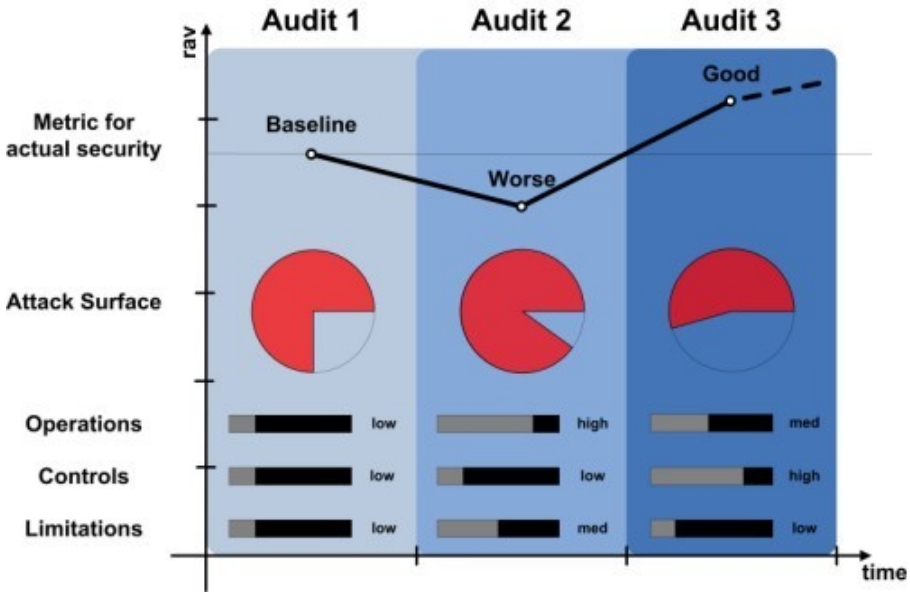

Ilustración 3.18: Ejemplo de comparación de STARs [50]

Referente a las dos versiones comentadas, la 2.2 engloba en cinco canales los seis apartados siguientes: seguridad de la información, seguridad del proceso, seguridad en las tecnologías de Internet, seguridad de las comunicaciones, seguridad de las redes inalámbricas y seguridad física. En cambio la 3.0, aún englobando los cinco canales, prefiere cambiar el concepto, y enfocarse a la susceptibilidad al ataque, es decir, a lo expuesto que se está a este. Parece un progreso, pero se puede ver como volver a la metodología llamada "cáscara de huevo", donde la seguridad exterior es fuerte, pero internamente no.

Hay dos puntos a destacar que se proponen como carencias en esta metodología. El primero, es la *`Human-Computer Interaction Security´*, tratada en el Capítulo 5, que no se menciona en ninguna parte del documento. No hay un enfoque para el binomio seguridad – usabilidad, quedándose como una mera guía de pautas a usar, como si fueran capa sobre capa. La segunda, no se analizan tampoco los malos usos del sistema. No se analiza por lo que es, sino por lo que podría ser, y esto comportar un sesgo de la información real. En ocasiones irá bien, pero no siempre.

Por ello, existen otras metodologías, como la propuesta por la fundación *`Open Web Application Security Project´*, destinada a páginas web.

3.5.3 EVOLUCIÓN: ESTIMULANDO LA MENTE

El paso adelante hay que darlo en la dirección de la educación, y como se han tratado ya maneras tecnológicas para ayudar a desarrollarla, se va a exponer un factor humano. Este factor psico-social recibe el nombre de *anchoring*, traducido a `anchaje´, y explota la relación indirecta entre conceptos en nuestra mente.

3.5.3.1 ANCHORING

El *anchoring* es una técnica estudiada por la rama de la P.N.L. Esta técnica determina que es posible enlazar conceptos a partir de insertar un disparador en la mente, que al notar un estímulo se dispare haciendo que se recuerde una sensación, un recuerdo o un estado mental. No hay que confundirlo con el *anchoring* psicológico, que es una tendencia a confiar demasiado en algo a la hora de tomar decisiones.

Son tres tipos de *anchoring* los que existen: visual, auditivo y kinestésico [51], los cuales tienen en cuenta los cuatro factores siguientes: la intensidad, el tiempo, las repeticiones y ser único. La técnica se divide en dos etapas, establecer el llamado *anchor* y activarlo. En un entorno de I.S. lo que se busca es encontrar un *anchor* ya establecido por otra persona y aprovecharlo. Este *anchor*, en Internet, se basa en algo visual. Se puede traducir a aprovecharse de las webs populares, de diseños originales, etc. Suplantar el *anchor* significa calcar la estructura, colores y estilos de la página web, pero teniendo en cuenta que tal web es la que produce un *anchor* a la víctima objetivo, porque la visita mucho, porque le gustan esos colores, etc.

Estudios de neurobiología [52] avalan por qué se produce el *anchoring*, lo cual respalda su uso en un ataque. El estudio se basa en las relaciones amorosas y lo difícil que es olvidarse de ellas. Se explica que hay una parte del cerebro que se encarga de recordar aquellas vivencias más intensas que se han tenido durante un largo periodo de tiempo, lo cual hace que olvidarlas también requiera de más tiempo.

Un ejemplo se produce al estar demasiado tiempo concentrado en un mismo proyecto, acaba no rindiendo suficiente. Es necesario desconectar porque involuntariamente se está produciendo el *anchoring*.

3.5.4 REFLEXIÓN

Primero se ha tratado la legislación y las normas en la seguridad informática, la ética a seguir y la competición C.T.F., destacando aquellos puntos que se vieron más relevantes de la I.S. Después se trataron los análisis de riesgo y la O.S.S.T.M.M. como un ejemplo de metodología. Finalmente se trato el *anchoring* como otro factor psico-social efectivo en Internet.

Este caso analiza el lado defensivo. Se han creado normas, leyes, metodologías, análisis, competiciones, y un largo etc. para combatir el cibercrimen, pero todavía existe. Puede que no sea esta la manera de solucionar las cosas, es decir, poner solución a través de parches, sino que haya que enfocar los esfuerzos a apoyar directamente la creación de nuevo software y plataformas mucho más organizadas, estructuradas y al mismo tiempo seguras. Aún así, estará siempre el factor humano, el cual es el que hace que toda protección tecnológica no valga absolutamente nada. Y no sólo existe como administrador del sistema, sino también como desarrollador.

Al lector ya se le ha insistido de que exija un *software* más competente, pero al mismo tiempo, debe saber que hace en cada momento. La mala configuración del *software* y la inocencia del usuario son los vectores de ataque del futuro, donde las plataformas serán mucho más conscientes del término seguridad.

4 VALIDACIÓN

Hasta ahora, se ha visto lo que es la I.S., algunos ejemplos reales y varias de sus proyecciones, pero sería interesante complementar tal dispersión de datos con algo más concreto. Para ello se concentran los esfuerzos en dos puntos a la hora de formalizar tanto la viabilidad como la metodología de tal técnica.

El primero de ellos trata de mostrar un ejemplo real y completo de I.S., aplicada a la red social Facebook, explicando las diferentes etapas y los factores psico-sociales que hay involucrados en cada etapa. El objetivo es demostrar que en un entorno fuertemente protegido aún es posible utilizarlo para fines inadecuados, y en este caso, se hará a partir de la confianza y la ingenuidad del usuario.

El segundo abarca un estudio sobre la realización de una recopilación con datos sociales, es decir, dado que a menudo, para poder llevar a cabo un ataque de I.S., es necesario hacerse la pregunta sobre como actuaría la gente referente a una acción, hay que conocer como obtener la respuesta de la forma adecuada. Para ello se ha elaborado un escenario que conduce a una pregunta específica, y se ha llevado a cabo un cuestionario para identificar la respuesta de la gente en general, a modo de ejemplo real.

Lamentablemente, estos dos puntos no abarcan todo lo que la I.S. puede ofrecer, pero aportan unos conocimientos muy decisivos, los cuales ya por si mismos permitirían crear un ataque de características devastadoras. Aprenderlos aportará una mejor visión para ampliar el tema en un futuro.

4.1 INGENIERÍA SOCIAL VERSUS TECNOLOGÍA

Una vez encontrada una vulnerabilidad técnica, es posible comprometer el sistema, pero hay entornos que han invertido mucho en su seguridad. En una situación tan supervisada, es fácil detectar el ataque y denunciarlo a las autoridades. En cambio, en un ataque de I.S., el responsable de hacer una acción peligrosa es el mismo empleado. Todas las medidas y toda la supervisión desaparecen ya que no contemplan esa capa de ataque.

Para verificar que se cumple se propone un ejemplo en Facebook, persuadiendo al usuario hasta una web externa y allí se intenta que desvele su contraseña.

4.1.1 PROPÓSITO

Para dar constancia de como un ataque de I.S. en Internet es posible, se propone una prueba de concepto. No se ha realizado para no poner en riesgo la privacidad y seguridad del usuario final, pero, en otras ocasiones, todos los conceptos presentes han sido probados satisfactoriamente, por lo que la suma de todos ellos implica una mejora del ataque.

Lo importante es transmitir al lector la facilidad que puede ser verse manipulado, perdiendo sus datos, sin realizar ninguna acción sospechosa o diferente a lo “normal”. Tanto las etapas del proceso como los factores psico-sociales podrían haber sido otros totalmente diferentes, pero se eligieron los expuestos debido a su sencillez y su gran aparición en casos de I.S.

Se espera que el lector se conciencie y reflexione sobre que datos introduce y donde, dándoles el valor que considere adecuado.

4.1.2 ENTORNO

El ejemplo se desarrolla en la red social Facebook. Esta red ha sido escogida por su tamaño y su experiencia como red social. En el año 2011 tiene más de 500 millones de usuarios activos, el 50% se loguea a diario y cada usuario tiene alrededor de 130 amigos por media. Se consumen 700 mil millones de minutos por mes en Facebook [53].

Facebook se preocupa de la seguridad de sus millones de usuarios, realizando auditorias y premiando a aquellos que le reportan vulnerabilidades. Al final es una plataforma de la cual es muy difícil encontrar una vulnerabilidad técnica, por lo que inmediatamente se convierte en un buen entorno para jugar con la I.S.

Sus términos de servicio prohíben expresamente realizar cualquier tipo de ataque a sus sistemas, pero también prohíbe estrictamente apoderarse de información de los usuarios, almacenándola más de X días, dependiendo del tipo de información. Es otro motivo más por el cual no intentar la intrusión. Lamentablemente para los usuarios y para Facebook, que un usuario realice una aplicación, se guarde los datos y no los borre posteriormente, es imposible de conocer.

De esta demostración se puede extraer que si es factible en Facebook, cualquier otra red social que comparta un sistema de aplicaciones externas, también es vulnerable.

4.1.3 PROCESO

Desde el punto de vista de la víctima, el ataque se define como un proceso natural, sin intención de ser percibido como una amenaza. Se resume en las siguientes etapas:

1) La víctima, poseedora de una cuenta de Facebook, toma contacto con una aplicación que le incita a buscar otras aplicaciones, a modo de juego, con premio final incluido.
2) La víctima, al encontrar todas las aplicaciones, es redirigido a una página donde se le piden sus datos personales para enviarle el regalo.

Este proceso es dirigido por una cadena de factores psico-sociales para motivar constantemente a la víctima a finalizarlo.

Desde el punto de vista del atacante, montar el sistema es un poco más complejo. No sólo requiere conocimientos técnicos avanzados del sistema de Facebook, sino también requiere remarcar los factores psico-sociales para su fácil asimilación por el usuario. Dependiendo de los objetivos que se persiguen, hay diferentes maneras de organizar y montar los factores, cada una con resultados diferentes. Este ejemplo sólo muestra una posible combinación, utilizando tres

factores tratados ya anteriormente: *misdirection*, *rapport* y *anchoring*.

A continuación, a partir de los tres factores se explica el ataque. En cada apartado, se concreta de manera más precisa las acciones a tomar y las consecuencias que estas pueden causar, formando en su conjunto todo el proceso malicioso que englobaría un ataque de I.S.

4.1.3.1 MISDIRECTION

El inicio del ataque parte de la ventaja de conocer el terreno. Es imprescindible saber qué herramientas ofrece Facebook y cómo funcionan para después encontrar otros usos. En este caso se aprovecha que se permitan añadir aplicaciones externas, acercando la víctima a la trampa. Que se permitan estas aplicaciones es algo ya natural y los usuarios lo utilizan con fluidez, cuando en realidad se genera un serio problema de privacidad. El desconocimiento y la despreocupación se utilizan como medio para obtener el objetivo en este caso, porque dado que mucha gente lo usa, suponen que alguien ya se habrá preocupado de revisar que esa aplicación o enlace externo no será malicioso.

Teniendo en cuenta los conceptos anteriores, se diseña la primera etapa del ataque. En esta se generan varias aplicaciones de Facebook. El número a crear se debe fijar acorde al número de datos sensibles que se quieran robar al usuario, pero teniendo en cuenta también que se busca ser ágil. Por cada aplicación, se deben pedir permisos de uno o dos datos sensibles de la víctima, evitando las sospechas de haberlos pedido todos juntos. Un número razonable de aplicaciones son tres o cuatro. Cada aplicación debe cumplir dos características para poder considerar que se está realizando un ataque a través de *misdirection*.

1) Debe consistir en algo entretenido. Se puede elaborar un juego de inteligencia, un juego en Flash, un sistema de encuesta, un *puzzle* mental, etc. Cuanto más relacionado este con otras aplicaciones de Facebook que ya existan, menos sospechoso resulta. En algún momento hay que mostrar que los permisos que se han pedido son explotados, manteniendo el sistema sea coherente e incrementando el *rapport* con la víctima.
2) Debe convencer a la víctima para que encuentre las demás aplicaciones. Se debe guiar a la víctima para que siga el proceso, facilitándole el camino, con pistas sencillas y directas, ya que el usuario tiene ganas de disfrutar sin pensar. El mejor señuelo es obsequiarle con un premio al completar el proceso. Es importante

relacionar estrechamente el premio con lo que se está haciendo. Por ejemplo, si se está ofreciendo un juego que tiene cosas bonitas y tiernas, como peluches, chucherías, etc. se puede regalar un llavero con un osito que lleve un corazón con un "te quiero" escrito. Añadir la marca de Facebook en el regalo también ayuda con el factor de *anchoring*.

4.1.3.2 RAPPORT

Mientras el usuario va completando las aplicaciones y se recoge información, se debe formalizar y mantener el contacto para crear *rapport*. La propuesta es utilizar el correo electrónico, una vía muy común a día de hoy y que todo internauta tiene. Por lo tanto, cada vez que la víctima complete una actividad, se le envía un correo electrónico animándole en la búsqueda de más aplicaciones, valorando su ímpetu, dándole pistas y aconsejando que se dé prisa, que quedan pocas existencias del premio mencionado.

Es importante que para toda la exaltación de calificativos se usen las palabras adecuadas. El vocabulario debe ser bastante simple e informal, contemplando hacer el proceso multilingüe. No sólo se pueden obtener más víctimas, sino que se da la sensación de complejidad, factor que también favorece la sensación de seguridad.

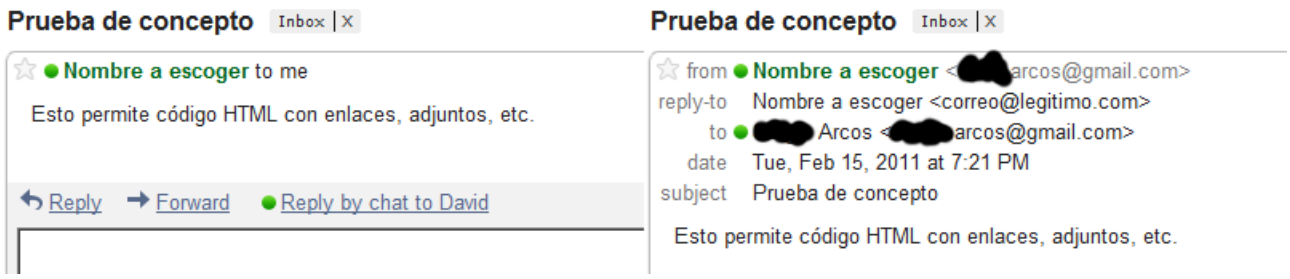

Ilustración 4.1: Prueba de concepto sobre Gmail. A la izquierda, sin detalles. A la derecha, con detalles. El correo se recibió en la bandeja de entrada sin advertencias.

Que el ataque sea un éxito depende en un 70% del *rapport*. Mientras se ha elaborado este trabajo se han descubierto vulnerabilidades en Facebook, Gmail y Android buscando problemas en la confianza de los componentes internos [54][55][56]. Facebook permitía saltarse un filtro de control si se compartía una nota que tenía incrustada una imagen, dado que el filtro no contemplaba que esa imagen no estuviera saneada. Gmail permitía enviar un correo saltando el

filtro anti-spam si se suplantaba como el emisor una cuenta de un amigo del emisario. Ejemplo en la figura 4.1. Android permitía infectar aplicaciones ya existentes si en vez de instalar la aplicación se movía directamente al directorio de aplicaciones instaladas.

4.1.3.3 ANCHORING

Hasta el momento sólo se ha tenido a la víctima jugando entre aplicaciones. El último paso es mover a la víctima fuera de Facebook, llevándolo a un web personalizada. Se le debe informar tanto mediante la aplicación como desde un correo de los que se le han estado enviando. La víctima debe rellenar sus datos para recibir el premio en casa.

Lo primero que hay que determinar es lo que se le quiere robar. Este ejemplo se plantea para robar contraseñas, ya que sus datos personales ya se han ido robando. La solución ideal seria crear plantillas dependiendo del dominio de la cuenta que utiliza el usuario objetivo. Si esta persona tiene un correo "@gmail.com", lo óptimo sería mostrar el diseño parecido a la página de registro de Gmail. A partir del efecto *anchoring* el usuario tenderá a repetir la contraseña.

Se puede perfeccionar el método teniendo en cuenta dos cosas: imitar las restricciones que imponen los otros gestores de correos, para que a la víctima se le vayan las ganas de pensar una buena contraseña, y complementar con un error que falle a propósito una o dos veces si la contraseña que introduce no es la de ese correo. Hay que ir con cuidado de no agobiar al usuario.

Lo importante en este paso es, que una vez que la víctima envía sus datos, sean guardados mientras se le garantiza que se cuida su privacidad. En ningún momento la víctima debe dudar que se está recopilando su información en una base de datos. De ser así, tampoco podría hacer nada, pero siempre es mejor evitarlo. Esto mismo sucede en las aplicaciones de Facebook.

Hay que finalizar el proceso con un correo a la víctima unos días después, dándole las gracias, informando que en breve se volverán a ofrecer más premios, pero que lamentablemente no ha sido uno de los afortunados. Una buena excusa es decirle que otros fueron más rápidos que él. Nunca hay que quemar al usuario y siempre ser educado, sin necesidad de destapar la trampa.

4.1.4 CONCLUSIONES

Se ha propuesto un ataque de I.S. en la plataforma de Facebook, dividiéndolo en tres etapas correspondiendo a tres factores psico-sociales.

Aunque no se haya realizado la prueba de concepto, existen ejemplos “parecidos” que sí se han llevado a cabo [57]. Se mencionan cifras de 165.000 usuarios infectados, es decir, usuarios a los que les han podido copiar ciertos datos. Resulta que tal ataque sólo utiliza una combinación de *misdirection,* para incrementar la propagación de la infección, y una falsa pero seductora promesa, que es saber quien mira su perfil, aunque en la realidad es técnicamente imposible. Este ataque se ha publicado en un medio por su mediocridad, pero los ataques de I.S. bien realizados no salen a la luz.

El método expuesto en este trabajo no intenta ser algo cutre, sino que pretende al menos mostrar y denunciar que las grandes plataformas están primando la cantidad de usuarios presentes a costa de la privacidad de estos, a partir de no invertir suficiente en el proceso de ingeniería de software. A raíz de esta tendencia a primar beneficios, se están inculcando malos hábitos a los usuarios de Internet que luego costará mucho más corregir. Este tipo de ejemplos deberían de ayudar a dejar de culpar al usuario y replantearse el problema de fondo.

4.2 ESTIMANDO TENDENCIAS CON ESTADÍSTICAS

Al planificar un ataque I.S., el atacante ha de asegurarse que el comportamiento de las personas será el previsto. A veces es suficiente con intuiciones, pero si se quiere una mejor aproximación, es conveniente hacer una estimación antes de llevarlo a la práctica. En este apartado se va a realizar un ejemplo de cómo estimar la tendencia de la gente.

Primero se define el objetivo de la prueba de concepto. A continuación se explica como surgió la pregunta analizada posteriormente. Y después ya se trata el proceso completo para llegar a obtener los resultados, con los factores más relevantes.

4.2.1 PROPÓSITO

El problema principal de esta estimación es el procedimiento. No es lo mismo poder obtener datos objetivos, por ejemplo, meteorológicos, que datos donde interfieren aspectos humanos. Esto se debe a que la cantidad de variables que influyen son muchas y los pesos varían para cada persona y momento. Las modas de ropa son un ejemplo de que las personas no siempre piensan igual.

Dado que la estimación realizada en este ejemplo es válida científicamente, se detalla el proceso para poder volverse a reproducir y comparar los resultados con otra muestra en la posterioridad. Los recursos que se han podido dedicar han sido escasos, pero el resultado que se obtiene es suficientemente fiable para saber la orientación final. Por eso, lo importante no será la cantidad de datos recogidos, sino enfocarla en la dirección correcta.

4.2.2 ENTORNO

La pregunta a estimar en esta prueba de concepto esta basada en dos conceptos que ya se vieron en el Capítulo 3.

El primero es Honeybot [9], el bot que intercepta comunicaciones situándose entre dos usuarios en IRC. Se extrae esa idea de suplantar cuentas, intentando avanzar automáticamente

sin interacción. El segundo concepto es la red social Facebook. Juntando ambos se tiene la posibilidad de crear un chatbot que suplante cuentas de Facebook, poniendo en contacto varios usuarios sin crear sospechas. Hay detalles a nivel técnico que se han de tener en cuenta para implementarlo, pero gracias a las botnets sería posible realizarse.

Este nuevo chatbot plantea la misma pregunta que se hizo sobre Honeybot de cara a la víctima/defensor: "¿Cómo es posible averiguar que se está chateando con una persona real y no con una máquina?" La respuesta pasa por localizar algún elemento que no sea coherente. Dado que el IRC tiene pocos elementos, las opciones son reducidas, e incluso aún más si el chatbot lo tiene en cuenta. En Facebook hay muchos más elementos que componen la información del usuario, por lo que hay que priorizarlos.

El atacante con esta respuesta será capaz de optimizar el chatbot y el defensor será capaz de concienciarse para no realizar preguntas convencionales, por lo que ambos tienen algo que aprender.

Para realizar esta prueba de concepto se requieren algunos conocimientos de PHP y MySQL para crear la encuesta, conocimientos en medios de difusión de información para promocionar la encuesta y conocimientos de estadística y minería de datos (lenguaje/programa R) para plasmar las gráficas con los resultados.

4.2.3 PROCESO

Se divide el proceso en tres grupos, cada uno centrado en una función. En cada uno se explica primero desde un punto de vista neutral, y después desde el enfoque de la pregunta.

4.2.3.1 PUNTOS CLAVE

Para sacar el máximo provecho a un experimento como el propuesto, hay que guiarse por al menos las siguientes tres indicaciones en ese orden propuesto.

Focalizar la pregunta

Hay que delimitar lo que se busca y formular la pregunta en un lenguaje que todo el mundo comprenda. Hay muchos tipos de usuarios e interesa minimizar el error en la comprensión. Como para llegar a contestar una misma pregunta siempre hay diferentes caminos viables, lo mejor es escoger el más sencillo de todos, como dice la metodología KISS (*Keep It Simple, Stupid*).

En la prueba realizada, aunque la pregunta original era "¿Cómo es posible averiguar que se está chateando con una persona real y no con una máquina?", al usuario no se le podía trasladar directamente. Lo que se ha hecho es modificar la pregunta a "¿De donde sacarías la información para preguntarle sobre él?" donde él es el perfil dudoso. La pregunta y el contexto que se explican son más sencillos y resuelven el mismo problema.

Prever la respuesta

Lo primero es hacer un esfuerzo para acotar el rango de respuestas posibles. Si se realiza una encuesta donde se preguntan cosas distantes, el resultado no será aplicable, por eso, hay que ajustar las respuestas posibles con la realidad social y con la posible implementación de ello. También hay que contemplar la afinidad que se quiere conseguir en la respuesta, es decir, el número de muestras suficientes para obtener una suficiente aproximación.

En la prueba realizada se ha recogido una muestra de 67 individuos respecto a una población de 500 millones. Al haber tan bajo número, por no disponer de suficientes recursos, se ha puesto énfasis a otros condicionantes. Uno ha sido seleccionar los elementos más relevantes de Facebook, descartando el resto, así que se han seleccionado 12, divididos en sietes elementos de miniatura y cinco páginas. También se ha buscado gente de diferentes partes del mundo, intentando obtener unos datos no demasiado influenciados por la sociología.

Formalizar el escenario de pruebas

Cuando se diseña el proceso para recoger los datos hay que garantizar que no se ve demasiado interferido por usuarios malintencionados, por lo que hay que tomar ciertas medidas de protección, las cuales mantengan el sesgo lo más bajo posible. Es así, dado que se está en Internet, donde es fácil enviar varias peticiones o otros ataques que pueden desajustar los datos. También hay que escoger la técnica adecuada dependiendo de la pregunta y la respuesta, para finalmente implementarlo de la forma más sencilla posible, guiando al usuario.

En la prueba realizada se garantizó tecnológicamente que fueran imposible dos cosas. La

primera, recibir más de una petición desde un mismo ordenador. Eso significa que si alguien quería falsear la información, sólo podría manipular una sola muestra, no el conjunto. Y la segunda, recibir datos de entradas incorrectos o incompletos. Dada la sencillez de la prueba, era fácil conseguir esto. A nivel de diseño, se hizo una primera página que explicaba y contextualizaba a la persona, y al pulsar aceptar empezaba la encuesta, dirigiéndola a una imagen 4.2, donde se tenían que ir seleccionando los diferentes apartados. Con un lenguaje dinámico, javascript, se iban inutilizando las áreas que ya había marcado y se iban mostrando en el lateral, hasta haber seleccionado todas las opciones en orden. En cualquier momento podía repetir la prueba.

Ilustración 4.2: Encuesta – Distribución/Numeración de Facebook

4.2.3.2 ESTRATEGIA SOCIAL

Una vez se ha preparado la encuesta se debe distribuir para poder obtener resultados. Se han empleado varios canales para llegar a los usuarios. Entre ellos se han potenciado: Facebook, Twitter y otras dos comunidades, A y B. En Facebook y Twitter se ha logrado dispersión entre

amigos de amigos, mientras que la comunidad A y B fue destinada a un público más aleatorio.

Se ha observado la evolución durante los 20-30 días que ha estado publicada la encuesta, y, a partir de esa observación, se extrajeron las siguientes afirmaciones:

- La comunidad B no mostró interés por la encuesta. Es difícil convencer de hacer encuestas a gente que no tiene interés por ello y no esta adaptada para ese sector concreto, así que el resultado fue bajo.
- La comunidad A participó, pero fueron los que menos empeño pusieron a la encuesta. Algunos intentaron manipular los datos sin éxito.
- Twitter fue muy efectivo para expandirlo entre amigos de amigos. La gente se la tomaba enserio, igual que paso con Facebook, dada la confianza que ya tenían con la cuenta con la que se expandía.
- Facebook no recibió apenás expansión entre amigos, pero es donde la gente llegó a participar más seriamente.

La encuesta acabó realizándose sobre 67 personas, suficientes para obtener la orientación hacia una tendencia generalizada. De otra manera la encuesta debería haber estado mejor elaborada, ofreciendo información al mismo punto que captaba, usando elementos más creativos y educativos para motivar a la gente a participar.

A continuación lo que se puede observar es de donde procede la gente que realizó la encuesta. A parte de recoger las opciones que se marcaban con una puntuación del 12 al 1, se recogió la IP, que servía para dos cosas: proteger el sistema contra abusos y saber de donde procedían los votos. Esta se ha mantenido secreta para proteger la privacidad de los encuestados.

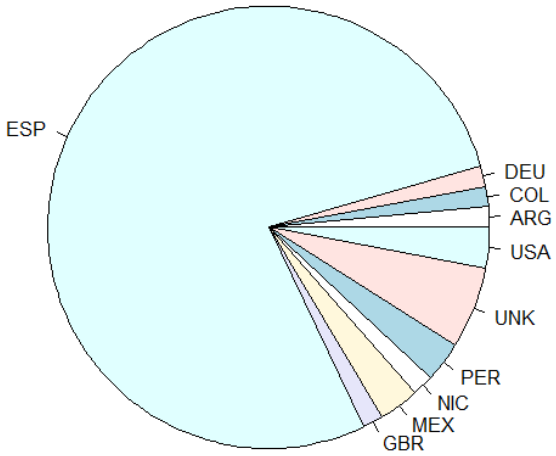

Ilustración 4.3: Encuesta - Distribución de los participantes por países

En la figura 4.3 se observa que de España proceden el 77.6% de las encuestas realizadas. El resto se distribuyen por diferentes países, donde hay cuatro direcciones IPs que no se han logrado identificar, marcadas como UNK. El resto engloba 1/4 del total, provenientes de las dos comunidades nombradas anteriormente.

En este ejemplo no importa demasiado de donde procedan, dado que usuarios de Facebook pueden ser todos, pero en una encuesta real habría que controlar geográficamente los datos, igual que su demografía, para ser correctos. Como se ha visto durante todo el proyecto, el aspecto sociológico condiciona el resultado.

4.2.3.3 INTERPRETACIÓN DEL RESULTADO

Los resultados se pueden encontrar en el apéndice A de este proyecto. La encuesta pedía que el usuario seleccionara en orden de prioridad las diferentes opciones de Facebook. El sistema asignaba a cada opción una puntuación empezando por el 12, por lo tanto, la primera opción elegida tenía 12 puntos, la segunda 11 puntos, y así hasta la última, que tenía 1 punto, sin repeticiones.

En la encuesta se marcaron las miniaturas con un valor numérico y las páginas con una letra (X1-X7 y A-E respectivamente.). Con los datos en crudo, se generó una primera gráfica:

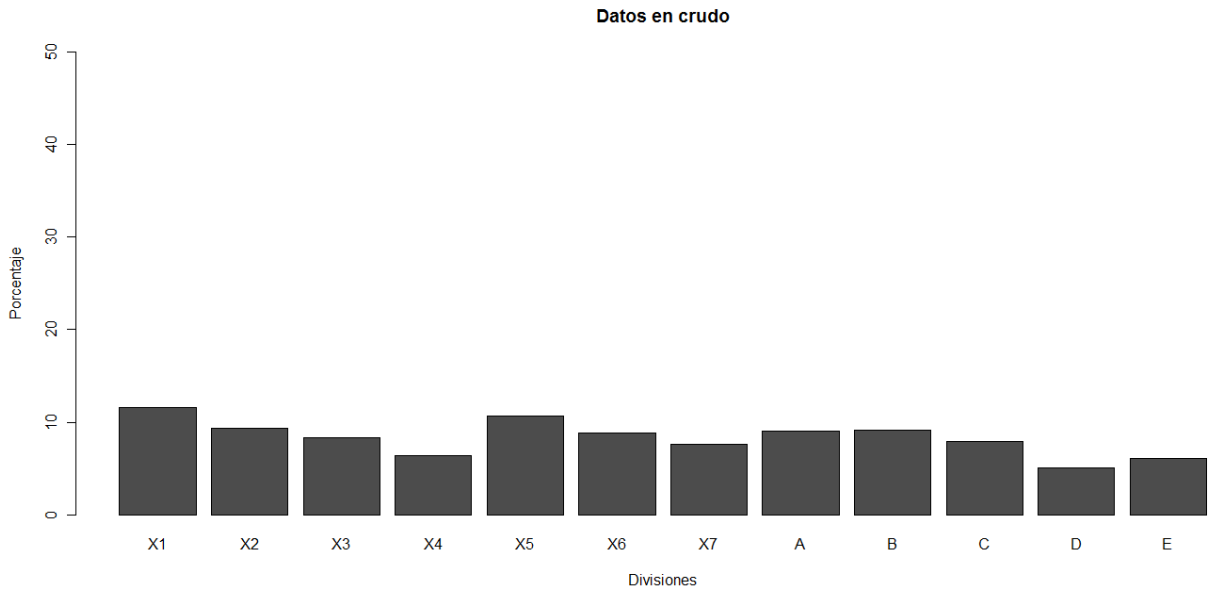

Ilustración 4.4: Encuesta - Datos en crudo

La gráfica esta compuesta por el eje Y, que representa el porcentaje de peso de cada división, y por el eje X, que representa a esas divisiones. Las divisiones son las opciones que se dejaban escoger, y son las que se vieron en la anterior imagen 4.4. Dado lo poco discriminatorio que resulta el gráfico, es difícil responder a la pregunta que se había formulado. Para poder extraer la tendencia, se crea otra gráfica a partir de, únicamente, el primer valor escogido por los usuarios:

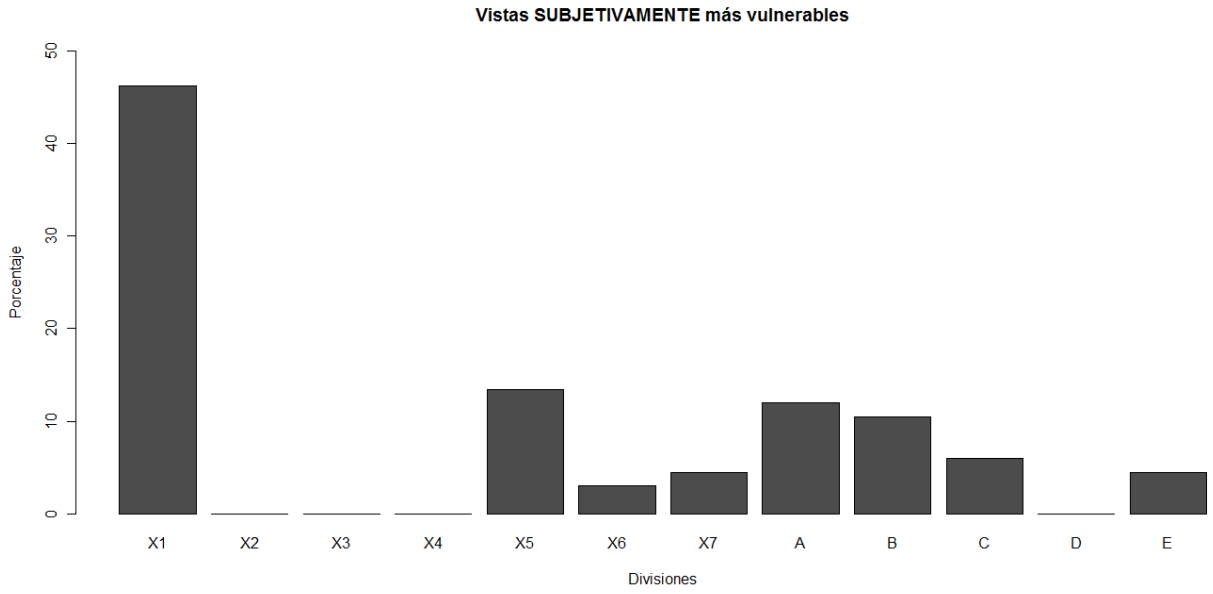

Ilustración 4.5: Encuesta - Representación de la primera opción escogida

Esta nueva gráfica muestra el número de veces que se ha escogido cada división como primera opción. En total hay 67 votos de primera opción, repartidos entre las 12 opciones, y normalizados al tanto por cien. Mirándola se puede extraer la respuesta a la pregunta: "¿Cómo es posible averiguar que se está chateando con una persona real y no con una máquina?". Si se quiere realizar un bot que suplante cuentas de Facebook, se debe insertar en su base de conocimiento las opciones X1 y X5, que son respectivamente la foto de perfil y los datos básicos de su información personal, como donde vive, que estudia, etc. Las páginas, que también reciben un número considerable de votos, pueden llegar a suprimirse a través de las opciones de privacidad. Considerándolas que existen, ya se ha abarcado más del 60% de contestaciones.

Lo importante es ver la gran diferencia entre la gente que preguntaría por la foto de perfil y el resto. Eso lleva a otra pregunta, sin relación con la anterior: "¿Donde hay más información de la persona?" o formulada de otra manera "¿De donde debería la gente realmente preguntar?". Para contestarla, se ha elaborado otra gráfica con información extraída dependiendo del tamaño del subconjunto de cada opción.

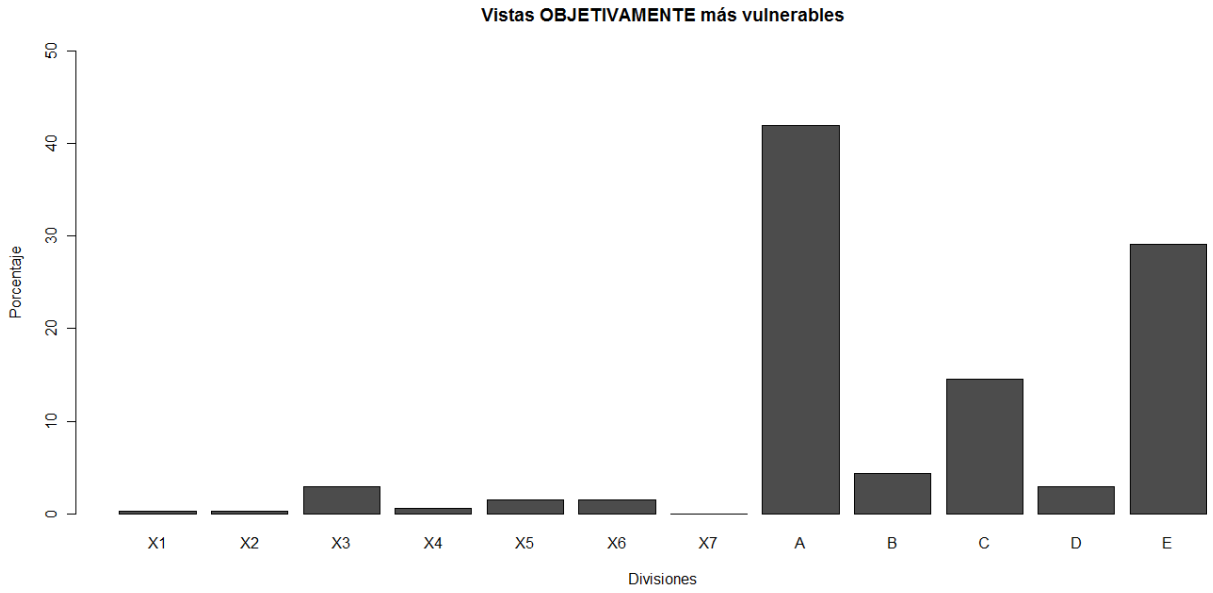

Ilustración 4.6: Encuesta - Representación de los datos "objetivamente" por el número de elementos que contienen información sobre la persona (de manera aproximada)

En la nueva gráfica 4.6 se han insertado de una manera aproximada el numero de valores que hay en cada sección. Al mismo tiempo muestra la herencia que existe en la propia visualización de Facebook. Se remarca que se distribuye por paginas (las letras) y que, al mismo tiempo, hay pequeñas secciones, que son categorías de estas páginas, que muestran una

información en concreto (los números), o pre-visualización. Llegado a este punto, se observan dos detalles. El primero, que es normal que la foto de perfil sea la primera en apreciarse, dado que Facebook ya lo ha organizado de manera que sea así. Y la segunda, que hay una clara diferencia entre lo que haría la gente y lo que debería hacer, apreciable con la siguiente figura:

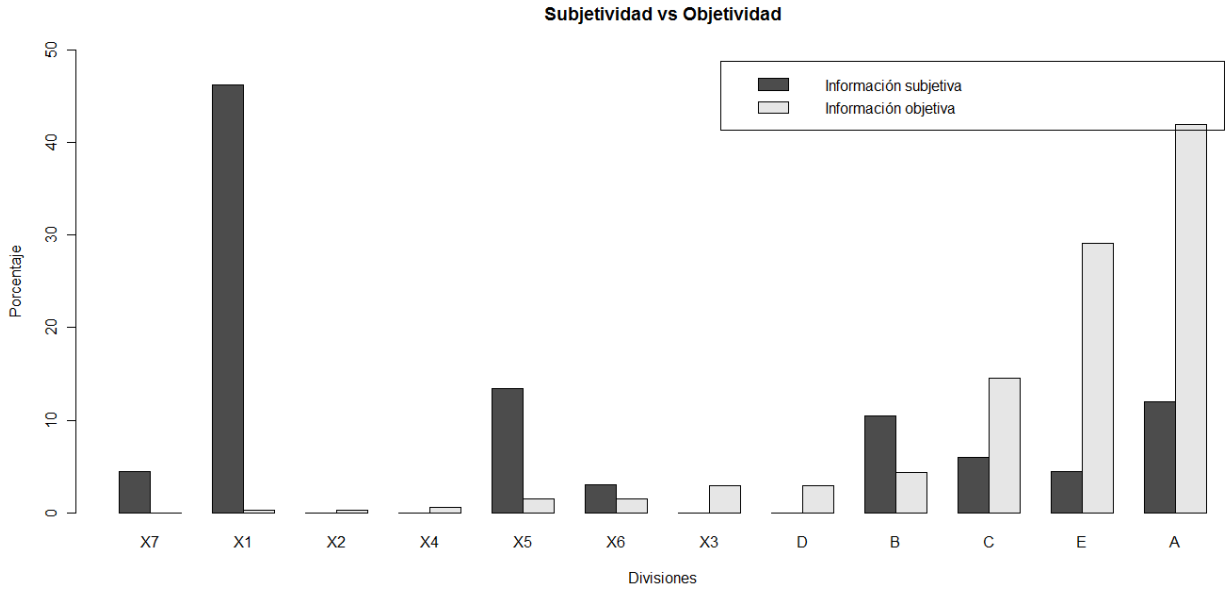

Ilustración 4.7: Encuesta - Conjunto de las dos gráficas, ordenada por la información "objetiva"

Estos dos puntos que se acaban de ver, son muy importantes para seguir estudiando el campo de I.S., ya que a partir de cómo se han diseñado los sistemas, se ha cambiado el comportamiento de las personas. El área que Facebook y otras empresas están tratando, recibe el nombre de Interacción Humano-Computador.

4.2.4 CONCLUSIONES

Se ha visto un ejemplo sobre como gestionar una encuesta que contiene opiniones subjetivas, dando las pautas para realizar cualquier otro tipo de método. En este caso la pregunta formulada ha sido “¿Cómo es posible averiguar que se está chateando con una persona real y no con una máquina?”, pudiendo aprovechar la información tanto un defensor como atacante, a través de conocer las debilidades del sistema. La respuesta fue la foto de perfil con una afinidad del 45%, pero lo importante es que la foto de perfil es la tendencia.

Se destaca el requisito de elaborar y documentar suficientemente bien cada paso para poderse volver a realizar. En la encuesta realizada, se ha contado con pocos recursos y no se ha podido obtener una mejor muestra, por lo que ha quedado demasiado orientativa. Se podría haber elaborado también una mejor encuesta, o cambiar directamente el método, para poder aplicar conceptos estadísticos más avanzados. Hubiera sido importante definir una distribución estadística y haber podido marcar unos valores de confianza, pero los datos con los que se trabajaba eran demasiado simples. En cualquier caso, es importante saber justificar por qué se ha hecho cada cosa.

En ningún caso se realizó la encuesta para que nadie saque benefició de esta aproximación. Markus Huber ya apuntaba en su trabajo que la foto de perfil había sido un factor importante, por lo que fue meramente redundante, buscando otros detalles de los que aprender y mejorar, como ha sido la introducción al mundo de la `Interacción Humano-Computador”.

5 INTERACCIÓN HUMANO-COMPUTADOR SEGURA

La Interacción Humano-Computador es una disciplina destinada a entender la comunicación entre un usuario y un ordenador. A partir del conocimiento adquirido, se pueden diseñar interfaces que facilitan, por parte del usuario, optimizar la usabilidad y el control del ordenador.

La Interacción Humano-Computador Segura, abreviada HCI-Sec, se desarrolla buscando el equilibrio entre usabilidad y seguridad. En primer lugar, parecen ser términos opuestos, pero justamente esta rama es la que se preocupa de conseguir su equilibrio. En el 2011, aún con una bibliografía exclusiva sobre el tema [58], es necesario expandir su conocimiento más allá del ámbito de la investigación. Son los ingenieros de software los encargados de aplicar estos conocimientos y fomentar el uso correcto y seguro de los ordenadores. Como caso anecdótico, la comunidad Social-Engineer.Org, aún teniendo una relación e interés en la HCI-Sec, todavía no contiene ninguna entrada de tales palabras en su red de contenidos.

La sección 5.1 y la sección 5.2 discuten desde un punto de vista defensivo los enfoques de la usabilidad y la seguridad respectivamente, dejando la sección 5.3 para discutirlo desde un punto de vista de I.S., exponiendo algunos ejemplos.

5.1 DISEÑO CENTRADO EN LA USABILIDAD

El objetivo primario de la gran mayoría de aplicaciones expuestas al público es ser usable. Para ello se desarrollo el D.C.U., un proceso para comprender la interacción y mejorarla. La definición original es Diseño Centrado en el Usuario, pero cambiando usuario por usabilidad hace más fácil

la lectura y la comparación de los términos en este trabajo. Hay muchas más alternativas de diseños como, por ejemplo, centrado en el rendimiento, en el ahorro energético, etc.

El proceso de la D.C.U. se basa en un ciclo de prueba y error, simplemente acotando bien los requisitos detectados al inicio del proyecto, e iterando sobre un prototipo que se va probando al mismo tiempo. Por esto, se determinan tres etapas, como se ve en la figura 5.1, y cada etapa tiene diferentes técnicas. Al final, lo más importante del proceso es escoger las técnicas correctas, consiguiéndose mediante *feedback* obtenido por cualquier medio.

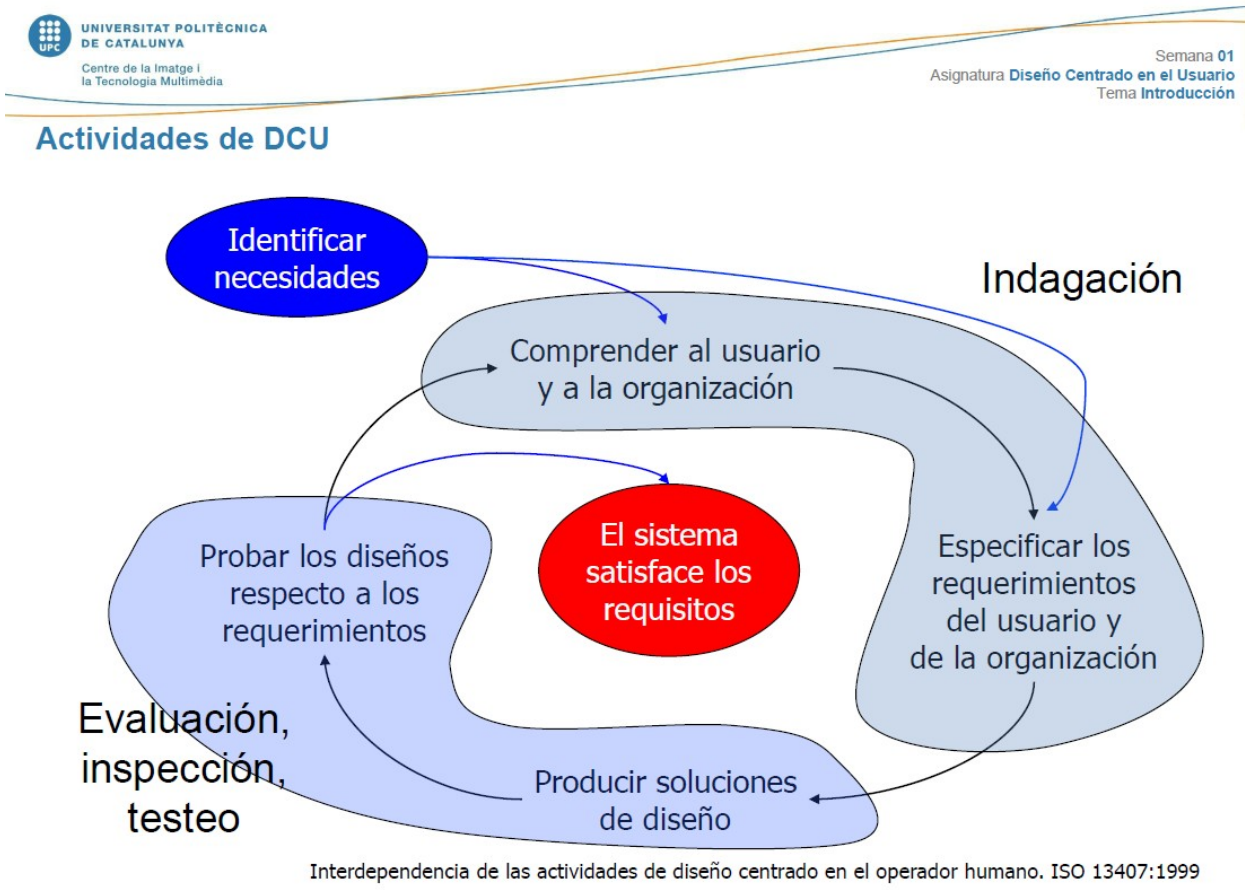

Ilustración 5.1: Ciclo de actividades del D.C.U. [59]

- La indagación. Se basa en conocer al usuario, desde sus aspiraciones hasta sus necesidades. Requiere medir hasta que punto está satisfecho, y finalmente, establecer los requisitos básicos en todo el proyecto. Se utilizan algunas de las siguientes técnicas: estudios de campo, tanto entrevistas como observación contextual, entrevistas clásicas, cuestionarios, encuestas, focus group, cardsort,

websort, etc.

- La inspección. Se basa en ajustar los límites acordados de usabilidad, a partir de los objetivos que se han marcado y de conocer el entorno a adaptar. A veces el sistema esta sobreajustado y hay que volver a una etapa inicial. Se utilizan algunas de las siguientes técnicas: evaluación heurística, inspecciones formales, paseos cognitivos, etc.
- La evaluación. Aquí se utiliza el sistema con sujetos diferentes a los originales para comprobar que es eficiente. Se utilizan algunas de las siguientes técnicas: test exploratorio, test experimental, etc.

5.2 DISEÑO CENTRADO EN LA SEGURIDAD

Simson L. Garfinkel realiza en 2005 su tesis doctoralbuscando un punto de cohesión en el conflicto entre la usabilidad y la seguridad. Se le puede considerar el pionero del HCI-Sec, aunque él mismo ya señale que desde el 1978 había alguna mención al tema. Simson argumenta [60] la siguiente frase como conjunto de su tesis: *"Usability and security can be made synergistic by redesigning systems with specific principles and through the adoption of well-defined patterns"* traducido como "la usabilidad y la seguridad se pueden hacer sinergia mediante el rediseño de los sistemas con los principios específicos y mediante la adopción de patrones bien definidos". Esta frase esconde dos ideas a concretar, explicadas a continuación.

La primera y más importante, es la necesidad de diseñar patrones y añadirlos en las etapas de ingeniería de software. La intención es dar soporte a lo que actualmente existe, con nuevos elementos y conceptos que, sin necesariamente alterar el comportamiento del programa, ofrecen una mayor estabilidad y seguridad al usuario. Un ejemplo sería acompañar a las aplicaciones con pantallas *wizzard*, o utilizar firmas digitales y cifrado por defecto en todos los correos electrónicos. Estos elementos deben ser una capa invisible para el usuario.

Esta segunda idea, se basa en enseñar bien a la gente. Simson, con sus experimentos, observa que una parte de la culpa es el pasotismo de la gente, pero otra de la mala información que proveen los programas. Aparte de instar a una mejor calidad, insiste en que sea coherente lo que dice el programa con lo que hace. Él llama a hacer una "seguridad automática, entendible y auditable" [60]. Establece que a partir de buenas prácticas de conducta ofrecidas por unos resultados "naturales", y del uso correcto de los programas, se acabe aumentando notablemente la seguridad personal, objetivo de la HCI-Sec.

Simson destaca el problema de que “muchos desarrolladores están entrenados en seguridad o en usabilidad, pero no en ambas” [60]. Al final, para demostrar y argumentar sus afirmaciones, lo que utiliza son ejemplos y estadísticas de casos reales llevados a cabo por él.

En el siguiente apartado, con un enfoque ya hostil, se localizan algunos ejemplos maliciosos de como utilizar los conocimientos de HCI-Sec para provocar un ataque de I.S.

5.3 VISIÓN OFENSIVA

Aunque existan páginas como Facebook, las cuales disponen de fuertes medidas extraídas de la disciplina HCI-Sec, el resto no lo tiene apenas en consideración. Esto provoca que, analizando la seguridad de estas webs mejor protegidas, un atacante pueda aprender métodos de ataque que todavía no conocía. Por lo tanto, que una web implemente mejores medidas de seguridad debilita al resto de páginas dada la información que ofrece al defenderse.

Actualmente los expertos de HCI-Sec están muy preocupados y ponen sus recursos en los temas de privacidad. Se estudia desde hace ya décadas, pero con las redes sociales y otras nuevas tecnologías han aparecido más debilidades. Se menciona que sea imprescindible un control posible por parte del usuario en todas las redes sociales, permitiendo decidir con quién y qué se comparte con cada persona. Para el atacante significa que si tal control no existe, se podrá aprovechar para hacer ataques, por ejemplo, de *rapport* y de `*spear phishing*´.

Para valorar la gravedad del asunto, se muestran una serie de ejemplos de ataques de I.S. dirigidos al entorno de HCI-Sec. Se han separado aquellos que pertenecen a aplicaciones de software, incluyendo el sistema operativo, y aquellos que pueden lograrse desde periféricos.

5.3.1 SOFTWARE: CUANDO LO QUE SE VE NO ES LO QUE ES

Interfaz visual – UI - Android

Igual que un estafador telefónico utiliza las palabras, la aplicación utiliza su interfaz para comunicar la información al usuario, pero esta interfaz es muy heterogénea y fácil de suplantar.

En programación se separa la parte gráfica del resto, desacoplando el sistema por capas, por lo que sólo hay que extraer la capa de las vistas. En plataformas como Android se puede copiar la estructura y los gráficos decompilando cualquier aplicación y extrayendo la carpeta de *resources*. De igual manera se puede en los *widgets* de escritorio, de los cuales a veces se dispone de su código original, y en páginas web, que como ya se vio en el Capítulo 3, se clonan con la herramienta `*Social Engineer Toolkit*´. Pero un usurpador es capaz de utilizar más técnicas, logrando infectar código malicioso en una aplicación ya en marcha o manipular la firma del fichero, dejando de poder verificar la integridad y la autoridad. En tal caso, el atacante deberá evaluar que elementos pueden ser más valiosos, donde hay carencia de protección, y desplazar al usuario al mejor escenario.

Iconos y carpetas - Windows

Los sistemas operativos se organizan a través de ficheros y carpetas, identificados por iconos. Dada la diversidad de sistemas operativos, cada uno utiliza unos iconos con un significado diferente. El problema reside cuando la asociación entre el icono y lo que ejecuta no es fiable.

En el sistema operativo Windows, se ha encontrado que las extensiones `.scf´, que pertenecen a los llamados `Windows Explorer Command´ (*Shell Command File*), auto-ocultan su extensión, sin una manera fácil de mostrarse. Esta extensión permite también redefinir el icono que utiliza y permite ejecutar un comando predefinido al hacer doble click. Esto es idóneo para nombrar un fichero `explorer.exe.scf` y que al pulsarlo ejecute el comando en cuestión. Por suerte, ese comando sólo puede ser de una lista concreta. Una persona experta en HCI-Sec debe ser capaz de darse cuenta que tiene otras aplicaciones, como añadir cualquier tipo de datos al final de este fichero. Un caso similar existe con las carpetas en Windows, porque utilizando un fichero llamado `desktop.ini´ dentro de una carpeta, se pueden cambiar varias propiedades como, por ejemplo, la ruta a donde apunta su icono. Esto hace que cualquier usuario que accede a esa carpeta se vaya a otra, por ejemplo, a la papelera de reciclaje, quedándole totalmente oculta la real. Estas incoherencias del sistema potencian todo tipo de virus y troyanos para no ser detectados ni borrados por usuarios estándar, mientras que proporcionan puntos de apoyo a los I.S. para sorprender al usuario que no lo sabe.

A día de hoy ya se están intentando estandarizar algunos iconos característicos, con proyectos como el Drumbeat de Mozilla. Igual que en el mundo físico se ha logrado un lenguaje universal de signos, en Internet resulta más difícil que se vaya a implementar por parte de todos. Al final ser diferente y original es un valor añadido valorado en la red.

Visual *Spoofing* – Dominios de Internet

El visual *spoofing* es el arte de encontrar palabras muy parecidas visualmente, pero que no sean las mismas. Normalmente ataca a los servidores de dominios, encargados de resolver las direcciones escritas en IPs pero se puede extrapolar a más campos. Para conseguir esa similitud visual, se pueden explotar dos maneras:

La primera es usando las fuentes tipográficas. Un ejemplo es la similitud entre el número `1´ y la letra `l´. Otro ejemplo es la similitud entre la letra `m` y el conjunto `rn`, que son dos letras juntas. Si en estos ejemplos se modifican las fuentes, se obtienen resultados incluso más parecidos.

La segunda es usando la codificación con los caracteres *unicode*. Estos caracteres especiales por tener tanta similitud han recibido el nombre de `*The confusables*´, divididos en tres escalas, *single-script*, *mixed-script* y *whole-script*, dependiendo de la magnitud de caracteres de otro *charset* o codificación utilizada. Un ejemplo sería la palabra `*scope*´, que se visualiza igual con la combinación de *unicodes* en *charset* `Latin´: %u0073 %u0063 %u006F %u0070 %u0065, que con la combinación de *unicodes* en *charset* `Cyrillic´: %u0455 %u0441 %u043E %u0440 %u0435, sin coincidir ningún carácter. Tal ataque, destinado a los nombres de dominio principalmente, ha recibido el nombre de `*IDN homograph attack*´, saneado posteriormente creando el *punycode*, que no es más que una expansión de los caracteres *unicode* en los dominios de Internet, añadiendo caracteres visibles muy notorios.

¿Cómo protegerse?

Como se vio en el Capítulo 3.3.2.3, con la página confianzaOnline.es, no sería una forma segura de protegerse. Existen otro servicios similares como Trust-e, que disponen de otra implementación, pero que tampoco son infalibles.

La idea que persiguen estos sitios que venden confianza es ofrecer una garantía de que el sitio que tienen como cliente es verídico y es lo que dice ser, siendo esto un servicio, porque ellos mismos han auditado el proceso. Esa es la idea, pero al final la implementación no es la adecuada. El primer problema reside en el usuario, que no conoce que debe pulsar el icono para verificar que esa página es correcta de verdad. El segundo problema es que, en la página, muchas veces ni pone el icono ni el enlace correctamente. Y el tercero, que la página suministradora de enlaces no comprueba la referencia de donde se procede, por lo que enviando al usuario desde una página falsa se le podría engañar si no comprueba realmente el dominio.

5.3.2 HARDWARE: HUMAN INTERFACE DEVICES

USB Human Interface Device (H.I.D.) Teensy

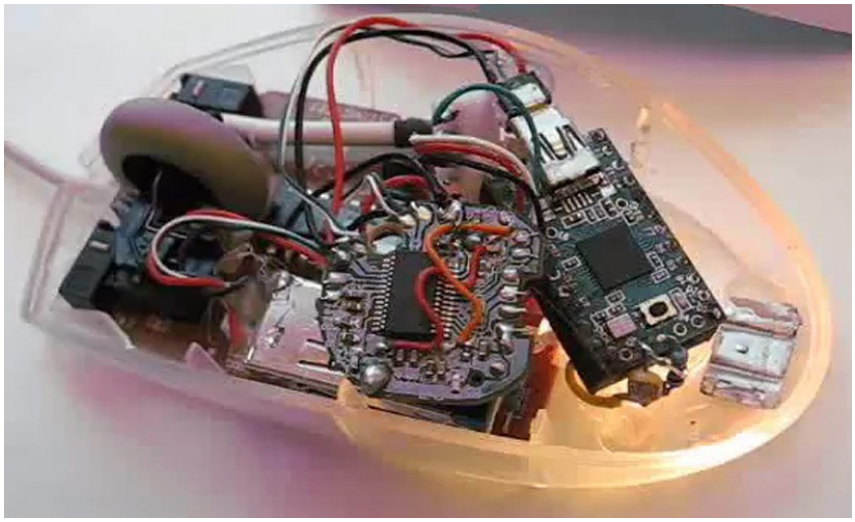

Ilustración 5.2: Raton con Teensy incrustado [61]

Teensy es un microcircuito programable, con salida mini-usb tipo B, un led y un botón. Dado su pequeño tamaño, enganchándolo a un adaptador de mini-usb tipo B macho a mini-usb tipo A macho, se puede transformar en un microcircuito con apariencia de *pendrive*. Dado que es un H.I.D., es decir, que se puede emular el comportamiento de la señal del usb, puede ser reconocido como un teclado, un ratón o una memoria. Hacerse pasar por otro dispositivo le permite enviar teclas, movimientos de ratón, etc. de forma automática, sin que el usuario pueda reaccionar a tiempo. Además, también se puede incrustar en periféricos ya reconocibles, como Adrian Crenshaw muestra [61] con un ratón (figura 5.2), y conseguir que las ordenes programadas se produzcan un tiempo después de la conexión del dispositivo, aumentando el impacto y la sutileza del ataque. También es interesante proporcionarle una carcasa divertida si el objetivo es hacerlo pasar como *pendrive*, como ha hecho la gente de Hak5 [62], usando un patito de goma.

Al final del trabajo se anexa un código realizado para mostrar por pantalla la consola de Windows y un teclado virtual, destinado para utilizar en kioskos con entrada USB sin teclado físico.

Hishing

El Hishing se comenta en el Capítulo 3, mezclando hardware y *phishing*. Las posibilidades son muchas, pero la falta de un enfoque con una visión HCI-Sec ha hecho que todavía se mantenga bastante desconocido. Un ejemplo es comprar un móvil de segunda mano y que este lleve un *firmware* modificado, el cual roba de forma invisible la información que va gestionando. Otro ejemplo, que muestra todavía más la seriedad del asunto es el lector de tarjetas de certificados digitales EMV, el cual puede llegar a hacer compras sin consentimiento ni conocimiento del usuario [63]. Tales lectores de tarjetas manipulados pesan aproximadamente 100g más que sus modelos originales, y todos ellos suelen proceder de lugares como China o Taiwan. Otro ejemplo son los llamados ePad, tablets chinos muy baratos con virus incorporados de fábrica [64].

5.4 NEGOCIO: FALTAN RECURSOS

Se han comentado cosas que resultan muy interesantes a la hora de poder crear nuevos puestos de empleo entorno a la seguridad informática. Algunas de ellas son las siguientes:

- Es posible crear patrones destinados a complementar aquellos ya existentes en el área de la ingeniería del *software*. No haría falta que los programadores aprendieran más conocimientos del lenguaje, sólo modificar y ampliar el catálogo de patrones aplicables. Provocaría que no se arrastraran ciertos problemas a capas superiores. Por esto hay que asesorar en temas de seguridad en el proceso de creación del *software,* recomendando los patrones más adecuados para cada proyecto.
- Es necesario seguir concienciando a los usuarios. Cada vez hay más manos utilizando sistemas informáticos, lo cual aumenta la posibilidad de vulnerabilidades, errores y despistes. Para erradicar tal problema, son necesarias buenas maneras de trabajar por parte del usuario. Hay que elaborar directrices simples y repetitivas, y sin entrar en confusiones. Elaborando un sistema más usable se fortalece este punto.
- Es interesante que las empresas de auditorias en seguridad informática contraten expertos en HCI-Sec, los cuales no se dediquen a la parte técnica del sistema, como sería comprobar parámetros, sino a entender el sistema, intentando atacarlo por lo que es en su fundamento. Ver el objetivo con otros ojos implica detectar y prevenir la mala utilización, determinando que riesgo se pueden producir. Al final es un valor añadido a ofrecer a los clientes, dado que detectar elementos que supongan una fuga de datos o robo de contraseñas puede dar un mal prestigio.
- Es y será cada vez más lógico pensar en crear grupos independientes que verifiquen y

ofrezcan la confianza de que los productos que se están utilizando, ni están alterados, ni se están aprovechando del usuario. Dada la gran demanda de productos *outsourcing*, sería adecuado crear esta última capa de seguridad y evaluación en el país de origen, creando más empleo local y de calidad, incidiendo como tema clave la privacidad.

Estas son algunas de las expansiones que puede ofrecer el campo de la HCI-Sec, pero seguro que el valor que añada un profesional que conozca tanto de usabilidad como de seguridad puede favorecer en otros muchos campos. En la Universitat Oberta de Catalunya se ha creado un nuevo postgrado de Interacción Persona-Computador, que será un tema de gran relevancia en los próximos años.

6 CONCLUSIONES

El proyecto se ha empezado definiendo la I.S. y dictando unas pautas identificables en todos los ataques. A continuación se han tratado diversos elementos relacionados con la I.S., por tal de cambiar el enfoque de la mente del lector, mostrando el peligro de su alrededor y, en ocasiones, como aprovecharlo. Después se ha propuesto una prueba de concepto sobre Internet, sin el atacante presente, y se ha estructurado sobre como obtener información de sujetos para adaptar cada ataque. Finalmente se ha llevado la I.S. hasta el campo de la `Interacción Humano-Computador Segura´, mostrándolo como un nuevo modelo de negocio, explotable tanto para atacante como defensor.

Durante el proyecto, se han desarrollado dos pruebas de concepto. Uno ha sido para hacer la encuesta que recogía los datos de la gente, y otro ha sido para hacer una prueba de concepto del periférico Teensy, dentro del área de `Interacción Humano-Computador Segura´.

Al principio del proyecto se comentó que cada persona sacaría unas conclusiones diferentes, dado que cada persona debería valorar sus debilidades frente a ataques tan transparentes y emocionales. Lo importante es que haya servido de aportación al público en general, haciendo meditar o simplemente ofreciendo nuevas ideas para seguir trabajando en el tema.

Aunque se han explicado muchos temas, hay otros que no se han podido dar, o simplemente han sido tratado de manera superficial. Es por eso que este proyecto puede servir como base para otros nuevos proyectos, mucho más específicos, pero ya teniendo una pequeña idea de lo que actualmente esta ya funcionando por el mundo. Algunos conceptos, se quedarán obsoletos al ser publicado el trabajo, o incluso mientras se ha estado haciendo, pero al tratar todo como

ideas, estas perdurarán, como ha sido el caso del virus `I love you´, pudiendo evolucionar y adaptarse a los nuevos mercados. Lo importante a partir de ahora es saber coger un nuevo caso de I.S. e identificar todos aquellos elementos que logran que se materialice, no memorizar los casos que hubo, porque nunca se repetirán.

En definitiva, la I.S. es un ejercicio de innovación, mezclando un proceso con factores humanos, y si se mira un poco más allá de la seguridad informática y los sistemas de seguridad, se puede extrapolar dichas características a todo aquello que nos rodea, pudiendo analizar y decidir nuestras acciones con un poco más de información.

6.1 AMPLIACIONES Y TRABAJOS FUTUROS

Uno de los objetivos del trabajo era introducir el tema, pero se sigue requiriendo de mucho más estudio, tanto en expansión como en profundidad. Se aportan un listado de ideas extraídas de entre los elementos mencionados durante el trabajo, relacionadas con las nuevas tecnologías:

- Dispositivos móviles. La nueva generación de teléfonos móviles no dejan de ser pequeños ordenadores de bolsillo. Estos se catalogan en unos pocos grandes sistemas operativos (webOS, iOS, Android, …), los cuales han sido diseñados y comercializados a gran velocidad. Utilizar técnicas de intrusión tecnológicas a veces puede ser bastante difícil, pero engañar al usuario a través de pantallas y factores psico-sociales es relativamente sencillo. Se puede hacer un estudio específico de este tema cogiendo todos los sistemas operativos y creando una tabla comparativa con maneras de burlar, no su seguridad, sino su usabilidad, atacando directamente al usuario.
- Minería de datos. Se ha tratado en este proyecto un pequeño ejemplo de encuesta donde se han visto que los datos objetivos y los subjetivos eran diferentes. Las personas no se guían por lo que se debería hacer, sino por lo que creen que deben hacer. A partir de esto, se pueden realizar otras pruebas mucho más complejas, estableciendo una metodología de como llevarlo a cabo correctamente. Sería interesante conseguir que tal metodología necesitara un número mínimo de muestras, pero diera una orientación del resultado bastante exacta, por tal de hacer un ataque lo más sigiloso posible.
- Interacción Humano-Computador Segura. Se ha introducido levemente este tema en el proyecto, enfocándolo como una posible panacea de protección contra la I.S. y muchos otros ataques. Se resume en hacer bien las cosas, pero ello conlleva un esfuerzo y una complejidad muy grande, teniendo que valorar por un lado que entiende el usuario y por

otro que se está ejecutando por la capa de debajo. No es tan diferente a lo que supone llevar a cabo un ataque de I.S., pero en este caso, se ayudaría a la sociedad de otra manera. Dentro de esto, se puede enfocar en varios casos concretos y hacer patrones de ingeniería de software exclusivos.

- Factores humanos en la red. Igual que se ha comentado de los móviles, también hay muchos otros protocolos a ser examinados, como, por ejemplo, la web. Se pueden estudiar cuales son los prejuicios más importantes que existen y como llegan a manipular la conducta de la persona. Si un candado impide que el usuario pulse un botón, o contrariamente le da más ganas. Esto crearía una base de conocimientos que ayudaría a tomar mejores decisiones en otros proyectos externos. Además, se puede plantear el crear una herramienta automática que estudie, detecte y extraiga conclusiones de estos factores humanos.

7 PLANIFICACIÓN Y PRESUPUESTO

Tomando que un crédito son 20 horas y que el proyecto son 37.5 créditos, deberían haberse hecho un total de 750 horas, que al final han sido aproximadamente 760 horas, contando unas 200 horas pérdidas por problemas de planificación. En el apartado de planificación se desglosan las horas por mes y en el de presupuesto se calcula cuanto podría costar un proyecto similar.

7.1 PLANIFICACIÓN

El proyecto lo escogió el autor libremente, llevándole más tiempo del previsto el organizarse a nivel de objetivos y la recopilación de información. Tuvo varios contratiempos decidiendo los objetivos, igual que el creciente progreso en la investigación del tema. A continuación se muestra el Gantt 7.1 y se describen sus etapas.

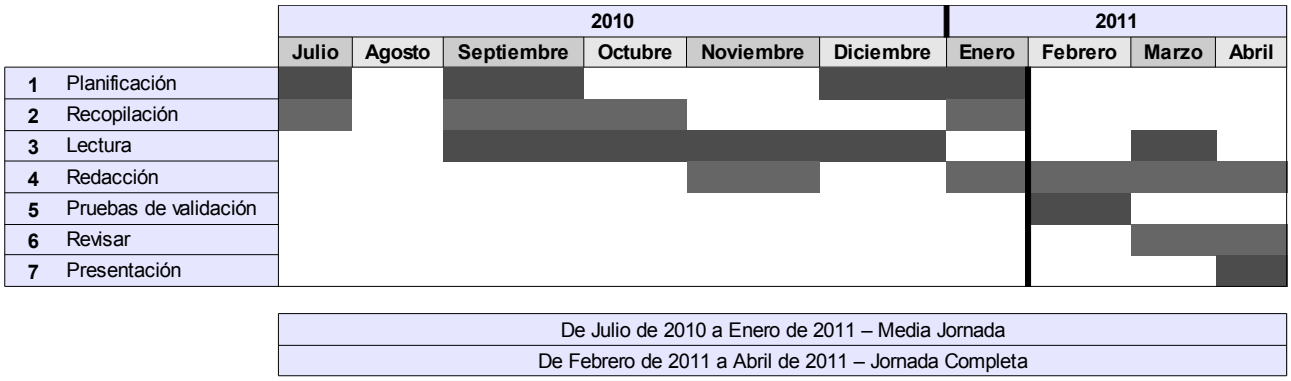

Ilustración 7.1: Gantt del proyecto

La etapa de planificación constó en definir y concretar los objetivos del proyecto. Es la parte más problemática por la dificultad de enfocar los objetivos, que duró hasta navidad, fecha en la que se elaboró una pequeña presentación que sirvió de guía para el resto del proyecto. Abarca también la necesidad de dedicar horas a presentar el informe previo, quedar con la tutora, y otros detalles.

La etapa de recopilación sirvió para encontrar todo tipo de documento o multimedia de personajes relevantes de la I.S. Se empezó con los libros de Kevin Mitnick, pero a medida que el proyecto avanzaba se fue pasando a descubrir comunidades de I.S., documentos de

investigación universitarios, reportes de empresas de auditoria, otros autores de P.N.L., la temática de HCI-Sec, etc. Fue difícil tener que readaptar el proyecto en tantos puntos por la dificultad de reunir todo el material.

La etapa de lectura consistió en dedicar tiempo a leer todo aquello que se había recopilado. Se recortó bastante y aún así se dedicó un tiempo considerable del proyecto. Aún en las últimas fechas se estaba complementando la memoria con información reciente.

La etapa de redacción se agrupó al final del proyecto. Se dedicaron muchas horas, de las cuales se perdieron muchas por el gran número de revisiones que sufrió el proyecto. La sola estructuración del conjunto ya consumió bastante tiempo.

La etapa de las pruebas de validación fue la etapa más amena. Se pudo hacer un cambio de contexto, dejando a parte el leer, redactar y gestionar y pasar a tocar código. A diferencia del resto, ofrecía resultados palpables, terminándose con bastante agilidad en un corto período.

La etapa de revisión como se ha comentado en la etapa de redacción hizo que cambiará bastante en poco tiempo. Por suerte cada revisión aportó una mejora considerable a la consolidación de todos los apartados individualmente y como grupo.

La última etapa, la de presentación, son horas destinadas a realizar la presentación final. Se primó crear una presentación creativa por tal de poder transmitir lo mejor posible el riesgo de la I.S. a cualquier tipo de público.

Todo el proceso se ha plasmado desde el primer día en una libreta dedicada exclusivamente para el proyecto, y se han ido elaborando revisiones de los documentos para mantener el registro del proceso, pero sólo se han acabado incorporando aquellas partes más relevantes.

7.2 PRESUPUESTO

Se desglosa en una tabla 7.1 todo el material y el coste de personal que produciría un proyecto de estas características, contando que una persona es capaz de llevarlo a cabo.

Presupuesto PFC de Ingeniería social			
Costes de personal			
Investigador	760 horas	30 € / hora	22.800 €
Costes de material			
Fungibles	1 unidad	10 € / unidad	10 €
Hosting	2 meses	10 € / mes	20 €
Teensy	1 unidad	25 € / unidad	25 €
Teclado bluetooth	1 unidad	15 € / unidad	15 €
		Total:	22.870 €

Tabla 7.1: Costes del proyecto

En una ampliación del proyecto sería conveniente invertir más dinero para realizar pruebas de concepto mucho más precisas, con una mejor plataforma para recoger los datos y más medios para promocionarla a una mayor población. Por ejemplo, Simson Garfinkel en su proyecto de tesis sobre la HCI-Sec [60] llegó a comprar casi 240 discos duros para analizar el comportamiento de la gente.

8 GLOSARIO

0day: Vulnerabilidad no corregida oficialmente.

Adware: Programa *spyware* que simula al usuario pulsando ciertos *banners* de publicidad.

Anchoring (anclaje): Termino de la P.N.L. que posibilita enlazar conceptos a partir de insertar un disparador en la mente, y que al notar un estimulo se dispare, haciendo que se recuerde una sensación, un recuerdo o un estado mental.

Authority (Autoridad): Poder que permite controlar a la gente de la misma organización pero de puestos inferiores.

Barnum Statements: Frases genéricas con cabida en cualquier situación.

Bogon: Es un nombre informal que recibe un paquete de Internet que dice ser de un área que no existe.

Bot/Zombie: Ordenador infectado por algun tipo de *malware* que lo controla remotamente.

Botnet: Red de ordenadores *zombies* bajo el control de un *botmaster*.

Captcha: Imagen distorsionada legible para un humano pero no para una máquina.

Clickjacking: Ataque malicioso dirigido al navegador del usuario que simula estar pulsando sobre algo pero que en realidad pulsa sobre otro sitio.

Cold reading: Técnica para extraer información de alguien sin saber nada de él.

Commitment (Responsabilidad): Forma de ser considerado sujeto de una deuda u obligación.

Consistency (Consistencia): Argumentación de todas las dudas posibles.

Cracker: Persona que hace delitos en Internet.

Cross-Site Request Forgery (CSRF): Ataque malicioso que se produce al acceder externamente desde otra web a una web que no válida correctamente la procedencia.

Domain Name System (DNS): Sistema de nomenclatura jerárquica para ordenadores.

Dumpster Diving: Técnica que consiste en buscar en la basura documentos oficiales de una empresa.

E-crime: Crimen electrónico. Cualquier tipo de crimen que se produce en Internet.

Exchangeable Image file Format (EXIF): Formato de las imágenes JPG que permite incrustar en su interior información oculta (metadatos).

Exploit: Programa que se utiliza para obtener privilegios en un sistema local o remoto.

File Transfer Protocol (FTP): Protocolo de transferencia de datos en plano basado en el sistema cliente-servidor.

Harvesting: Nombre que recibe en seguridad informática el proceso de recoger información de algo o alguien. También recibe el nombre de *footprinting*.

Hishing (Hardware Phishing): Denominación de ataque de phishing basado en utilizar periféricos para ocultar troyanos o otro tipo de *malware* debido a la poca sospecha que levanta.

Hoax: Bulo sin fin económico.

Ingeniería social (I.S.): Arte de persuadir a alguien para que haga algo sin su voluntad.

Ingeniería social automatizada (I.S.A.): Término utilizado para referirse a lograr que se efectúe un ataque de I.S. sin depender de una persona.

Internet Protocol (IP): Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a un interfaz de un dispositivo dentro de una red que utilice el protocolo IP, que corresponde al nivel de red del protocolo TCP/IP.

Keylogger: Programa spyware que captura las teclas introducidas.

Ley de Protección de Datos Personales (L.O.P.D.): Ley española de obligado cumplimiento que legisla los datos personales.

Liking (Simpatía): Acción o efecto de conectar con otra persona, sufriendo o alegrándose ambas juntas.

Malware: Conjunto de programas maliciosos para un internauta.

Man-In-The-Middle: Ataque que consiste en interceptar la comunicación entre dos individuos.

Man-Left-In-The-Middle (MLITM): Ataque a partir de la variable “Referer” del navegador que compromete la seguridad del servidor o de otro atacante.

Misdirection (Distracción): Ataque psico-social aprovechándose de las limitaciones perspectivas y de conocimiento de la víctima.

Payload: Encapsulado de datos que permite enviar acciones específicas por el atacante.

Pharming: Ataque a través de la suplantación de servidores DNS para robar credenciales.

Phisher: Persona que envía *phishing*.

Phishing: Engaño masivo a través de correos electrónicos.

Phreaker: Persona con avanzados conocimientos en redes telefónicas y su funcionamiento a la cual le gusta ponerlas a prueba.

Pickpocking: Disciplina que se encarga de estudiar como usurpar cualquier objeto de una persona sin que esta lo perciba.

Programación neurolingüística (P.N.L.): parte de la psicología no clínica, que estudia los procesos mentales con el fin de obtener un modelo formal y dinámico de cómo funciona la mente y la percepción humana.

Proxy: Es un programa o dispositivo que realiza una acción en representación de otro.

Psico-social: Mezcla de factores psicológicos y sociológicos que influencian sobre algo o alguien.

Rapport (acompasamiento): Termino utilizado cuando dos o más personas se sincronizan porque se sienten comprendidos mutuamente.

Reciprocity (Reciprocidad): Expectativa social que la gente responde acorde a lo recibido.

Rootkit: Troyano que invade la parte más crítica del sistema operativo.

Scam: Engaño personalizado con fin económico.

Scambaiter: Persona que contesta los *scams* para engañar al *scammer*.

Scammer: Persona que envía *scams*.

Scarcity (Escasez): Factor social que influencia por ser único.

Shoulder Surfing: Técnica que consiste en espiar por encima del hombro a la víctima cuando introduce una contraseña o un dato sensible.

SmiShing (SMS *Phishing*): Denominación de ataque de *phishing* basado en enviar un mensaje de texto haciéndose pasar por una entidad oficial.

Social Proof (Aprobación social): Fenómeno psicológico donde se asume que las acciones de otros reflejan el correcto comportamiento de una situación concreta

Spear Phishing: Denominación de ataque de *phishing* basado en personalizar en un nivel muy elevado el contenido del mensaje, enviándose en el momento más adecuado.

Spider: Programa que se encarga de hacer peticiones a servidores web y recoge información clave, como sus enlaces, imágenes, etc.

Spoofing: Técnicas de suplantación de identidad generalmente con usos maliciosos o de investigación.

Spyware: Programa espía que recopila información del usuario para enviarla a una empresa externa. Esta dentro del conjunto de *malware*.

Tabnapping: Ataque malicioso dirigido al navegador del usuario que simula una nueva pestaña pidiendo las credenciales sin que el usuario la haya abierto.

Troyano: Programa malicioso que obtiene el control sobre un sistema remoto.

Virus: Programa malicioso que se propaga por si sólo en la red.

Vishing (VoIP *Phishing*): Denominación de ataque de *phishing* basado en lanzar ataques a través de la vía telefónica a través de un mensaje que se repite.

Voice Over IP (VoIP): Grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP.

Web-Bug: Fichero o imagen inocua que recopila información de quien la visita.

Whisphing (Whale Phishing): Denominación de ataque de *phishing* basado en usar una técnica concreta sobre una población muy amplia.

9 APÉNDICE A. ENCUESTA Y CREACIÓN DE GRÁFICAS

~/www/structure.sql

```
CREATE TABLE IF NOT EXISTS `data` (
  `ip` int(11) NOT NULL,
  `1` int(2) NOT NULL,
  `2` int(2) NOT NULL,
  `3` int(2) NOT NULL,
  `4` int(2) NOT NULL,
  `5` int(2) NOT NULL,
  `6` int(2) NOT NULL,
  `7` int(2) NOT NULL,
  `A` int(2) NOT NULL,
  `B` int(2) NOT NULL,
  `C` int(2) NOT NULL,
  `D` int(2) NOT NULL,
  `E` int(2) NOT NULL,
  PRIMARY KEY (`ip`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

~/www/index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html><head><title>WebSort [facebook]</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link rel="stylesheet" type="text/css" href="encuesta.css">
<script type="text/javascript">
var count = 0;
var str = "";
function writeText(txt)
{
    var map = document.getElementById("map")
    var id  = document.getElementById(txt);
    map.removeChild(id);
    document.getElementById("desc").innerHTML                                    =
document.getElementById("desc").innerHTML+" "+txt;
    str = str + txt;
    count++;
    if (count==12) {
        showExit();
    }
}
function showExit() {
    document.location="recogerDatos.php?s="+str;
}
function showDialog() {
    document.getElementById("dialog").style.display = 'block'
}
function closeDialog() {
    document.getElementById("dialog").style.display = 'none'
}
```

```
</script>
</head>

<body onLoad="showDialog();">

<div id="dialog">
    <h1>WebSort de Facebook</h1>
    <p>Has llegado a la página web de Sergio Arcos, un estudiante de Ingeniería informática realizando su Proyecto Final de Carrera.</p>
    <p>Me gustaría invitarte a contribuir en este realizando una pequeña actividad. Serán apenás 5 minutos.</p>
    <h2>¿Qué debo hacer?</h2>
    <p>En la siguiente pantalla te vas a encontrar una imagen representando una cuenta ajena de Facebook. En ella hay 12 carácteres, divididos en 7 números y 5 letras.</p>
    <p>Los números representan los elementos comunes en casi cualquier pantalla.</p>
    <p>Las letras son las páginas internas de Facebook donde puedes navegar.</p>
    <p>Aún así, no le prestes atención a esto, sólo es orientativo.</p>
    <br />
    <p>Te pido lo siguiente: Imagina que acabas de conocer a esta persona y te ha dado curiosidad por mirar su perfil. Bien,</p>
    <h2>¿De donde sacarías la información para preguntarle sobre él?</h2>
    <p>Debes seleccionar en orden de "primero" (más importante) a "último" (menos importante) los 12 elementos que hay (con un click encima basta).</p>
    <p>Te pido que te metas profundamente en el papel de "pícaro" y visualices esta imagen como si fuera un caso real (imagina donde clickarias en la cuenta de tu mejor amigo o amiga).</p>
    <p>Tomate tu tiempo, o hazlo bien rápido. Se natural. (Si ves que lo has hecho mal, termina la encuesta y vuelve a esta página: puedes volver a rellenarla).</p>
    <a href="javascript:closeDialog();">EMPEZAR</a>
</div>

<img src="facebook.png" width="996" height="768" alt="[facebook_map]" usemap="#fbmap" />

<map id="map" name="fbmap">
<area id="1" shape="rect" coords="0,65,189,204" href="javascript:writeText('1');" alt="[foto de perfil]" />
<area id="2" shape="rect" coords="0,325,189,418" href="javascript:writeText('2');" alt="[relacion]" />
<area id="3" shape="rect" coords="0,418,189,574" href="javascript:writeText('3');" alt="[amistades]" />
<area id="4" shape="rect" coords="0,574,189,768" href="javascript:writeText('4');" alt="[familia]" />
<area id="5" shape="rect" coords="189,65,701,136" href="javascript:writeText('5');" alt="[datos personales]" />
<area id="6" shape="rect" coords="189,137,701,204" href="javascript:writeText('6');" alt="[previsualizacion de fotos]" />
<area id="7" shape="rect" coords="723,137,967,204" href="javascript:writeText('7');" alt="[relacion tu y yo]" />
<area id="A" shape="rect" coords="278,263,511,368" href="javascript:writeText('A');" alt="[pagina muro]" />
<area id="B" shape="rect" coords="278,394,511,537" href="javascript:writeText('B');" alt="[pagina informacion]" />
<area id="C" shape="rect" coords="598,259,830,484"
```

```
href="javascript:writeText('C');" alt="[pagina imagenes]" />
<area           id="D"           shape="rect"          coords="278,556,511,709"
href="javascript:writeText('D');" alt="[pagina canciones]" />
<area           id="E"           shape="rect"          coords="598,505,830,724"
href="javascript:writeText('E');" alt="[pagina amigos]" />
</map>

<p id="desc"></p> </body></html></body></html>
```

~/www/recogerDatos.php

```
<?php

function insertValues() {
    if (!isset($_GET['s'])) {
        return false;
    }

    $ip = ip2long($_SERVER['REMOTE_ADDR']);
    $s = strtoupper($_GET['s']);

    if (strlen($s)!=12) {
        return false;
    }

    $arr = array();
    $check = array('1','2','3','4','5','6','7','A','B','C','D','E');

    for ($i=0; $i<12; $i++) {
        $pos = strpos($s,$check[$i]);
        if ($pos === false) return false;
        $arr[$i] = 12-$pos;
    }

    $db = mysql_connect('localhost', '266474_websort', 'websort.');
    mysql_select_db('martes_zxq_websort', $db);


    $R = mysql_query("INSERT INTO `data` (
    `ip`,
    `1`, `2`, `3`, `4`, `5`, `6`, `7`,
    `A`, `B`, `C`, `D`, `E` )
    VALUES (
        $ip,
        '$arr[0]','$arr[1]','$arr[2]','$arr[3]','$arr[4]','$arr[5]','$arr[6]',
        '$arr[7]','$arr[8]','$arr[9]','$arr[10]','$arr[11]'
    )
    ON DUPLICATE KEY UPDATE
    `1`='$arr[0]', `2`='$arr[1]', `3`='$arr[2]', `4`='$arr[3]',
    `5`='$arr[4]', `6`='$arr[5]', `7`='$arr[6]', `A`='$arr[7]',
    `B`='$arr[8]', `C`='$arr[9]', `D`='$arr[10]', `E`='$arr[11]'", $db);

    // echo mysql_error();

    return true;
}
```

```
if (insertValues()) {
    require("ok.html");
} else {
    header("location: index.html");
}
?>
```

~/www/analyze.r

```
## dd = datos registrados
dd     <-     read.table("facebook.dat",
header=T, sep=" ")
dd <- dd[2:13]
dm <- dd
dd                                      <-
data.frame(sum(dd[1]),sum(dd[2]),sum(dd
[3]),sum(dd[4]),sum(dd[5]),sum(dd[6]),s
um(dd[7]),sum(dd[8]),sum(dd[9]),sum(dd[
10]),sum(dd[11]),sum(dd[12]))

## dm = datos registrados - el máximo

##     dt     =     analisis     evaluado
"objetivamente"
dt   <-   read.table("facebookTree.dat",
header=T, sep=" ")


dd <- 100*(dd/sum(dd))
dt <- 100*(dt/sum(dt))
colnames(dd) <- colnames(dt)

ncon <- ncol(dm)
for (i in 1:ncon) {
    dm[i] <- nrow(dm[dm[i]==12,])
}
dm = dm[1,]
dm <- 100*(dm/sum(dm))



barplot(as.matrix(dd),     ylim=c(0,50),
main="Datos          en          crudo",
xlab="Divisiones", ylab="Porcentaje")
barplot(as.matrix(dm),     ylim=c(0,50),
main="Vistas     SUBJETIVAMENTE     más
vulnerables",         xlab="Divisiones",
ylab="Porcentaje")
barplot(as.matrix(dt),     ylim=c(0,50),
main="Vistas     OBJETIVAMENTE     más
vulnerables",         xlab="Divisiones",
ylab="Porcentaje")
```

```
m                                       <-
matrix(as.matrix(rbind(dm,dt)),ncol=12)

colnames(m)<-colnames(dd)
rownames(m)<-c("sub","obj")

m <- as.table(m)
ms <- m[ , order(m[2,])]

barplot(ms,                ylim=c(0,50),
legend.text=c("Información
subjetiva","Información     objetiva"),
main="Subjetividad   vs   Objetividad",
xlab="Divisiones",   ylab="Porcentaje",
beside=T)




## Gráfico de localizaciones
loc    <-    read.table("location.dat",
header=T, sep=" ")

plot(loc,    main="Paises    de    los
parcipantes",      legend.text=c("77.6%
españoles",       "22.4%       resto"),
xlab="País",ylab="N°                  de
participantes" )

pie(loc)
```

10 APÉNDICE B. PROGRAMACIÓN DE TEENSY

El siguiente código es capaz de abrir una ventana de comandos del sistema y un teclado virtual.

~/teensy/example/keyboard/keyboard.pde

```
void setup() {
}

void loop() {

   // Esperamos a que este activo el
teclado
  while(Keyboard.isInit()){
    Keyboard.set_key1(KEY_NUM_LOCK);
    Keyboard.send_now();
    delay(500);
  }

  // Ctrl + R -> Ventana de Windows de
Ejecutar comando
  Keyboard.set_modifier(MODIFIERKEY_GUI
);
  Keyboard.set_key1(KEY_R);
  Keyboard.send_now();
  sendNull();
  delay(200);

  // Introducimos el siguiente comando
  // cmd /C for %i in (i h g f e d) do
@%i:\b.bat 2>nul
  Keyboard.print("cmd ");
  // /
  Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_4);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_7);
  Keyboard.send_now();
  sendNull();

  Keyboard.print("C for ");
  // %
  Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_3);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_7);
  Keyboard.send_now();
  sendNull();

  Keyboard.print("i in ");
  Keyboard.print("i");
  sendNull();

  // :
  Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_5);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_8);
  Keyboard.send_now();
  sendNull();

  // \
  Keyboard.set_key1(KEY_LEFT);
  Keyboard.send_now();
  Keyboard.set_key1(KEY_RIGHT);
  Keyboard.send_now();

  Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_9);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_2);
  Keyboard.send_now();
  sendNull();

  Keyboard.print("b");
  // .
  Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_4);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_6);
  Keyboard.send_now();
  sendNull();

  Keyboard.print("bat 2");
  // >
  Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_6);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_2);
  Keyboard.send_now();
  sendNull();
```

```
  // (
   Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_4);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_0);
  Keyboard.send_now();
  sendNull();

  Keyboard.print("i h g f e d");
  // )
   Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_4);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_1);
  Keyboard.send_now();
  sendNull();

  Keyboard.print(" do ");
  // @
   Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_6);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_4);
  Keyboard.send_now();
  sendNull();
  // %
   Keyboard.set_modifier(MODIFIERKEY_ALT
);
  Keyboard.set_key1(KEYPAD_3);
  Keyboard.send_now();
  Keyboard.set_key1(KEYPAD_7);
  Keyboard.send_now();
  sendNull();
  Keyboard.print("nul");

  // Pulsamos el enviar
  Keyboard.set_key1(KEY_ENTER);
  Keyboard.send_now();
  sendNull();

  delay(1000);
  // Reestablecemos valores originales
  Keyboard.set_key1(KEY_NUM_LOCK);
  Keyboard.send_now();
  sendNull();

  // Esperamos eternamente

  while(true);
}

// Funcion que simula soltar una tecla
void sendNull() {
  Keyboard.set_modifier(0);
  Keyboard.set_key1(0);
  Keyboard.send_now();
}
```

~/teensy/example/keyboard/disk/b.bat

```
start osk
start /MAX cmd
exit
```

11 INDICE DE FIGURAS

12 INDICE DE TABLAS

13 BIBLIOGRAFIA

[1] Diccionario de la lengua española. Definición. *Real Academia Española, 2011.* URL http://rae.es/social [Visto 2011-04-30]

[2] Marcus Nohlberg. Securing Information Assets: Understanding, Measuring and Protecting against Social Engineering Attacks. *Stockholm University - University of Skövde, December 2008.* ISBN 978-91-7155-786-5.

[3] Kevin D. Mitnick & William L. Simon. The Art of Deception: Controlling the Human Element of Security 1st. *John Wiley & Sons, Inc., New York, NY, USA ©, 2002.* ISBN 0-7645-4280-X.

[4] Kevin D. Mitnick & William L. Simon. The Art of Intrusion: The Real Stories Behind the Exploits of Hackers, Intruders and Deceivers. *Wiley Publishing, Inc., 2005.* ISBN 0-7645-6959-7

[5] Joe Chappele. Hackers 2: Takedown. (*Película) Wilmington, North Carolina, USA, 2000.*

[6] Dale Pearson. Social Engineering, Mentalism, Hypnosis, Misdirection and Influence. *Blog, 2010.* URL http://www.subliminalhacking.net/ [Visto 2011-04-30]

[7] David Gragg. A Multi-Level Defense Against Social Engineering. *SANS Institute, GSEC Option 1 version 1.4b, December 2002.*

[8] Markus Huber. Automated Social Engineering: Proof of Concept. *DSV SecLab, Stockholm University/Royal Institute of Technology, Mar 2009.*

[9] Tobias Lauinger, Veikko Pankakoski, Davide Balzarotti, Engin Kirda. Honeybot, Your Man in the Middle for Automated Social Engineering. *EURECOM, Sophia-Antipolis, France, 2010.*

[10] (ISC)^2. C.I.S.S.P. Test Exam: Questions and Answers. *Online, 1994.* URL http://itsecuritylive.com/PDF/CISSPc.pdf [Visto 2011-04-29]

[11] BBC News. Love Bug suspect held. *News, (Monday) 8 May, 2000.* URL http://news.bbc.co.uk/2/hi/science/nature/740623.stm [Visto 2011-04-30]

[12] TheRegister News. No 'sorry' from Love Bug author. *News, 11 May, 2005.* URL http://www.theregister.co.uk/2005/05/11/love_bug_author/ [Visto 2011-04-30]

[13] Chan Robles Virtual Law Library. REPUBLIC ACT NO. 8792. *Electronic Commerce Act of 2000.* URL http://www.chanrobles.com/republicactno8792.htm [Visto 2011-04-30]

[14] BBC News. Police close in on Love Bug culprit. *News, (Saturday) 6 May, 2000.* URL http://news.bbc.co.uk/2/hi/science/nature/738537.stm [Visto 2011-04-30]

[15] HeadQuarters United States Army Forces Command. FORSCOM “ILOVEYOU” Virus Lessons Learned Report. *Department of the army, July 2000.* URL http://www.iwar.org.uk/iwar/resources/call/love.pdf [Visto 2011-04-30]

[16] Robert B. Cialdini. Influence: The Psychology of Persuasion. *Collins; Revised edition (October 7, 1998).* ISBN 978-0688128166

[17] Team Cymru Research NFP. Malicious Activity Maps. *Burr Ridge, IL 60527 | USA, 2011.* URL http://www.team-cymru.org/Monitoring/Malevolence/maps.html [Visto 2011-04-30]

[18] The measurement Factory. Mapping the IpV4 Address Space. *Online, © 2009.* URL http://maps.measurement-factory.com/ [Visto 2010-11-14]

[19] Anónimo. Vulnerabilidades en ListaRobinson.es. *SecurityByDefault, octubre de 2010.* URL http://www.securitybydefault.com/2010/10/vulnerabilidades-en-listarobinsones.html [Visto 2011-04-30]

[20] Johnny Long. No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing. *Syngress (February 21, 2008).* ISBN 978-1597492157

[21] Nart Villeneuve. KOOBFACE: Inside a Crimeware Network. *Report, JR04-2010*. URL http://www.infowar-monitor.net/reports/iwm-koobface.pdf [Visto 2011-04-30]

[22] Daniel G. Siegel. On the New Threats of Social, Engineering Exploiting Social Networks. *FAKULTÄT FÜR INFORMATIK, TECHNISCHE UNIVERSITÄT MÜNCHEN, Bachelorarbeit in Informatik, August 2009.*

[23] Stephen Lattanzio. ATM Skimming and Phishing or Trapping. *GLG Research, The #1 ATM Security Concern, July 2006.* URL http://www.glgroup.com/News/ATM-Skimming-and-Phishing-or-Trapping-2250.html [Visto 2011-04-30]

[24] Axelle Apvrille. Android DroidDream Uses Two Vulnerabilities. *Fortinet, March, 2011.* URL http://blog.fortinet.com/android-droiddream-uses-two-vulnerabilities/ [Visto 2011-04-30]

[25] Adrian Latorre Crespo. Soy estúpido y tacaño, solo costaba 1$. *El android libre, 31 marzo 2011.* URL http://www.elandroidelibre.com/2011/03/soy-estupido-y-tacano-solo-costaba-1-no-robes-como-hice-yo.html [Visto 2011-04-30]

[26] 419eatear. Don't fall for common scams like this - fight them!. *Community, 2003.* URL http://www.419eater.com/ [Visto 2011-04-30]

[27] Vicente Díaz. eCrime - las nuevas mafias. *S21SEC, FirstConference, 2008.* URL http://www.oissg.org/images/files/vicente_diaz__ecrime_las_nuevas_mafias__s21sec.pdf [Visto 2011-04-30]

[28] Panda Security. Nace el Consejo Nacional Consultor sobre Cyber-Seguridad. *Prensa, 2009.* URL http://www.pandasecurity.com/spain/about/corporate-news/new-70.htm [Visto 2011-04-30]

[29] Eduard Punset. Lenguaje Corporal. *Redes, 2009.* URL http://www.youtube.com/watch?v=tL7DJtprLw8 [Visto 2011-04-30]

[30] Alan Lazalde. Stuxnet y el sombrío 9 de mayo de 1979. *Alt1040, la guía del geek. Octubre, 2010.* URL http://alt1040.com/2010/10/stuxnet-y-el-sombrio-9-de-mayo-de-1979 [Visto 2011-04-30]

[31] Ruben Santamarta. Legionella vs SCADA/HVAC. *48bits, Octubre 2010.* URL http://blog.48bits.com/2010/10/25/legionella-vs-scadahvac/ [Visto 2011-04-30]

[32] Larry McVoy. BK2CVS problem. *Linux-Kernel Archive Nov 2003.* URL http://lkml.indiana.edu/hypermail/linux/kernel/0311.0/0621.html [Visto 2011-04-30]

[33] Peter Bright. Anonymous speaks: the inside story of the HBGary hack. *Ars Technica, Feb 2011.* URL http://arstechnica.com/tech-policy/news/2011/02/anonymous-speaks-the-inside-story-of-the-hbgary-hack.ars/3 [Visto 2011-04-30]

[34] Team Cymru Research NFP. A taste of http botnets. *Burr Ridge, IL 60527 | USA, 2011.* URL http://www.team-cymru.com/ReadingRoom/Whitepapers/2008/http-botnets.pdf [Visto 2011-04-30]

[35] Ryan Naraine and Dancho Danchev. Brazilian ID thieves using Twitter as botnet command channel. *ZDNet, August 2009.* URL http://www.zdnet.com/blog/security/brazilian-id-thieves-using-twitter-as-botnet-command-channel/4060 [Visto 2011-04-30]

[36] James O'Gorman. How to lie: Three tips. *Social-Engineer Newsletter, Vol 02 Issue 14*. URL http://www.social-engineer.org/newsletter/SocialEngineerNewsletterVol02Is14.htm [Visto 2011-04-30]

[37] A. M. Turing. Computing Machinery and Intelligence. Journal (Paginated) , 1950. URL http://cogprints.org/499/ [Visto 2011-04-30]

[38] David Cole. The Chinese Room Argument. *The Stanford Encyclopedia of Philosophy (Winter 2009 Edition), Edward N. Zalta (ed.)*. URL http://plato.stanford.edu/archives/win2009/entries/chinese-room/. [Visto 2011-04-30]

[39] Elizabeth Montalbano. Phishing Education Called Inadequate. *IDG News, PCWorld, Oct 2007.* URL http://www.pcworld.com/article/138243/phishing_education_called_inadequate.html [Visto 2011-04-30]

[40] Lorrie Cranor, Alessandro Acquisti, Julie Downs, Jason Hong and Norman Sadeh. Anti-Phishing Filtering & Education. *CUPS, Carnegie Mellon, May 2008.* URL http://cups.cs.cmu.edu/trust/May-2008-handout.pdf [Visto 2011-04-30]

[41] PhishGuru™. PhishGuru™. *Wombat Security Technologies, 2008-2011.* URL http://www.wombatsecurity.com/phishguru [Visto 2011-04-30]

[42] Sun Tzu. El arte de la guerra. *Planeta Madrid S.A., M.R. Ediciones, 1999.* ISBN 978-84-270-2499-1

[43] Agencia Española de Protección de Datos. Guía para el ciudadano. *AGPD*. URL

https://www.agpd.es/portalwebAGPD/canaldocumentacion/publicaciones/common/Guias/GUIA_CIUDADANO_OK.pdf [Visto 2011-04-30]

[44] Secretaría de estado de telecomunicaciones y para la sociedad de la información. La ley de Internet fácil. *Ministerio de Industria, Turismo y Comercio, LSSI: Ley 34/2002, de 11 de julio.* http://www.mityc.es/dgdsi/lssi/Documents/ltriptico.pdf [Visto 2011-04-30]

[45] Real Decreto-ley 14/1999, de 17 de septiembre, sobre firma electrónica. URL http://noticias.juridicas.com/base_datos/Admin/l59-2003.html [Visto 2011-04-30]

[46] David Imizcoz Etxeberria. Metodología de análisis de riesgos para abordar una certificación ISO/IEC 27001. *S21sec, Febrero 2008.* URL http://blog.s21sec.com/2008/02/metodologa-de-anlisis-de-riesgos-para.html [Visto 2011-04-30]

[47] Centro Criptológico Nacional. Esquema de certificación. *CCN-Cert, España.* URL https://www.ccn.cni.es/index.php?option=com_content&view=article&id=8&Itemid=12&lang=es [Visto 2011-04-30]

[48] Álvaro Ramón. (Ciber)ética vs tecnología. *S21sec, Abril 2009.* URL http://blog.s21sec.com/2009/04/ciberetica-vs-tecnologia.html [Visto 2011-04-30]

[49] Christopher J. Hadnagy, Mati Aharoni, James O'Gorman. Social Engineering Capture the Flag Results. *Social-Engineer.org, Defcon 18, 2010.* URL http://www.social-engineer.org/resources/sectf/Social-Engineer_CTF_Report.pdf [Visto 2011-04-30]

[50] Pete Herzog. Open Source Security Testing Methodology Manual. *ISECOM, Institute for security and open methodologies.* URL http://www.isecom.org/osstmm/ [Visto 2011-04-30]

[51] Ken Ward. Ken Ward's Mind Mastery Course - NLP Anchoring. *Trans4mind, Diciembre 2009.* URL http://www.trans4mind.com/personal_development/mindMastery/anchoring.htm [Visto 2011-04-30]

[52] Gisele Sousa Dias. La ciencia revela por qué es tan difícil olvidar un gran amor. *Clarin, sociedad, Agosto 2010.* URL http://www.clarin.com/sociedad/ciencia-revela-dificil-olvidar-amor_0_317368405.html [Visto 2011-04-30]

[53] Facebook. Estádisticas. Facebook, 2011. URL http://www.facebook.com/press/info.php?statistics [Visto 2011-04-30]

[54] Sergio Arcos Sebastián. Web-bug en Facebook. *Blog personal, 2011.* http://martes13.net/2011/03/19/web-bug-en-facebook/ [Visto 2011-04-30]

[55] Sergio Arcos Sebastián. Bypass del filtro de spam de Gmail. *Blog Personal, 2011.* http://martes13.net/2011/03/20/bypass-del-filtro-de-spam-de-gmail/ [Visto 2011-04-30]

[56] Sergio Arcos Sebastián. Infectando aplicaciones Android. *Blog personal, 2011.* http://martes13.net/2011/03/30/infectando-aplicaciones-android/ [Visto 2011-04-30]

[57] Internet. Lanzan más de diez millones de invitaciones fraudulentas en Facebook. *La vanguardia, Abril, 2011.* URL http://www.lavanguardia.es/internet/20110405/54137387007/lanzan-mas-de-diez-millones-de-invitaciones-fraudulentas-en-facebook.html [Visto 2011-04-30]

[58] Ponnurangam K. HCISec Bibliography. *July,2008.* URL http://gaudior.net/alma/biblio.html [Visto 2011-04-30]

[59] Centre de la Imatge i la Tecnologia Multimedia. Diseño Centrado en el Usuario. *Presentacion, Universitat Politècnica de Catalunya, [sesion1,parte1,pagina36]*.

[60] Simson L. Garfinkel. Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable. *S.B., Massachusetts Institute of Technology (1987)*. URL http://simson.net/thesis/ [Visto 2011-04-30]

[61] Adrian Crenshaw. Programmable HID USB Keystroke Dongle: Using the Teensy as a pen testing device. *Defcon, IronGeek, 2010.* URL http://www.irongeek.com/i.php?page=security/programmable-hid-usb-keystroke-dongle [Visto 2011-04-30]

[62] Internet. USB Rubber Ducky. *Hak5, 2010.* URL http://www.hak5.org/forums/index.php?showforum=56 [Visto 2011-04-30]

[63] Ross Anderson, Mike Bond, and Steven J. Murdoch. Chip and spin. *Computer Laboratory, University of Cambridge, JJ Thompson Avenue, CB3 0FD, UK, March 2006*. URL http://www.chipandspin.co.uk/problems.html#middle [Visto 2011-04-30]

[64] Luis Argente Castro. ¿Me pone un Mac? De los baratos si puede ser. Blog Personal, Abril, 2011. URL http://macbarato.blogspot.com/ [Visto 2011-04-30]

14 BIBLIOGRAFIA NO REFERENCIADA

[-] Marco Canepa. El show de Juanelo. *Viñetas, Juanelo.cl, 1999.* URL http://www.juanelo.cl/ [Visto 2011-04-30]

[-] Dale Pearson. Head Hacking: The Magic of Suggestion and Persception. *Presentación #hashdays 2010*. URL http://www.youtube.com/watch?v=Gd0Q56iJOL0 [Visto 2011-04-30]

[-] Kevin D. Mitnick, Emmanuel Goldstein. The Next HOPE (2010), Social Engineering. *Presentación H.O.P.E. 2009*. URL http://www.youtube.com/watch?v=Ubz9RNqBzu4 [Visto 2011-04-30]

[-] Juan Muñoz, Fèlix Vázquez, Fran Elizabarreta, Francisco Tirado. Influència Social i Grups. *Clases en la Universitat Autónoma de Barcelona, Psicología.*

[-] Paul Stoffregen, Robin Coon. Teensy Board Development. *PJRC Store, 2011.* URL http://www.pjrc.com/teensy/teensyduino.html [Visto 2011-04-30]

[-] Michale Murray. NLP for Social Engineers. *Michale Murray and Associates, 2010.* URL http://episteme.ca/nlp-for-social-engineers/ [Visto 2011-04-30]

[-] Stuart Schechter. Common Pitfalls in Writing about Security and PrivacyHuman Subjects Experiments, and How to Avoid Them. *Microsoft Research, SOUPS, 2010.* URL http://cups.cs.cmu.edu/soups/2010/howtosoups.pdf [Visto 2011-04-30]

[-] Cristian Borghello. El arma infalible: la Ingeniería Social. *ESET, LLC 610 West Ash Street, Suite 1900, Abril 2009.*

[-] ICO. Framework code of practice for sharing personal information. *Information Commissioner's Office, Octubre, 2007.*

[-] Logan. Information Gathering Report. *Social-Engineer.org, 2010.*

[-] Abel Lozano Prieto. Análisis teórico-experimental detécnicas y herramientas dephishing y delitos electrónicos. *Proyecto Final de Carrera, Universidad Carlos III de Madrid, Octubre 2009.*

[-] Yusef Hassan Montero y Sergio Ortega Santamaría. Informe APEI sobre usabilidad. *Ministerior de Cultura, Informe APEI 3, 2009.*

[-] Audun Jøsang. Human Factors in IT Security. *NISNet Winter School, Finse, April 2010.* URL http://www.nisnet.no/filer/Finse10/HumFact-ITSec.pdf [Visto 2011-04-30]

[-] Internet. Scams Message Board. *Marquis Communications Inc, 2004.* URL http://www.scam.com/ [Visto 2011-04-30]

[-] Internet. Common Vulnerabilities and Exposures. *Mitre, 2011.* URL http://cve.mitre.org/ [Visto 2011-04-30]

[-] Kevin Poulsen. Hackers Assault Epilepsy Patients via Computer. *Wired Magazine, March, 2008.* URL http://wired.com/politics/security/news/2008/03/epilepsy [Visto 2011-04-30]

[-] Internet. Russian Business Network. *2010*. URL http://rbnexploit.blogspot.com/ [Visto 2011-04-30]

[-] ANONYMOUS. AnonOps Communications. *2011*. URL http://anonops.blogspot.com/ [Visto 2011-04-30]

[-] Asociación. Sell de Çonfianza Online. *Pl. Manuel Gómez Moreno, s/n. 28020 Madrid, 2011.* URL http://www.confianzaonline.es/ [Visto 2011-04-30]

[-] David Kennedy. SecManiac: Home of the Social-Engineer Toolkit. *Blog Personal, 2010*. URL http://www.secmaniac.com/download/ [Visto 2011-04-30]

[-] Wade Alcorn. Browser Exploitation Framework. *Bindshell.net, 2010.* URL http://www.bindshell.net/tools/beef/ [Visto 2011-04-30]

[-] Gobierno. Portal Administración electrónica. *Ministerio de política territorial y administración pública, 2011.* URL http://administracionelectronica.gob.es/ [Visto 2011-04-30]