Universidad Internacional de La Rioja (UNIR)

ESIT

Máster universitario en Seguridad Informática

Desarrollo de Aplicación Web Open Source SETA (Security Education and Training Awareness) para Phishing

Trabajo Fin de Máster

presentado por: Iglesias Touceda, Óscar

Director/a: Pimienta García, Víctor Andrés

Ciudad: Madrid, España
Fecha: Julio 2021

Resumen

Nota: el objetivo de la contribución es construir una aplicación web open source tipo SETA (Security Education Training and Awareness), y ponerla a disposición de las organizaciones para poder incrementar su resiliencia ante eventuales ataques de ingeniería social mediante la técnica de phishing. Apoyándose en el API Python del aplicativo open source GoPhish, la herramienta hace uso del framework Django basado en Python, y permitirá de manera cómoda realizar una prueba de concepto de entrenamiento de concienciación de phishing para una organización, cubriendo los siguientes aspectos: creación y gestión de campañas de phishing simulado extremo a extremo, y entrenamientos de refuerzo y concienciación frente a ataques de ingeniería social mediante un módulo de cuestionarios de autoevaluación personalizables. La herramienta generará asimismo dos indicadores que permitirán medir el grado de madurez de la organización frente a este tipo de ataques (versus otras entidades, o comparando frente a una línea temporal).

Palabras Clave: phishing, SETA, toolkit, concienciación, ingeniería social.

Abstract

Note: contribution goal is building a SETA (Security Education and Training Awareness) open source web application so that organizations can increase their resilience against unexpected social engineering attacks using the phishing technique. Leveraging the GoPhish open source tool python API, the application will use the web development framework Django based on Python, and it will allow making a proof of concept of simulated phishing awareness campaigns end-to-end on a handy manner, covering all relevant aspects: creation and management of phishing simulated campaigns, and training employees through tailored self-assessment questionnaires. The web application will create also two KPIs which will show the security awareness maturity level of a given company against those attacks, measuring values against another organisations or throughout time.

Keywords: phishing, SETA, toolkit, awareness, social engineering.

Agradecimientos:

A Laura, por ser mi faro en la niebla, y darme una hija maravillosa.

A Vera, por enseñarme que se puede ser feliz con nada.

A mi hermano Borja, porque no me olvido de ti a pesar de la distancia.

Y por supuesto a mis padres, por habérmelo dado todo sin pedir nada a cambio. Ahora lo comprendo, pero nunca podré agradecerlo lo suficiente.

Contenido

Índice de ilustraciones

Índice de tablas

1. Introducción

1.1 Presentación

En el ámbito de los Sistemas de Información, la ingeniería social consiste en el arte de embaucar o manipular a otras personas con el objetivo de sacar beneficio propio de dicha situación. Bien sea haciéndoles revelar información privada, o que de manera inadvertida ejecuten software en un sistema que eventualmente comprometa alguna de las dimensiones habituales de la Seguridad de la Información (confidencialidad, integridad, disponibilidad, trazabilidad y no repudio).

La Seguridad de la Información de un sistema entendida desde el punto de vista holístico, es tan alta, como la de su componente más débil. Como se verá en las siguientes secciones, por diversas causas el ser humano suele desempeñar este último papel, y por tanto ser el objetivo primordial de los ataques de usuarios o entidades maliciosas.

La ingeniería social es una técnica que puede llevarse a cabo en cualquier fase de un ataque, es decir, durante la búsqueda de vulnerabilidades, o para ejecutar la explotación en sí misma. Existen diversas familias de técnicas que pueden emplearse para ello, como recogen los manuales de Análisis de Vulnerabilidades de Martínez (2019):

- Pasivas: consisten habitualmente en la mera observación de un usuario para recabar información confidencial (por ejemplo “shoulder surfing”).
- No presenciales (haciendo uso del email, correo ordinario, teléfono, fax).
- Presenciales no agresivas (conversando con desconocidos, haciendo uso de técnicas de vigilancia, seguimiento, empleo de acreditaciones falsas o tácticas de desinformación para conseguir un objetivo).
- Presenciales agresivas (suplantación de identidad, chantaje o extorsión, despersonalización mediante sustancias tóxicas como alcohol o drogas, y/o presión psicológica).

En el trabajo de Salahdine, F. & Kaabouch, N. (2019), se describen las principales tipologías de ataques de ingeniería social que predominan actualmente:

Ilustración 1. Ataques de ingeniería social (Salahdine & Kaabouch, 2019)

De entre todas las variantes descritas, las más ampliamente utilizadas son aquellas que se pueden ejecutar remotamente, por varios motivos:

- La facilidad de comisión de este tipo de ataques. En general, su dificultad técnica es baja con las herramientas actuales.
- Su componente internacional. Las víctimas se encuentran ubicadas en cualquier lugar del globo.
- La falsa sensación de impunidad asociada a la lejanía física con respecto a las víctimas potenciales. Al ser ejecutados desde países con legislaciones más permisivas o medios técnicos de rastreo inferiores, complica las labores de investigación por parte de las autoridades competentes (Interpol, Europol, etc).

Dentro del subconjunto de ataques de ingeniería social realizados de manera remota, actualmente uno de los predominantes, es la técnica de *phishing*. Ésta consiste en suplantar

la identidad legítima de un tercero, para que un usuario realice una acción de manera inadvertida que favorezca al atacante. Generalmente se lleva a cabo mediante correo electrónico, por ser un mecanismo muy simple, barato, y normalizado para la gran mayoría de usuarios (casi la totalidad de las personas con conexión a Internet hoy en día poseen al menos una identidad de correo electrónico). El objetivo será engañar al usuario para, desde redirigirle a un formulario en el que obtener credenciales de acceso a servicios o bancarias, hasta vulnerar el propio sistema mediante la instalación de malware, keyloggers, exfiltrar información o infectar otros sistemas en la misma red, etc.

Según el informe ISTR 2019 de Symantec (2019), el número de usuarios que recibieron intentos de phishing en el año anterior fue de casi un 30%. Y en los ataques dirigidos, el vector de ataque principal (65% de los casos), fue el empleo de esta técnica para el robo de credenciales.

1.2 Motivación

El objetivo del presente Trabajo Fin de Master, será intentar mitigar las consecuencias de los ataques de ingeniería social mediante la técnica de phishing, centrándonos principalmente en entornos corporativos.

Los riesgos asociados a dichos ataques son altos, puesto que si bien es cierto que los ataques de otra índole que intentan asaltar las barreras electrónicas (mecanismos de identificación de amenazas, monitorización o vigilancia como pueden ser firewalls, IDS u otros) pueden ser mejorados de manera sistemática mediante acciones de mantenimiento tales como actualizaciones de reglas o bases de datos de firmas (por ejemplo), recablear el cerebro humano para hacer a las personas resilientes a los ciberataques de ingeniería social, no es posible (al menos de forma programática).

Adicionalmente, aunque en España desde instituciones como INCIBE se hacen denodados esfuerzos para divulgar y difundir la cultura de ciberseguridad en todas las capas de la sociedad (desde el mundo empresarial hasta los hogares), el grado de conocimiento y concienciación en general en este ámbito, es más bien todavía escaso.

Los ataques de ingeniería social no van a desaparecer, sino que tienden a ir a más. Según el Anti-Phishing Working Group (2020):

- Los impulsores de estos ataques (los llamados ingenieros sociales), se centran en aquellos aspectos que son más susceptibles de resultar atractivos a las personas

(actualmente, lo relativo a la COVID-19, instituciones sanitarias, o prestaciones por desempleo, por ejemplo).

- El crecimiento de websites maliciosos de phishing entre enero'20 y marzo'20, ha sido del 9.6%.
- El crecimiento de la actividad de phishing en un país con gran afectación pandémica como Brasil en Q1 de 2020, ha sido del 24%.
- Los servicios objetivo predilecto de los hackers, son webmail y soluciones Software-as-a-Service, tipo Google Suite u Office 365 (de uso masivo por la sociedad, y que alberga gran cantidad de información confidencial de los individuos), suponiendo un tercio del total del universo de targets, como se indica en la figura:

Ilustración 2. Sectores objetivo de campañas de phishing Q1-2020 (APWG, 2021)

Por otra parte, si nos focalizamos en los perjuicios económicos que pueden causar las brechas de seguridad asociadas a este tipo de ataques (al margen de problemas reputacionales o regulatorios asociados a eventuales sanciones), vemos en el reporte anual de Coste del Cibercrimen (Accenture, 2019), que el phishing + malware (este último puede ser incluido en adjuntos en los emails fraudulentos), representan el problema principal:

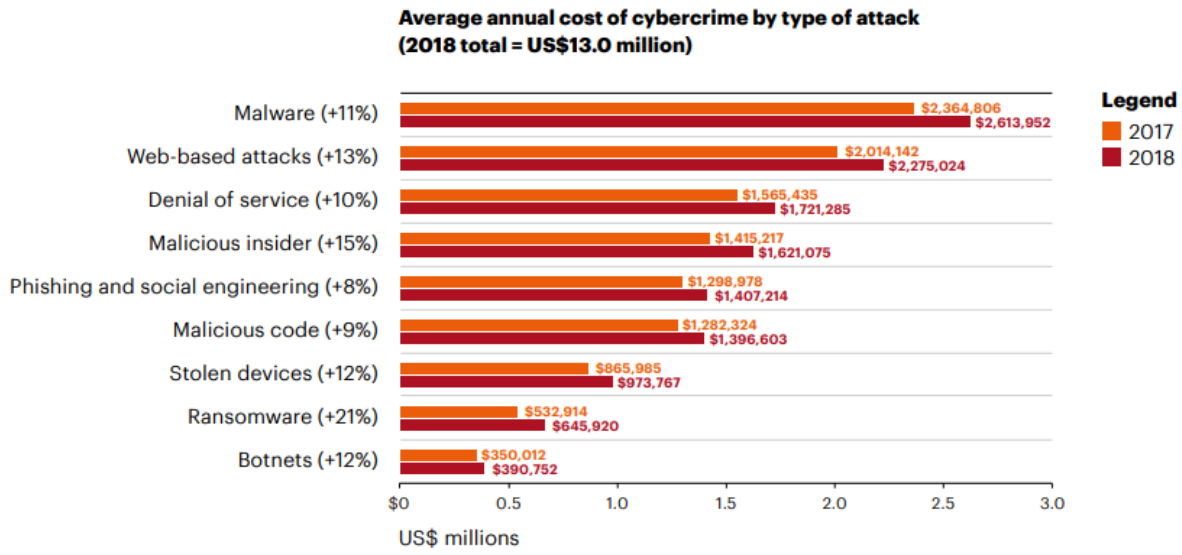

Ilustración 3. Coste anual del cibercrimen por tipo de ataque (Accenture, 2019)

1.3 Objetivos generales

Para mitigar en la medida de lo posible los efectos indeseados de los ataques de ingeniería social mediante la técnica de phishing, podemos observar un framework de gestión de la Seguridad de la Información que utilice los drivers de negocio para guiar las actividades de ciberseguridad dentro de las compañías, y considere los riesgos asociados a la tecnología como parte fundamental del proceso de gestión de riesgos general. Hablamos del Cybersecurity Framework del Instituto Nacional de Estándares y Tecnologías de Estados Unidos (NIST, 2014).

En el Core de dicho Framework, se definen un conjunto de actividades de ciberseguridad y resultados esperados, comunes a las organizaciones de cualquier sector. Se indican una serie de estándares de la industria y guías de buenas prácticas, que permiten la adecuada ejecución y comunicación de actividades de ciberseguridad a lo largo y ancho de una compañía, desde la capa ejecutiva, a la de implementación y operaciones. El Core define cinco Funciones concurrentes y continuas (que se descomponen posteriormente en Categorías y Subcategorías), que cuando se consideran como un todo, proporcionan una visión estratégica y sistemática para la gestión adecuada del ciclo de vida y las operaciones de gestión de riesgos de ciberseguridad de las organizaciones.

Las cinco Funciones y sus Categorías asociadas, se presentan a continuación:

Function Unique Identifier	Function	Category Unique Identifier	Category
ID	Identify	ID.AM	Asset Management
		ID.BE	Business Environment
		ID.GV	Governance
		ID.RA	Risk Assessment
		ID.RM	Risk Management Strategy
		ID.SC	Supply Chain Risk Management
PR	Protect	PR.AC	Identity Management and Access Control
		PR.AT	Awareness and Training
		PR.DS	Data Security
		PR.IP	Information Protection Processes and Procedures
		PR.MA	Maintenance
		PR.PT	Protective Technology
DE	Detect	DE.AE	Anomalies and Events
		DE.CM	Security Continuous Monitoring
		DE.DP	Detection Processes
RS	Respond	RS.RP	Response Planning
		RS.CO	Communications
		RS.AN	Analysis
		RS.MI	Mitigation
		RS.IM	Improvements
RC	Recover	RC.RP	Recovery Planning
		RC.IM	Improvements
		RC.CO	Communications

Ilustración 4. Framework Ciberseguridad NIST (NIST, 2014)

La propuesta de este Trabajo Fin de Master, es incrementar el grado de resiliencia de las organizaciones ante ataques de ingeniería social mediante técnicas de phishing por correo electrónico, mediante el desarrollo de una aplicación web que, ubicado en la categoría de Awareness y Training dentro de la función de Protección de activos del framework NIST presentado anteriormente, permite a los gestores de una compañía ejecutar acciones periódicas de concienciación en dos planos:

- Práctico: ejecución de campañas de phishing simuladas, con diferente grado de verosimilitud, para medir el nivel de concienciación de sus empleados con respecto a esta tipología de ataques de ingeniería social.

- Teórico: lanzamiento de cuestionarios/encuestas personalizadas de manera sencilla, de tal modo que se pueda evaluar el conocimiento de los integrantes de la organización también a nivel teórico, de manera totalmente adaptada al contexto empresarial concreto.

Ambas acciones podrían lanzarse conjuntamente o de forma independiente, generando un par de indicadores de interés (Training-score y Awareness-Score, se describirán posteriormente) para su seguimiento en el tiempo.

Gracias a esta herramienta, las organizaciones se podrían beneficiar en varios aspectos:

- Disminuir los costes asociados a estos ciberejercicios, dado que la herramienta se licencia como software libre, de sencilla instalación y uso intuitivo.
- Mejorar su grado de resiliencia ante esta tipología de ataques, disminuyendo el nivel de ciber-riesgo general.
- De ser adoptada masivamente, el posible perjuicio total causado asociado a brechas de seguridad vinculadas al vector de ataque mencionado en el conjunto de la industria, se vería igualmente reducido.

De manera más concreta, el alcance de lo que se propone realizar en el TFM, es:

- Analizar el estado del arte, estudios realizados, etc., en materia de concienciación de seguridad frente a ataques de ingeniería social.
- Desarrollar una herramienta que pueda ser utilizada para incrementar el nivel de ciber-resiliencia de las organizaciones ante este tipo de ataques, concretamente mediante la técnica de email phishing.
- Generar métricas asociadas a dicha herramienta, que permitan medir el nivel de concienciación de las compañías, y evaluar el grado de eficacia del propio aplicativo.
- Realizar una prueba de concepto de la aplicación.
- Hacer pública y disponible la solución a cualquier organización interesada de manera gratuita, como software open source.

1.4 Estructura del documento

Se describe a continuación la estructura del presente documento:

- **Capítulo 1. Introducción**. Se presenta el trabajo, su motivación, y objetivos generales a alcanzar.

- **Capítulo 2. Estado del arte**. Se mencionan los orígenes de la ingeniería social, condicionantes de la psicología humana que favorecen dichos ataques, y contexto actual, además de un análisis de trabajos anteriores similares, y la descripción a modo de visión general del panorama de soluciones actuales en el mercado frente a esta problemática.
- **Capítulo 3. Objetivos y metodología del trabajo**. Se detalla el objetivo global del Trabajo Fin de Máster, y se describe someramente el proceso llevado a cabo para su elaboración.
- **Capítulo 4. Desarrollo específico de la contribución**. Se describe de manera pormenorizada la solución propuesta, desde la identificación de sus requisitos funcionales, pasando por el funcionamiento de sus distintos componentes, hasta las pruebas de evaluación realizadas.
- **Capítulo 5. Conclusiones y trabajos futuros**. Incluye a modo de resumen, las conclusiones del TFM realizado, y posibles líneas futuras de investigación o actuación.
- **Capítulo 6. Referencias**. Contienen las referencias bibliográficas utilizadas a lo largo de la elaboración del Trabajo.
- **Capítulo 7. Anexos**. Recoge información extra y evidencias que se considera relevante mencionar, como pueden ser ficheros de prueba empleados, capturas de pantalla del aplicativo, emails generados, etc.

2. Contexto y estado del arte

2.1 Introducción

"Si no te conoces a ti mismo ni a tu enemigo, sucumbirás en todas las batallas" (Sun-Tzu, Griffith, 1964, p.11).

También algunos atribuyen a Albert Einstein, la cita apócrifa de que sólo hay dos cosas infinitas en el universo, el universo y la estupidez humana, pero que no estaba seguro de lo primero.

El incremento exponencial de la digitalización de la sociedad y el uso de servicios electrónicos, genera una creciente exposición a riesgos por parte de los usuarios. El grado de interconexión y el número de dispositivos conectados a Internet, es cada vez mayor desde la introducción del iPhone en 2007, hito que cambió el curso de la historia en este sentido.

Tampoco ayuda a mitigar el problema el hecho de que el perímetro de las organizaciones se difumina con la introducción de estos dispositivos y las políticas BYOD (Bring Your Own Device), que hacen que los equipos estén dentro o fuera de la red corporativa indistintamente, y se entremezclen las esferas personales y profesionales en un mismo equipo. Esto facilita las cosas enormemente a los ciberdelincuentes.

Todo lo anterior, no hace más que acrecentarse con tecnologías relativamente recientes, como es la introducción del Internet de las Cosas (IoT) y últimamente las redes de comunicaciones móviles de última generación (5G), que gracias a IPv6 hacen pensar en una explosión extra de número de equipos conectados a la red de redes.

En este sentido, los avances tecnológicos mencionados junto con nuevas técnicas como la inteligencia artificial o el machine learning, van generando vulnerabilidades en aplicaciones, dispositivos y redes (de diseño, configuración o implementación), que claramente van por delante de la capacidad de respuesta de los profesionales de la Seguridad, que se están viendo abrumados y en la necesidad de generar mecanismos de respuesta y defensa de Activos de Información semiautomáticos.

Pero al margen de esta carrera armamentística en lo tecnológico, no podemos olvidarnos de nuevo de los seres humanos. El escenario descrito (mayor número de víctimas potenciales, con plataformas más estandarizadas pero además un bajo conocimiento medio en cuestiones de Ciberseguridad), es el caldo de cultivo perfecto para el cibercrimen. Sus autores intentarán aprovecharse de nuestros sesgos cognitivos para, haciendo uso del camino de mínima resistencia que constituyen las personas, obtener un rédito (principalmente económico, pero también de otra índole como ventajas competitivas o mejor posición geoestratégica, etc.), mediante ataques de ingeniería social.

En su libro El Arte del Engaño (Mitnick, 2002), el popular hacker estadounidense hace referencia en su introducción a esta realidad, en algunos de sus párrafos:

> Una compañía puede tener la mejor tecnología de seguridad que el dinero puede comprar, entrenar tan bien a su plantilla que cierran bajo llave todos sus secretos cuando se van a casa de noche, y contratar a vigilantes de seguridad de la mejor firma del sector.
>
> Aun así, esa compañía sigue siendo totalmente vulnerable.
>
> Los individuos pueden seguir todas las mejores prácticas y recomendaciones de los expertos en Ciberseguridad, instalar sistemáticamente todos los productos de seguridad recomendados, y ser absolutamente diligentes en la configuración adecuada de sus sistemas y la aplicación de todos los parches de seguridad sugeridos.
>
> Esos individuos son todavía completamente vulnerables.

> (…)
>
> ¿Por qué? Porque el factor humano es verdaderamente el punto débil de la seguridad.
>
> (…)
>
> Como indica el conocido consultor de seguridad Bruce Schneier, “La seguridad no es un producto, es un proceso”. Además, no es un problema tecnológico, sino de gestión y de personas.
>
> Mientras que los fabricantes desarrollan sin cesar nuevas tecnologías de seguridad, que dificultan la explotación de vulnerabilidades técnicas, los atacantes se enfocan cada vez más en las personas. Vulnerar un firewall humano es con frecuencia muy sencillo, no requiere inversión más allá del coste de una llamada telefónica, e implica un riesgo mínimo.

Al último pasaje añadiríamos el envío de un email en la técnica de phishing, algo que resulta ser completamente gratis para los delincuentes.

En el libro Introduction to Social Engineering de Ozkaya (2018), se menciona un caso real que ejemplifica a la perfección todo lo indicado anteriormente. En 2015, la compañía californiana de hardware de comunicaciones Ubiquiti Networks sufrió un ataque de phishing en el que, tras un análisis de inteligencia por parte de los ciberdelincuentes, consiguieron hacerse con información suficiente acerca del CEO de la compañía como para poder suplantar su identidad de cara a otro empleado de alto rango mediante correo electrónico. Persuadieron a este último mediante solicitudes ilegítimas para realizar transferencias bancarias desde una cuenta hongkonesa a países como Rusia, China, Hungría y Polonia. Se llegaron a realizar 14 transacciones en 17 días, hasta que el FBI detectó el incidente y se contuvo el problema. El saldo final de la brecha fue de 46.7 millones de dólares.

El anterior, era un ataque dirigido. En cuanto a los ataques masivos o automatizados, en la misma obra (Ozkaya, 2018) se muestra una figura basada en el Data Breach Report de Verizon de 2015, que muestra la gravedad del problema en cuanto a la falta de concienciación de las personas acerca del impacto que en la Seguridad de la Información tienen este tipo de ataques:

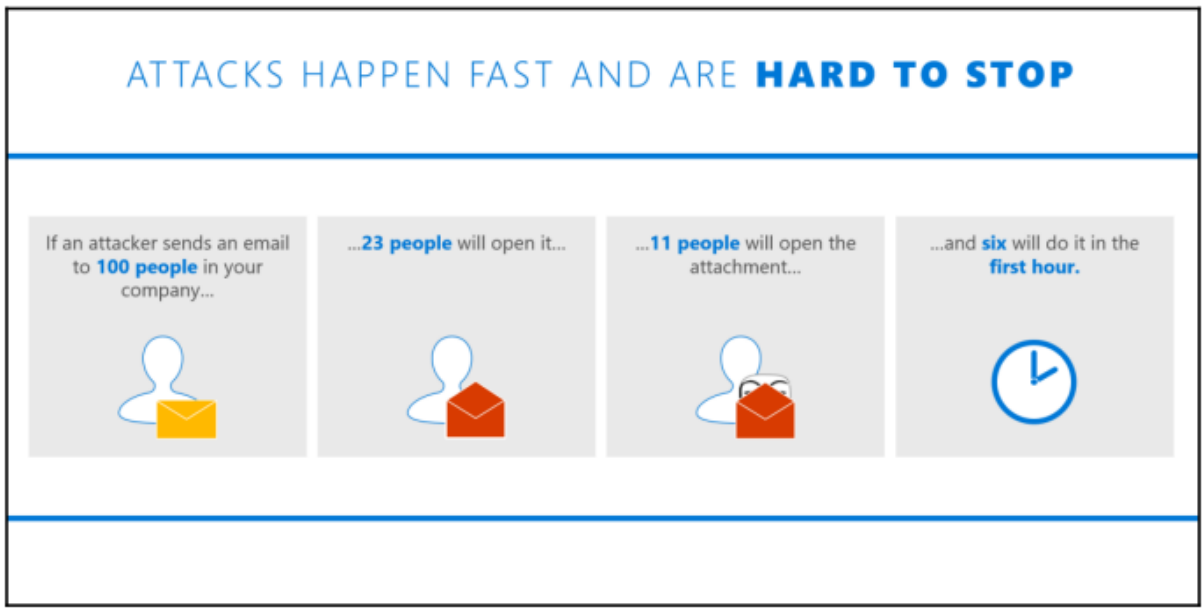

Ilustración 5. Facilidad de empleo de ingeniería social mediante phishing (Ozkaya, 2018)

Como se puede apreciar en la figura anterior, la susceptibilidad de las personas a ser víctimas de este tipo de ataques, es elevadísima. Además de nuestra curiosidad innata y otra serie de rasgos cognitivos que se verán más adelante, hay estudios que conjeturan acerca de cuáles son los motivos que provocan esta reacción por parte de los humanos, o qué factores pueden provocar una susceptibilidad mayor.

En Algarni, Xu, & Chan (2017), se realiza un estudio empírico de susceptibilidad a técnicas de ingeniería social en Facebook. Se analizan trece propiedades de un sujeto, que mapean a cuatro niveles de percepción y dan lugar al grado de susceptibilidad de victimización de un tercero mediante un ataque de ingeniería social, protagonizado por el individuo descrito anteriormente.

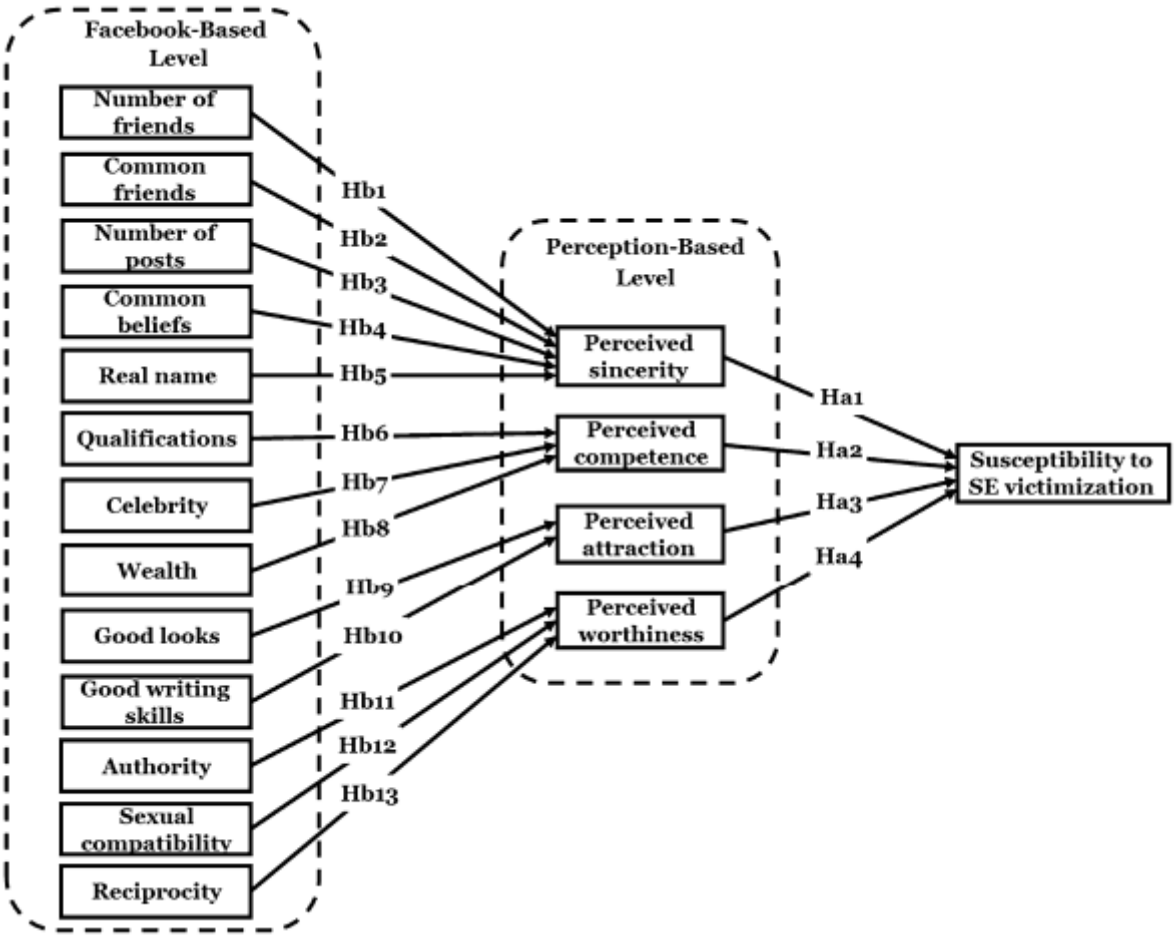

Ilustración 6. Dimensiones de credibilidad de fuente de ingeniería social en Facebook (Algarni, Xu, & Chan, 2017)

Las conclusiones son que las dimensiones que más impactan en la credibilidad, son, por este orden: sinceridad (basado en número de contactos, amigos comunes, y grado de afinidad principalmente), grado de competencia percibido, atracción, e interés propio. Esto es conocido y será empleado por los ingenieros sociales, obviamente.

Este exceso de confianza en la sinceridad de terceros, es otra de las causas principales de susceptibilidad a ataques mediante phishing, como se sugiere en la investigación de Albladi & Weir (2020) que apunta a otros estudios específicos sobre phishing en la misma línea.

Por si todo lo anterior no es suficiente para entender el fuerte impacto que en la Seguridad de la Información pueden tener las acciones de Ingeniería Social sobre las personas, en Hadnagy (2010) se recogen algunas estadísticas relevantes:

- Según la información del equipo de prevención de pérdida de datos de Symantec, 1 de cada 500 emails contiene información confidencial.
- El 62% de los incidentes que ponen en riesgo información sensible, están relacionados con robo o suplantación de identidades.

- El 32% de los empleados de las compañías, desconoce las políticas de protección de datos en vigor (añadimos: entre las que se incluyen aquellas destinadas a la prevención de ataques de email phishing).

2.2 Antecedentes

2.2.1 Historia de la ingeniería social

Es interesante conocer el camino recorrido desde sus inicios, de las técnicas de ingeniería social en Ciberseguridad. En Wang, Sun & Zhu (2010), se detalla cronológicamente la evolución en este ámbito desde el último cuarto del siglo pasado.

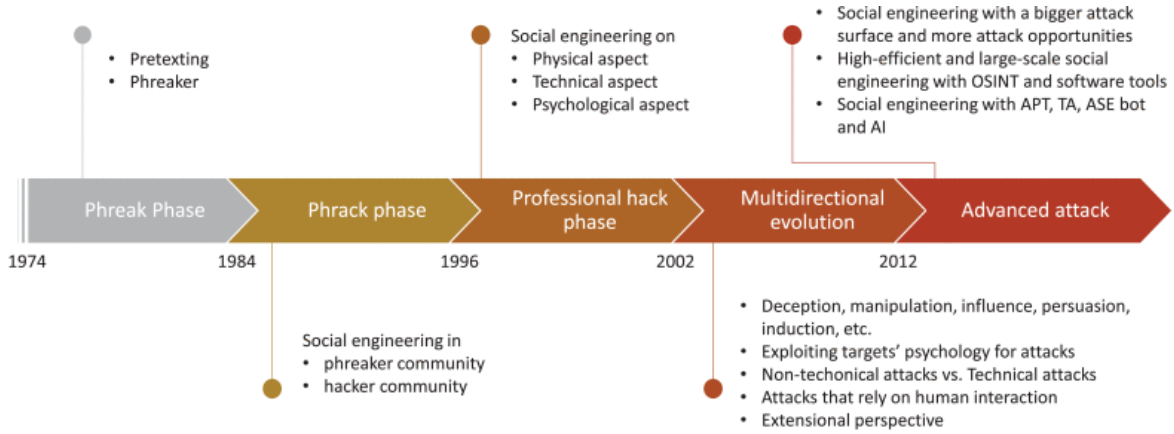

Ilustración 7. Evolución conceptual de la ingeniería social en Ciberseguridad (Wang, Sun & Zhu, 2010)

Como se ve, básicamente la historia de la ingeniería social en Ciberseguridad se puede dividir en cinco etapas:

- Fase phreack (1974-84). En un artículo llamado *More on Trashing* en la revista *The Hacker´s Quarterly*, aparece por primera vez acuñado el término ingeniería social. Está relacionado con la información confidencial y accionable que podía encontrarse en la basura de los gigantes de las telecomunicaciones, como Bell Telephone Company. En otro artículo en la misma revista, se comenta la posibilidad de suplantar la identidad de un trabajador del sector telco para engañar a las operadoras de los centros de conmutación, y obtener información confidencial de usuarios. Posteriormente se descubre que ya existían comunidades desde aproximadamente 1974 utilizando estas técnicas de impersonación y persuasión.
- Fase phrack (1984-1996). Se genera un híbrido entre estas tretas en el ámbito de las compañías telefónicas, que continúan, y el descubrir que mediante engaños a las personas es mucho más sencillo comprometer sistemas que, por ejemplo, rompiendo

contraseñas mediante fuerza bruta (máxime con la potencia de cómputo de la época). Se utilizan técnicas como provocar un fallo de red, y llamar al "cliente" haciéndose pasar por integrante del equipo de Soporte, para sonsacarle información confidencial. La ingeniería social amplía sus técnicas y alcance potencial.

- Fase del hack profesional (1996-2001). Se comienzan a profesionalizar los ataques, y los presenciales ganan más relevancia (trashing, shoulder surfing, talking, tailgating…). Aparecen los primeros fraudes por Internet, los troyanos, y el primer ataque de phishing en 1996, haciéndose pasar por el equipo de Soporte de AOL (América Online), para robar credenciales y datos bancarios de los usuarios. Comienzan a utilizarse técnicas de subversión psicológica, al reconocer que el usuario es, de largo, el eslabón más débil de la cadena de Seguridad.
- Evolución multidireccional (2002-2012). Surgen trabajos como los de Mitnick & Simon (2002), el concepto ingeniería social adquiere gran relevancia, y comienzan a publicarse numerosos artículos y estudios acerca de la psicología y confianza sociales, psicología del lenguaje, emocional, y de las expresiones.
- Ataques avanzados (2012+). La base potencial de víctimas es gigantesca debido al crecimiento de Internet y la hiperconexión, y se democratiza el acceso a tecnologías que implementan técnicas de ingeniería social. Se desarrollan herramientas que facilitan enormemente las tareas como Maltego para OSINT, Simple Phishing Toolkit, GoPhish, o Social-Engineering Toolkit (SET), que soportan múltiples vectores de ataque. Combinadas con técnicas como big data, machine learning e inteligencia artificial, posibilitan realizar ataques altamente eficientes. El uso masivo de las redes sociales agudiza el problema, multiplica la superficie de ataque y la cantidad de datos personales expuestos. La eficacia de los cibercriminales en sus ataques, es más alta que nunca.

En Singh (2013) se muestran algunas de las herramientas que, como se indicó en la última etapa de la evolución histórica de la ingeniería social, han simplificado enormemente la ejecución de ataques, como es el Social-Engineering Toolkit (SET). Entre otras cosas, permite realizar ataques de spear-phishing incluso clonando sitios web, para simulando ser páginas legítimas, llevar a cabo el robo de credenciales.

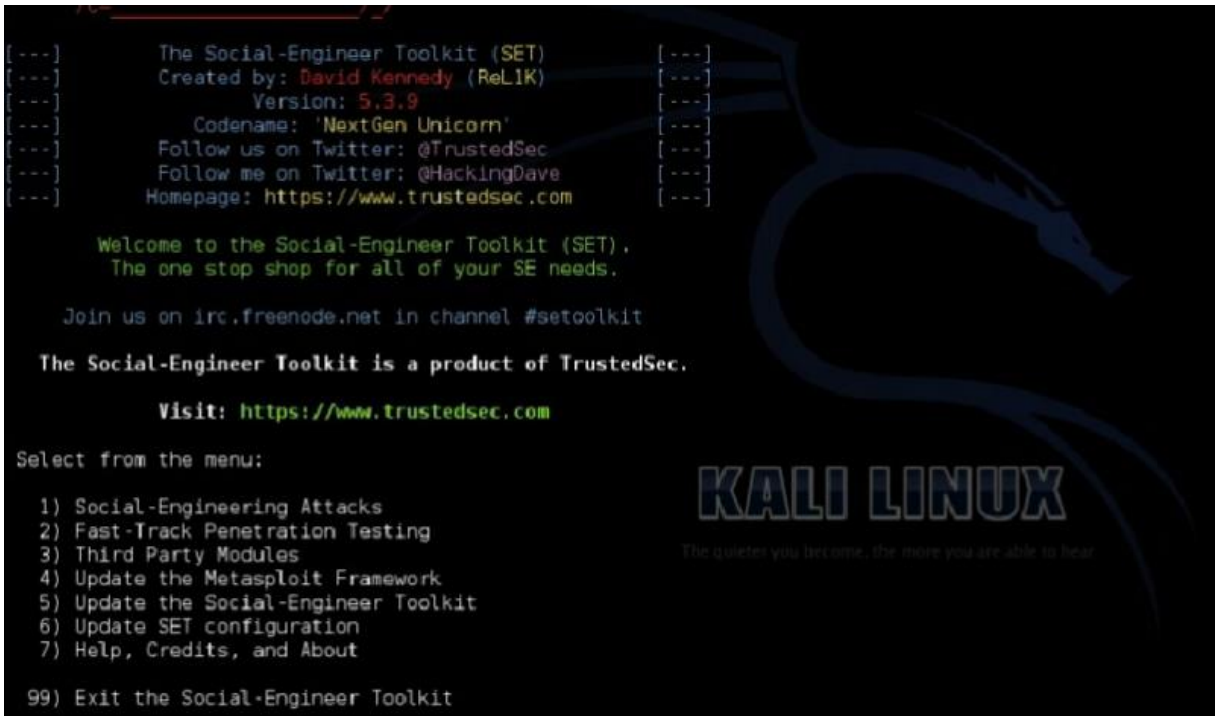

Ilustración 8. Lista de ataques disponibles en SET (Singh, 2013)

Otro claro ejemplo lo encontramos en Maltego (anteriormente mencionado), que permite mediante técnicas de OSINT recabar información de la víctima para poder preparar ataques personalizados, como se ve en Ozkaya (2018):

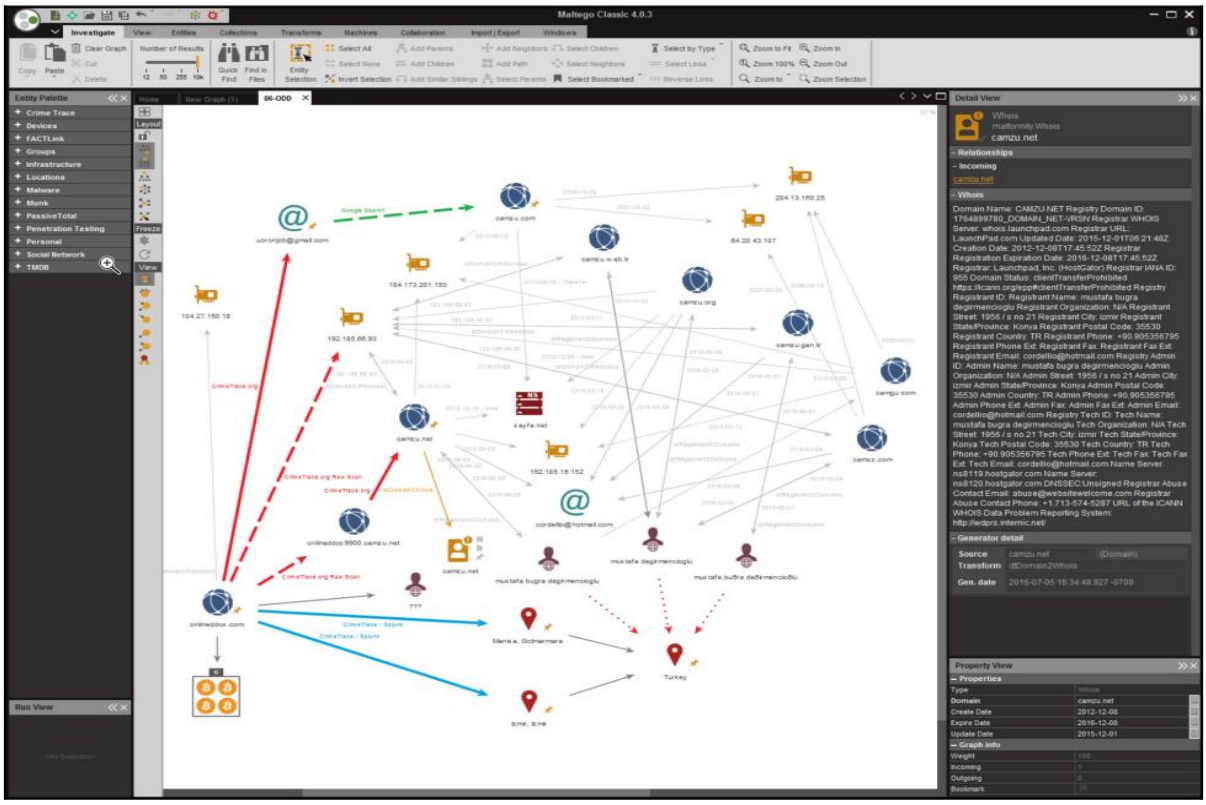

Ilustración 9. Diagrama de relaciones en Maltego (Ozkaya, 2018)

2.2.2 Framework de ataques y modelo ontológico

En cuanto al modus operandi de los ingenieros sociales, el framework de esta tipología de ataques se describe en Mouton, Leenen & Venter (2016), aunque se ilustra mejor en Ozkaya (2018).

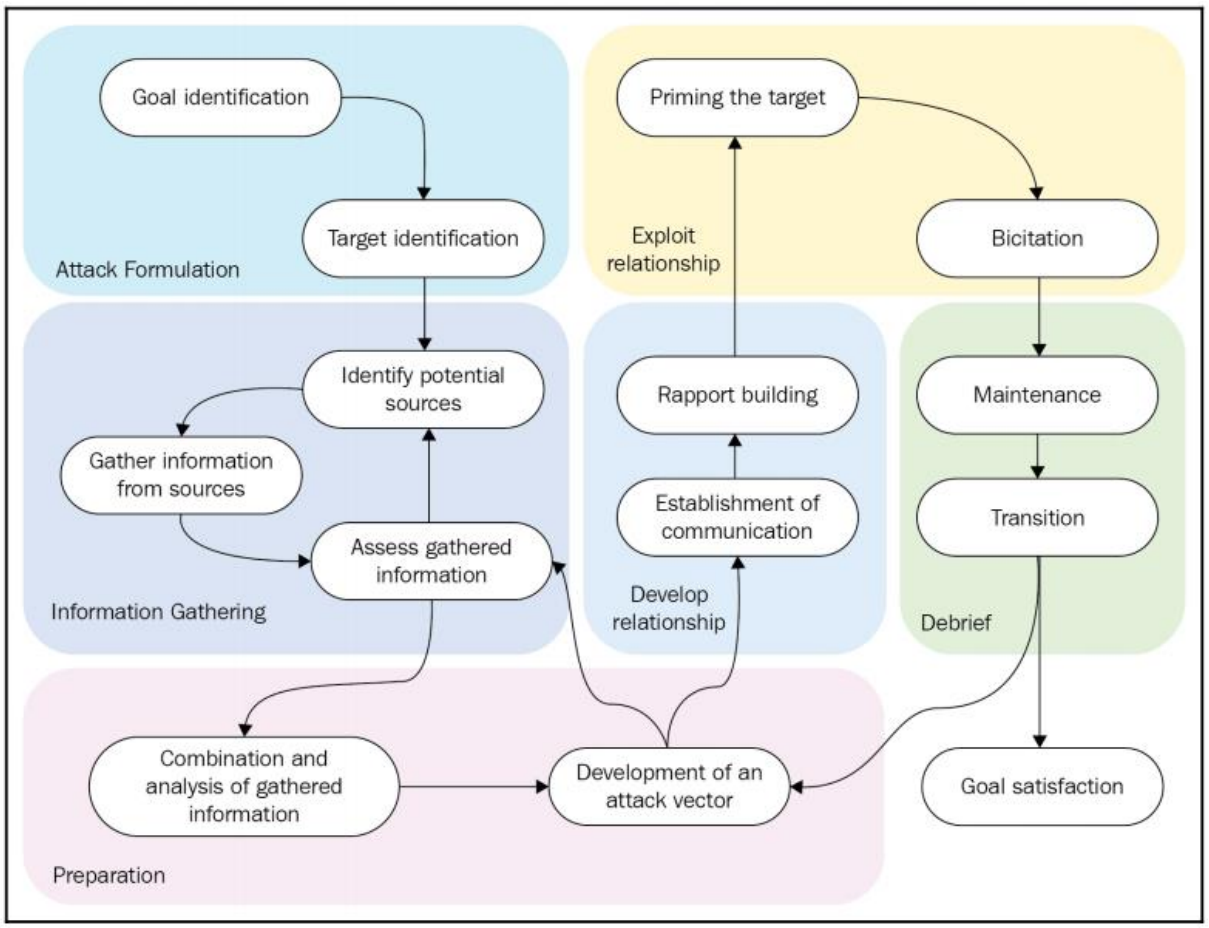

Ilustración 10. Framework de ataques de ingeniería social (Ozkaya, 2018)

Existen seis fases principales:

- **Formulación del ataque**: se identifican tanto objetivo, como víctima/s.
- **Recopilación de información**: se identifican fuentes de información, públicas o no, de los elementos de la fase anterior. Aquí se harán uso de las cientos de herramientas disponibles por ejemplo en la suite de Kali Linux, u otras desarrolladas a medida.
- **Preparación**: se combina la información obtenida en el paso anterior, y se desarrolla el vector de ataque. En esta etapa, es donde se habrán identificado ya todos los elementos del modelo ontológico:

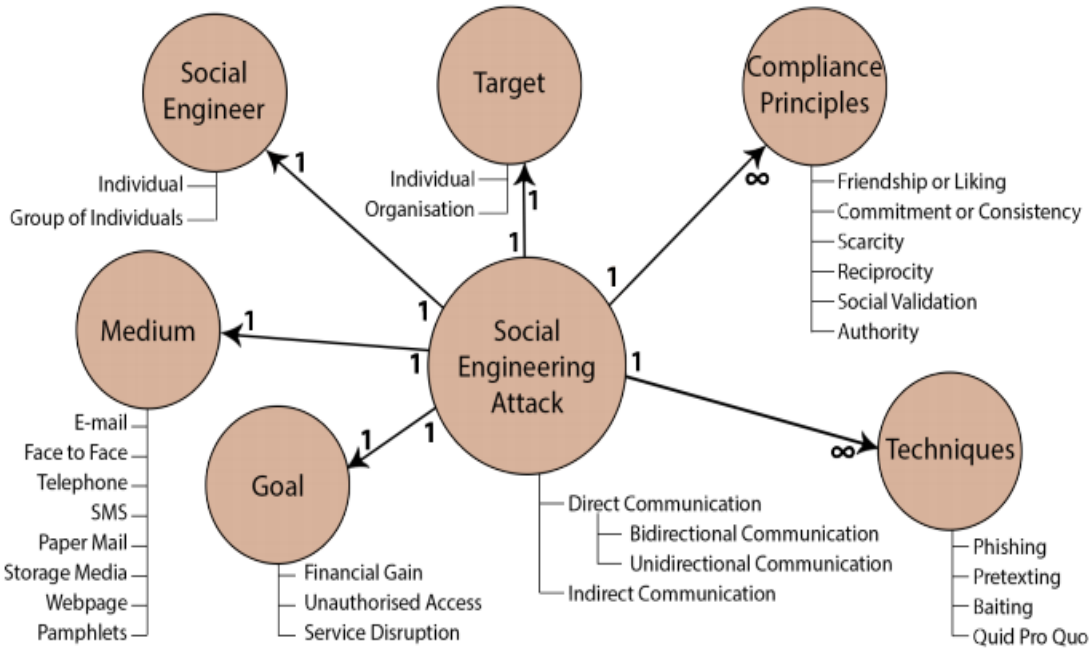

Ilustración 11. Modelo ontológico de ataque de ingeniería social (Mouton et al., 2016)

- **Desarrollo de relación**: se establece contacto con la víctima para intentar ganar su confianza. En el caso de un ataque de phishing, ese contacto será aséptico, pues al no haber interlocución directa, consistirá en un mero intento de engaño a distancia.
- **Explotación:** inducción a la víctima a realizar una acción que, de manera inadvertida, favorezca al atacante, como puede ser pinchar un enlace, abrir un adjunto, o introducir sus credenciales en un formulario supuestamente legítimo.
- **Post-Explotación:** ya sin necesidad de interaccionar con la víctima, el atacante a priori ha conseguido su objetivo y por tanto puede de manera puntual o sostenida en el tiempo, realizar acciones ilícitas.

2.2.3 Programación neurolingüística

Una vez descrito el marco histórico y el proceso habitual llevado a cabo por los ingenieros sociales para la consecución de sus objetivos, nos detendremos en las investigaciones realizadas en el marco del análisis de la psicología humana, y los principios que permiten persuadir a las personas en este contexto de ingeniería social, para la consecución de los objetivos propios.

Desde tiempos inmemoriales se ha intentado desbloquear la mente humana, encontrar esos patrones que permitan realizar un *buffer overflow* de tal modo que un sujeto se pliegue a las

exigencias de un tercero. Los motivos para ello pueden ser variados, lícitos e ilícitos, como puede ser un interrogatorio policial, etc. No entraremos en los aspectos éticos.

En Hadnagy (2010), se indica que, para alterar la conducta de un sujeto, es necesario comprender cómo estos piensan. Se considera que existen tres tipos de pensamiento: visual, auditivo, y kinestésico (asociado a los sentimientos), cada uno con diferentes submodalidades asociadas a los matices que se pueden percibir en dichos ámbitos (por ejemplo: intensidad de luz, brillo, color, tamaño de objetos, grado de enfoque, etc., afectan al modo de pensamiento visual).

En las interacciones de ingeniería social en directo, es fundamental detectar cuál es el tipo de pensamiento dominante de la víctima, normalmente en base a las microexpresiones (de su rostro u otras partes del cuerpo), para o bien conseguir el objetivo deseado (como que una recepcionista inserte un USB en su PC para de manera inocente realizar una fotocopia de manera urgente a un desconocido), o detectar engaños, y obtener información de forma indirecta.

El término programación neurolingüística (NLP) se acuña en torno a 1970. Estudia cómo las personas piensan y experimentan el mundo que les rodea, pero también es un sistema terapéutico para intentar mejorar la autoconsciencia y cambiar patrones mentales y emocionales.

Los ingenieros sociales pueden apalancarse en la NLP para refinar sus técnicas, mediante por ejemplo, la selección de las expresiones escritas que maximicen la probabilidad de éxito de un ataque de phishing. Para ello, insertarán “comandos” en los mensajes del email que irán directamente al subconsciente del individuo a hackear. De nuevo, en el capítulo 5 de Hadnagy (2010), hay multitud de ejemplos relacionados con otros ámbitos de actividad.

2.3 Trabajos relacionados

2.3.1 Kevin Mitnick

No sería justo pasar por alto a una de las referencias en ingeniería social en la actualidad. El informático norteamericano es una figura controvertida, como muchas otras en el entorno del mundo hacker. Emerge de la curiosidad por la tecnología, que le lleva a hacer cosas cuasi inocentes como suplantar la identidad de los empleados de McAuto en McDonalds a distancia en bromas a los conductores, y el reto del intento de acceso a los sistemas protegidos de grandes corporaciones, lo que le lleva a cometer delitos por los que pasaría un tiempo en

prisión. Finalmente volverá al lado de la luz, y actualmente se dedica a la consultoría. Más detalle, en un artículo autobiográfico escrito por él mismo en (Mitnick, 2019).

La importancia de éste en el ámbito de la ingeniería social se materializa principalmente en su obra inicial, El Arte del Engaño (Mitnick & Simon, 2002). La parte cuatro de dicho tratado, es fundamental.

Indica que no hay barreras electrónicas posibles contra los ataques de ingeniería social. Si son lo suficientemente sofisticados, tienen unas tasas de éxito de casi el 100%, y la única manera de mitigarlos es combinar la tecnología con políticas de seguridad que establezcan reglas claras de comportamiento por parte de los empleados en los diferentes escenarios que pueden plantearse (interacciones físicas, por email, etc.), además de un entrenamiento adecuado. Es fundamental que éste exista, y sea sostenido en el tiempo.

El concepto anterior, es piedra angular del presente TFM. Indica además que hay organismos que proponen que un 40% del presupuesto de Seguridad de la Información de las compañías, se dedique a los programas de concienciación (Función de Protección del framework NIST, como vimos).

Deben ser entrenados tanto en ser conscientes de que habrá entidades externas que tratarán de manipularles para conseguir algo a cambio, como en el conocimiento escrupuloso de las normas de actuación. Ambas cosas podrán realizarse de manera simple y personalizada, mediante la aplicación web objeto del presente Trabajo Fin de Máster.

Mitnick referencia en su libro la investigación de Robert B. Cialdini en Scientific American en 2001, que presenta seis tendencias básicas en la naturaleza humana. Los ingenieros sociales se apoyarán en ellas para conseguir el engaño:

- **Autoridad**. Las personas tienden a acceder a peticiones realizadas por personas con un rango de autoridad elevado. Por ello, probablemente realicen lo que el hacker les solicita, si consigue simular que la petición proviene de un conocido de mayor rango.
- **Afinidad.** Tendemos a complacer a terceros con los que hay cierta afinidad (comparte gustos, aficiones, criterios o actitudes con nosotros). El atacante puede haber hecho una investigación previa para conocernos mejor, y aprovechar esta debilidad mimetizando ser alguien que conecta con nuestra forma de ser.
- **Reciprocidad.** Cuando se nos ha prometido o dado algo que consideramos de valor, nos inclinamos a corresponder en la misma medida. Incluso cuando no habíamos solicitado nada de antemano.

- **Consistencia.** Una vez que nos comprometemos públicamente a realizar alguna acción, intentamos ser consistentes con ella para no dar imagen de poca fiabilidad hacia terceros.
- **Validación social.** Tendemos a complacer a terceros cuando ello es lo que se espera de nosotros, porque otras personas actúan (o nos hacen creer que lo hacen), del mismo modo.
- **Escasez.** Tendemos a realizar acciones más compulsivas cuando se nos indica que algo tiene existencias limitadas, o que una oferta sólo está disponible temporalmente (por ejemplo).

2.3.2 Programas de Entrenamiento y Concienciación de Ciberseguridad

De manera muy sintética, de nuevo en Mitnick & Simon (2002) se desgranan las claves de un programa de entrenamiento y concienciación de empleados.

El objetivo es influir al personal para que cambie su actitud y motivarles para que cumplan su cometido para la protección de los Activos de Información de la compañía.

Generalmente, los programas de entrenamiento se customizarán según el público objetivo. No tiene sentido tratar por igual al vigilante de seguridad, que al responsable de TI. Todo el personal debería no obstante recibir formación periódica, y no se debería proporcionar acceso a ningún sistema o PC a empleado alguno, hasta realizar el curso inicial.

Si son sesiones presenciales, deben ser cortas, al grano. Señalar la importancia de la seguridad incluso con ejemplos reales, y las políticas a cumplir. Para dinamizar las sesiones, se pueden apoyar en vídeos, entrenamientos en ordenador, cursos online o materiales escritos.

Posteriormente se programarían sesiones específicas de tipologías de ataque, más extensas y dedicadas, y sesiones de refuerzo para que no bajen la guardia. Huelga decir que deben realizarse en horario laboral, y que ante un cambio de puesto que requiera un diferente nivel de permisos, el empleado debería inmediatamente realizar el curso de refresco que corresponda al mismo.

Tras ello, se pueden realizar tests online (como se propone en este TFM), para validar que realmente los conceptos han sido asimilados, incluso proveyendo un certificado al empleado para motivarle. También es conveniente que firmen un acuerdo de adherencia a las mencionadas buenas prácticas de protección de activos, puesto que esto se ha observado que refuerza en gran medida el compromiso por su parte.

En cuanto al recordatorio de la importancia de esta actividad: no sólo se reforzará en las sesiones posteriores programadas, sino que debe estar siempre presente en la mente de los individuos, mediante mecanismos como: distribución de copias de folletos, posters, boletines de noticias, recordatorios por email, pegatinas…

La amenaza es constante, y por tanto los recordatorios, también deben serlo.

2.3.3 Otros trabajos relacionados

En estudios adicionales, como los de Aldawood & Skinner (2019), se habla de las limitaciones actuales de los programas de entrenamiento y concienciación:

- Entorno de negocio. El difuminado del perímetro de la organización y el grado de interconexión con proveedores, incrementa la superficie de ataques. Se ve además un grado de correlación elevado entre ataques de ingeniería social, y uso de redes sociales por parte de los empleados.
- Entorno social. Por un lado, la experiencia de cliente, la competencia feroz y la propia sociedad demandan un trato más cercano y personalizado, pero por otro, ciertas prácticas constituyen una rendija por la que pueden colarse ataques en caso de no tomar las debidas precauciones.
- Entorno regulatorio. Se sugiere que quizás debería incluirse en la legislación la obligatoriedad de realizar campañas de concienciación, dado el grave impacto que pueden tener estos ataques incluso en la privacidad de los usuarios. Se muestran ejemplos. En Turquía, la mayor brecha de seguridad (2016) afectó datos personales de 50 millones de individuos (64% de la población). En China (2015), un 78,2% de los usuarios de Internet habían sufrido alguna filtración de datos personales. En USA (2015), más de 253 millones de habitantes habían visto expuesto alguno de sus datos personales.
- Entorno económico. El entrenamiento del personal requiere presupuesto, que debe ser incrementado o bien reasignado desde otras áreas. Esto es un problema en un escenario de cierta contracción como el que nos encontramos en la actualidad.
- Entorno personal. Como se ha comentado ya con anterioridad, la susceptibilidad ante ataques de ingeniería social varía entre individuos, en función de factores ambientales y propios.

En otro estudio de Lim, Park & Lee (2016), se propone un sistema de entrenamiento en Seguridad contra técnicas de Phishing y Smishing. Se consigue en una prueba de concepto con usuarios reales, unas reducciones de apertura de correos ilegítimos del 47 al 33%, y de la tasa de clic en enlaces maliciosos de un 16 a un 12%.

Se muestra a continuación el esquema típico de un ataque de phishing:

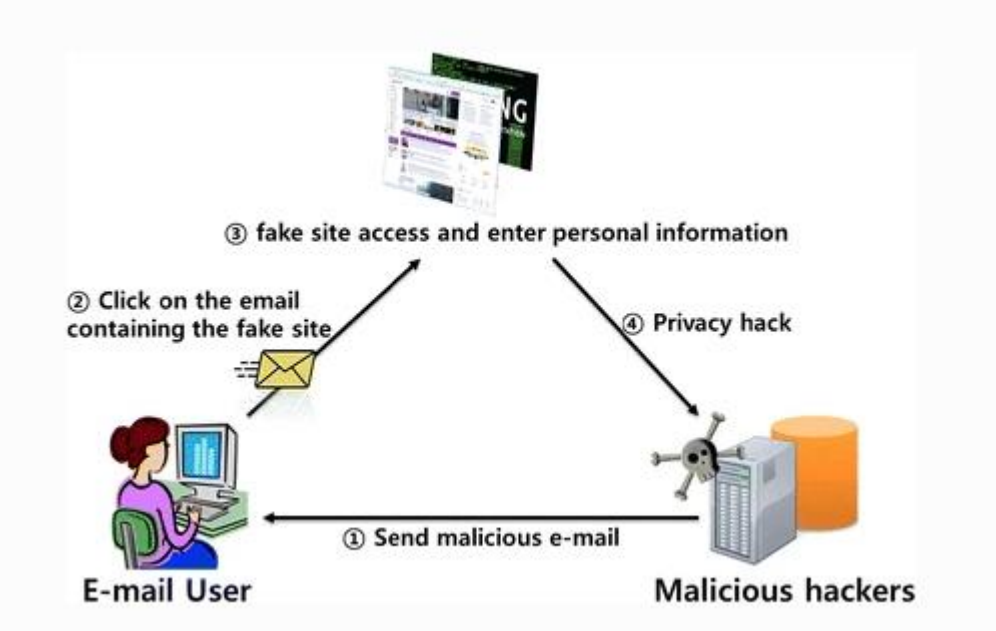

Ilustración 12. Escenario clásico de phishing (Lim et al., 2016)

El esquema construido para el experimento, permitía básicamente realizar campañas simuladas de phishing y smishing para medir el grado de concienciación y detectar tanto si abrían o no el email, como si pinchaban o no el link (en el caso del email phishing), y la mejora en las campañas sucesivas.

Dado que la metodología es parecida a la que se propone con la aplicación web objeto de este TFM, pero en nuestro caso combinada con un módulo de encuestas de refuerzo/autoevaluación, se podría suponer que los resultados de una prueba de concepto en condiciones similares, arrojase incluso resultados más positivos con la solución propuesta aquí.

2.3.4 Defensas y retos futuros

Como es lógico, aunque los entrenamientos en materia de concienciación de Ciberseguridad frente a ataques de ingeniería social son la herramienta claramente más efectiva para mitigar este tipo de amenazas, no son la única existente.

En Salahdine & Kaabouch (2019), se exponen las principales opciones que pueden plantearse. Idealmente, se combinarán todas ellas, para aplicar una defensa en profundidad. La siguiente tabla, indica los tipos de técnicas, basadas en las personas, o en la tecnología:

Tabla 1. Comparativa de contramedidas basadas en personas vs tecnología (Salahdine & Kaabouch, 2019)

Techniques	Description	Advantages	Limitations
Human Based	Education Training Awareness	- Easy to train humans what to do - Low number of victims	- Humans can be influenced emotionally - Tendency to o trust - Greed - Relative human decisions
Computer Based	Software, systems, and tools	- Efficient - Accurate	- Expensive products - Limited by the human unawareness - Very specific

La que se indica a continuación, desgrana las variantes para las técnicas basadas en el empleo de tecnología:

Tabla 2. Contramedidas tecnológicas para ataques de ingeniería social (Salahdine & Kaabouch, 2019)

Techniques	Description	Advantages	Limitations
Filtering tools	Anti-phishing tools (McAfee filter, Microsoft filter, and Web sense)	- Can block phishing emails and websites	- Not efficient - Attackers can send internally emails - Limited by human unawareness - Expensive tools
Alerting and scanning software	Anti-virus, anti-spams, anti-scams	- Efficient in alerting - Efficient in scanning - Strong products with security measures	- Expensive products - Alerts ignored by Humans
Biometric solutions	Based on biological traits	- Distinguish real profiles from fake profiles through their biological traits - Efficient	- Can be mimicked
Artificial intelligence-based	Based on adaptive learning systems	- Efficient - Adaptive	- Complex
Machine learning-based	Learning-based	- Achieve very good results - Effective -Online learning	- Complex
Anti-social engineering framework	Social Engineering Centered Risk Assessment (SERA)	- Efficient - High probability of attacks' detection	- Very expensive
Threshold-based	Use threshold to detect attacks	- Easy	- Not efficient - Limited by the threshold value
Phone-based	Use phones	- Easy	- Phone companies are still not able to stop Robocalls
Flow whitelisting	Identifying legitimate traffic from malicious traffic coming to the company's network	- Efficient - Learning-based - Able to distinguish between legitimate traffic from malicious traffic	- Limited by the human awareness - Ignoring alarms
IDS-based	Intrusion detection system	- Able to detect suspicious activities	High false alarm rates

Es relevante comentar que, aunque como se ha indicado ya en diversas ocasiones el componente humano y su refuerzo de concienciación es clave en este tipo de ataques, la aplicación de la inteligencia artificial está contribuyendo a la evolución de la ingeniería social a planteamientos más avanzados. Como se recoge en Wang et al. (2010), a modo de ejemplo:

- HoneyPhish, es una prueba de concepto del uso de cadenas de Markov para el procesamiento de lenguaje natural, que genera emails de phishing de manera automática (Gallagher, 2016).
- En otro estudio con DeepPhish, se recoge cómo basándose en redes de memoria de largo-corto plazo, se pueden generar URLs para phishing que evadan los mecanismos de detección automáticos (Bahnsen, Torroledo, Camacho & Villegas, 2018).
- Otro trabajo, describe cómo apoyarse en redes adversarias generativas (GAN), para construir un generador de dominios de malware que esquive también los detectores de dominios basados en deeplearning (Anderson, Woodbridge & Filar, 2016).

2.4 Estado actual

A continuación, se describirá la foto del panorama actual en cuanto a soluciones de entrenamiento de concienciación en Seguridad en general, y soluciones de esta índole anti-phishing en particular.

2.4.1. Organismos oficiales

2.4.1.1 CSA (Singapur)

Sede de uno de los centros de INTERPOL, Singapur es un polo tecnológico fundamental en el continente asiático, y como tal el nivel de concienciación en materia de Seguridad en la región es elevado. En 2015 se funda la Cyber Security Awareness Alliance de Singapur (CSA, 2015), para promover la concienciación y la adopción de buenas prácticas en el ámbito de la ciberseguridad.

Entre los materiales a disposición de particulares y empresas, se encuentran (CSAb, 2015): handbooks, relatos de historias de cibercrimen, libros electrónicos interactivos, flyers, password checkers para verificar cuánto tardaría un criminal en averiguar contraseñas con una composición determinada, una sección completa dedicada a la privacidad de datos, un conjunto de proveedores de servicios de ciberseguridad de confianza, otros enlaces de interés, y un kit de ciberseguridad del empleado.

Éste último, se centra en un cuestionario inicial para conocer el grado de concienciación actual de una organización, y a partir de ahí sugerir un programa de entrenamiento teórico, con presentaciones y materiales auxiliares de utilidad.

2.4.1.2 CDSE (USA)

El CDSE (Centre for Development of Security Excellence), se crea en 2010 como una escisión del Servicio de Seguridad y Defensa (DSS) dentro del Instituto de Seguridad del Departamento de Defensa (DoDSI), establecido en 1972 en Virginia (USA).

Esta institución gubernamental se centra en proporcionar educación en Seguridad de la Información mediante educación, entrenamiento y certificaciones, para el DoD y la industria americana en general.

Entre los recursos que proporcionan (CDSE, 2010), se encuentra información relativa a casos reales estudiados, soporte y buenas prácticas para diferentes posiciones dentro del ámbito de la ciberseguridad, decenas de cursos de awareness gamificados, posters, píldoras multimedia, artículos cortos, webinares, y toolkits de herramientas.

Dentro de estos toolkits, existe una sección entera (CDSEb, 2010), dedicada a SETA (Security Education Training and Awareness). Ésta, contiene multitud de materiales y guías sobre cuatro aspectos primordiales:

- Cómo crear un programa de entrenamiento.
- Cómo generar de manera efectiva la concienciación en Seguridad.
- Cómo formarse (cursos y certificados oficiales) para convertirse en un profesional en este ámbito.
- Cómo crear adecuadamente los briefings de Seguridad y otros materiales.

2.4.1.3 NCSC (UK)

El NCSC (National Cyber Security Centre) es un organismo creado en 2016 en Londres con profesionales de los centros de Aseguramiento de Información y Evaluación de Amenazas, y del Centro para la protección de Infraestructuras Nacionales. Su objetivo es proporcionar respuesta temprana a incidentes de Seguridad, siendo punto de contacto para sector público, industria, PyMEs y particulares en general.

Si bien es cierto que su web (NCSC, 2016) está llena de contenidos útiles en muchos frentes, no se ha encontrado como tal ningún programa de cyber awareness más allá de artículos

concretos referidos a situaciones generadas a raíz del COVID-19 (NCSCb, 2016) y una guía acerca de la problemática del phishing (NCSCc, 2016), en la que exhaustivamente se describe dicha técnica y cómo mitigarla.

2.4.1.4 NICE (USA)

NICE (2014), National Initiative for Cybersecurity Education. Es una iniciativa del NIST americano a raíz de un partnership entre gobierno, centros educativos y sector privado para fomentar la educación, el entrenamiento y el desarrollo de carreras profesionales dentro del mundo de la Ciberseguridad.

La cantidad de recursos que ponen a disposición del público es ingente. En NICEb (2014), se listan literalmente decenas de opciones de material y contenidos para aprendizaje y concienciación tanto gratuitos como de coste contenido, segmentados en desarrollo profesional, entrenamiento para instructores, juegos educacionales, y entrenamientos de concienciación.

En este último apartado, concienciación de Ciberseguridad, los ítems disponibles son:

- El enlace a todos los recursos del CDSE vistos anteriormente.
- Juegos educacionales para el ámbito familiar.
- Cursos de bajo coste no técnicos para trabajadores remotos.
- Cursos de concienciación en aspectos de privacidad y seguridad en gestión de correo electrónico.
- Cursos de concienciación en gestión de la Información y el uso de la tecnología.
- Curso de Wizer security gratuito de concienciación, con vídeos, cuestionarios, reportes de progreso y certificados acreditativos.
- Enlace a un servicio comercial para gestión de phishing awareness mediante cuestionarios online. Curso con coste asociado (15 libras/usuario, impuestos no incluídos).

Adicionalmente, en NICEc (2014), se incluyen otros recursos adicionales internos, de otros organismos oficiales, o proveedores de servicios de Seguridad, además de una lista de proveedores de servicios de formación y entrenamiento de las que las empresas pueden disponer (pagando las tarifas correspondientes).

2.4.1.5 INCIBE (España)

El Instituto Nacional de Ciberseguridad fue creado en 2006, y se enfoca exclusivamente en temas relativos a la Seguridad de la Información desde 2014. Ofrece servicios y desarrolla tecnologías en este ámbito, apoya al Ministerio de Industria, y presta soporte directo a ciudadanos y empresas.

INCIBE también pone a disposición de particulares y tejido empresarial gran cantidad de activos para mejorar el grado de concienciación en Seguridad y frente a ataques de ingeniería social. En este aspecto, destacaríamos:

- El blog de concienciación (INCIBE, 2006), con multitud de posts de gran valor, relevantes y actualizados conforme a las nuevas amenazas que van surgiendo.
- Un pdf con Políticas de seguridad para las PyMEs, con una checklist de controles en lo relativo a concienciación y formación en ciberseguridad (INCIBEb, 2006).
- Lo más potente desde mi punto de vista: el kit de concienciación (INCIBEc, 2006). Un archivo comprimido de unos 500Mb, autocontenido. Incluye el manual de implantación de un plan de concienciación para una organización, cubriendo todos y cada uno de los aspectos básicos: cronograma, recursos formativos, trípticos, posters, herramientas para ejecutar ataques de ingeniería social dirigidos (phishing, malware y USB infectados), y plantillas para cuestionarios y encuestas de satisfacción. La duración del plan es de unos 50 días, aproximadamente.

Ilustración 13. Kit de concienciación (INCIBEc, 2006)

2.4.2. Soluciones de big players del sector

2.4.2.1 Check Point

El fabricante israelí, uno de los líderes mundiales en Firewalling, ofrece soluciones para la protección electrónica de email en Cloud para software colaborativo (office 365 y G Suite).

Adicionalmente ofrece la generación de campañas de Concienciación en uso de email de forma segura, entendemos que a partir de campañas de phishing simuladas, o bien de su combinación con formaciones específicas. No queda del todo claro en su web, sin solicitar un presupuesto previo (Check Point, 1993).

2.4.2.2 Cisco

El proveedor californiano no aparenta ofrecer servicio alguno en este ámbito. Desde el punto de vista técnico sí tiene soluciones de Email security en su sitio web, pero en cuanto a concienciación, sólo hemos hallado en (Cisco, 1984) información relativa a cómo exportar su know-how en cuanto a cómo desarrollan sus campañas internas de concienciación en Seguridad, por si puede ser de ayuda a terceros. Se ofrece de modo altruista.

2.4.2.3 Fortinet

Al igual que sus competidores, los americanos también tienen soluciones automáticas contra ciertos ataques de ingeniería social. Pero a diferencia de Cisco, van más allá en su oferta de concienciación en esta materia, ofreciendo dos variantes:

- Curso individual de autoconcienciación (Fortinet, 2000). Gratuito, pero de alcance muy limitado.
- Servicio de entrenamiento profesional para empresas, de pago. Consta básicamente de tres componentes (Fortinetb, 2000):
    - Activos de concienciación: posters, plantillas de recordatorios, vídeos, hojas de checklists, salvapantallas customizables…
    - Entrenamiento. Vídeos fomativos con encuestas embebidas de autoevaluación.
    - Módulo de administración, para hacer seguimiento del grado de avance y fechas de los hitos relativos a las campañas formativas.

2.4.2.4 Google

La subsidiaria de Alphabet no presta ningún servicio de concienciación frente a ataques de ingeniería social. En cambio, su incubadora JigSaw sí plantea un servicio interesante (Google, 2019), que permite a un usuario individual evaluar su capacidad de identificación de emails de phishing, mediante una aplicación web que plantea una serie de supuestos prácticos cuya dificultad se incrementa progresivamente.

Es un ejercicio recomendable, que puede perfectamente incluirse en el programa de un itinerario formativo en materia de concienciación frente a ciberataques de ingeniería social mediante phishing.

Ilustración 14. PhishingQuiz (Google, 2019)

2.4.2.5 IBM

Los neoyorquinos, al igual que Fortinet, son más fuertes en sistemas de detección automática. No obstante, sí plantean de manera similar a Fortinet un programa personalizado de entrenamiento y concienciación (IBM, 2020) que aparentemente, cubriría los aspectos de eLearning, gamificación, y simulaciones sobre todo centradas en phishing, pero también acerca de otros ataques de ingeniería social (no se menciona más detalle, se requiere contacto comercial).

2.4.2.6 KnowBe4

La novel empresa norteamericana, cuyo Chief Hacking Officer y copropietario es el propio Kevin Mitnick, es uno de los líderes claros del sector en cuanto a entrenamiento de concienciación en ataques de ingeniería social, según Gartner. De hecho, ésta es su actividad principal.

Ofrecen servicios como, por ejemplo:

- Campañas simuladas de phishing (KnowBe4, 2020).
- Programas de e-learning customizados de forma automática en base a una encuesta inicial (ASAP, Automated Security Awareness Program), incluyendo temario, encuestas y dashboard de seguimiento, entre otros (KnowBe4b, 2020).
- Una extensa librería específica de contenidos de concienciación, desde módulos interactivos, de compliance, vídeos, juegos tipo trivial, posters, newsletters… (KnowBe4c, 2020).

2.4.2.7 McAfee

El proveedor californiano, a pesar de ser famoso por su solución de antivirus, tiene en su portfolio una solución para securizar el email en servidores de correo.

No obstante, en cuanto a concienciación en Ciberseguridad y prevención de ciberataques, disponen de lo que llaman McAfee online safety program (McAfee, 2020), mediante el cual empleados comprometidos de la firma colaboran con su comunidad realizando cursos de concienciación en escuelas, organizaciones juveniles y entidades sin ánimo de lucro, mediante workshops en los que hacen uso de material que luego reparten entre los asistentes.

2.4.2.8 Microsoft

El gigante de Washington, en el top 3 mundial de compañías por capitalización bursátil, tiene disponibles para sus clientes diversas soluciones para protección del correo electrónico, tanto tradicional como en entornos cloud.

En cuanto a concienciación en seguridad y concretamente ataques de phishing, proporcionan un servicio interesante en su partnership con la compañía Terranova Security (Microsoft, 2020). Llamado Gone Phishing Tournament, las organizaciones participantes (sin coste), llevan a cabo una campaña de phishing simulada en el período de tiempo que ellos consideren

y con los usuarios target que predefinan, recibiendo tras el ciberejercicio un reporte de personalizado acerca del grado de ciberresiliencia de la compañía en sí misma, y la comparación con otras empresas del sector similares (Terranova, 2020).

2.4.2.9 Proofpoint

Otro fabricante californiano con soluciones de detección de amenazas vía email, pero que presenta también un portfolio bastante completo de soluciones de concienciación frente a ataques de ingeniería social.

Además de proveer a sus clientes con gran cantidad de contenidos multimedia online en diferentes formatos en un programa sistemático (Proofpoint, 2020), tienen un su catálogo un programa específico contra ataques de ingeniería social mediante phishing y malware, principalmente, que incluyen tanto formación teórica, como simulaciones prácticas para medir el grado de ciberresiliencia (Proofpointb, 2020).

2.4.2.10 RSA

Esta filial de Dell Technologies tiene básicamente el mismo portfolio en cuanto a concienciación en Ciberseguridad que Proofpoint. De hecho, en (RSA, 2016), hacen referencia a las opciones de su portfolio en este ámbito, y aparecen los datasheets de Proofpoint, por lo que da la sensación de que existe algún tipo de partnership en este sentido.

No sólo realizan ataques de phishing, sino también de USB infectado y Smishing.

2.4.2.11 Sophos

Los británicos ofrecen soluciones de seguridad para email entre otras, y además tienen en su catálogo de soluciones el producto para realizar campañas de phishing simuladas, de cara a medir el nivel de concienciación de los usuarios objetivo dentro de una organización. La solución se llama Sophos Phish Threat (Sophos, 2020).

2.4.2.12 Trend Micro

Por último, el fabricante américo-japonés dispone de soluciones de detección automáticas como el resto, pero además también de un programa (Phish Insight) consistente en campañas de phishing simuladas en un entorno Software-as-a-service, combinadas con módulos de

training teóricos interactivos, de manera similar a muchos de sus competidores (TrendMicro, 2020).

2.4.3. Otras soluciones de proveedores especialistas

Hay literalmente decenas de proveedores de nicho disponibles en Internet, especializados en proporcionar servicios de concienciación frente a ataques de ingeniería social mediante phishing.

Algunos ejemplos relevantes, pueden ser:

2.4.3.1 Cofense

La antigua plataforma conocida como PhishMe, fue adquirida por Cofense. Éstos ofrecen servicios de detección de phishing, además de entrenamiento en concienciación mediante campañas simuladas haciendo uso del aplicativo mencionado (Cofense, 2020).

2.4.3.2 Hoxhunt

Proporciona entrenamiento online a empleados, personalizado en función del resultado de la campaña de phishing simulada, y ofrece herramientas para hacer seguimiento de la misma (Hoxhunt, 2020).

2.4.3.3 Ironscales

Plantea diversos servicios de identificación de amenazas, threat hunting, protección anti-malware, etc., y al igual que Hoxhunt, la posibilidad de generar campañas simuladas de phishing y entrenamientos relacionados (Ironscales, 2020).

2.4.3.4 Mimecast

Ofrece servicios similares al anterior, y su programa de concienciación considera el phishing como una pieza más. En este contexto, un módulo SaaS permite orquestar las campañas simuladas de manera sencilla, como se indica en (Mimecast, 2020).

2.4.3.5 Phriendlyphishing

Como su propio nombre indica, se focaliza únicamente en este tipo de ataques y proporcionar servicios de concienciación asociados, ya sea bien sólo teóricos, o teórico-prácticos (Phriendlyphishing, 2020).

2.4.3.6 Phishingbox

Además de proporcionar un módulo de identificación de amenazas en correo electrónico para O365, ofrece un servicio similar al anterior. Como novedad, tienen no sólo un entrenamiento orientado a Phishing, sino también focalizado en Ingeniería Social en general, y en Seguridad de la Información en general.

2.4.3.7 Otros

Como se indicó, existen muchos otros proveedores similares. Algunos de ellos y una comparativa asociada, se puede consultar en (G2, 2020).

2.4.4. Soluciones Open Source

Existen un puñado de soluciones open-source en el mercado para entrenamiento en concienciación sobre ataques de ingeniería social mediante email phishing. Se listan a continuación los principales, y sus características. Remarcar que únicamente se centran en la generación de las campañas simuladas, no en el entrenamiento teórico asociado que complementa a las mismas.

2.4.4.1 Gophish

Una de las soluciones más populares. Escrita en GO, es de uso extremadamente simple, y permite de manera intuitiva crear una campaña end-to-end: desde la creación de grupos de usuarios, páginas de spoofing, plantillas de email, perfiles de correo, y las propias campañas.

Incluye además un API Python para uso desde terceras aplicaciones, y se puede descargar como paquete binario compilado (disponible en Windows, Mac OSX y Linux) sin dependencias asociadas, para mayor facilidad de instalación. Licencia tipo MIT.

Para el desarrollo de la aplicación web propuesta en este TFM, nos apoyaremos en esta plataforma (Gophish, 2013).

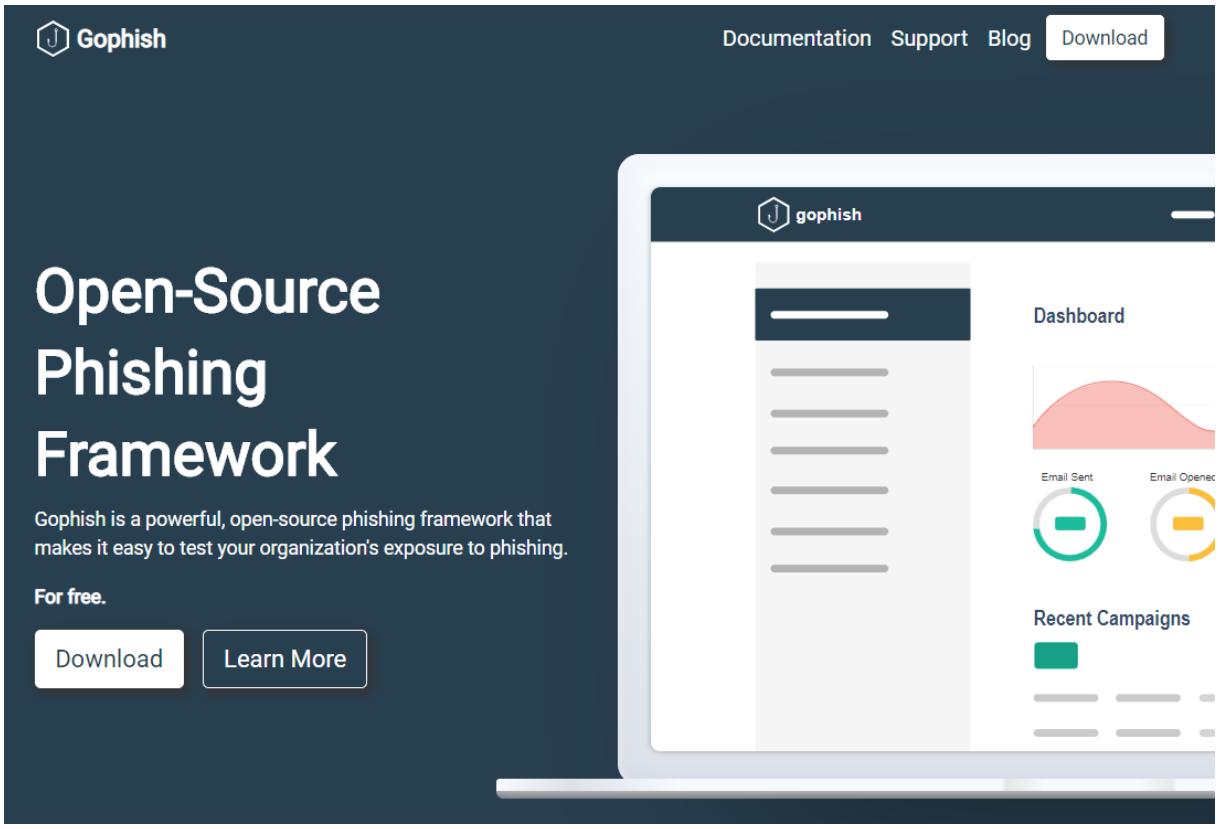

Ilustración 15. Sitio web Gophish (Gophish, 2013)

2.4.4.2 HiddenEye

Herramienta que, permitiendo realizar lo mismo que lo anterior, tiene soporte Android y funcionalidades extra como Keylogger y plantillas incorporadas para más de 30 de las redes sociales más habituales. Licencia tipo “The Unlicense”.

La contrapartida es que el uso no es tan intuitivo, y la instalación es más compleja, teniendo que descargar el código del repositorio (HiddenEye, 2020).

2.4.4.3 King Phisher

Entre las más conocidas junto con GoPhish. Escrito en Python, soporta campañas de más de 10K usuarios, y no tiene interfaz web, lo que reduce su exposición ante ataques tipo XSS, por ejemplo (KingPhisher, 2018).

Permite visualizar resultados en formato gráfico, adjuntar imágenes en las plantillas de email, utilizar 2FA, clonar páginas web o geolocalizar usuarios (todo incluído también en GoPhish, excepto el 2FA). Otra funcionalidad interesante son los avisos por SMS del progreso de las campañas. Sólo compatible con SO Linux. Licencia BSD modificada (3 cláusulas).

2.4.4.4 Lucy

La versión open source de este producto comercial, permite ejecutar cierta funcionalidad mediante un script Debian en Linux, pero con muchas limitaciones. El aplicativo full, permite testeo de infraestructura, campañas de entrenamiento online, etc. (LucySecurity, 2020).

La modalidad free, está ciertamente limitada: no permite programar campañas o crearlas/exportarlas mediante un fichero, por ejemplo, y el tamaño máximo de la campaña son 100 empleados. No es la más recomendable.

Sólo uso en Linux o entornos virtualizados. Tipo de licencia desconocida (no se ha hallado la información).

2.4.4.5 Phishing Frenzy

Aplicación originalmente dedicada a pentesting, pero con un módulo para ataques de phishing. Escrita en Ruby on Rails, se licencia como GNU Public License v3.0. Sólo disponible para Linux, y la instalación no es trivial (PhishingFrenzy, 2017).

Permite gestionar campañas, obtener credenciales, y visualizar estadísticas, entre otras funcionalidades.

2.4.4.6 SecurityIQ PhishSim

La herramienta Infosec IQ de Infosec, permite entre otras cosas lanzar campañas simuladas de phishing. Tiene un gran número de funcionalidad, y más de 1000 plantillas disponibles para los emails, adjuntos y páginas de spoofing, además de un constructor de template.

La versión gratuita de Infosec IQ permite lanzar dichas campañas, pero requiere licencia, que debe ser proporcionada por la propia compañía (suponemos que a cambio de los datos de contacto para envío futuro de ofertas asociadas).

No hemos encontrado información acerca del tipo de licencia de uso. (InfosecIQ, 2020).

2.4.4.7 Simple Phishing Toolkit (sptoolkit)

Este caso es curioso. Se trata de una herramienta originalmente escrita en 2012, principalmente en PHP, y que no sólo permitía realizar ataques de phishing, sino que poseía un módulo de campañas educativas y vídeos de concienciación.

A pesar de su versatilidad, incomprensiblemente el proyecto fue abandonado, y ahora mismo es difícil de encontrar la documentación asociada, o el propio código fuente. Esto hace que haya caído en desuso, y que su instalación no sea sencilla.

El repositorio original se encuentra en (Sptoolkit, 2012).

2.4.4.8 Social-Engineer Toolkit (SET)

La archiconocida aplicación para ataques de ingeniería social incluida por defecto en la suite de Kali Linux, codificada en Python. Ya mencionada con anterioridad, es multiplataforma, permite integración con módulos de terceros, y realizar ataques de spear-phishing además de clonar sitios web para el robo de credenciales (entre muchas otras cosas).

La licencia de uso es personalizada. Más información en (SET, 2012).

2.4.4.9 SpearPhisher BETA

Otro caso extraño. Herramienta creada en 2013 por TrustedSec, para Windows, de interfaz extremadamente sencilla que permitía orquestar campañas de phishing de manera muy ágil. No llegó a generarse versión estable, y el proyecto fue abandonado, desapareciendo incluso la página web.

No hay información de tipo de licencia de uso, y desde el blog se puede simplemente descargar el ejecutable (SpearPhisher, 2013).

2.4.4.10 SpeedPhish Framework (SPF)

Aplicación en Python dedicada primordialmente al Pentesting, pero que puede configurarse para lanzar ataques de phishing customizando plantillas de email y páginas web. No sería ni de lejos la primera opción para alguien buscando herramientas de concienciación.

De uso en sistemas Linux, licencia BSD modificada con 3 cláusulas. (SPF, 2014)

2.4.5. Panorama del sector en España

Por último, analizaremos someramente el estado actual del panorama en Espala en cuanto a proveedores tradicionales de soluciones de comunicaciones o especialistas en seguridad, que

podrían prestar servicios de concienciación en materia de ataques de ingeniería social y concretamente email phishing, a empresas.

Podemos considerar cuatro grandes grupos, principalmente:

- **Empresas y operadores de telecomunicaciones**. Algunos proveedores de servicios esenciales, como por ejemplo Telefónica con su división ElevenPaths (2013), ofrecen este tipo de servicios a medianas y grandes empresas. Otras como Orange (2020) y Vodafone (2020) están abriéndose camino en esta materia, y también ofrecen soluciones a medida en este sentido.

  Existen otros pequeños operadores, de alcance más regional, que también pueden prestar este tipo de servicios, pero su presencia en este ámbito es más residual.

- **Empresas de ciberseguridad**. Éstas ofrecen diferentes servicios de manera especializada. No tienen gran fuerza comercial, y muchas de ellas llegan a acuerdos con terceros locales que revenden su actividad.

  Suelen recibir apoyo de grandes operadores telco, o especialistas del sector, y atacan a sectores específicos, como pueden ser las Administraciones Públicas. Ejemplos pueden ser S21Sec (2000), Wise Security (2019), OneSeq (1991), Factum (2009), SecureIT (2009), SIA (2004) recientemente adquirida por Indra, o Aiuken (2012).

- **Centros locales de ciberseguridad**, como INCIBE (2006), Tegra (2018), OneCyber (2020), Basquecybersecurity (2020) o Cybereop (2020).

- **Otras pequeñas empresas locales** que ofrecen servicios en el ámbito de las telecomunicaciones o Sistemas de Información sin ser especializadas, a pequeñas empresas. En algunos casos, son representantes o distribuidoras. Ofrecen servicios básicos como los de concienciación, análisis de vulnerabilidades o test de intrusión, por ejemplo.

3. Objetivos y metodología de trabajo

3.1 Objetivo

3.1.1 Objetivo general

El principal objetivo de la contribución que representa este TFM, como ya se indicó en la sección Introducción, es realizar una aportación en el campo de la protección de activos de Información. Concretamente, mediante una solución software que permite aumentar el grado de concienciación de los usuarios frente a ataques de ingeniería social mediante email phishing, en el ámbito corporativo.

Como se indicó en el apartado Antecedentes dentro de Contexto y Estado del arte, desde sus inicios en los años 70 del pasado siglo, la prevalencia de esta tipología de ataques se ha incrementado enormemente por su facilidad de ejecución, y la susceptibilidad de los individuos, por motivos psicoculturales, y falta de concienciación.

Por otro lado, y tal y como se reflejó en Trabajos relacionados y Estado actual, existen multitud de iniciativas a disposición de individuos y entidades empresariales para mitigar ataques de ingeniería social. Centrándonos en el email phishing, resumiendo a muy alto nivel, hay iniciativas tanto del sector público como del privado, ofreciendo soluciones que se centran en dos aspectos, mayoritariamente:

- Plano teórico: realizar cursos de formación/concienciación de empleados, online u offline. Para mejorar su grado de conocimientos sobre estas técnicas de engaño, cómo se realizan, ganchos que suelen utilizarse, trucos para desenmascararlos, y cómo proceder al detectarlos. Haciendo uso de material audiovisual diverso (cuestionarios, libros, etc.).
- Plano práctico: efectuar campañas simuladas de email phishing mediante envíos reales, para medir el grado de resiliencia frente a los ataques, tanto pre como post entrenamiento.

Planteamos un desarrollo software que cubre un hueco en este ámbito de la Ciberseguridad, como se verá en la siguiente sección.

3.1.2 Objetivo específico

Para las corporaciones, los costes son importantes. Para las pequeñas, lo son más. En el tejido productivo español, dejando los autónomos al margen, más del 99% de las empresas

tienen menos de 250 empleados, según datos del Ministerio de Industria, Comercio y Turismo (PlataformaPYME, 2020).

Desgraciadamente, aun hoy en día la inversión en Ciberseguridad en este segmento empresarial se ve como un lastre, en lugar de como un habilitador de negocio. Muchos de los gestores de estas PyMEs, intentan buscar soluciones gratuitas para sus necesidades ofimáticas y de seguridad (antivirus gratuitos, etc.), para maximizar sus beneficios.

Para aquellos concienciados con la seguridad y la importancia e impacto potencial de los ataques de ingeniería social, pero con presupuesto ajustado, las opciones en cuanto a soluciones disponibles, son básicamente tres:

- Utilizar o desarrollar si tienen el conocimiento para ello, alguna herramienta que les permita hacer un envío masivo de correos de phishing a través de un servidor de correo concreto, y quizás realizar algún curso de concienciación manualmente.
- Emplear alguna de las plataformas de phishing open-source existentes. Como hemos visto, la mayoría tienen ciertas limitaciones y/o su instalación es dificultosa. En cualquier caso, los entrenamientos teóricos deberán realizarse manualmente, excepto en el caso de Simple Phishing Toolkit. Por desgracia, ésta ya no recibe soporte, y la documentación es muy escasa.
- Hacer uso de versiones demo de soluciones comerciales. Muchas de ellas son interesantes, pero la funcionalidad es temporal y muy limitada. El coste de las opciones con coste que hemos analizado, oscila entre los 15€ y los 50€ por usuario, lo que para muchas empresas puede resultar prohibitivo, sobre todo si se realizan varias campañas anuales (cosa diferente sería si analizamos el binomio coste/beneficio de la inversión, pero ello excede el objetivo del presente TFM).

Si además se quiere optar por mantener la privacidad de los datos, la última opción debe descartarse por motivos obvios (se trata de soluciones SaaS gestionadas por un tercero).

Por todo ello, para cubrir ese gap detectado, lo que se propone es una herramienta web open-source, sencilla y autocontenida, que permita a una PyME implementar con éxito de forma gratuita los dos principales pilares de un programa de concienciación frente a ataques de ingeniería social mediante técnicas de phishing, como son:

- Realizar campañas simuladas de email phishing, sin límite de empleados.
- Llevar a cabo cuestionarios de formación/evaluación teórica de conocimientos al respecto, totalmente personalizados de manera simple, al contexto de dicha organización.

Todo ello en formato online, con gestión centralizada y total flexibilidad, manteniendo la privacidad de sus datos, pudiendo visualizar y exportar los resultados, y generando indicadores para medir el grado de consecución de resultados para un evento particular, o la evolución de la ciberresiliencia frente al tiempo.

3.2 Metodología de trabajo

La metodología seguida para la consecución de los objetivos mencionados en el epígrafe anterior, ha sido la siguiente:

1) **Análisis de contexto.** Se lleva a cabo tanto desde el punto de vista de la bibliografía disponible respecto a la problemática de los ataques de ingeniería social, como de trabajos relacionados y soluciones existentes en el mercado: de entidades oficiales, grandes actores del mundo de la Ciberseguridad e Internet en general, pequeños proveedores, y soluciones open-source, particularizando también para el escenario español.
2) **Definición de requerimientos funcionales y casos de uso de la solución**. Descripción de qué debe poder realizar la herramienta.
3) **Diseño de casos de prueba**. Pruebas a realizar para verificar que la funcionalidad anterior realmente está cubierta.
4) **Selección de motor phishing open-source**. Dado que se han analizado las soluciones open-source para el lanzamiento de las campañas de phishing, se decide integrar uno de los motores existente, por simplicidad y economía de tiempo.
5) **Selección de tecnología a emplear**. En base a los dos anteriores, selección de tecnología a emplear (tipo de herramienta, y lenguaje/s de programación).
6) **Diseño de arquitectura** para soportar los requerimientos.
7) **Diseño de modelo de datos** que se ajuste a las necesidades requeridas.
8) **Desarrollo del interfaz del motor de phishing**, para lanzar las campañas simuladas.
9) **Pruebas** del interfaz del motor.
10) **Desarrollo de la aplicación web, documentación de los módulos** y todos sus componentes.
11) **Pruebas de los componentes**, para verificar su correcto desarrollo.
12) **Pruebas E2E**, para comprobar el funcionamiento global del aplicativo.
13) **Selección de licencia Open Source** más adecuada para la distribución del software..
14) **Documentación auxiliar** (ayuda, guía de instalación, paquetes para subir a repositorio público…).
15) **Recopilación de feedback** de terceros.

4. Desarrollo específico de la contribución

4.1 Análisis de contexto

Descrito en el epígrafe 2 del presente TFM (Contexto y estado del arte), cuyo objetivo a raíz del mencionado análisis, se materializa en los objetivos concretos de desarrollo de Software planteados en la sección 3 (Objetivo general, y específico).

4.2 Definición de requerimientos funcionales

A continuación, se indican los requerimientos funcionales del desarrollo software a implementar. Simultáneamente a la definición de los mismos, se profundiza en ciertos conceptos del diseño.

RF01: existirá autenticación y control de acceso de usuarios para la parte de administración de la herramienta. Sólo los usuarios logados podrán acceder para gestionar el sitio, y lo harán mediante un formulario web a tal efecto.

RF02: los usuarios no relacionados con la administración, es decir, los que participan en las campañas de entrenamiento y concienciación, no necesitarán autenticarse para poder completar los cuestionarios, pues serán automáticamente reconocidos por el sistema para la recopilación de estadísticas en base a algún identificador.

RF03: los usuarios no logados que accedan al aplicativo, podrán visualizar sólo el contenido no protegido (páginas estáticas de información sobre el sitio, licencia, contacto y ayuda). En la página principal, verán una descripción de la funcionalidad básica de la aplicación, pero no podrán hacer uso de ella. Si se autentican correctamente, podrán tener acceso a todas las opciones disponibles.

RF04: desde la sección de administración, podrán gestionarse todos los objetos asociados a la funcionalidad de los cuestionarios de concienciación. Por el contrario, los objetos asociados al módulo de Phishing, se gestionarán desde el propio aplicativo (una vez autenticados), o desde la propia interfaz web del motor de Phishing.

RF05: para la funcionalidad de ataques de phishing simulados, desde el aplicativo podrá controlarse el end-to-end del proceso, existiendo una sección entera dedicada a ello y sus componentes.

RF06: grupos de usuarios. Respecto a este objeto, una vez logado en el aplicativo a desarrollar, un usuario administrador podrá:

- Crear grupos nuevos de usuarios objetivo de la campaña, de manera simple. Para ello, a través de un formulario sólo deberá cargar un archivo CSV delimitado por comas con un formato determinado (nombre, apellidos, email de contacto y posición en la compañía). A dicho grupo, le asignará un nombre que será su identificador human-friendly en la base de datos.
- Gestionar los grupos existentes. Desde la sección de Listado de grupos, podrá visualizar todos los creados en la base de datos del motor de Phishing, ver su ID en base de datos, su nombre, y el número de usuarios que contiene.
- Desde la sección de listado de grupos, podrá también eliminar cualquiera de los existentes.
- Desde la misma ubicación, podrá también consultar información ampliada de un grupo de usuarios concreto. Al pulsar esta opción, accederá a una nueva ventana donde visualizará los detalles anteriores, más el listado completo de usuarios y sus datos particulares.

RF07: plantillas de email de phishing. Respecto a esta entidad, una vez logado en el aplicativo a desarrollar, un usuario administrador podrá:

- Crear nuevas plantillas de email, de manera simple. Para ello, a través de un formulario deberá indicar un asunto para el correo, y subir un archivo HTML con el texto a incluir en el email gancho, enlaces, etc. Podrán utilizarse etiquetas tanto en el asunto, como dentro del código HTML, que serán sustituidas por las variables reales asociadas a cada usuario concreto, para personalizar los emails. El catálogo de etiquetas será el que provee el motor de phishing por defecto. A dicha plantilla de email, el administrador le asignará un nombre en el formulario que será su identificador human-friendly en la base de datos.
- Gestionar las plantillas de email existentes. Desde la sección de Listado de emails, podrá visualizar todos los creados en la base de datos del motor de Phishing, ver su ID en base de datos, su nombre y asunto.
- Desde la sección de listado de emails, podrá también eliminar cualquiera de los existentes.
- Desde la misma ubicación, podrá también visualizar información ampliada de una plantilla concreta. Al pulsar esta opción, accederá a una nueva ventana donde se verán los detalles anteriores, y el código HTML asociado a la plantilla.

RF08: plantillas de páginas de aterrizaje. Éstas, son a las que se redirigirá al usuario que pinche el enlace del email de phishing, para intentar de forma simulada robarle las

credenciales u otra información confidencial. En lo que respecta a esta entidad, una vez logado en el aplicativo a desarrollar, un usuario administrador podrá:

- Crear nuevas plantillas de páginas de aterrizaje, de manera simple. Para ello, a través de un formulario deberá indicar si se requiere o no la captura de credenciales en base de datos (indicando una de las siguientes opciones: captura de usuario, captura de usuario y password, o no captura), y subir un archivo HTML con el código de la página de aterrizaje a mostrar. Se indicará también si se quiere redirigir a una nueva URL al usuario tras introducir las credenciales, para dar la sensación de que simplemente se ha producido un error de login, pero está intentando acceder a la página legítima. A dicha plantilla, el administrador le asignará un nombre en el formulario que será su identificador human-friendly en la base de datos.
- Gestionar las plantillas de páginas de aterrizaje existentes. Desde la sección de Listado de páginas de aterrizaje, podrá visualizar todas las creadas en la base de datos del motor de Phishing, ver su ID en base de datos, su nombre, si capturan o no credenciales (usuario y/o password), y la URL a la que redirigen (si existe).
- Desde la sección de listado de páginas, podrá también eliminar cualquiera de las existentes.
- Desde la misma ubicación, podrá también visualizar información ampliada de una plantilla concreta. Al pulsar esta opción, accederá a una nueva ventana donde se verán los detalles anteriores, más el código HTML asociado a la plantilla.

RF09: perfiles de correo SMTP (Simple Mail Transfer Protocol). Son las posibles configuraciones de servidor SMTP utilizadas para el envío masivo de emails en cada campaña a través de Internet. Respecto a esta entidad, una vez logado en el aplicativo a desarrollar, un usuario administrador podrá:

- Crear nuevos perfiles SMTP, de manera simple. Para ello, a través de un formulario deberá indicar el host a utilizar (por ejemplo smtp.gmail.com), el username (id de correo, como puede ser uno de Gmail), username y password de autenticación, la información que se quiere visualizar en el campo origen de los emails enviados (remitente: tanto nombre, como dirección de correo), y si los errores de certificado SSL deben ser ignorados o no. A dicho perfil, el administrador le asignará un nombre en el formulario, que será su identificador human-friendly en la base de datos.
- Gestionar los perfiles SMTP existentes. Desde la sección de Listado de perfiles SMTP, podrá visualizar todos los creadas en la base de datos del motor de Phishing, ver su ID en base de datos, su nombre, y el host.

- Desde la sección de listado de perfiles, podrá también eliminar cualquiera de los existentes.
- Desde la misma ubicación, podrá también visualizar información ampliada de un perfil SMTP concreto. Al pulsar esta opción, accederá a una nueva ventana donde se verán todas las propiedades indicadas en el formulario de creación.

RF10: campañas de Phishing. Son las ejecuciones reales de campañas simuladas, para verificar el grado de resiliencia frente a estos ataques de ingeniería social de un grupo de usuarios, de manera práctica y objetiva. Referente a esta entidad, una vez logado en el aplicativo a desarrollar, un usuario administrador podrá:

- Crear nuevas campañas, de manera sencilla. Para ello, a través de un formulario deberá indicar:
    - El nombre de la campaña, que será su identificador human-friendly en la base de datos.
    - Seleccionará de entre los ya existentes (creados previamente) mediante desplegables:
        - Un grupo de usuarios
        - Una plantilla de email phishing
        - Una página de aterrizaje
        - Un perfil de correo SMTP
        - La URL en la que se ubica la página de phishing
        - La fecha de lanzamiento de la campaña, en horario GMT y formato DD/MM/YYY@HH:MM. En caso de dejarla vacía, el lanzamiento sería inmediato.
- Gestionar las campañas existentes. Desde la sección de Listado de campañas, podrá visualizar todas los creadas en la base de datos del motor de Phishing, ver su ID en base de datos, su nombre, el número de usuarios incluídos en la campaña, las fechas de creación y de inicio de la campaña (no coincidirán en caso de ejecución diferida), y el estado de ejecución de la misma. Los estados posibles son en espera (Queued), en curso (In progress), o completada (Completed), y se visualizan en rojo, verde y azul, respectivamente.
- Desde la sección de listado de campañas, se podrá tanto finalizar (pasar a estado Completed, punto a partir del cual se dejan de recopilar estadísticas asociadas a la interacción de los usuarios con los emails), como eliminar pasando a Cancelled (momento a partir del cual se borran de la base de datos y dejan de aparecer en el

listado). Lógicamente las opciones estarán disponibles o no, en función del estado actual de una campaña dada.

- Desde la misma ubicación, se podrá también visualizar información ampliada del estado de una campaña concreta. Al pulsar esta opción, accederá a una nueva ventana donde se podrá consultar:
    - Información general de la campaña
        - Id en base de datos.
        - Nombre asociado en el formulario de creación.
        - Fecha de creación en formato ISO.
        - Fecha de lanzamiento en formato ISO.
        - Estado actual (de entre los posibles, indicados anteriormente).
    - Componentes de la campaña
        - Plantilla de email phishing utilizada.
        - Página de aterrizaje empleada.
        - Perfil de correo SMTP en uso.
        - URL de phishing.
    - Estadísticas de Phishing
        - Número de emails totales en la campaña (coincide con el número de usuarios del grupo objetivo).
        - Emails enviados.
        - Emails abiertos.
        - Emails en los que se ha clicado el enlace de phishing.
        - Emails que han generado una captura de credenciales.
        - Emails que han dado lugar a errores de transmisión.
    - Estadísticas de usuario
        - Listado de usuarios e información asociada (user ID, nombre, apellidos, email, posición, y estado de Phishing (de los cinco descritos en estadísticas en el punto anterior).
    - Descargas
        - Opción de descarga de estadísticas generales de la campaña en fichero CSV. Contiene información general de la campaña, componentes, y estadísticas de phishing. El nombre del archivo será Campaign_<CAMPAIGN_NAME>_stats_DDMMYYY_HHMMSS.csv.
        - Opción de descarga de estadísticas de usuarios en la campaña, en fichero CSV. El nombre del archivo será Campaign_<CAMPAIGN_NAME>_user_stats_DDMMYYY_HHMMSS.csv.

RF11: cuando se active la campaña de phishing, se realizará el envío masivo de correos electrónicos a los usuarios, y se actualizará convenientemente en la base de datos del motor de phishing los estados de cada uno de los usuarios en la campaña, en función de las interacciones de los mismos con el email recibido. En el momento que la campaña se complete o cancele, esto último dejará de suceder.

RF12: para la gestión de la comunicación entre el motor de Phishing y el aplicativo, se utilizará un API. Existirá un fichero de configuración en el que se indique al aplicativo las propiedades del mismo, en cuanto a clave de API a utilizar para acceso al motor, URL de la web principal del motor, perfiles SMTP, o si deben o no usarse certificados SSL.

RF13: para la funcionalidad de cuestionarios de autoevaluación y aprendizaje en materia de concienciación frente a ataques de ingeniería social, desde el aplicativo podrá controlarse el end-to-end del proceso, existiendo una sección entera del mismo dedicada a ello y sus componentes.

RF14: cuestionarios de autoevaluación y aprendizaje. Serán tests educativos, que los usuarios objetivo de la campaña de concienciación podrán realizar online. Podrán crearse tantos tests como se considere oportuno, así como realizar también múltiples campañas. Será otro módulo del aplicativo independiente, totalmente ajeno al motor de Phishing open-source, y con base de datos propia.

RF15: El formato de los cuestionarios será de preguntas y respuestas, totalmente personalizable y dinámico. El administrador que desee crear un nuevo test, no tiene más que subir un archivo CSV delimitado por punto y coma al aplicativo a través del formulario a tal efecto, que debe reunir las siguientes características:

1. Contendrá un número indefinido de filas, pero sólo dos columnas.

2. La primera columna de cada fila, contendrá:

    - Una Q, si se trata de una pregunta,
    - Una T, si se trata de una respuesta y además es válida,
    - Una F, si se trata de una respuesta y además es falsa.

3. La segunda columna, contendrá el texto de la pregunta o respuesta, según corresponda.

4. Deberá haber al menos dos respuestas por pregunta.

5. Deberá haber al menos una respuesta válida por pregunta.

Se entiende que, hasta que no aparezca en el archivo una nueva pregunta al inspeccionar las filas en orden descendente, todas las respuestas pertenecen a la última pregunta detectada.

De esta manera, como se ve, la estructura del cuestionario es dinámica. Podrán existir tantas preguntas como desee el instructor, tantas respuestas por pregunta como considere (pregunta a pregunta), y del mismo modo puede jugar con cuántas respuestas válidas habrá por cada una de las cuestiones.

RF16: gestión de plantillas de cuestionarios. Una vez logado en el aplicativo a desarrollar, un usuario administrador podrá:

- Crear nuevas plantillas de cuestionarios de aprendizaje y autoevaluación o encuestas, de manera simple. Para ello, a través de un formulario deberá subir un archivo CSV cumpliendo las restricciones indicadas en el apartado anterior. En caso contrario, se generará un error de validación de datos. A dicha plantilla, el administrador le asignará un nombre en el formulario, que será su identificador human-friendly en la base de datos.
- Gestionar las plantillas de test existentes. Desde la sección de listado de encuestas, podrá visualizar todas las creadas en la base de datos del aplicativo, ver su ID en base de datos, su nombre, y su fecha de creación GMT.
- Desde la sección de listado de encuestas, podrá también eliminar cualquiera de las existentes.
- Desde la misma ubicación, podrá también visualizar información ampliada de una plantilla de test concreta. Al pulsar esta opción, accederá a una nueva ventana donde se verán los detalles anteriores, y el listado de todas las preguntas y respuestas, marcando con un tick o un aspa cuáles son verdaderas o falsas, respectivamente.

RF17: gestión de campañas de concienciación. Una vez logado en el aplicativo a desarrollar, un usuario administrador podrá gestión su ciclo de vida. Para lanzar una campaña de aprendizaje basada en cuestionarios, existirán dos enfoques:

- Campaña basada en grupo de usuarios. Se genera en base a un grupo de usuarios predefinido en la base de datos del motor de phishing, haciendo uso de la funcionalidad descrita en la sección de Phishing del aplicativo. Será totalmente independiente de las campañas de phishing, e incluso puede llevarse a cabo sobre usuarios diferentes, sin solape alguno.
- Campaña de aprendizaje basada en una campaña de phishing ya existente. Como se indica, reaprovecha un grupo de usuarios inmersos en una campaña de phishing ya generada, pero además permite generar un indicador o KPI de concienciación

adicional que puede ser útil para la organización (más detalle posteriormente). Este KPI, no estaría disponible para campañas de aprendizaje basadas en grupos, en lugar de en campañas de phishing previas.

RF18: Si se decide crear una campaña de entrenamiento basado en grupo, se llevará a cabo de manera simple mediante un formulario en el que se indicará el nombre de la campaña, el grupo de usuarios seleccionándolo de un desplegable (de entre los existentes en la base de datos del motor de Phishing), y la plantilla de test, seleccionándola de un desplegable, en este caso con los nombres de los tests creados en la base de datos propia del aplicativo, e independiente de la del motor.

RF19: Si se decide crear una campaña de entrenamiento basado en campaña de phishing, se llevará a cabo de manera simple mediante un formulario en el que se indicará el nombre de la campaña, la campaña de phishing objetivo seleccionándola de un desplegable con las existentes vivas en la base de datos del motor de Phishing, y la plantilla de test, seleccionándola de un desplegable, en este caso con los nombres de los tests creados en la base de datos propia del aplicativo, e independiente de la del motor.

RF20: Una vez creadas las campañas de entrenamiento (de uno de los dos modos indicados en el requisito funcional anterior), se podrá visualizar el listado de todas las existentes en una sección específica a tal efecto.

- Se mostrará el listado de campañas vivas, su ID en base de datos, el nombre, su estado (ACTIVE o COMPLETED) en color verde o azul respectivamente, el número de usuarios en la campaña, la fecha de creación en formato GMT, y dos KPIs que se verán posteriormente (T-score y A-score).
- Se podrán completar las campañas activas, y eliminar las activas o completadas.
- Se podrá acceder a la sección de detalle de una campaña de entrenamiento concreta.

RF21: Al visualizar la información de detalle de una campaña de entrenamiento, se le mostrará al usuario administrador:

- Detalles generales asociados a la campaña
    - Id en base de datos.
    - Nombre de la campaña de entrenamiento.
    - Número de usuarios.
    - Nombre del grupo objetivo de la campaña (grupo de usuario, o campaña de phishing).
    - Tipo de objetivo (Grupo o Campaña).

    - Nombre de la plantilla del cuestionario.
    - Fecha de creación.
    - Estado.
- Estadísticas de los usuarios
    - User ID.
    - Nombre de usuario.
    - Apellidos de usuario.
    - Email.
    - Posición en la compañía.
    - Estado de phishing (de entre los posibles: email enviado, abierto, clicado, credenciales enviadas, o error. Sólo aplica si el entrenamiento se crea a partir de una campaña de phishing, en caso contrario, se mostrará N/A).
    - Resultados de la encuesta (contiene las respuestas marcadas por parte del usuario en caso de haber respondido ya al cuestionario, en formato cadena de caracteres, con 1´s para respuestas verdaderas y 0´s para las falsas, intercalando un guion medio entre respuestas. N/A, en caso de no haber respuesta todavía).
    - Puntuación obtenida por parte del usuario en la encuesta (entre 0 y 100, o N/A en caso de no haberse realizado. Se detalla el cálculo en otro requisito).
- Descargas.
        - Permite la descarga en archivo CSV de toda la información anterior (general de la campaña, más la específica de usuario). El nombre del archivo será Training_<TRAINING_NAME>_stats_DDMMYYY_HHMMSS.csv.

RF22: una vez creada una campaña de entrenamiento, se generará inmediatamente un envío de email a todos los integrantes del mismo, independientemente de que se originasen en un grupo de usuarios, o una campaña de phishing. La configuración SMTP para realizar el envío de correo se recuperará de un fichero de configuración específico para el módulo de cuestionarios del aplicativo.

RF23: La plantilla de email a enviar para la campaña, será única para todo el aplicativo, y tendrá dos tags para su customización: {PERSON_NAME} y {SURVEY_LINK}. El primero de ellos, se sustituirá por el nombre de usuario. El segundo de ellos, será el enlace que el usuario debe clicar para acceder a la página del aplicativo que le permitirá llevar a cabo la encuesta de evaluación.

RF24: {SURVEY_LINK} se construirá en base a dos componentes:

- La URL base del aplicativo, que se obtendrá del fichero de configuración mencionado anteriormente para el módulo de tests.
- Un slug, o identificador unívoco de 50 caracteres alfanuméricos pseudoaleatorios que se concatenará al anterior. Este slug se guardará en base de datos, será único por campaña y usuario, y servirá para identificar al registro de base de datos asociada a ellos.

RF25: Una vez el usuario reciba el email con la invitación para realizar el test de autoaprendizaje, podrá pinchar la URL con el slug para acceder a su instancia de la encuesta. Se le presentará el formulario con todas las preguntas y respuestas, y podrá seleccionar las que considere verdaderas antes de enviar los datos mediante POST. Una vez hecho esto, se le indicará la nota obtenida (0-100) en verde o rojo en función de si sobrepasa o no el umbral de puntuación 50 respectivamente, y se le presentará nuevamente la encuesta indicando cuáles eran las opciones de respuesta correctas y falsas.

RF26: El slug sólo podrá ser consumido una vez, y sólo mientras la campaña siga estando activa. Si la campaña está terminada o cancelada, o bien el usuario ya ha contestado al test, la URL no estará disponible y se le presentará un error. Se entiende que el email no es transferible a terceros, aunque sería conveniente advertirlo en la plantilla de envío.

RF27: Puntuación de los tests. Para no complicar demasiado la lógica de cómputo, se ha optado por el siguiente algoritmo:

i) Se reparte equitativamente el máximo de puntos (100), entre todas las preguntas.
ii) Los puntos de cada pregunta, se dividen de nuevo equitativamente entre todas las respuestas válidas dentro de la misma.
iii) Si el usuario marca una respuesta falsa, no resta puntuación.
iv) Si el usuario marca una respuesta correcta, se le añaden los puntos que correspondan al aplicar i) y ii), al subtotal.

La limitación de este enfoque, es que un usuario que marque absolutamente todas las opciones acabaría obteniendo una puntuación de 100. Se ha preferido mantener esta flexibilidad a no ceñirnos al esquema simple de sólo una respuesta verdadera por pregunta y/o darla por errónea en caso de algún error.

No obstante, la lógica puede modificarse en el código fuente, sin más que actualizar el módulo correspondiente (*create_survey_scoring).*

RF28: KPIs (Key Performance Indicators). Como se indicó con anterioridad, al visualizar la lista de entrenamientos activos, se muestran dos indicadores: T-score y A-score. Se definen a continuación:

- T-score: Training score o puntuación de entrenamiento. Representa el valor promedio de las puntuaciones obtenidas por los usuarios que ya han consumido el test de autoaprendizaje. Es por tanto, un valor entre 0 y 100.
  Sólo se muestra si el número de usuarios que ya han respondido a la encuesta, es suficientemente significativo. En caso contrario, se visualiza In Progress. El criterio porcentual de encuestas ya respondidas para considerar significativo el valor del indicador, se define en el fichero de configuración del módulo de test, y se parametriza entre 0 y 100.
- A.score: Awareness score, o puntuación de concienciación. Sólo aplica a entrenamientos derivados de una campaña de phishing previa o en curso, en caso contrario se muestra NA.
  Se calcula como el porcentaje de usuarios que han aprobado la encuesta de training(*), menos el porcentaje de usuarios que han sido víctimas del phishing (considerando como tal, clicar el enlace y/o introducir credenciales en la web falsa). Por tanto, es un valor entre -100 y 100.
  Sólo se muestra, si de nuevo el número de usuarios que han respondido ya al test es suficientemente significativo según el criterio del fichero de configuración.

  (*) El criterio de aprobado, es diferente al del 50% que se indica orientativamente al usuario a la hora de rellenar el test. En este caso, se emplea el criterio que determina el administrador, también en el fichero de configuración, entre 0 y 100.

  El A-score pretende sintetizar en un número, el grado de resiliencia de la organización frente a ataques de phishing. Con el objetivo de compararlo con otras similares, o bien con iteraciones futuras de campañas de concienciación teórico-prácticas mediante el uso del aplicativo propuesto.
  - Un valor cercano al 100, indicaría que los usuarios obtienen muy buenas calificaciones en los test teóricos, y además no han sido víctimas del ataque simulado de phishing.
  - Un valor cercano al -100, indicaría todo lo contrario: pésimas notas en los cuestionarios, y la gran mayoría habrían sido phisheados en el ejercicio práctico.

- Un valor cercano al punto intermedio, el 0, puede indicar que las fuerzas de ambos componentes se anulan: notas decentes en teoría, pero regulares en la práctica (phisheados), por ejemplo. En este caso, para conocer qué componente ha tenido más peso de los dos, usaríamos el T-score para aportar mayor claridad y saber dónde hay que poner más foco dentro de la organización para aumentar la ciberresiliencia frente a este tipo de ataques.

Resumiendo, en base a los valores de ambos indicadores, podríamos construir un cuadro como el siguiente, para decidir cómo proseguir dentro de la organización en cuanto al entrenamiento de concienciación en técnicas de ingeniería social.

Tabla 3. Relación A-score y T-score, y acciones recomendadas (elaboración propia)

A-Score	T-Score	% Phished	Recomendación	Motivo
Alto	Alto	Bajo	Continuar ritmo actual de entrenamientos de concienciación	Está dando buenos resultados, no se requieren cambios de rumbo
Bajo	Bajo	Se infiere que será más alto de lo deseable	Incrementar frecuencia o calidad de los entrenamientos	Situación totalmente indeseada. No se ve a los empleados preparados a nivel teórico ni práctico para afrontar ataques de ingeniería social
Intermedio	Alto	Se infiere que será alto	Reforzar campañas simuladas de entrenamiento phishing	Será la situación más frecuente, si las campañas de phishing son difíciles de detectar. Los usuarios demuestran buen conocimiento a nivel teórico, pero fallan en la detección práctica
Intermedio	Bajo	Alto o bajo	Incrementar frecuencia o calidad de los entrenamientos	Situación anómala, a nivel teórico hay problemas, pero en la práctica los errores tampoco son excesivamente elevados
Intermedio	Intermedio	Intermedio	Incrementar frecuencia o calidad de los entrenamientos. Introducir más variedad	Se está en tierra de nadie. Es necesario mayor compromiso

RF29: adicionalmente al fichero de configuración para la sección de phishing, existirá otro para el módulo de tests de autoaprendizaje. Dado que éste también hace uso de la base de datos del motor de phishing, requerirá los mismos parámetros que el anterior (API key, URL del sitio web del motor). Adicionalmente, contendrá los necesarios para que el módulo de encuestas funcione adecuadamente, a saber:

- Plantilla de correo a utilizar para el envío de los enlaces con los tests de concienciación.
- Perfil SMTP para dichos envíos (host, origen, usuario y contraseña).
- Título del email a enviar.

- URL base de la encuesta, a la que luego se concatenará el slug creado en base de datos (descrito anteriormente).
- Umbrales para los indicadores A-score y T-score:
    - Porcentaje mínimo de tests completados por los usuarios, a partir del cual se considera que los resultados globales son significativos a efectos de mostrar los valores de los KPI (entre 0 y 100).
    - Nota mínima obtenida por los usuarios en el test, a efectos de considerarlo apto para el cálculo del A-score (recordemos, A-score = % aptos - % phised, valor entre -100 y 100).

RF30: Tanto en el módulo de phishing como en el de tests de autoaprendizaje, cualquier error a la hora de rellenar un formulario se le indicará al administrador en pantalla.

RF31: El intento de acceso a un objeto inexistente en base de datos, mostrará un error.

RF32: Además de las funcionalidades necesarias para dar cobertura a los requisitos funcionales de los módulos de phishing y de cuestionarios de autoaprendizaje, existirán secciones estáticas que no requerirán login:

- Sección de ayuda, con una guía de instalación.
- Sección about, con una introducción básica de en qué consiste el aplicativo, y su funcionalidad.
- Sección licencia, indicando el detalle de la licencia Open Source seleccionada para el proyecto.
- Sección contacto, con información del autor.

La primera de ellas, se ubicará en la cabecera del aplicativo, y estará siempre visible. Ídem para las otras tres, pero en este caso, se localizan en el footer.

RF33: El aplicativo debe poder ser instalable al menos, desde entornos Windows, aunque en la práctica lo es también en Linux

RF34: El aplicativo debe poder ser usable desde un PC. No se requiere web responsiveness.

RF35: Debe generarse una Prueba de Concepto, que demuestra que el aplicativo tiene utilidad práctica. Su paquetización e instalación en un servidor y entorno corporativo reales, excede el alcance del TFM.

4.3 Diseño de casos de prueba

A partir de los requisitos funcionales mencionados en el apartado anterior, se derivan los casos de prueba necesarios para verificar que, efectivamente, la funcionalidad solicitada está cubierta por el aplicativo.

Se indican a continuación, requisito a requisito.

TC_RF01: se verificará que sólo las páginas estáticas y la principal, son accesibles para usuarios anónimos.

Se probará asimismo que en el intento de acceso a una página que requiera autenticación, se muestra el formulario de login y tras la introducción exitosa de credenciales, se redirige de nuevo a la página deseada.

TC_RF02: se verificará que, una vez recibido el correo con el enlace para realizar el test de autoaprendizaje, cualquier usuario puede acceder al recurso sin necesidad de logarse, ya que es reconocido por el slug (limitación: reenvío del correo a un tercero, o ataque por fuerza bruta para adivinar slugs).

TC_RF03: se verificará que en el menú principal, los usuarios sólo verán la información general (en lugar de las opciones disponibles de administración), a no ser que estén autenticados.

TC_RF04: se verificará que, desde la sección de administración, podrán efectivamente tratarse los objetos asociados a la funcionalidad de los cuestionarios de concienciación. También que los objetos asociados al módulo de Phishing en cambio, sólo se podrán gestionar desde el propio aplicativo (una vez autenticados), o desde la propia web del motor de Phishing.

TC_RF05: se verificará que para la funcionalidad de ataques de phishing simulados, todas las acciones necesarias se pueden llevar a cabo sin salir del aplicativo.

TC_RF06: grupos de usuarios. Respecto a esta entidad, se verificará que una vez logado en el aplicativo a desarrollar, un usuario administrador podrá crear grupos nuevos de usuarios a partir de ficheros CSV, gestionar los grupos existentes, eliminarlos, y acceder a su información de detalle.

TC_RF07: plantillas de email de phishing. Referente a estos objetos, se verificará que una vez logado en el aplicativo a desarrollar, un usuario administrador podrá crear nuevas plantillas de email de manera simple a partir de un formulario, gestionar las existentes, eliminarlas, y acceder a su información de detalle.

TC_RF08: plantillas de páginas de aterrizaje. Respecto a esta entidad, se verificará que una vez logado en el aplicativo a desarrollar, un usuario administrador podrá crear nuevas plantillas de páginas de aterrizaje de manera simple a partir de un formulario, gestionar las existentes, eliminarlas, y acceder a su información de detalle.

TC_RF09: perfiles de correo SMTP (Simple Mail Transfer Protocol). Referente a éstos, se verificará que una vez logado en el aplicativo a desarrollar, un usuario administrador podrá crear nuevos perfiles de correo de manera simple a partir de un formulario, gestionar los existentes, eliminarlos, y acceder a su información de detalle.

TC_RF10: campañas de Phishing. Referente a esta entidad, se verificará que una vez logado en el aplicativo a desarrollar, un usuario administrador podrá crear una nueva campaña en base a los inputs definidos en el requisito RF10, gestionar las campañas existentes visualizando los elementos indicados en el mismo requisito, y transicionar los estados conforme a las reglas allí descritas. Se comprobará asimismo que en el detalle de la campaña se obtiene toda la información solicitada y estadísticas asociadas, y que las descargas en fichero CSV están disponibles en el formato convenido.

TC_RF11: se validará que en la creación de campañas se realiza el envío masivo de emails, y que los estados asociados a las mismas se actualizan conforme a los eventos producidos, sólo hasta su cierre.

TC_RF12: se validará la existencia del fichero de configuración para el uso del API del motor de phishing, con los parámetros indicados, y su funcionalidad efectiva.

TC_RF13: se verificará que para la funcionalidad de gestión de cuestionarios de autoaprendizaje, todas las acciones necesarias se pueden llevar a cabo sin salir del aplicativo.

TC_RF14: se comprobará que efectivamente se pueden realizar dichos tests, que su generación no tiene limitaciones, y que su base de datos es propia e independiente de la del módulo de phishing.

TC_RF15: se verificará que la estructura de los cuestionarios, es conforme a la definición del requisito funcional RF15. Que no hay limitaciones en cuanto a números de preguntas y respuestas, y que se observan las cinco restricciones mencionadas, además de que efectivamente pueden ser creados mediante un archivo CSV con el formato convenido.

TC_RF16: gestión de plantillas de cuestionarios. Referente a esta entidad, se verificará que una vez logado en el aplicativo a desarrollar, un usuario administrador podrá crear nuevas plantillas de manera simple a partir de un formulario cargando un archivo CSV (generando un error en caso de no observar las restricciones del RF15), gestionar las existentes, eliminarlas,

y acceder a su información de detalle, que mostrará entre otros elementos, qué respuestas son verdaderas o falsas.

TC_RF17: gestión de campañas de concienciación. Respecto a estos objetos, se verificará que una vez logado en el aplicativo a desarrollar, un usuario administrador podrá gestionarlos extremo a extremo, pudiendo ejecutar una nueva basada en una campaña existente de phishing, o un grupo de usuarios.

TC_RF18: se verificará que la creación de una campaña de entrenamiento basada en tests de autoaprendizaje de un grupo de usuarios, se crea tal y como se indica en el requisito funcional RF18.

TC_RF19: se verificará que la creación de una campaña de entrenamiento basada en tests de autoaprendizaje de una campaña de phishing, se crea tal y como se indica en el requisito funcional RF19.

TC_RF20: se comprobará que en la lista de campañas de entrenamiento de autoaprendizaje en curso, se muestran todos los elementos y de la forma indicada en el RF20, se puede acceder al detalle de las campañas, y se pueden eliminar o finalizar las campañas en curso, según corresponda en base a la definición indicada en el propio requisito.

TC_RF21: campañas de entrenamiento. Se comprobará que, en el detalle de la campaña, se obtiene toda la información solicitada y estadísticas asociadas de campaña general y usuarios particulares, y que las descargas en fichero CSV están disponibles en el formato convenido.

TC_RF22: se verificará que las campañas de test de autoaprendizaje generan el envío masivo de emails en todos los casos, independientemente del origen de la campaña. Y que lo hacen en base a la configuración indicada en el fichero correspondiente.

TC_RF23: se comprobará que las plantillas de email que contienen el enlace al cuestionario, se comportan correctamente, muestran en texto adecuado, y formatean las variables customizadas (nombre de usuario, y enlace del test).

TC_RF24: se comprobará que los slugs se generan adecuadamente para cada usuario en base de datos (en tiempo y formato), y se incluyen correctamente en las plantillas a la hora de generar los emails de envío para realizar los test.

TC_RF25: se verificará que los usuarios reciben el email con el enlace correcto, que pueden acceder al mismo para realizar la encuesta, que las preguntas y respuestas se muestran adecuadamente, y que tras hacer el POST de la misma reciben su nota, en base a la lógica

definida, así como que visualizan la plantilla de test nuevamente, indicando preguntas y respuestas verdaderas y falsas.

TC_RF26: se comprobará que el enlace de la encuesta sólo puede ser consumido una vez, y siempre y cuando la campaña siga activa. En caso contrario, se le presentará un error al usuario.

TC_RF27: se comprobará que el algoritmo de puntuación de los tests, es conforme a la definición del requisito funcional RF27.

TC_RF28: se verificará que existen los dos indicadores, A-score y T-score, que sólo se muestran cuando aplica en función del tipo de campaña y/o de las estadísticas en base a lo indicado del fichero de configuración, y que el cómputo de los mismos es correcto en base al algoritmo descrito en el RF28.

TC_RF29: se comprobará la existencia del fichero de configuración para el módulo de test de autoaprendizaje, que existen los parámetros definidos, y que aplican convenientemente en los casos necesarios.

TC_RF30: se verificará que se muestran errores en los formularios de configuración, en caso de ser rellenados de forma incorrecta.

TC_RF31: se comprobará que el intento de acceso a un id de objeto inexistente en base de datos, muestra un error.

TC_RF32: se revisará que existen las secciones estáticas solicitadas en el requisito funcional RF32 (ayuda, about, licencia, contacto), y que se ubican en la región adecuada de la pantalla.

TC_RF33: se comprobará que el aplicativo puede instalarse en un entorno Windows10.

TC_RF34: las pruebas de validación, se llevarán a cabo desde un PC estándar.

TC_RF35: se realizará la prueba de concepto (PoC) en un servidor de pruebas en PC, dentro de una LAN doméstica.

4.4 Selección de motor phishing Open Source

Como se comenta en el requisito funcional RF12 y otros relacionados, para desarrollar un aplicativo software open-source, es de gran ayuda apoyarse en soluciones existentes y probadas. Puesto que hemos analizado en la sección 2.4.4 las principales opciones en cuanto a motores open-source para realizar campañas de phishing, integraremos en nuestra solución

la que consideramos más óptima, desde el punto de vista de la simplicidad y economía de tiempo.

Si analizamos el listado de las 10 soluciones propuestas, vemos que la más adecuada para nuestros objetivos, es Gophish (2013). Los motivos, son varios:

- Es una de las plataformas más populares y utilizadas, con una muy importante comunidad de usuarios haciendo contribuciones, y recibiendo soporte continuo de su creador.
- Es multiplataforma. Existen archivos binarios compilados de instalación sin dependencias, tanto para Windows, como Linux y MacOS. De este modo, cumplimos al menos en este ámbito, el requerimiento RF33.
- Lo más importante, quizás: GoPhish dispone de un API (Gophishb, 2013) desarrollado en Python (2020), para poder comunicarse con el motor desde aplicaciones de terceros. Esto es perfecto para nuestros intereses, cubriendo así el requerimiento RF12.

  Además, python es un lenguaje de programación interpretado de alto nivel, muy popular en la actualidad (ver figura a continuación), utilizado profusamente en muchas de las compañías más importantes del planeta, y que se ha puesto de moda para su empleo en IA y Machine learning. Su gran difusión, permite que las opciones de desarrollo del aplicativo que conecte con el motor, sean muy amplias.

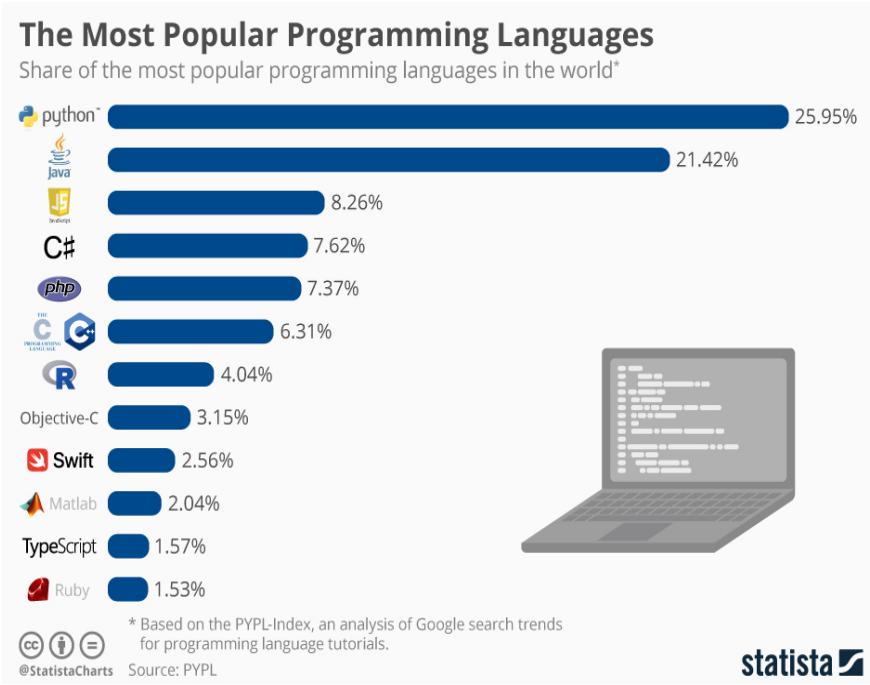

Ilustración 16. Lenguajes de programación más populares (Statista, 2019)

4.5 Selección de tecnología a emplear

En base a los requerimientos funcionales, en los que se requiere por una parte capacidad de gestión del aplicativo por parte de un administrador, pero por otro lado también interacción remota por parte de los usuarios que llevan a cabo encuestas de autoaprendizaje, parece claro que la opción natural para diseñar la solución sea una aplicación web.

Además, dado que se solicita interacción con el motor de phishing Gophish, escrito en Go pero con una interfaz API desarrollada en Python, utilizaremos un framework de desarrollo de aplicaciones web basado en este último lenguaje, llamado Django (2005).

Se trata de un framework bastante popular, licenciado como BSD en 2005, dentro del top 10 de uso en 2020 según Statista ((2020), ver figura a continuación). Sus principales características son que permite construir aplicaciones con rapidez, que incorpora gran cantidad de funcionalidad integrada que habilita por ejemplo la gestión de autenticación, administración de contenidos, etc., y proporciona también muchas funciones de seguridad, para evitar por ejemplo ataques de XSS o CSRF. Además, es versátil y escalable.

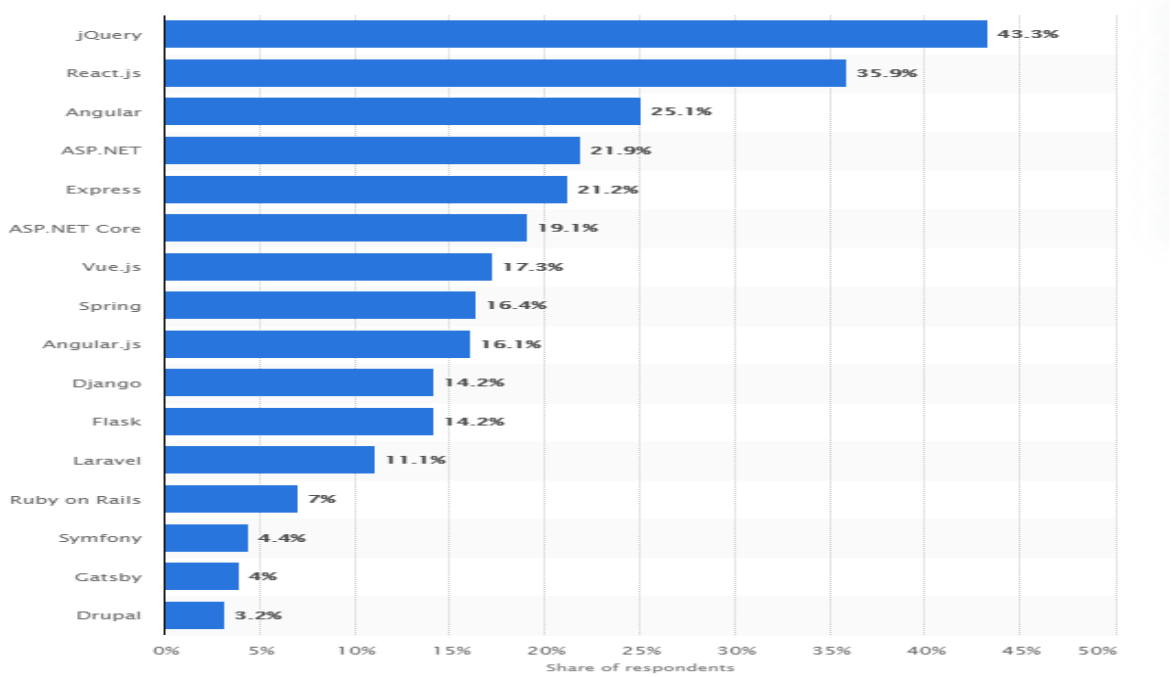

Ilustración 17. Frameworks web más populares (Statista, 2021)

4.6 Diseño de arquitectura

Los desarrollos de la aplicación web, como se indicó, se apoyarán en el framework Gophish para realizar las campañas de phishing, y se codificarán en Django. Django, utiliza el patrón de diseño MVC (modelo-vista-controlador).

Éste es un patrón reconocido como buena práctica de programación, dado que permite desacoplar los componentes del código entre sí, de modo que un cambio en uno de ellos no requiera alterar aquellos otros con los que se relaciona, otorgando gran flexibilidad y modularidad a la solución:

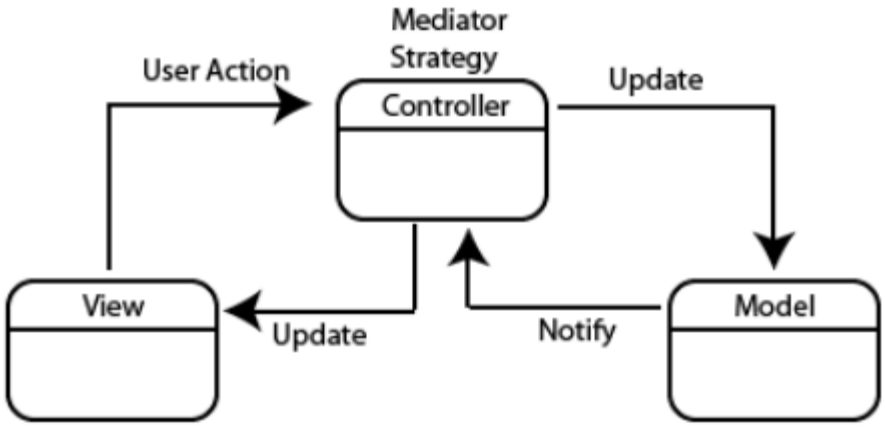

Ilustración 18. Patrón de diseño MVC (Blanco, 2021)

El modelo, contiene la información que reside en la aplicación, como es la asociada al esquema de base de datos. Dicha información es modificada por el controlador, que a su vez recibe información del propio modelo para procesarla.

El controlador, media entre el modelo y las vistas, y contiene la lógica asociada, los algoritmos que procesan la información de modelos y vistas en base a los inputs recibidos de base de datos y usuarios, respectivamente.

Las vistas, son los componentes con los que interactúa el usuario, llamadas templates en Django. Éstas muestran la información de las páginas web en base a lo recibido del controlador y el usuario, y manejan la disposición de los diferentes elementos en pantalla.

En cuanto a la arquitectura de la aplicación web en sí, a la que llamaremos Veraphish, es la siguiente:

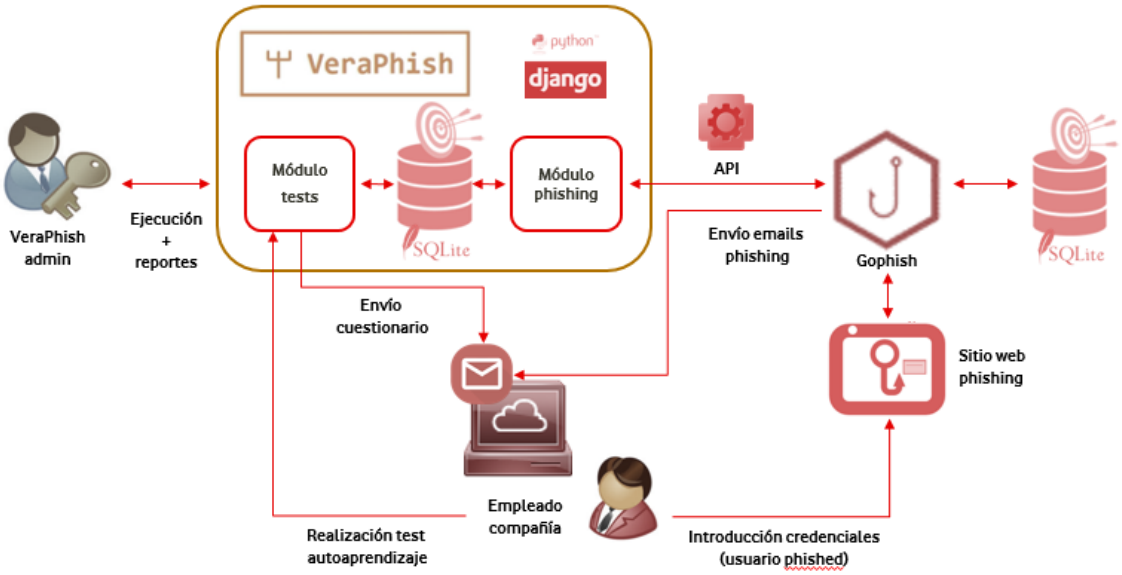

Ilustración 19. Diseño arquitectura Veraphish (elaboración propia)

Como se puede apreciar, la aplicación web Veraphish tiene dos módulos diferenciados, para cubrir los dos objetivos buscados en el marco de la concienciación de empleados frente a ataques de ingeniería social mediante técnicas de email phishing: el módulo de encuestas o test de autoaprendizaje, y el módulo para realizar ataques de phishing simulados.

Ambos están desarrollados en Django/python, y hacen uso de la base de datos SQLite del propio aplicativo. Dicha base de datos contiene los modelos que se verán en el siguiente apartado, que dan soporte tanto a las dos funcionalidades específicas, como a los módulos de administración de Django (relativos al funcionamiento del servidor de pruebas, el control de cambios en base de datos, o la autenticación de usuarios, por ejemplo).

Los usuarios administradores, se conectarán al aplicativo para, de manera autenticada, poder llevar a cabo las labores de gestión: de usuarios, creación, gestión, borrado y exportación de resultados de campañas de phishing, e idénticamente con respecto a los test de autoaprendizaje y las posteriores campañas asociadas a los mismos.

Recordemos que, aunque no se ha dibujado en el esquema por simplicidad, la configuración del aplicativo pasa también por parametrizar los ficheros de configuración mencionados en los requisitos funcionales. A priori, esta es una labor que se realiza únicamente durante la instalación del entorno.

Tenemos así, recordando en base a los requisitos, una funcionalidad de gestión de campañas de phishing en la aplicación web, que conectada mediante el API python al motor open-source de Gophish, permite orquestar la ejecución de las mismas. El motor tiene su propia base de

datos también SQLite, para albergar los grupos de usuarios, plantillas de email, perfiles SMTP, páginas de aterrizaje, y estadísticas asociadas a las campañas vivas.

Como se aprecia en el esquema, la creación de una campaña de phishing dispararía el envío de emails a los empleados de la compañía. En caso de que éstos cayesen en el engaño y pinchasen los enlaces del mail supuestamente fraudulento, serían redigiridos al sitio web de phishing, donde introducir credenciales u otros datos confidenciales a robar. Las interacciones con el email por parte de empleados, ya sea apertura de email, clic en el enlace o introducción de datos, serían convenientemente recogidos por el motor de phishing y su base de datos, y por tanto se reflejarían en el módulo de phishing de Veraphish cuando el usuario administrador consultase el estado de dicha campaña.

Por otro lado, tenemos el módulo de tests de autoaprendizaje. Respecto a los empleados, en este caso para una nueva campaña creada, éstos recibirían un email con el enlace de invitación para la realización del test. El enlace sería único por persona, y se almacenaría en la base de datos por el aplicativo en el momento del lanzamiento de la campaña. Una vez consumido por parte de dicho usuario, le permitiría (una única vez, y siempre y cuando la campaña siga en curso), rellenar el test, obtener su puntuación, y verificar cuáles eran las respuestas acertadas.

Con respecto al administrador de la aplicación, conforme los usuarios vayan completando los test recibirán los resultados en base de datos y podrán exportarlos individual o colectivamente, además de dispones de los dos indicadores A-score y T-score, siempre y cuando se cumplan las condiciones indicadas en la definición en requisitos funcionales de los mismos, y de la parametrización del fichero de configuración al efecto.

4.7 Diseño de modelo de datos

Como ya se indicó anteriormente en el diseño de arquitectura, Veraphish contiene dos bases de datos independientes. La asociada la funcionalidad de phishing, es la propia de la solución estándar de Gophish, y por tanto no ha sido definida en el marco del TFM.

Consultando la documentación disponible (Gophishb, 2013), podemos visualizar los modelos de datos de cada componente que interviene en la creación de las campañas de phishing, a saber:

gophish.models.User

Atributos

- id (int): User ID
- first_name (str): Nombre
- last_name (str): Apellidos
- email (str): Email
- position (str): Puesto de trabajo

gophish.models.Group

Un grupo contiene uno o más objetos models.User. El nombre debe ser único.

Atributos

- id (int): ID de Grupo
- targets (list(models.User)): Lista de models.User
- name (str): Nombre del grupo
- modified_date (optional: datetime.datetime): Fecha de creación diferida de Grupo

gophish.models.Template

Una plantilla incluye nombre y contenido del propio email. El nombre debe ser único.

Atributos

- id (int): ID de plantilla
- name (str): Nombre de plantilla
- html (str): HTML de la plantilla
- text (str): Texto de la plantilla
- modified_date (optional: datetime.datetime): Fecha de creación diferida de la plantilla
- attachments (list(models.Attachment)): Lista de adjuntos

gophish.models.Page

Páginas de aterrizaje. El nombre debe ser único.

Atributos

- id (int): ID de página
- html (str): HTML de la página

- name (str): Texto de la página
- modified_date (optional: datetime.datetime): Fecha de creación diferida
- capture_credentials (bool default:False): Si la página capturará o no credenciales
- capture_passwords (bool default:False): Si la página capturará o no contraseñas
- redirect_url (str): URL a la que redirigir a los usuarios tras la introducción de datos

gophish.models.SMTP

Perfil de correo SMTP que incluye toda su configuración asociada. Nombre único.

Atributos

- id (int): ID de perfil SMTP
- name (str): Nombre de perfil
- interface_type (str): Tipo de conexión (sólo SMTP disponible por el momento)
- host (str): Host:puerto del servidor SMTP
- from_address (str): Dirección del remitente del envío
- ignore_cert_errors (bool): Si se ignoran o no los errores de certificado SSL
- modified_date (optional: datetime.datetime): Sello temporal del último cambio en el perfil.

gophish.models.Result

Resultados individuales de un models.User en una campaña de phishing.

Atributos

- id (int): ID del resultado
- first_name (str): Nombre
- last_name (str):Apellidos
- email (str): Dirección de email
- position (str): Puesto de trabajo
- ip (str): Última dirección IP
- latitude (float): Coordenada latitud de dicha IP
- longitude (float): Coordenada longitud de dicha IP
- status (str): Estado del usuario dentro de la campaña de phishing

gophish.models.Campaign

Cada atributo de la campaña (Grupo, Plantilla, Página...), Gophish lo referencia por nombre. Por tanto, esos son los identificadores a utilizar a la hora de crear una campaña nueva a través del API.

Atributos

- id (int): ID de resultado
- results (list(models.Result)): Lista de resultados de la campaña
- name (str): Nombre de la campaña
- status (str): Estado de la campaña
- created_date (optional: datetime.datetime): Fecha de creación
- send_by_date (optional: datetime.datetime): Fecha de envío de emails
- launch_date (optional: datetime.datetime): Fecha de lanzamiento
- template (models.Template): Plantilla de email a utilizar
- page (models.Page): Página de aterrizaje a utilizar
- smtp (models.SMTP): Perfil SMTP a utilizar
- url (str): URL a usar al construir los emails de phishing

gophish.models.Stat

Estadísticas de resultados de una campaña. Se devuelven como parte del modelo CampaignSummary.

Atributos

- total (int): Número total de usuarios de la campaña
- sent (int): Número de emails enviados con éxito
- opened (int): Número de emails abiertos
- clicked (int): Número de emails en los que el usuario ha pinchado el enlace de phishing
- submitted_data (int): Número de usuarios que han introducido datos en el formulario
- error (int): Número de errores en el envío de emails

gophish.models.CampaignSummary

Vista resumida de una campaña. Muestra los resultados a alto nivel, de manera más eficiente que teniendo que consultar los resultados completos de la misma, sobre todo para un número de usuarios elevado. Este objeto se crea automáticamente con cada campaña.

Atributos

- id (int): ID de resultado
- name (str): Nombre de campaña
- created_date (optional: datetime.datetime): Fecha de creación
- send_by_date (optional: datetime.datetime): Fecha de envío de email
- launch_date (optional: datetime.datetime): Fecha de lanzamiento
- status (str): Estado actual de la campaña
- stats (list(models.Stat)): Lista de resultados (models.Stat)

En cuanto a la configuración del modelo de datos para la parte de los tests de autoaprendizaje, aprovecharemos la potencia de Django para definir modelos abstrayéndonos de la complejidad de lidiar con las sentencias SQL (mapeo objeto-relacional, ORM). Los modelos definidos, cinco en total, son los indicados a continuación:

survey.models.Survey

Objeto principal para representar una plantilla de test de autoaprendizaje.

- Survey name (charfield): Nombre de la plantilla
- Answer key (charfield): String de respuestas asociadas (verdaderas y falsas)
- Score key (charfield): Igual que el anterior, para las puntuaciones
- Date (datetime): Fecha de creación de la plantilla (automática)

Survey.models.Question

Objeto pregunta, dentro de un test de autoaprendizaje. Relacionado con éste, en relación de base de datos many-to-one.

- Survey (survey.models.Survey): Clave externa. Se aplica la propiedad on_delete=models.CASCADE, de tal modo que si se elimina el objeto Survey, se eliminan también de la tabla Question todos los objetos relacionados
- Text (charfield): Texto asociado a la pregunta en sí

Survey.models.Answer

Objeto respuesta, dentro de una pregunta del test de autoaprendizaje. Relacionado con ésta, en relación de base de datos many-to-one.

- Question (survey.models.Question): Clave externa. Se aplica la propiedad on_delete=models.CASCADE, de tal modo que si se elimina el objeto Question, se eliminan también de la tabla Answer todos los objetos relacionados
- Text (charfield): Texto asociado a la respuesta en sí
- Valid (boolean): Indica si la respuesta es verdadera (True) o falsa (False)

Survey.models.TrainingCampaign

Objeto en base de datos que representa una campaña de entrenamiento mediante test de autoaprendizaje.

- Training name (charfield): Nombre de la campaña de entrenamiento
- Target group name (charfield): Nombre del grupo de usuarios o campaña en base de datos de Gophish
- Number of users (integerfield): Número de usuarios en la campaña
- Survey template name (charfield): Nombre de la plantilla de test de autoaprendizaje
- Date (datetime): Fecha de creación
- Status (charfield): Estado de la campaña, ACTIVE o COMPLETED. En caso de eliminación, se suprime de la base de datos
- Source (charfield): Entrenamiento basada en grupo de usuarios (GROUP), o campaña de phishing (CAMPAIGN)

Survey.models.TrainingItem

Objeto que representa a un usuario individual, dentro de una campaña de entrenamiento dada. Relacionado con los Survey.models.TrainingCampaign en una relación de base de datos many-to-one.

- Training campaign (survey.models.TrainingCampaign): Clave externa. Se aplica la propiedad on_delete=models.CASCADE, de tal modo que si se elimina el objeto TrainingCampaign, se eliminan también de la tabla TrainingItem todos los objetos usuario relacionados
- Slug (charfield): Cadena de 50 caracteres alfanuméricos pseudoaleatorios para generar un link de test de aprendizaje único por usuario
- First name (charfield): Nombre del individuo a encuestar
- Last name (charfield): Apellidos
- Email (charfield): Email del usuario
- Position (charfield): Puesto ocupado en la compañía, equipo al que pertenece
- Phishing status (charfield): Estado del usuario en la campaña de phishing asociada. Valor por defecto en creación si no existe tal estado, NA. En caso contrario, uno de los siguientes: Campaign created, Email Sent, Email opened, Clicked Link, Submitted Data
- Survey results (charfield): Cadena de caracteres indicando los resultados indicados por el individuo en el test, en formato binario (‘1’ indica respuesta marcada y ‘0’ lo contrario), separados por ‘-’. N/A en caso de no respuesta todavía.
- Score (charfield): Puntuación obtenida en el test en formato real (0-100), N/A en caso de no respuesta

Como se ve, existen tres tablas relacionadas entre sí para almacenar la información relativa a los test de autoaprendizaje, y dos adicionales para la gestión de las campañas de entrenamiento que hacen uso a su vez de las anteriores. El diagrama sería como sigue:

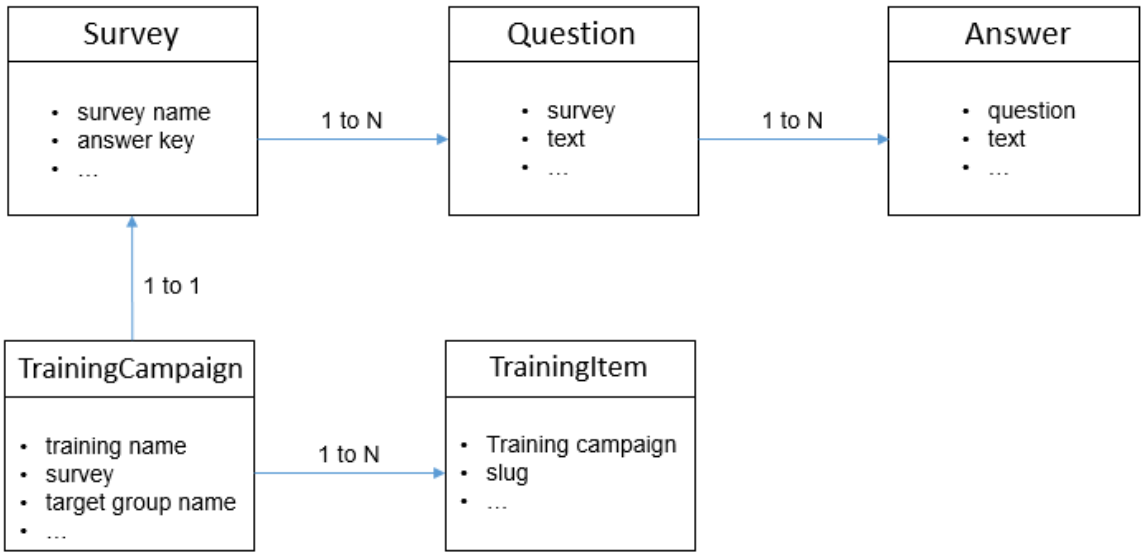

Ilustración 20. Modelo de base de datos de módulo de tests Veraphish (elaboración propia)

4.8 Desarrollo del interfaz del motor de phishing

El desarrollo de la interfaz que haría uso del API de Gophish mencionado con anterioridad (Gophishb, 2013), ha sido la primera pieza de código generada. Dado que el API es nativo en Python y el framework Django de desarrollo web se basa en el mismo lenguaje de programación, la elección natural era implementarlo con dicha tecnología.

Básicamente las etapas del desarrollo han sido las siguientes:

a) Análisis del API y el modelo de datos de cada componente (*)
b) Desarrollo del código en un script python aislado
c) Pruebas individuales de dicho componente del código fuente

(*) Los componentes desarrollados por orden cronológico han sido la interfaz para la creación, gestión y eliminación de los siguientes objetos: Grupo de Usuarios, Plantilla de Email, Template de Página de Aterrizaje, perfiles SMTP, y por último, Campaña de Phishing.

El script python de pruebas recibía los parámetros para gestionar las solicitudes realizadas por línea de comandos, y el entorno simplemente consistía en una máquina Windows 10 con python más el API de Gophish instalado, y el propio motor de Gophish (instalado, configurado y ejecutándose en local). Los test serían totalmente transferibles a un entorno Linux sin problema, puesto que el código es portable, y recordemos que Gophish es multiplataforma.

4.9 Pruebas del motor de phishing

Como se mencionó en la sección anterior, las pruebas de desarrollo del motor consistieron en verificar la funcionalidad correcta de las opciones implementadas en la interfaz:

a) Creación, gestión y eliminación de Grupos de Usuarios.
b) Creación, gestión y eliminación de Plantillas de Email.
c) Creación, gestión y eliminación de Páginas de Aterrizaje.
d) Creación, gestión y eliminación de perfiles SMTP.
e) Creación, gestión (incluyendo finalización y lanzamiento diferido), y eliminación de Campañas de Phishing.

El modus operandi consistía en:

1. Ejecutar la prueba desde el script generado ad-hoc, mediante línea de comandos, generando ficheros con datos de prueba al efecto cuando era necesario.
2. Verificar el resultado del script en la ventana de ejecución.
3. Comprobar en la GUI del motor de Phishing corriendo en local, que los resultados concordaban con lo esperado, para dar la prueba por válida.

4.10 Desarrollo de la aplicación web

Los pasos para el desarrollo del aplicativo, están íntimamente ligados con la idiosincrasia del framework Django. Grosso modo, las actividades llevadas a cabo han sido las siguientes:

a) Diseño de la aplicación. Listado de funcionalidades a incluir tanto en la sección de phishing como en la de tests, categorías jerárquicas en las mismas, páginas estáticas, layout y estilo de las páginas (todo ello a nivel teórico, en papel), anterior al propio desarrollo.
b) Creación de la estructura básica del desarrollo: proyecto de aplicación Django y configuración inicial de base de datos.
c) Creación de usuario de administración (se ha desarrollado un formulario para generación de usuarios extra, documentado a junto al código, pero que se deshabilita por defecto por motivos de seguridad).
d) Creación de app de administración para autenticación de usuarios (haciendo uso del framework de autenticación django.contrib.auth y sus modelos predefinidos).
e) Creación de sección de phishing (app SETAphish, en el argot Django).
    a. Gestión de Grupos de usuarios.
    b. Gestión de Plantillas de Email.

c. Gestión de Páginas de Aterrizaje.
d. Gestión de perfiles SMTP.
e. Gestión de campañas de Phishing extremo a extremo.

Para la creación de esta funcionalidad, reutilizamos todo el código para hablar con el motor de phishing y su base de datos asociada, cuya definición y desarrollo se describió en la sección 4.8.

En cada caso, el proceso de generación del código fuente consistía en:

1. Creación de las URLs e inclusión de las mismas en el fichero urls.py asociado del projecto Django. Vinculación de las mismas con la función del controlador.
2. Creación del controlador, con la lógica de negocio asociada a la funcionalidad, que hacía llamadas al módulo auxiliar para gestionar el motor de Gophish. En el propio código del controlador, se determina tanto la vista a utilizar para la gestión de la entrada/salida, como el contexto a trasladarle (diccionario de pares atributo-valor que la vista podrá utilizar).
3. Creación de la template con la vista, para el layout (*) de la parte visual de la web (código HTML y javascript), y el control de la entrada/salida de datos.

(*) En el desarrollo de la primera de las funcionalidades, se añade la actividad extra de definir las CSS que serían utilizadas en el proyecto. Se ha optado (por simplicidad y escasez de criterio estético por mi parte), por utilizar las conocidas CSS de Yahoo! (PureCSS, 2014) con ligeras customizaciones posteriores en local.

f) Creación de sección de tests de autoaprendizaje (app SETAsurvey, en argot Django).
   a. Gestión de Plantillas de Cuestionarios.
   b. Gestión de Campañas de entrenamiento.

En este caso, como se comentó, además de desarrollar toda la lógica de negocio de los controladores de cada subcomponente del módulo, se definió el modelo de datos específico (no se heredaba de Gophish, como en el caso anterior).

El proceso de desarrollo se llevó a cabo en tres pasos como en el apartado e), con la salvedad de que en este caso la cantidad de código necesario para la gestión de: plantillas de cuestionarios, sus validaciones, la generación y gestión de componentes de base de datos como los slugs, los envíos de emails, y los indicadores A-score y T-score asociados a las propias campañas, fue notablemente superior al caso de la sección de phishing, básicamente por dos razones:

- No se reaprovechaba un desarrollo open-source existente, sino que se partía desde cero.
- La lógica asociada a toda la gestión de los tests es más compleja.

g) Por último, se procedió a la creación de las páginas estáticas de acceso público. En este caso, el proceso se simplifica, puesto que la vinculación de las URLs con la vista es inmediata y no requiere código de controlador. Como se indicó en el requisito funcional RF32 en la sección 4.2, consistió en:
   a. Sección de ayuda, con una pequeña guía de instalación.
   b. Sección about, con una introducción básica de en qué consiste el aplicativo, y su funcionalidad.
   c. Sección licencia, indicando el detalle de la licencia Open-Source seleccionada para el proyecto.
   d. Sección contacto, con información del autor.

4.11 Pruebas de los componentes

En paralelo a la construcción de cada funcionalidad asociada a un módulo concreto (de Phishing o relativo a los Test de Autoaprendizaje), se fueron realizando las pruebas de cada uno de los componentes. En cada caso, se realizaban (al menos):

- Pruebas diferentes de tipo sunny day (escenario esperado o más frecuente de uso), y cuyo resultado era exitoso. Ejemplo: creación de un Grupo de usuarios a partir de un CSV bien conformado.
- Pruebas de escenarios cuyo resultado esperado era un error, como puede ser un formulario incorrectamente rellenado, la carga de un fichero con formato no esperado, el intento de acceso no autenticado a una funcionalidad que sí lo requería, o la invocación de una URL cuyo objeto era inexistente en base de datos.

4.12 Pruebas End-to-end

Una vez desarrollada toda la funcionalidad y probados los componentes de manera aislada, se procedió a llevar a cabo pruebas de funcionamiento extremo a extremo, tal y como si nos encontrásemos en el escenario real de una compañía queriendo llevar a cabo un entrenamiento de concienciación frente a ataques de ingeniería social.

Para ello, los pasos llevados a cabo, fueron los siguientes:

1. Reinstalación del aplicativo y componentes auxiliares (python, API Gophish y la propia plataforma de phishing). Esto sirvió para elaborar un sencillo manual de puesta en marcha.
2. Configuración de usuario administrador.
3. Parametrización de ficheros de configuración (de Gophish, módulo de phishing y módulos de tests).
4. Arranque de Gophish y servidor Django de pruebas.
5. Creación de un Grupo de usuarios.
6. Creación de una Plantilla de Email.
7. Creación de una Página de Aterrizaje.
8. Creación de un perfil SMTP.
9. Creación de dos campañas de Phishing, de lanzamiento inmediato y diferida.
10. Verificación de generación y envío de emails.
11. Verificación de interacción adecuada con los mismos por parte de los usuarios a entrenar, y de reflejo de la información adecuadamente en base de datos.
12. Verificación de que toda la funcionalidad asociada a los objetos creados anteriormente en Veraphish, se comportan conforme a lo esperado (visualización de detalles, exportación de estadísticas, eliminación…).
13. Creación de plantilla de Test de Autoaprendizaje.
14. Creación de dos campañas basadas en Cuestionario, para Grupos de usuarios, y Campañas en vuelo.
15. Verificación de envío de emails, y posibilidad de realizar los test conforme a lo esperado (una única vez, se muestra puntuación y resultados, y se almacenan convenientemente).
16. Verificación de que la funcionalidad asociada a los anteriores, también se comporta conforme a lo requerido (visualización de detalles de test y campañas, indicadores A-score y T-score se generan cuando procede y su valor es correcto, se pueden exportar las estadísticas, etc.).
17. Comprobación de que la página principal se puede visualizar sin estar autenticado, y se muestra la versión descriptiva de la misma. Ídem para las páginas estáticas, no requieren log in.

4.13 Selección de licencia open-source

Como se indicó con anterioridad, el objetivo del TFM es no sólo generar una solución software para la mejora del nivel de concienciación frente a ataques de ingeniería social mediante email

phishing para empresas, sino también ponerla a disposición de las mismas de manera altruista (para su uso, o explotación comercial).

Para ello, es conveniente que al hacer público el código fuente en un repositorio de acceso general, se licencie convenientemente, de tal manera que todos los usuarios conozcan los derechos y obligaciones que dicha licencia les otorga, y éstos estén orientados a la finalidad indicada en el párrafo anterior. Al fin y al cabo, el aplicativo no deja de ser un bien informático de naturaleza lógica.

Nos hemos decantado por aplicar una licencia de tipo open-source, de tal manera que se amplíen las cuatro libertades del software libre (uso para cualquier propósito, acceso a código fuente para su estudio y adaptación, libertad de distribución de copias, y mejora y publicación de las mismas (Loza, 2019), sin mantener condición alguna sobre el software derivado. Además, estas licencias son las más ventajosas desde el punto de vista empresarial, puesto que reducen los potenciales costes de desarrollo e innovación añadida, sin necesidad de abrir el código fuente modificado.

Resumiendo, el propósito es que la solución pueda tener la máxima difusión, para ser usado AS-IS por cualquier organización, o incluso construir una solución comercial sobre la misma.

Analizando el grado de popularidad y permisividad de las licencias open-source existentes hoy en día, decidimos utilizar la licencia MIT (2020). Sus principales características, son:

- Concesión de los siguientes derechos
    - Uso privado
    - Uso comercial
    - Modificación
    - Distribución
- Condiciones impuestas
    - Exige incluir una copia de la licencia y el copyright en el nuevo material licenciado
- Limitaciones
    - No hay responsabilidad alguna imputable al creador asociada al uso del software
    - El software se ofrece AS-IS, sin ningún tipo de garantía

4.14 Documentación auxiliar

El aplicativo está orientado de tal manera que su uso es absolutamente intuitivo. Tanto en la sección de ataques de phishing como en la de tests de autoaprendizaje, los elementos se disponen en el orden lógico de creación/gestión para que la funcionalidad pueda ser utilizada en un entorno real, es decir:

- Phishing: creación de grupos de usuarios, plantillas de email, páginas de aterrizaje, perfiles SMTP y, por último, campañas de phishing.
- Tests autoaprendizaje: creación de plantillas, y creación de campañas.

En cualquier caso, y dado que antes del uso se requiere, además de las habituales actividades de instalación y configuración previa de todo el entorno, una comprensión del alcance de la herramienta y su funcionalidad, se ha generado documentación adicional para soportar este ámbito.

En el paquete de Veraphish que se ubicará en un repositorio público, se incluirán los siguientes ítems:

- Archivo LICENSE, conteniendo la información asociada a la licencia MIT indicada en el apartado anterior.
- Archivo CODE of CONDUCT. Describe las normas de comportamiento que deben regir, en cuanto a la colaboración entre miembros de una eventual comunidad de usuarios que pudiese surgir en torno al proyecto.
- Archivo README. Incluye información útil con carácter general:
    - 0. Introducción. Contexto de elaboración del software, tecnología en que se basa, y funcionalidad a alto nivel que desarrolla.
    - 1. Licencia. Nuevamente, derechos y obligaciones que otorga la licencia MIT.
    - 2. Instalación. Procedimiento de puesta en marcha, checklist de 13 pasos.
    - 3. Sobre mí. Una breve reseña del autor, y datos de contacto.
- Directorio PROJECT. Contendrá todo el código fuente Django, ficheros de configuración y módulos auxiliares.
- Directorio DOCS. En él se incluirán una serie de archivos útiles para la mejor comprensión del aplicativo, y acelerar la curva de aprendizaje en cuanto a su uso. Proporciona ejemplos para todas las funcionalidades básicas del software desarrollado:
    - Fichero _README_docs.txt, explicando el contenido del directorio.
    - 0. [Gophish config file example] config.json, ejemplo de fichero json de configuración del motor de phishing GoPhish.

- 1. [SETAphish, fishing app config file example] SETAphish.cfg, ejemplo del fichero de configuración de Veraphish para phishing.
- 2. [SETAsurvey, training app config file example] SETAsurvey.cfg, ejemplo del fichero de configuración de Veraphish para encuestas de autoaprendizaje.
- 3. [Target group template example] group.csv, ejemplo de fichero para creación de Grupo de Usuarios a phishear.
- 4. [Email template example] email.html, ejemplo de fichero para creación de Plantilla de Email para Phishing.
- 5. [Landing page template example] page.html, ejemplo de fichero para creación de Plantilla de Página de Aterrizaje para Phishing.
- 6. [Email template for training surveys example] template.txt, ejemplo de plantilla para envío de emails para realización de Encuesta de Autoaprendizaje.
- 7. [Survey template example] survey.csv, y 8. [Survey template example2] survey2.csv: un par de ejemplos de plantillas de Tests de Autoaprendizaje.

Adicionalmente a lo anterior, como se comentó en los requerimientos funcionales (RF32), existen cuatro páginas web estáticas en el aplicativo con información adicional:

- Sección de ayuda, con guía de instalación.
- Sección about, con una introducción básica de en qué consiste el aplicativo, y su funcionalidad.
- Sección licencia, indicando el detalle de la licencia Open-Source seleccionada para el proyecto.
- Sección contacto, con información del autor.

En la siguiente referencia, se indica la ubicación del repositorio de código fuente (Github, 2021). A continuación, se muestran unas estadísticas básicas acerca del desarrollo, generadas automáticamente mediante SLOCCount (SLOCCount, 2001):

SLOC Directory SLOC-by-Language (Sorted)
1953 project (source_code) python=1953
0 docs (none)
0 top_dir (none)

Totals grouped by language (dominant language first):
python: 1953 (100.00%)

Total Physical Source Lines of Code (SLOC) = 1,953
Development Effort Estimate, Person-Years (Person-Months) = 0.40 (4.85)
*(Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))*
Schedule Estimate, Years (Months) = 0.38 (4.55)
*(Basic COCOMO model, Months = 2.5 * (person-months**0.38))*
Estimated Average Number of Developers (Effort/Schedule) = 1.06
Total Estimated Cost to Develop = $ 54,561
(average salary = $56,286/year, overhead = 2.40).

SLOCCount, Copyright (C) 2001-2004 David A. Wheeler

4.15 Recopilación de feedback

Debido al escaso margen temporal entre la finalización de los desarrollos y la entrega del TFM, no ha sido posible recoger feedback práctico real del uso de aplicativo en un entorno corporativo.

Sin embargo, el resultado del Trabajo ha sido compartido con: una persona de confianza dentro de mi actual organización, dedicado en el pasado a operaciones de cliente, desarrollo de software y actualmente Marketing de producto, y del mismo modo se pidió evaluarlo a la firma de consultoría en la que he desarrollado las prácticas académicas del Máster en Seguridad Informática de UNIR. A continuación, se indica el feedback recibido por ambas fuentes.

4.15.1 Comentarios del especialista

- *"Interpreto que la aplicación se debe instalar por cuenta de cada empresa que quiera utilizarla. Creo que de cara a su utilización, facilitaría mucho disponer de una versión portable o simplificar al máximo las tareas de configuración e instalación a realizar, pensando sobre todo en empresas que carezcan de recursos o conocimientos para llevarlas a cabo. Otra alternativa es que se ponga a disposición del público pero ampliando el control de la información para que un usuario sólo tenga acceso a la información propia".*

Como le indiqué, automatizar la instalación y despliegue de la solución en las instalaciones del cliente se considera una línea futura de trabajo, por falta de tiempo para abordar esta cuestión dentro del alcance del TFM.

- *De cara al uso de la misma, ¿cómo se puede asegurar a potenciales usuarios que es un software seguro, es decir, que es una aplicación segura que no va a hacer precisamente lo que dice prevenir, o de qué manera se puede fomentar su uso, ¿quizás desde INCIBE?*

Se le traslada al evaluador que el código fuente se subirá a un repositorio público con licencia MIT, y se compartirá con INCIBE por si desean mejorarlo en alguna medida. Cualquier organización podrá utilizarlo y modificarlo teniendo control total del aplicativo y sus riesgos, pero el software se entrega AS-IS.

En las líneas futuras de trabajo, se proponen un par de medidas para reducir la superficie de exposición, que son el uso de certificados TLS y el análisis estático de código fuente.

- *"Sería de interés poner a disposición de los usuarios una pequeña biblioteca de ejemplos tanto de campañas como de encuestas predefinidas, simplificaría y agilizaría su uso."*

Estaba contemplado, se incluye tanto en el paquete del código fuente que se subirá al repositorio público, como en los Anexos de la presente memoria.

- *"Se podría también incorporar información sobre el significado de los indicadores, algo similar a la Tabla 3. Relación A-score y T-score, de modo que facilite la interpretación de los resultados y acciones recomendadas. También quizás la posibilidad de disponer de información sobre la evolución de dichos indicadores a lo largo del tiempo y sucesivas campañas o encuestas."*

Me ha parecido una excelente idea. Se recoge en líneas futuras de actuación, puede ser implementado como quick-win.

- *"Pequeñas mejoras en la web, por ejemplo, ayuda con información sobre los campos a cubrir, o que no permita cargar ficheros sin haber fijado el nombre del grupo."*

Se incluye también como líneas de mejora del aplicativo. Complejidad algo superior al anterior, pero podría considerarse igualmente un quick-win.

Por último, mencionar que a modo de comentario final, indica:

"El TFM explica de forma clara el riesgo de seguridad existente y cuál es el componente clave del mismo, el factor humano, siendo la herramienta planteada necesaria y útil para fomentar la concienciación y permitir, sobre todo a empresas sin recursos, mejorar significativamente en tal aspecto."

4.15.2 Comentarios de la consultora

Se recibe asimismo feedback por parte de Óscar Rodríguez, consultor de Seguridad de la Información de la firma viguesa Inprosec (Inprosec, 2010). Su valoración es la siguiente:

- *"Personalmente como ya sabes, usamos habitualmente Gophish para campañas de phishing. Lo que me parece más interesante es lo de integrar los dos módulos, el de phishing y el de encuestas en una sola app, y ahí es donde yo le veo un grandísimo valor a la aplicación. Principalmente, por el hecho de realizar esas campañas de entrenamiento vinculando grupos de usuarios de entrenamientos de phishing, y sobre todo evaluar los resultados de las respuestas de manera centralizada y tan detallada. Nosotros no hacíamos encuestas de este tipo a nuestros clientes, alguna vez utilizábamos forms para lanzar encuestas puntuales sobre calidad de servicio, y nos estábamos planteando precisamente lanzar encuestas de concienciación, antes y después de una sesión formativa, para evaluar la efectividad. Tu app resuelve perfectamente esto y nos puede servir de gran ayuda al haberla licenciado como software libre.*

  Como conclusión, me parece una muy buena herramienta, que puede ser de gran utilidad para muchas organizaciones en la mejora de la concienciación en ciberseguridad. Como puntos positivos, destacaría:

  - *Integración con encuestas.*
  - *Facilidad de uso.*
  - *Gran detalle para analizar resultados.*
  - *Posibilidad de utilizar grupos del gophish en las encuestas.*

  Como puntos a mejorar:
  - *Obtener resultados tipo gráfico.*
  - *Añadir algún módulo que de alguna manera mida la efectividad de las sesiones de formación (que es el otro pilar de la concienciación).*

Los elementos indicados como posibles mejoras, se incluyen en la sección 5 (trabajos futuros).

5. Conclusiones y trabajos futuros

A lo largo del presente TFM se ha presentado la problemática de la ingeniería social, sus antecedentes e historia, así como los estudios y trabajos relacionados, definiciones formales, y posibles defensas tanto desde el punto de vista teórico como práctico, básicamente consistentes en desarrollar contramedidas electrónicas y programas de concienciación.

El objetivo de la aportación ha sido desarrollar una aplicación open-source SETA que permita a las organizaciones aumentar el grado de ciberresiliencia a estas amenazas, centrándonos especialmente en los ataques de email phishing. Para ello, se ha hecho un estudio exhaustivo de más de treinta fuentes diferentes, entre entidades gubernamentales, grandes actores del sector, partners de nicho, y soluciones open-source, poniendo el foco también en qué soluciones y proveedores están más al alcance de la mano de las PyMEs españolas.

Una vez identificado el contexto, se ve que en el mercado no existe una solución simple, que cubra tanto el aspecto teórico como práctico del entrenamiento de concienciación de forma personalizable, de fácil instalación, gratuito, y que vele por la privacidad de los datos de los usuarios. Este es el hueco que cubrimos con el aplicativo Veraphish.

Se han llevado a cabo pruebas de concepto en equipo local, de manera exitosa, validando la funcionalidad del aplicativo en un entorno controlado, como se indica en la sección anterior de Feedback.

Las líneas futuras de investigación y actuación, son variadas.

- Desde el punto de vista de la seguridad, se aconsejaría utilizar alguna herramienta de análisis tipo SAST (Static Application Security Analysis) para identificar y mitigar vulnerabilidades en el código de Veraphish. Cualquiera que soporte Python, podría ser válida, como Agnitio, Bandit, Checkmarx, Coverity, Fortify, SonarCloud o Veracode (ver OWASP, 2020).
- Utilización de certificados TLS para comunicación cifrada cliente-servidor.
- En cuanto a facilitar el despliegue seguro en un entorno productivo, se podría reaprovechar el trabajo llevado a cabo por otro alumno del Máster de Seguridad Informática de UNIR (Coma, 2017) para instalación segura de servidor usando aplicaciones de código abierto, automatizando el despliegue mediante Ansible. De esa manera, se ocultaría esa complejidad para el usuario en la medida de lo posible, ya que, dada la facilidad de uso del aplicativo, la instalación es la etapa que generará más fricción.
- Adición de funcionalidad de Django de internacionalización y localización, para poder traducir automáticamente el idioma del aplicativo a partir de las preferencias del navegador y aumentar su difusión.
- Hacer la aplicación web responsive, para optimizar su manejo desde dispositivos móviles.
- Incluir alguna capacidad para medir la eficacia de las campañas formativas, más allá de las propias evaluaciones prácticas mediante entrenamientos de phishing.

- En cuanto al aspecto funcional del aplicativo, se podrían mejorar sus capacidades en muchos ámbitos:
    - Inclusión de información en las webs de rendimiento de las campañas, de una explicación de los indicadores, para de manera estática o dinámica, facilitar la comprensión a los administradores de los KPIs mostrados.
    - Ayuda interactiva para el rellenado de los campos de formularios.
    - Refuerzo de las campañas teóricas de formación, no sólo generando cuestionarios autocontenidos, sino mediante píldoras formativas con vídeos e imágenes, que se puedan ser también incluidos en los planes de training y enviados a los usuarios con una cadencia determinada.
    - Reenfoque de los entrenamientos, creando una zona de usuario en el aplicativo, con un cronograma y tareas a completar por parte del empleado, permitiendo llevar a cabo un seguimiento del mismo por parte del instructor.
    - Inclusión de otro tipo de campañas teórico-prácticas de ingeniería social, llevando a cabo por ejemplo, ataques simulados mediante Smishing, Vishing, RRSS como Whatsapp, o USB infectados. Nótese que la parte teórica está cubierta con la funcionalidad ya desarrollada.
    - Reportes en formato gráfico para facilitar la visualización.

6. Referencias

Accenture. (2019). *The cost of cybercrime. Ninth annual report.* Recuperado 20 de Marzo de 2021, a partir de https://www.accenture.com/_acnmedia/PDF-96/Accenture-2019-Cost-of-Cybercrime-Study-Final.pdf#zoom=50

Aiuken. (2012). *SIA.* Recuperado 23 de Marzo de 2021, a partir de https://www.aiuken.com/en

Albladi, S. M., & Weir, G. R., S. (2020). Predicting individuals' vulnerability to social engineering in social networks. *Cybersecurity*, 3(1). Recuperado 01 de Abril de 2021, a partir de http://bv.unir.net:2145/10.1186/s42400-020-00047-5

Aldawood, H., & Skinner, G. (2019). Reviewing cyber security social engineering training and awareness Programs—Pitfalls and ongoing issues. *Future Internet, 11*(3), 73. Recuperado 12 de Abril de 2021, a partir de http://bv.unir.net:2145/10.3390/fi11030073

Algarni, A., Xu, Y., & Chan, T. (2017). An empirical study on the susceptibility to social engineering in social networking sites: The case of facebook. *European Journal of Information*

Systems, 26(6), 661-687. Recuperado 15 de Abril de 2021, a partir de http://bv.unir.net:2145/10.1057/s41303-017-0057-y

Anderson, H. S., Woodbridge, J. & Filar, B. (2016). DeepDGA: Adversarially-tuned domain generation and detection. *Proc. ACM Workshop Artif. Intell. Secur. (ALSec)*, pp. 13-21.

APWG, Anti-Phising Working Group. (2020). *Phishing Activity Trends Report 1st Quarter 2020.* Recuperado 22 de Marzo de 2021, a partir de https://docs.apwg.org/reports/apwg_trends_report_q1_2020.pdf

Bahnsen, A. C., Torroledo, I., Camacho, L. D. & Villegas, S. (2018). DeepPhish: Simulating malicious AI. *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, pp. 1-8.

Basquecybersecurity. (2020). *Basque CyberSecurity Centre.* Recuperado 27 de Marzo de 2021, a partir de https://www.basquecybersecurity.eus/es/

Blanco, A. (2020). *Modelo vista controlador en Wordpress.* Recuperado 03 de Abril de 2021, a partir de https://www.ablancodev.com/wordpress/modelo-vista-controlador/

CDSE. (2010). *Center for Development of Security Excellence.* Recuperado 04 de Abril de 2021, a partir de https://www.cdse.edu/index.html

CDSEb. (2010). *CDSE. SETA toolkit.* Recuperado 16 de Marzo de 2021, a partir de https://www.cdse.edu/toolkits/seta/index.php

Check Point. (1993). *Email Security Awareness to Employees.* Recuperado 20 de Abril de 2021, a partir de https://www.checkpoint.com/cyber-hub/threat-prevention/what-is-email-security/email-security-awareness-to-employees/

Cisco. (1984). *Security Education.* Recuperado 27 de Abril de 2021, a partir de https://www.cisco.com/c/en/us/about/security-center/security-programs/security-education.html#~plan

Cofense. (2020). *Cybersecurity Awareness Training for Employees.* Recuperado 06 de Abril de 2021, a partir de https://cofense.com/solutions/topic/security-awareness-solutions

Coma, A. F. (2017). *Instalación Segura de Servidor Usando Aplicaciones de Código Abierto.* (Trabajo Fin de Máster). Universidad Internacional de la Rioja, La Rioja. Recuperado 20 de Mayo de 2021, a partir de https://reunir.unir.net/handle/123456789/6596

CSA. (2015). *Cybersecurity Awareness Alliance.* Recuperado 26 de Marzo de 2021, a partir de https://www.csa.gov.sg/gosafeonline/content/cyber-security-awareness-alliance

CSAb. (2015). *Cybersecurity Awareness Alliance. Resources.* Recuperado 17 de Abril de 2021, a partir de https://www.csa.gov.sg/gosafeonline/resources

Cybereop. (2020). *CYBEREOP.* Recuperado 16 de Abril de 2021, a partir de https://www.cybereop.com/

ElevenPaths. (2013). *Eleven Paths.* Recuperado 10 de Abril de 2021, a partir de https://www.elevenpaths.com/index.html

Factum. (2009). *Factum.* Recuperado 23 de Marzo de 2021, a partir de https://factum-it.es/

Fortinet. (2000). *Information Security Awareness and Training Service.* Recuperado 29 de Marzo de 2021, a partir de https://www.fortinet.com/training/infosec-awareness

Fortinetb. (2000). *Information Security Awareness.* Recuperado 29 de Marzo de 2021, a partir de https://training.fortinet.com/local/staticpage/view.php?page=library_information-security-awareness

G2. (2020). *Best Security Awareness Training Software.* Recuperado 28 de Marzo de 2021, a partir de https://www.g2.com/categories/security-awareness-training

Gallagher, R. (2016). Where do the phishers live? Collecting phishers' geographic locations from automated honeypots. *Proc. ShmooCon.* Recuperado 13 de Abril de 2021, a partir de https://archive.org/details/Where_Do_The_Phishers_Live

Github. (2021). *Veraphish source code.* Recuperado 12 de Junio de 2021, a partir de https://github.com/oscigle/Veraphish

Google. (2019). *¿Puedes detectar cuándo te están engañando?* Recuperado 12 de Abril de 2021, a partir de https://phishingquiz.withgoogle.com/

Gophish. (2013). *Open-Source Phishing Framework*. Recuperado 15 de Marzo de 2021, a partir de https://getgophish.com

Gopshishb. (2013). *Python API Client.* Recuperado 15 de Marzo de 2021, a partir de https://docs.getgophish.com/python-api-client/

Hadnagy, C. (2010). *Social Engineering: The Art of Human Hacking*, John Wiley & Sons, Incorporated. ProQuest Ebook Central, recuperado 01 de Abril de 2021 de https://bv.unir.net:2056/lib/univunirsp/detail.action?docID=706746

HiddenEye. (2020). *HiddenEyeReborn*. Recuperado 03 de Abril de 2021, a partir de https://github.com/Open-Security-Group-OSG/HiddenEyeReborn

Hoxhunt. (2020). *Gamified Phishing Training.* Recuperado 07 de Abril de 2021, a partir de https://www.hoxhunt.com/gamified-phishing-training-platform

IBM. (2020). *Strengthening the first line of defense and the weakest link: people.* Recuperado 29 de Marzo de 2021, a partir de https://www.ibm.com/downloads/cas/ZVBO1GYJ

INCIBE. (2006). *INCIBE: Blog de concienciación.* Recuperado 25 de Marzo de 2021, a partir de https://www.incibe.es/etiquetas-blog/concienciacion

INCIBEb. (2006). *INCIBE: Concienciación y Formación. Políticas de Seguridad para la PyME..* Recuperado 25 de Marzo de 2021, a partir de https://www.incibe.es/sites/default/files/contenidos/politicas/documentos/concienciacion-y-formacion.pdf

INCIBEc. (2006). *INCIBE: Kit de concienciación.* Recuperado 26 de Marzo de 2021, a partir de https://www.incibe.es/protege-tu-empresa/blog/actualizate-ciberseguridad-el-nuevo-kit-concienciacion

InfosecIQ. (2020). *Try InfosecIQ for free*. Recuperado 28 de Abril de 2021, a partir de https://securityiq.infosecinstitute.com

Inprosec (2010). Web oficial. Recuperado 28 de Junio de 2021, a partir de https://inprosec.es/

KingPhisher. (2018). *King-phisher*. Recuperado 19 de Abril de 2021, a partir de https://github.com/rsmusllp/king-phisher

KnowBe4. (2020). *Phish-prone.* Recuperado 17 de Abril de 2021, a partir de https://www.knowbe4.com/phishing-security-test-offer

KnowBe4b. (2020). *Automated Security Awareness Program.* Recuperado 17 de Abril de 2021, a partir de https://www.knowbe4.com/automated-security-awareness-program

KnowBe4c. (2020). *ModStore.* Recuperado 17 de Abril de 2021, a partir de https://www.knowbe4.com/training-preview

Lim, I., Park, Y. & Lee, J. (2016). Design of Security Training System for Individual Users. *Wireless Pers Commun* 90**,** 1105–1120. Recuperado 19 de Abril de 2021, a partir de https://bv.unir.net:2133/10.1007/s11277-016-3380-z

Loza, M. (2019). *Tema 1. Contratación informática*. Material no publicado. Recuperado 20 de Abril de 2021, a partir de http://www.unir.net

LucySecurity. (2020). *Lucy*. Recuperado 22 de Abril de 2021, a partir de https://lucysecurity.com

Martínez, D. (2019). *Tema 1. Análisis de Vulnerabilidades*. Material no publicado. Recuperado 18 de Abril de 2021, a partir de http://www.unir.net

McAfee. (2020). *McAfee Online Safety Program.* Recuperado 15 de Abril de 2021, a partir de https://www.mcafee.com/enterprise/en-in/online-safety.html

Microsoft. (2020). *Why integrated phishing-attack training is reshaping cybersecurity.* Recuperado 26 de Abril de 2021, a partir de https://www.microsoft.com/security/blog/2020/10/05/why-integrated-phishing-attack-training-is-reshaping-cybersecurity-microsoft-security/

Mimecast. (2020). *Phishing Training.* Recuperado 18 de Abril de 2021, a partir de https://www.mimecast.com/content/phishing-training/

MIT. (2020). *MIT license.* Recuperado 20 de Mayo de 2021, a partir de https://choosealicense.com/licenses/mit/

Mitnick, K. (2019). When a hacker decides to turn ethical. *Gulf News.* Recuperado 22 de Abril de 2021, a partir de https://bv.unir.net:2257/docview/2201806106?pq-origsite=summon

Mitnick, K. D., & Simon, W. L. (2002). *The art of deception: Controlling the human element of security*. Recuperado 22 de Abril de 2021, a partir de https://repo.zenk-security.com/Magazine%20E-book/Kevin_Mitnick_-_The_Art_of_Deception.pdf

Mouton, F., Leenen, L. & Venter, H.s. (2016). Social engineering attack examples, templates and scenarios*. Computers & Security*, 59, 186 – 209. Recuperado 13 de Abril de 2021, a partir de https://www.researchgate.net/publication/299344351_Social_engineering_attack_examples_templates_and_scenarios

NCSC. (2016). *National Cyber Security Centre.* Recuperado 11 de Abril de 2021, a partir de https://www.ncsc.gov.uk/

NCSCb .(2016). *National Cyber Security Centre. Cyber Aware.* Recuperado 11 de Abril de 2021, a partir de https://www.ncsc.gov.uk/cyberaware/home

NCSCc. (2016). *NCSC: Phishing attacks: defending your organisation.* Recuperado 11 de Abril de 2021, a partir de https://www.ncsc.gov.uk/guidance/phishing

NICE. (2014). *National Initiative for Cybersecurity Education.* Recuperado 16 de Abril de 2021, a partir de https://www.nist.gov/itl/applied-cybersecurity/nice

NICEb. (2014). *NICE: Free and Low Cost Online Cybersecurity Learning Content.* Recuperado 16 de Abril de 2021, a partir de https://www.nist.gov/itl/applied-cybersecurity/nice/resources/online-learning-content

NICEc. (2014). NICE: *Education and Training Provider Resources.* Recuperado 16 de Abril de 2021, a partir de https://www.nist.gov/itl/applied-cybersecurity/nice/nice-framework-resource-center/education-and-training-provider

NIST. (2014). *Framework for Improving Critical Infrastructure Cybersecurity.* Version 1.1. Recuperado 29 de Marzo de 2021, a partir de https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf

OneCyber. (2020). *OneCyber.* Recuperado 17 de Abril de 2021, a partir de https://www.onecyber.es/

OneSeq. (1991). *OneSeq by Alhambra.* Recuperado 18 de Abril de 2021, a partir de https://www.oneseq.es/

Orange. (2020). *Orange Ciberseguridad.* Recuperado 31 de Marzo de 2021, a partir de https://www.orange.es/grandes-empresas/ciberseguridad.html

OWASP. (2020). *Source Code Analysis Tools.* Recuperado 01 de Abril de 2021, a partir de https://owasp.org/www-community/Source_Code_Analysis_Tools

Ozkaya, E. (2018). *Learn Social Engineering: Learn the Art of Human Hacking with an Internationally Renowned Expert*, Packt Publishing, Limited. ProQuest Ebook Central, recuperado 22 de Abril de 2021 de https://bv.unir.net:2056/lib/univunirsp/detail.action?docID=5379705.

Phishingbox. (2020). *Phishing Awareness Training.* Recuperado 03 de Abril de 2021, a partir de https://www.phishingbox.com/products-services/phishing-awareness-training

PhishingFrenzy. (2017). *PhishingFrenzy*. Recuperado 03 de Abril de 2021, a partir de https://github.com/pentestgeek/phishing-frenzy

Phriendlyphishing. (2020). *Phriendlyphishing.* Recuperado 04 de Abril de 2021, a partir de https://www.phriendlyphishing.com/

PlataformaPYME. (2020). *Cifras PyME. Datos Septiembre 2021.* Recuperado 29 de Marzo de 2021, a partir de https://plataformapyme.es/Publicaciones/Cifras%20PYME/CifrasPYME-septiembre%202020.pdf

Proofpoint. (2020). *Training modules datasheet.* Recuperado 11 de Abril de 2021, a partir de https://www.proofpoint.com/sites/default/files/pfpt-us-ts-interactive-training-modules-training-summary.pdf

Proofpointb. (2020). *Proofpoint Security Awareness Training datasheet.* Recuperado 11 de Abril de 2021, a partir de https://www.proofpoint.com/sites/default/files/pfpt-us-po-security-awareness-training-packages-summary.pdf

PureCSS. (2014). *Pure CSS. A set of small, responsive CSS modules that you can use in every web project.* Recuperado 13 de Mayo de 2021, a partir de https://purecss.io/

Python. (2020). *Python org.* Recuperado 20 de Mayo de 2021, a partir de https://www.python.org/

RSA. (2016). *Security Awareness Program.* Recuperado 01 de Abril de 2021, a partir de https://community.rsa.com/docs/DOC-40434

S21Sec. (2000). *S21Sec.* Recuperado 07 de Abril de 2021, a partir de https://www.s21sec.com/es/

Salahdine, F., & Kaabouch, N. (2019) Social Engineering Attacks: A Survey. *Future Internet 11(4), 89.* Recuperado 04 de Abril de 2021, a partir de https://www.researchgate.net/publication/332151597_Social_Engineering_Attacks_A_Survey

SecureIT. (2009). *Secure IT.* Recuperado 06 de Mayo de 2021, a partir de https://www.secureit.es/

SET. (2012). *The Social Engineering Toolkit.* Recuperado 04 de Abril de 2021, a partir de https://securitytrails.com/blog/the-social-engineering-toolkit

SIA. (2004). *SIA.* Recuperado 07 de Abril de 2021, a partir de https://www.sia.es/

Singh, R. (2013). *Kali Linux Social Engineering*, Packt Publishing, Limited. ProQuest Ebook Central. Recuperado 16 de Abril de 2021, a partir de https://bv.unir.net:2056/lib/univunirsp/detail.action?docID=1532001.

SLOCCount. (2001). *SLOCCount.* Recuperado el 12 de Junio de 2021, a partir de https://dwheeler.com/sloccount/

Sophos. (2020). *Sophos Phish Threat*. Recuperado 23 de Abril de 2021, a partir de https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-phish-threat-datasheet.pdf

SpearPhisher. (2013). *Introducing SpearPhisher.* Recuperado 23 de Abril de 2021, a partir de https://www.trustedsec.com/blog/introducing-spearphisher-simple-phishing-email-generation-tool

SPF. (2014). *SPF.* Recuperado 11 de Abril de 2021, a partir de https://github.com/tatanus/SPF

Sptoolkit. (2012). *Sptoolkit.* Recuperado 23 de Abril de 2021, a partir de https://github.com/chris-short/sptoolkit

Statista. (2019). *The most popular programming languages.* Recuperado 31 de Marzo de 2021, a partir de https://www.statista.com/chart/16567/popular-programming-languages/

Statista. (2020). *Most used web frameworks among developers worldwide, as of early 2021.* Recuperado 31 de Marzo de 2021, a partir de https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/

Sun-tzu, ., & Griffith, S. B. (1964). *The art of war*. Oxford: Clarendon Press.

Symantec. (2019). *Internet Security Threat Report.* Recuperado 29 de Marzo de 2021, a partir de https://docs.broadcom.com/doc/istr-24-2019-en

Tegra. (2018). *Nace TEGRA, el centro de ciberseguridad de Galicia.* Recuperado 22 de Marzo de 2021, a partir de https://www.gradiant.org/noticia/tegra-centro-ciberseguridad/

Terranova. (2020). *Gone Phishing Tournament.* Recuperado 03 de Abril de 2021, a partir de https://terranovasecurity.com/gone-phishing-tournament-2020/

TrendMicro. (2020). *Doing a Phishing Simulation is Easy with Phish Insight.* Recuperado 04 de Abril de 2021, a partir de https://phishinsight.trendmicro.com/en/simulator

Vodafone. (2020). *Vodafone Seguridad Digital.* Recuperado 07 de Abril de 2021, a partir de https://www.vodafone.es/c/empresas/autonomos/es/vodafone-para-tu-negocio/servicios-one-profesional/seguridad-digital/

Wang, Z., Sun, L. & Zhu., H. (2010). Defining Social Engineering in Cybersecurity, *IEEE Access*, vol. 8, pp. 85094-85115, doi: 10.1109/ACCESS.2020.2992807. Recuperado 22 de Marzo de 2021, a partir de https://ieeexplore.ieee.org/document/9087851

Wise Security. (2019). *WiseSecurity Global.* Recuperado 07 de Abril de 2021, a partir de https://www.wsg127.com/

Anexos

A continuación, se muestra el contenido de los documentos adicionales que acompañan al código fuente, y capturas de pantalla de la funcionalidad de Veraphish.

Documentos auxiliares

README_docs.txt

****

VeraPhish template examples

****

The list of examples within the folder shows both the configuration files plus examples ready to use to test the apps.

For the config files, it is important to remove the comments in the filename till '] ' included.

0: After installation, Gophish basic config.

1 and 2: Examples for setting up config for both the Django app SETAphish (phishing module) + SETAsurvey (training module). Notice some fields are the same in both files.

3: Example of target group of users to be fished. Used in Phishing section > Group creation. Can add as many users as you want.

4: Example of email template,. Used in Phishing > Email creation. Template rules available at: https://docs.getgophish.com/user-guide/template-reference

5: Example of landing page. Used in Phishing > Landing page creation. Perhaps you feel like using the Gophish template creator directly for further customisation which cannot be done through API. See reference manual (for Gophish in general, at: https://docs.getgophish.com/user-guide/).

6: Example of training email template. Placed at SETA_surveys/modules. It must include the SURVEY_LINK at least. Same for all surveys.

7 and 8: couple of survey templates. Templates can have as much questions as you want,

and variable number of answers per question. ONly thing you have to take care of is:

* Every question has at least two answers.

* For every question there is at least one true answer.

Take into account that for scoring purposes, the app only take care of True options. That means mistakes does not substract points.

T-score only will be shown if number of respondents in % is greater than the threshold in the config file.

A-score will be shown only if the above is true and the source of users for the training was a Phishing campaign.

[Gophish config file example] config.json

```
{
	"admin_server": {
		"listen_url": "127.0.0.1:3333",
		"use_tls": true,
		"cert_path": "gophish_admin.crt",
		"key_path": "gophish_admin.key"
	},
	"phish_server": {
		"listen_url": "0.0.0.0:80",
		"use_tls": false,
		"cert_path": "example.crt",
		"key_path": "example.key"
	},
	"db_name": "sqlite3",
	"db_path": "gophish.db",
	"migrations_prefix": "db/db_",
	"contact_address": "",
	"logging": {
```

```
        "filename": "",
        "level": ""
    }
}
```

[SETAphish, fishing app config file example] SETAphish.cfg

```
[MAIN]

# gophish api key from UI
api_key = d881d40c5b05e0adb6ee0ade537bf995798d94738dfc9facc9d9a4e35f79dfd0

# gophish dashboard url
host_url = https://127.0.0.1:3333

# smtp/sender url
sender_host = smtp.gmail.com

# certificate verification (True or False)
verify = False
```

[SETAsurvey, training app config file example] SETAsurvey.cfg

```
[MAIN]

# gophish api key from UI (64 chars string)
api_key = d881d40c5b05e0adb6ee0ade537bf995798d94738dfc9facc9d9a4e35f79dfd0

# gophish dashboard url, e.g. https://127.0.0.1:3333
host_url = https://127.0.0.1:3333

# SSL certificate verification (True or False)
verify = False
```

```
[TRAINING]

# training campaign template (template filename within modules folder)
email_template = template.txt

#training campaign smtp config

#e.g. smtp.gmail.com
email_host = smtp.gmail.com

#e.g. john.doe@gmail.com
email_origin = john.doe@gmail.com

#e.g. mypassword
email_password = joe_pass

#e.g. Phishing training survey
email_subject = Phishing training survey

# base survey url (e.g. http://127.0.0.1:8000/SETA_surveys/surveys/training/make_survey/)
training_base_url = http://127.0.0.1:8000/SETA_surveys/surveys/training/make_survey/

#thresholds for surveys

# minimum percentage (value: 0-100) of surveys answered for showing survey scores
rate_answers = 30

# minimum percentage (value: 0-100) grade to consider a test passed for the A-score calculation
pass_training = 50
```

[Target group template example] group.csv

John,Doe,john.doe@email.com,CEO

Anna,Barbera,anna.barb@outlook.com,CFO

Jim,Carrey,jim.carrey@gmail.com,Specialist

[Email template example] email.html

Hello {{.FirstName}},

The password for {{.Email}} has expired.

Please reset your password here, {{.URL}}.

Thanks. IT Team

[Landing page template example] page.html

```
<form action="action_page.php" method="post">

 <div class="container">
   <label for="uname"><b>Username</b></label>
   <input type="text" placeholder="Enter Username" name="uname" required>

   <label for="psw"><b>Password</b></label>
   <input type="password" placeholder="Enter Password" name="psw" required>

   <button type="submit">Login</button>
   <label>
     <input type="checkbox" checked="checked" name="remember"> Remember me
   </label>
 </div>
```

```
  <div class="container" style="background-color:#f1f1f1">
    <button type="button" class="cancelbtn">Cancel</button>
    <span class="psw">Forgot <a href="#">password?</a></span>
  </div>
</form>
```

[Email template for training surveys example] template.txt

```
Dear ${PERSON_NAME},

Please conduct the phishing awareness survey in here:
${SURVEY_LINK}

Have a great day!

The cybersecurity team.
```

[Survey template example] survey.csv

```
Q;Cuáles de las siguientes son técnicas de ingeniería social?
T;Phishing
T;Smishing
T;Vishing
F;Ninguna de las anteriores
Q;Qué debes hacer en caso de detectar un email fraudulento?
T;Reportarlo al equipo de Seguridad inmediatamente
F;Borrarlo y olvidarme del asunto
Q;Es conveniente mezclar correo personal y profesional?
F;Sí
T;No
Q;Cuáles podrían ser las consecuencias de abrir un adjunto de origen desconocido?
T;Compromiso del equipo
T;Ejecución de código remoto
```

T;Las anteriores, y muchas otras adicionales

Q;Cuáles es el eslabon más débil en la ciberseguridad de una organización?

F;El firewall exterior

T;Las personas

F;Las impresoras

Nótese en este caso que, el fichero es CSV delimitado por punto y coma, a diferencia del resto. El motivo es evitar problemas de análisis sintáctico del archivo en caso de que el contenido de las preguntas o respuestas del test, contenga comas.

Capturas de pantalla de Veraphish

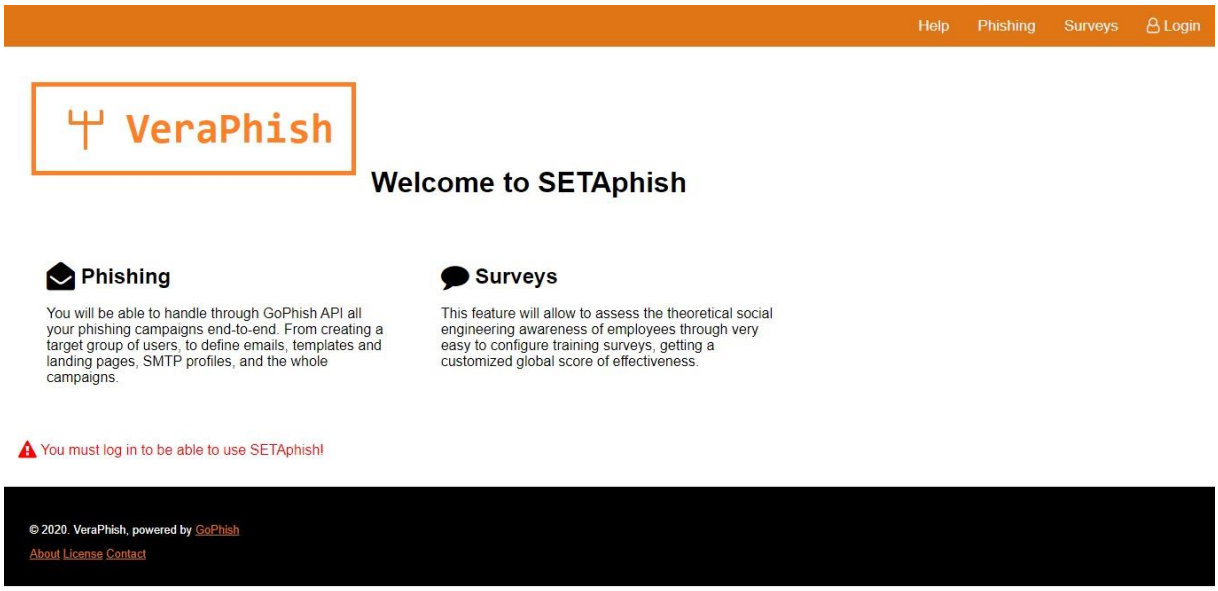

Ilustración 21. Veraphish: página principal de usuario no autenticado

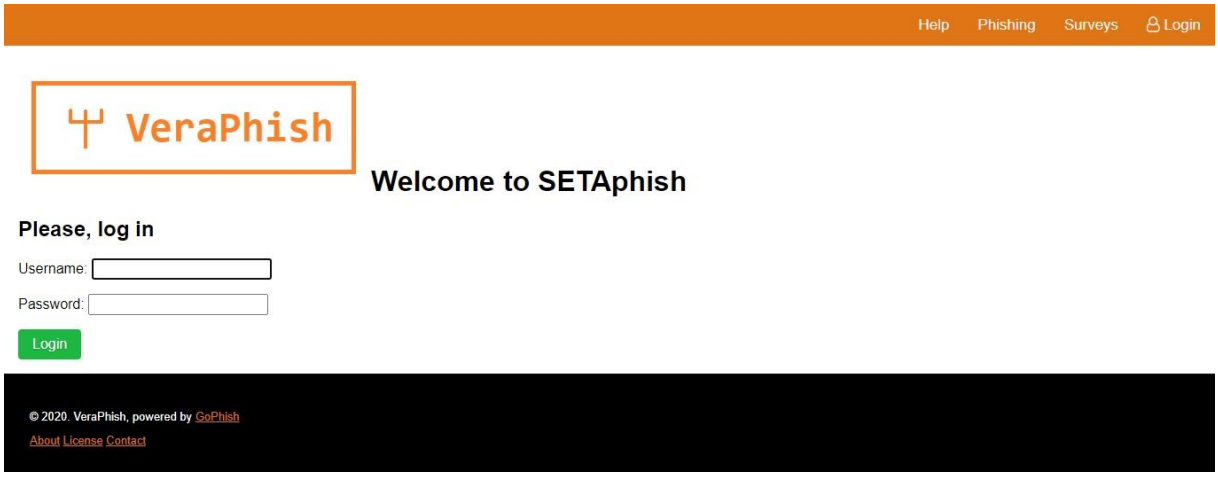

Ilustración 22. Veraphish: ventana de log in

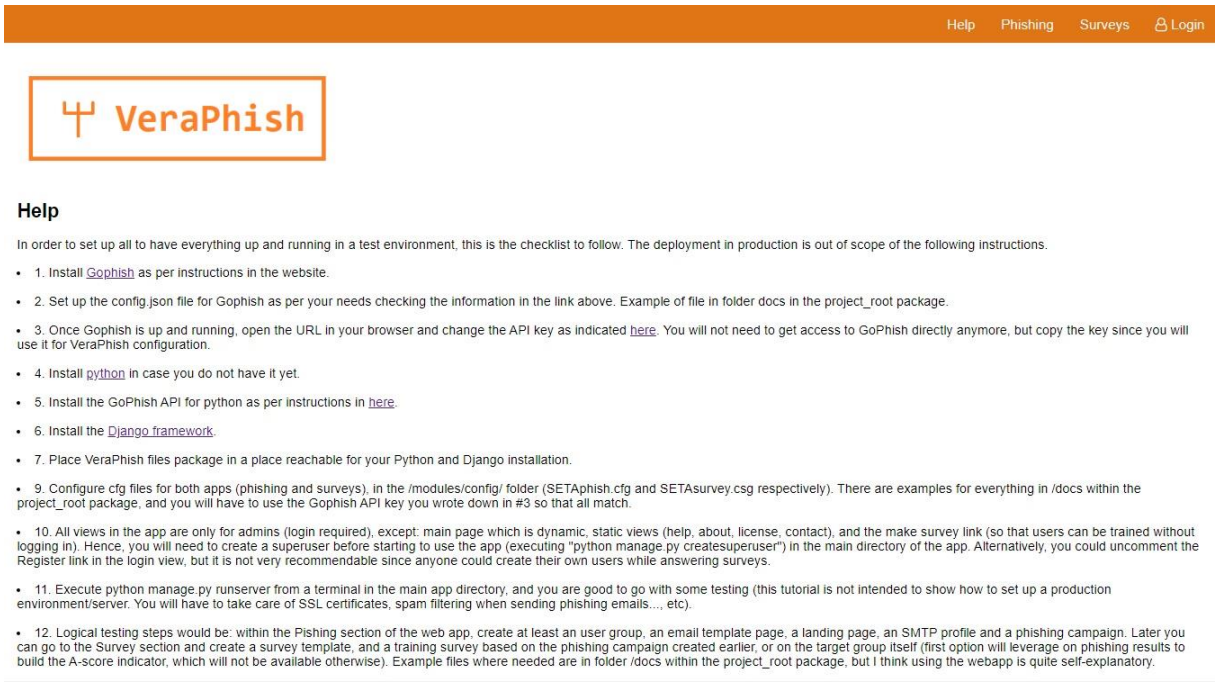

Ilustración 23. Veraphish: página de Ayuda (extracto)

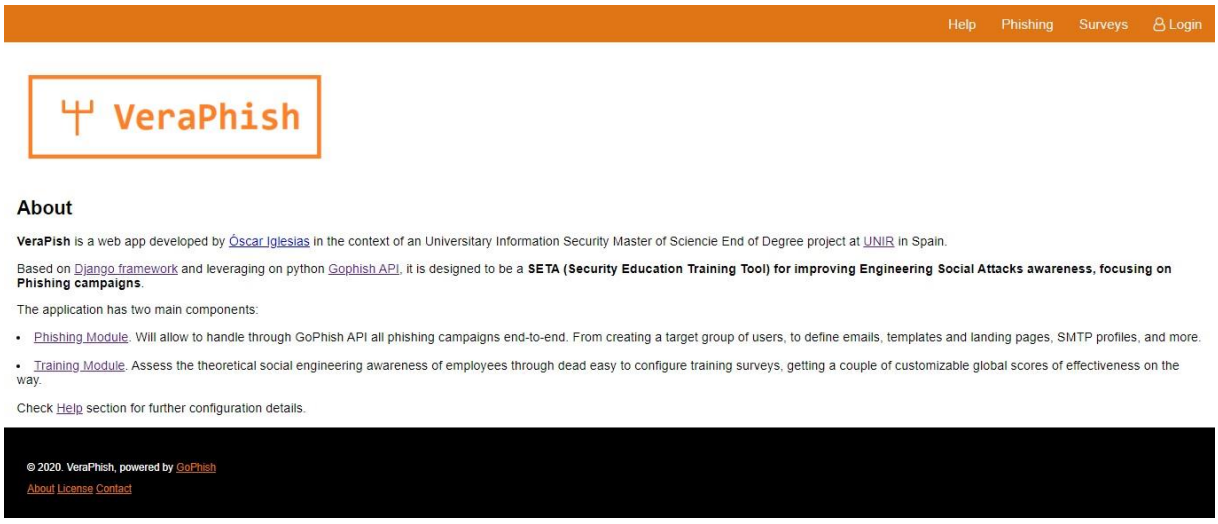

Ilustración 24. Veraphish: página About

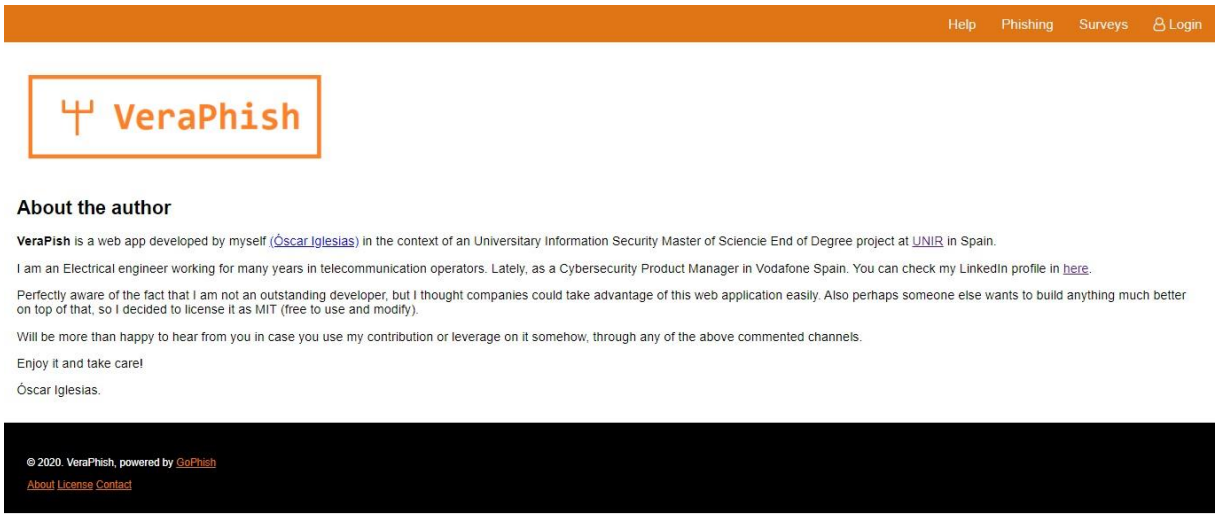

Ilustración 25. Veraphish: página de Contacto

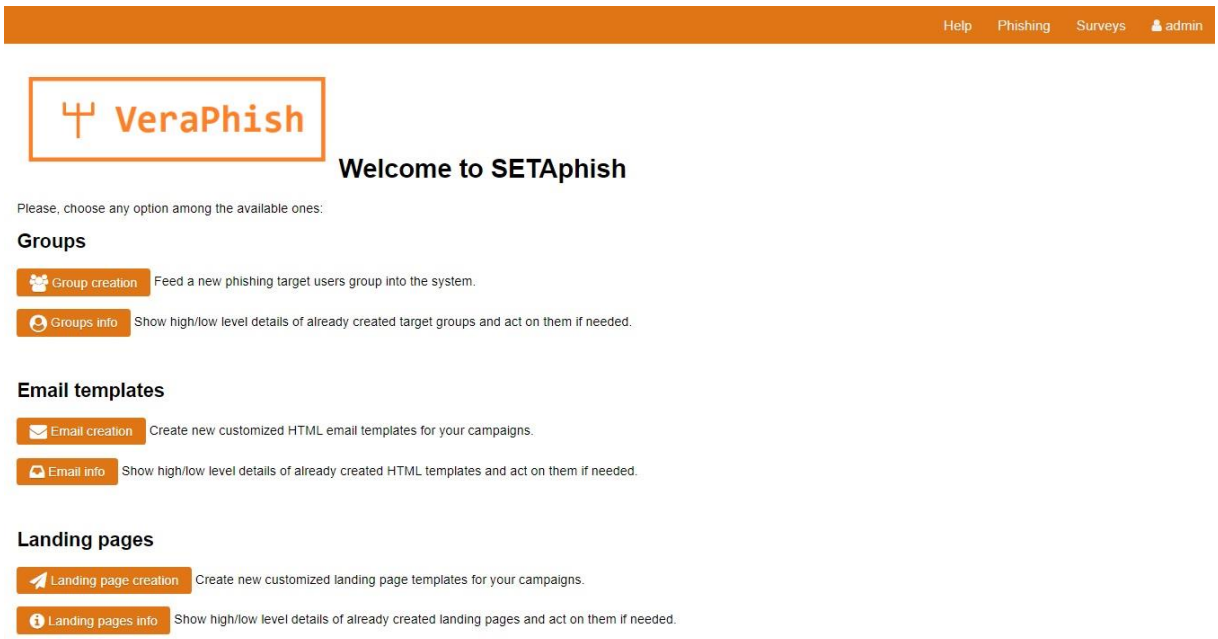

Ilustración 26. Veraphish: menú sección Phishing (1/2)

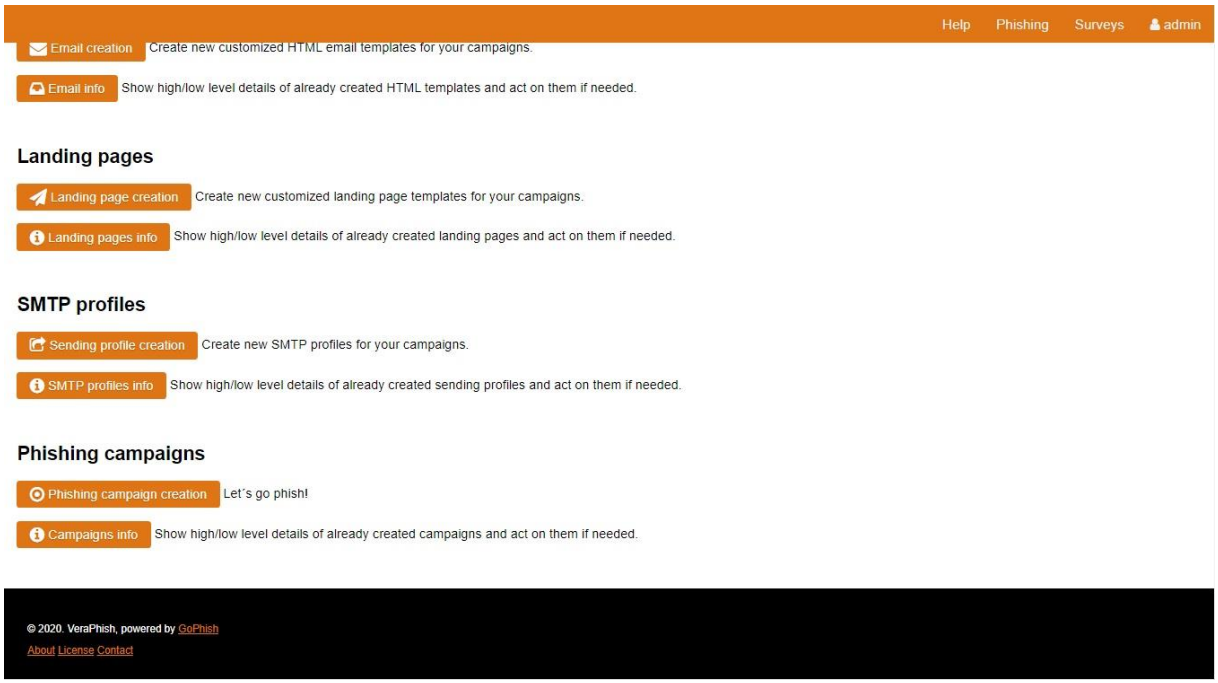

Ilustración 27. Veraphish: menú sección Phishing (2/2)

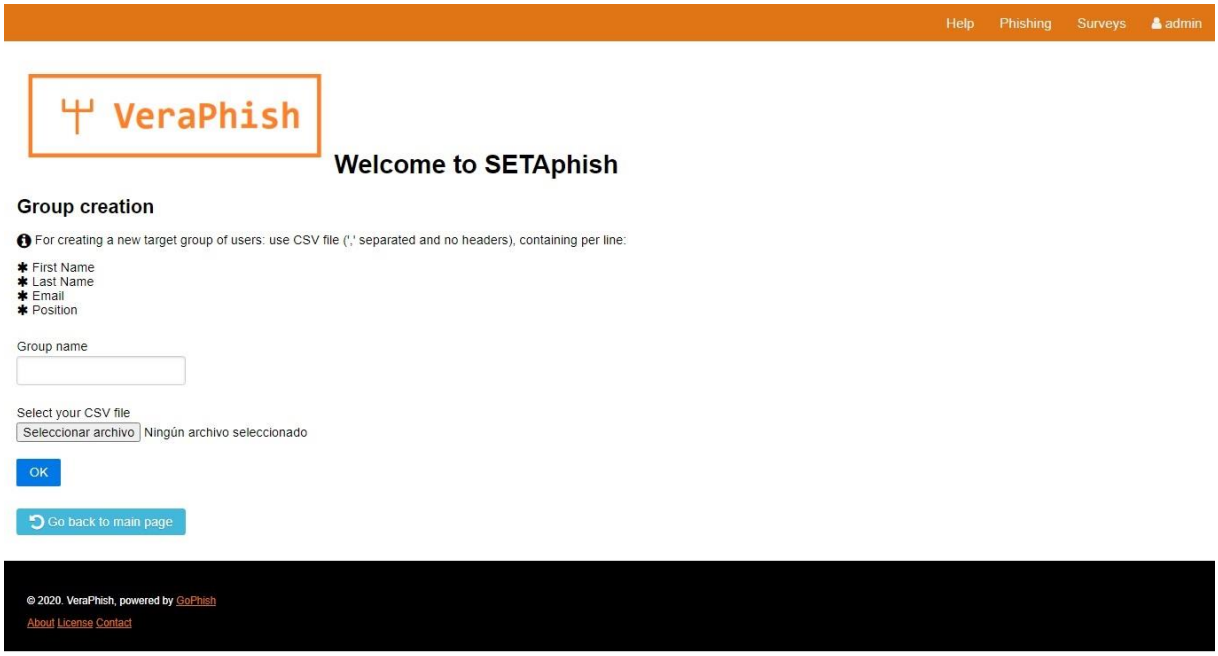

Ilustración 28. Veraphish: creación de Grupo de Usuarios

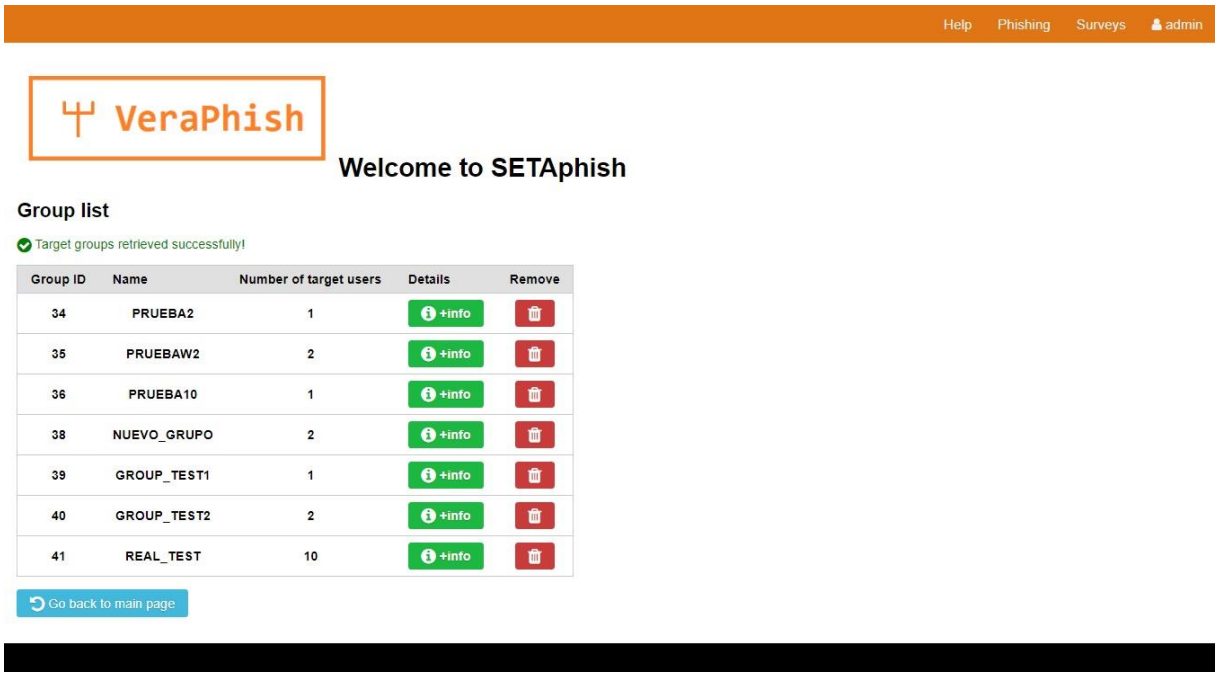

Ilustración 29. Veraphish: listado de Grupos de Usuarios

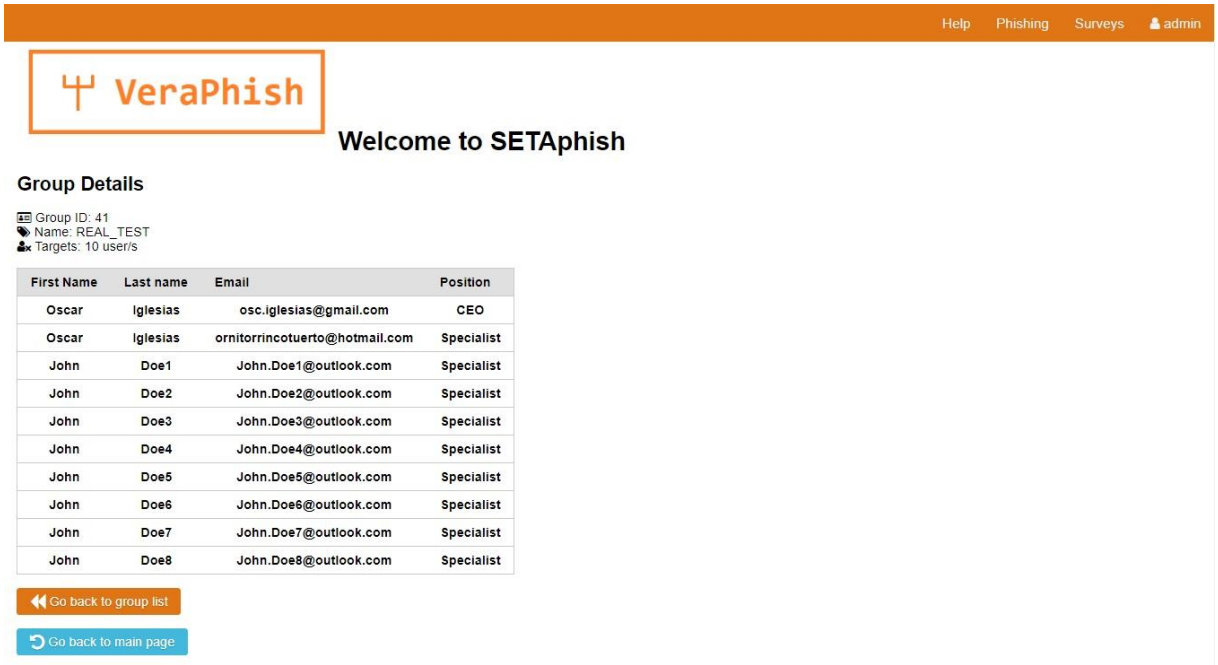

Ilustración 30. Veraphish: detalle de Grupo de Usuarios

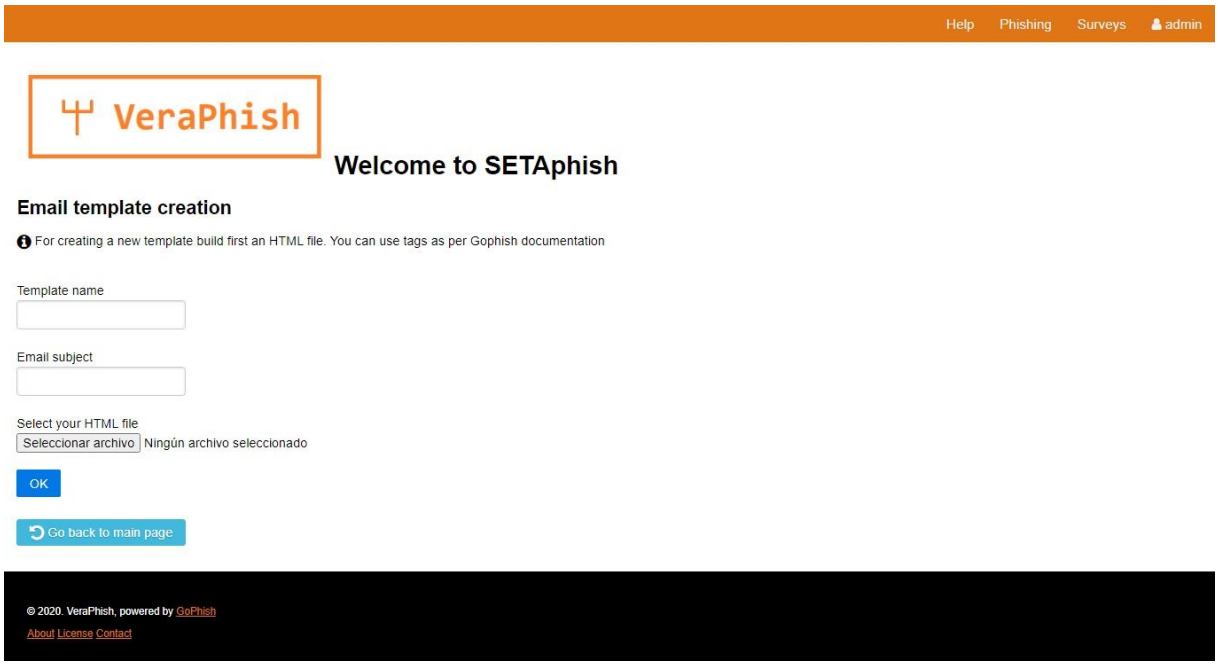

Ilustración 31. Veraphish: creación de Plantilla de Email phishing

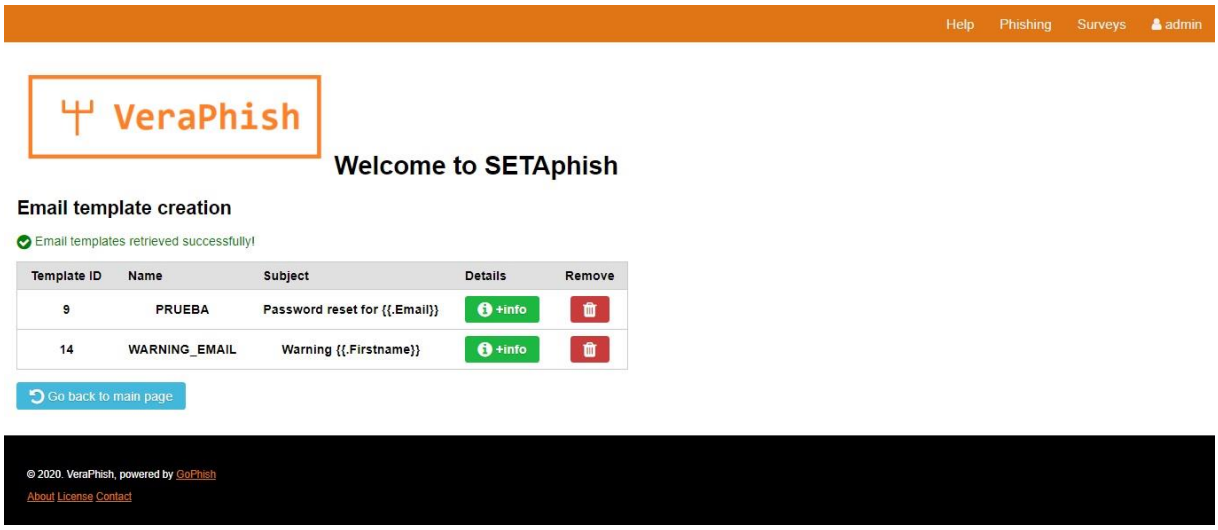

Ilustración 32. Veraphish: listado de Plantillas de Email phishing

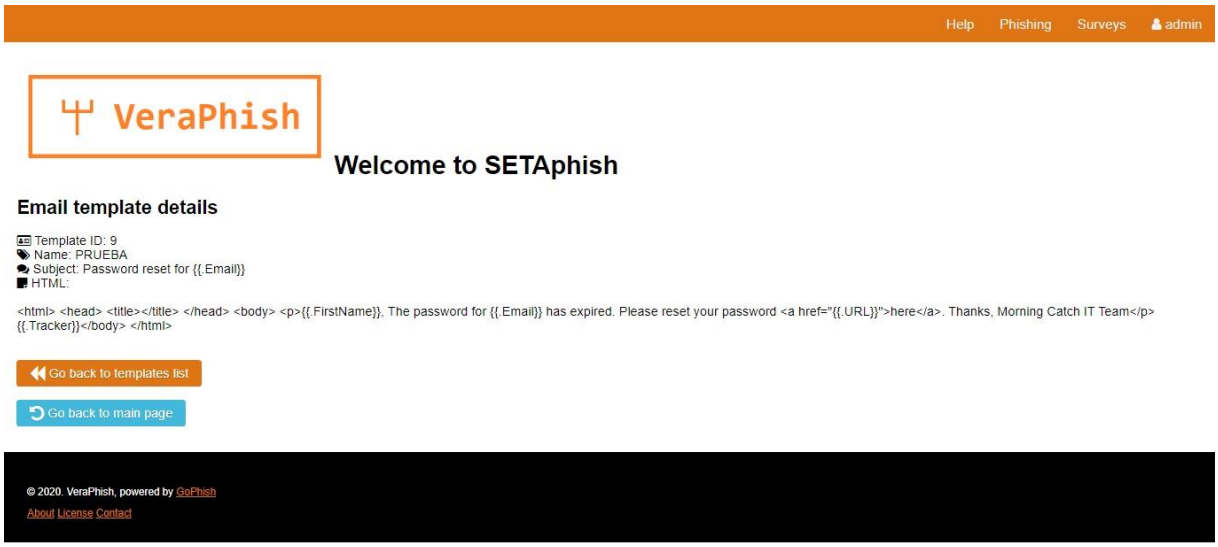

Ilustración 33. Veraphish: detalle de Plantilla de Email phishing

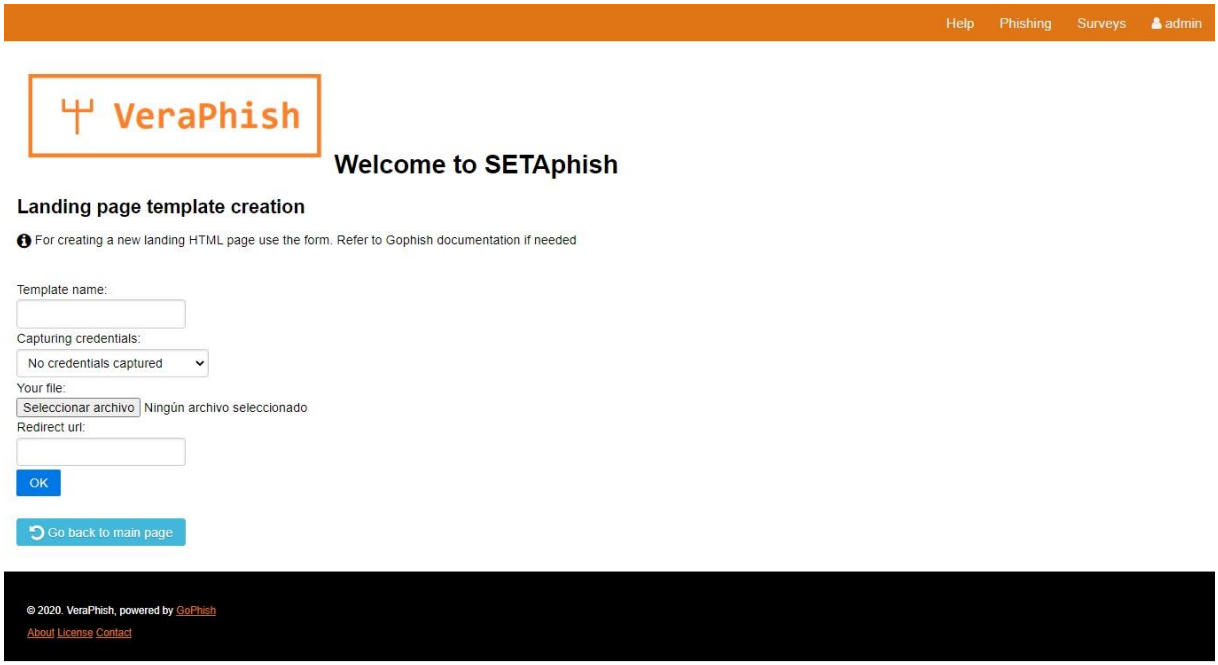

Ilustración 34. Veraphish: creación de plantilla de Página de Aterrizaje

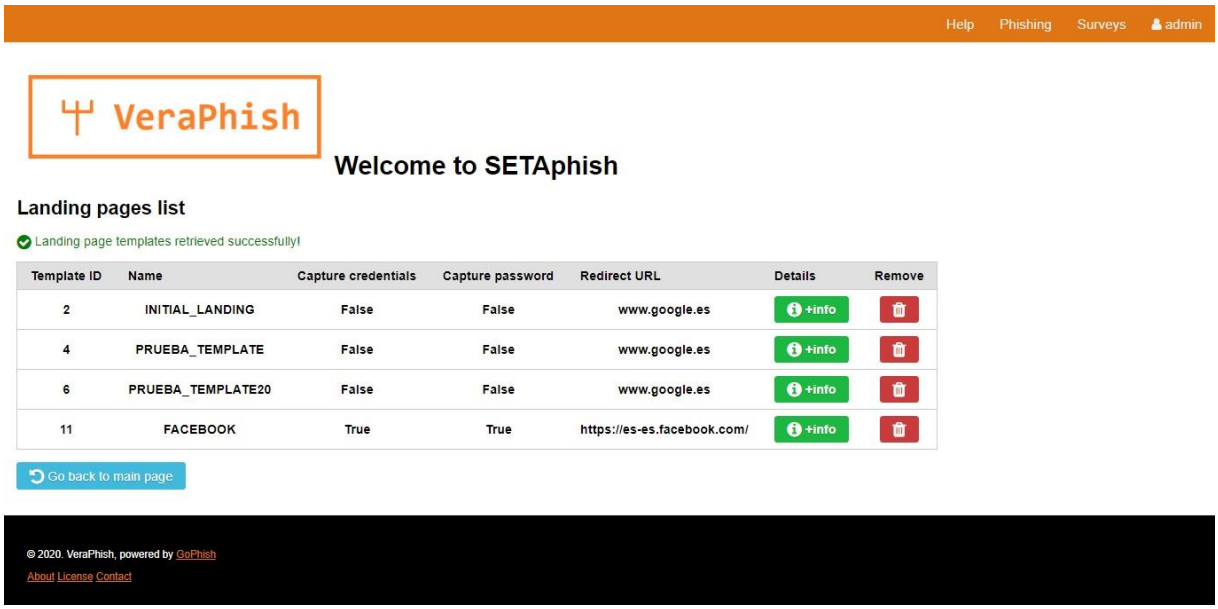

Ilustración 35. Veraphish: listado de plantillas de Página de Aterrizaje

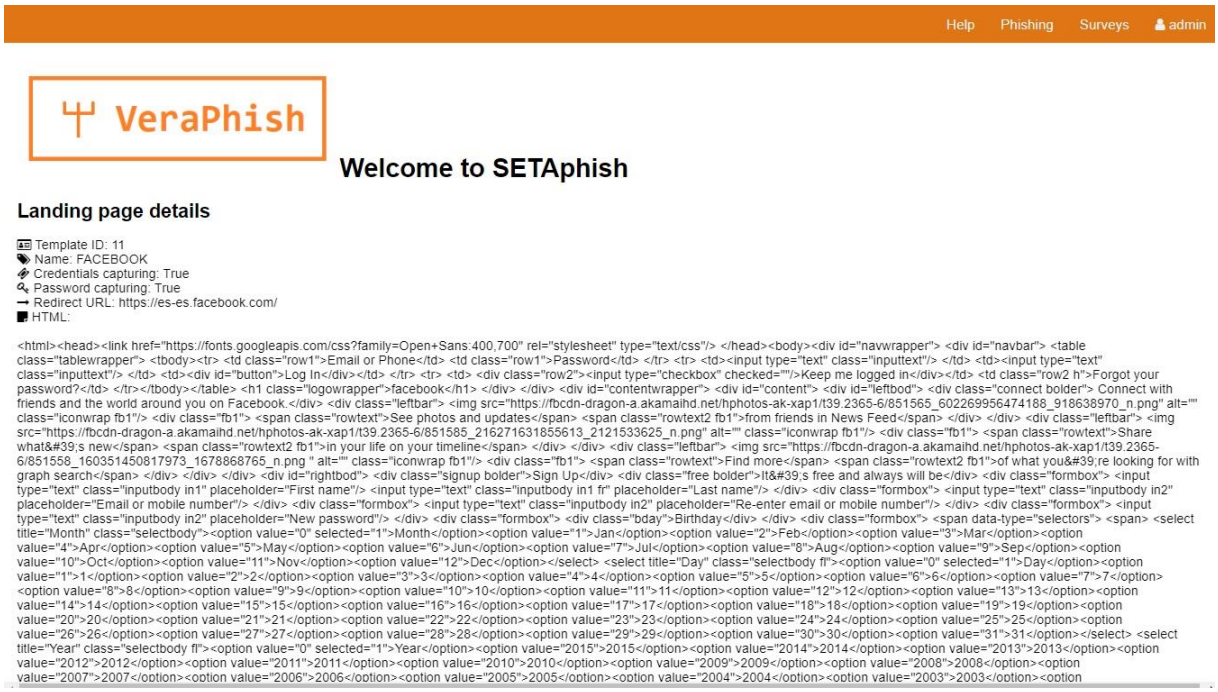

Ilustración 36. Veraphish: detalle de plantilla de Página de Aterrizaje (extracto)

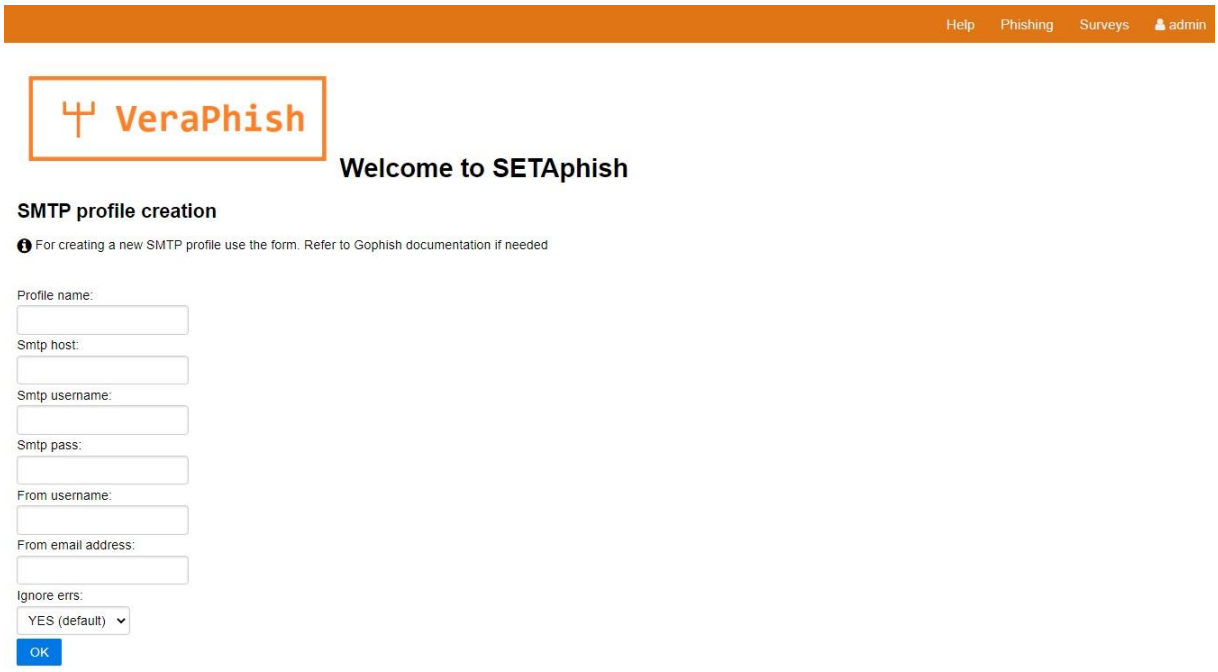

Ilustración 37. Veraphish: creación de perfil SMTP

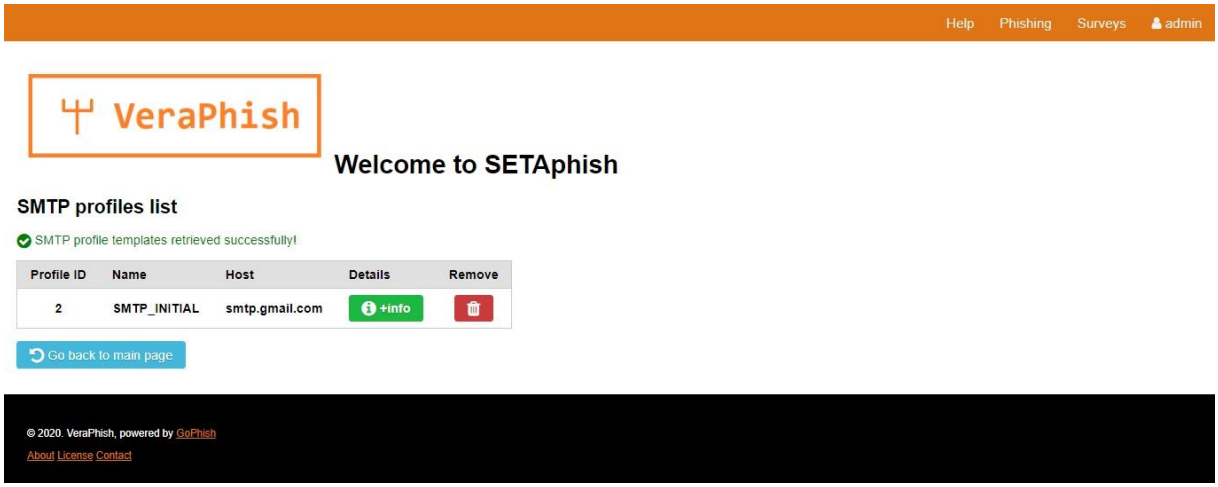

Ilustración 38. Veraphish: listado de perfiles SMTP

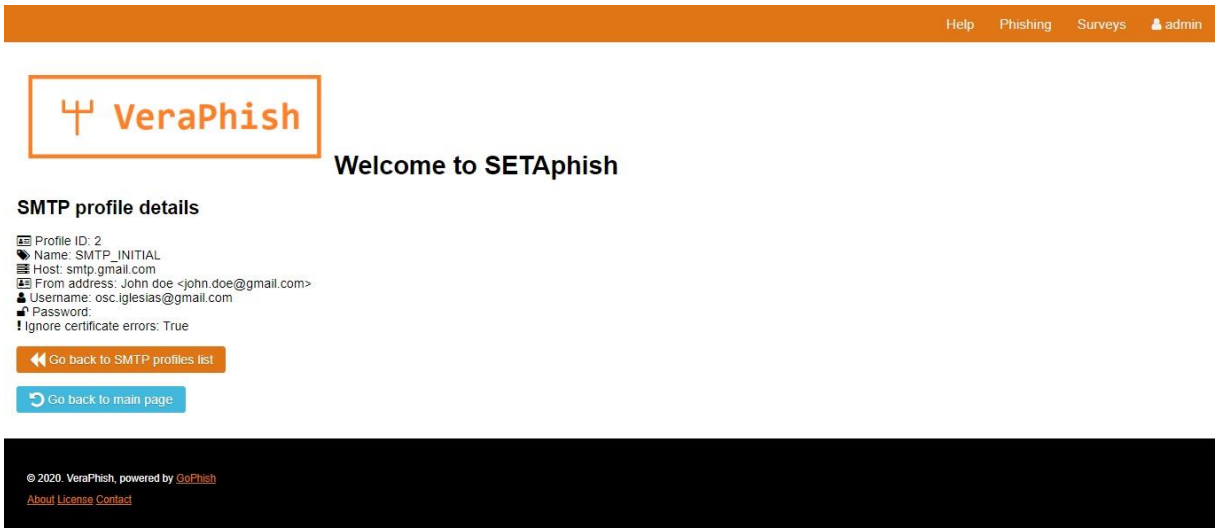

Ilustración 39. Veraphish: detalle de perfil SMTP

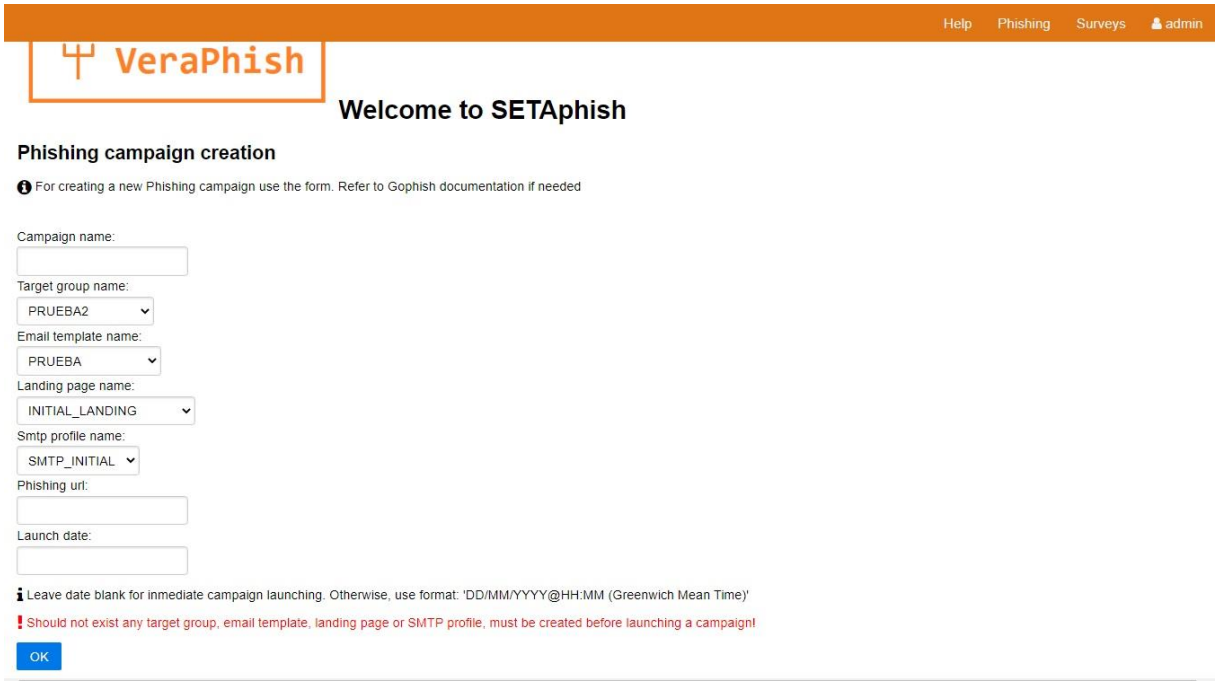

Ilustración 40. Veraphish: creación de Campaña de Email phishing

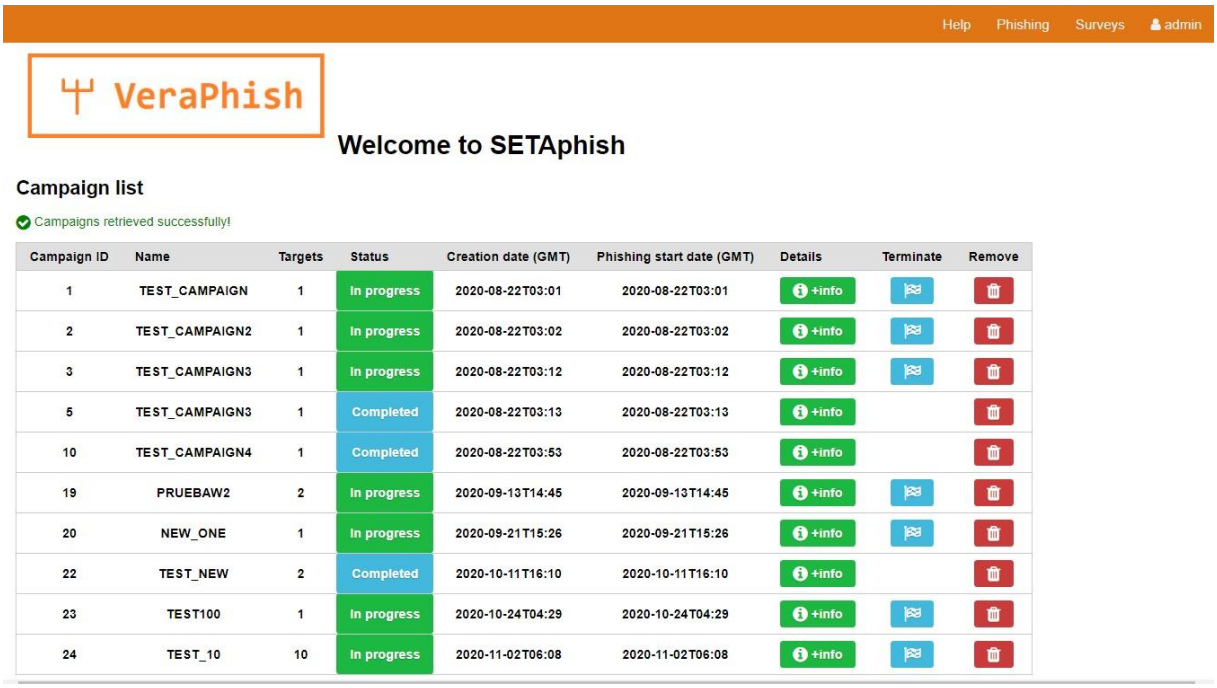

Ilustración 41. Veraphish: listado de Campañas de Email phishing

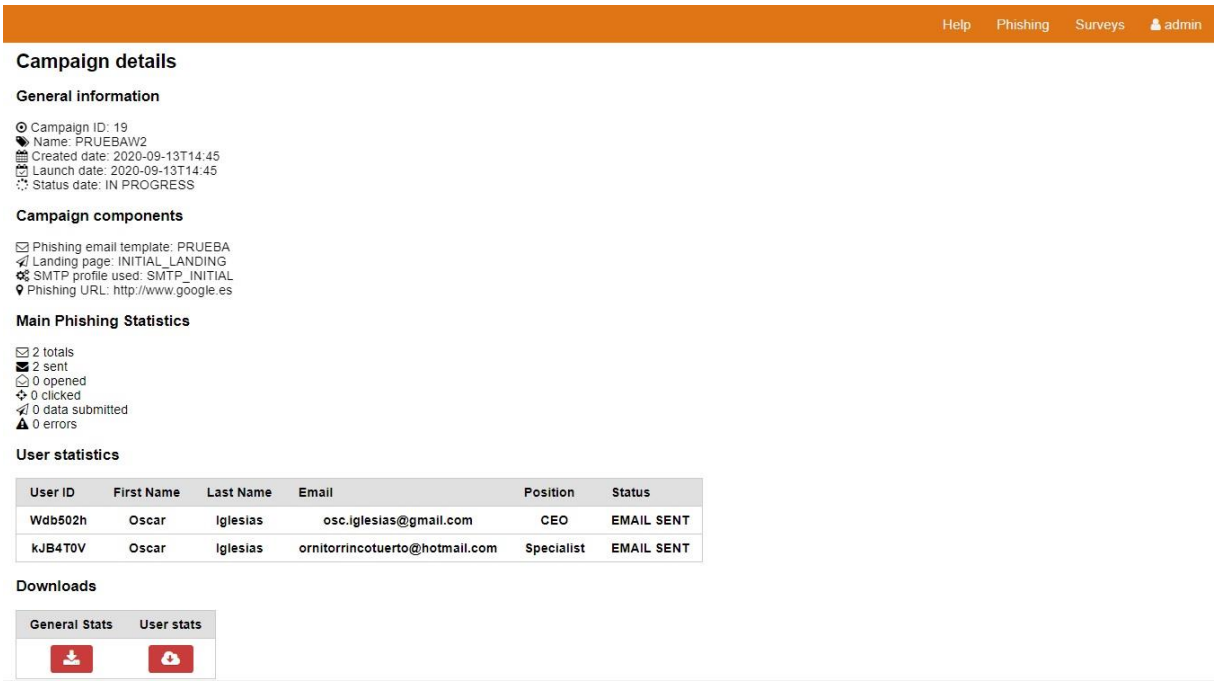

Ilustración 42. Veraphish: detalle de Campaña de Email phishing

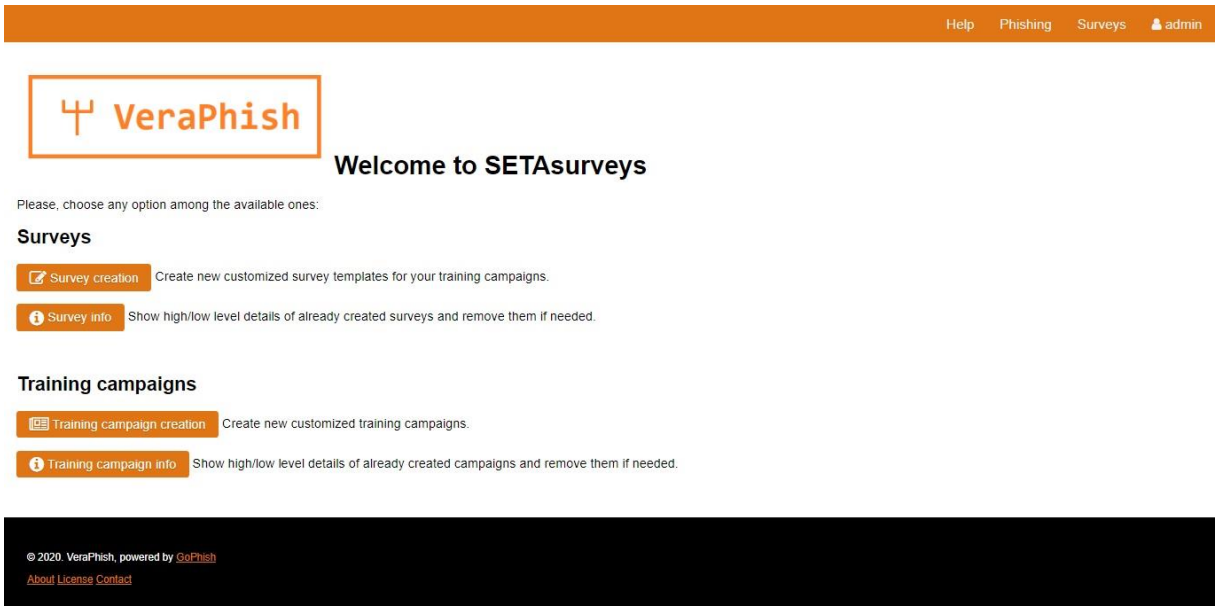

Ilustración 43. Veraphish: menú sección Tests de aprendizaje

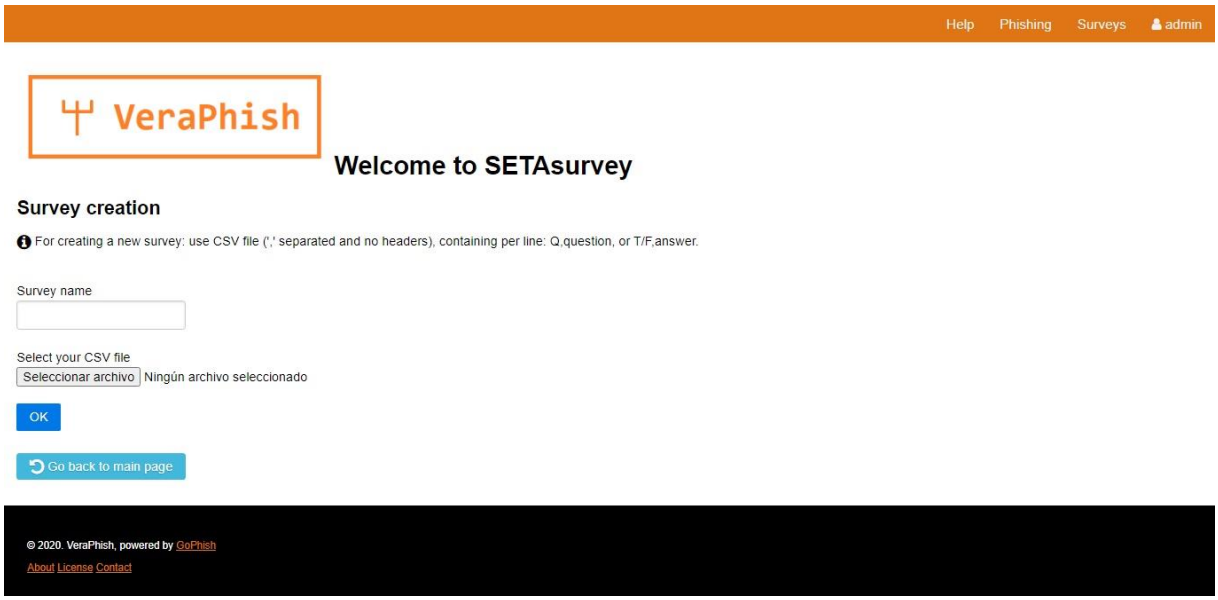

Ilustración 44. Veraphish: creación de Test de aprendizaje

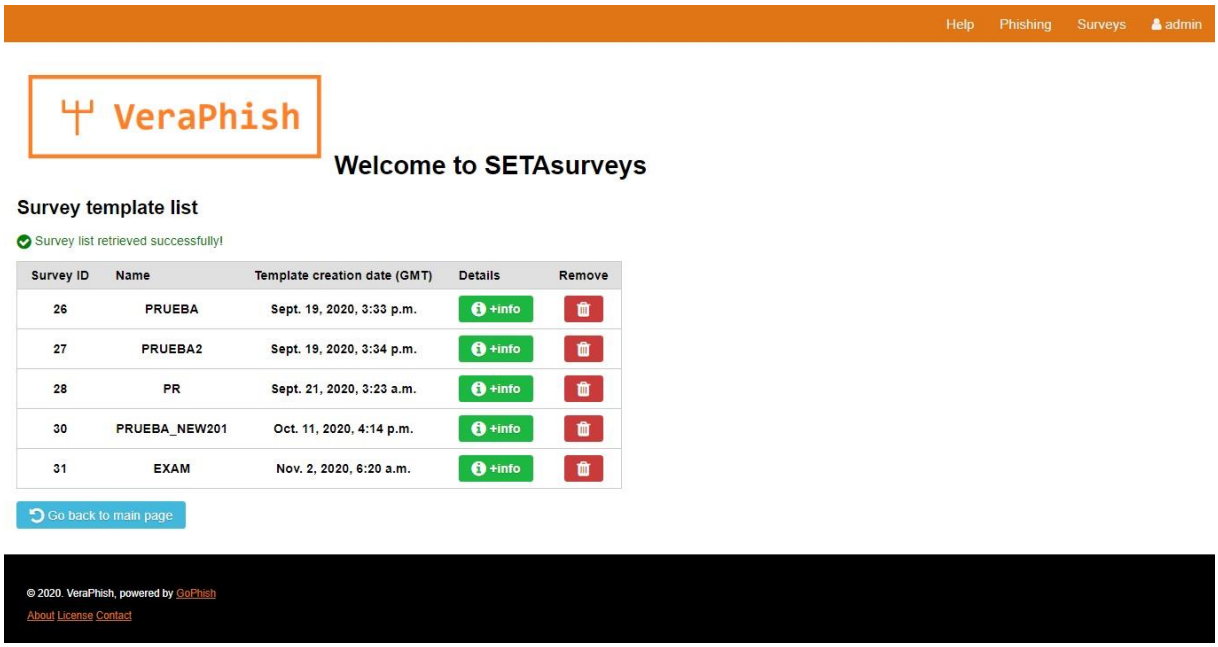

Ilustración 45. Veraphish: listado de Tests de aprendizaje

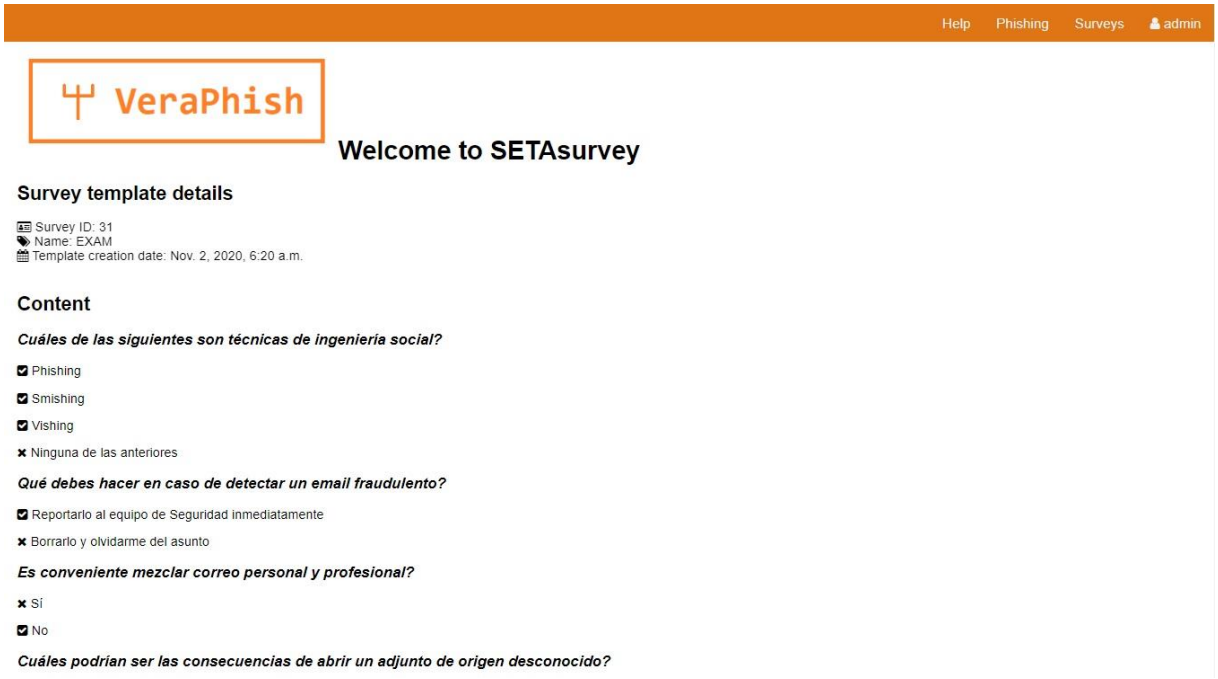

Ilustración 46. Veraphish: detalle de Test de aprendizaje (extracto)

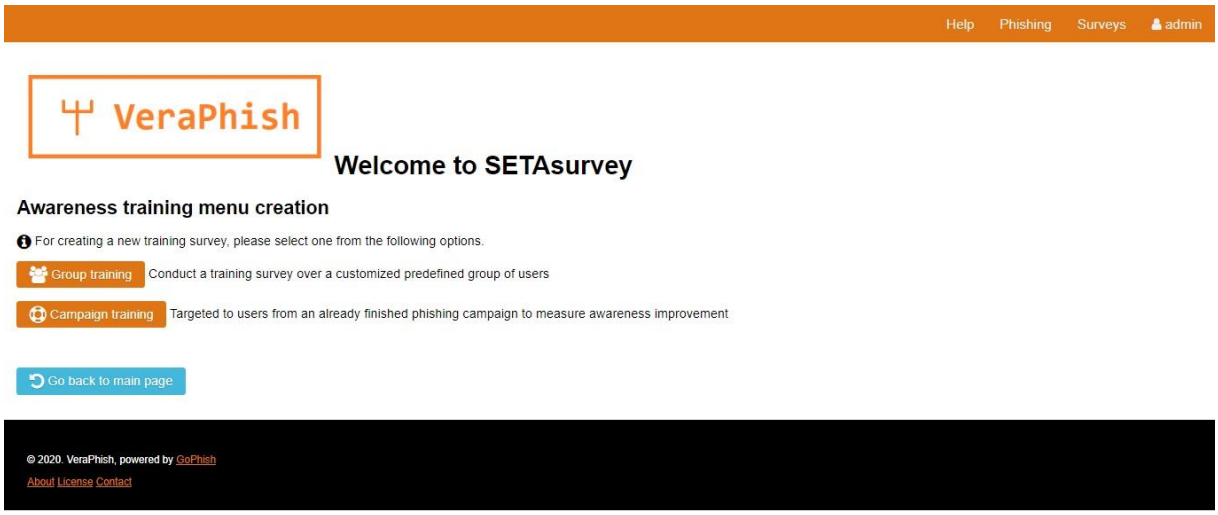

Ilustración 47. Veraphish: menú principal de creación de Campaña de Formación

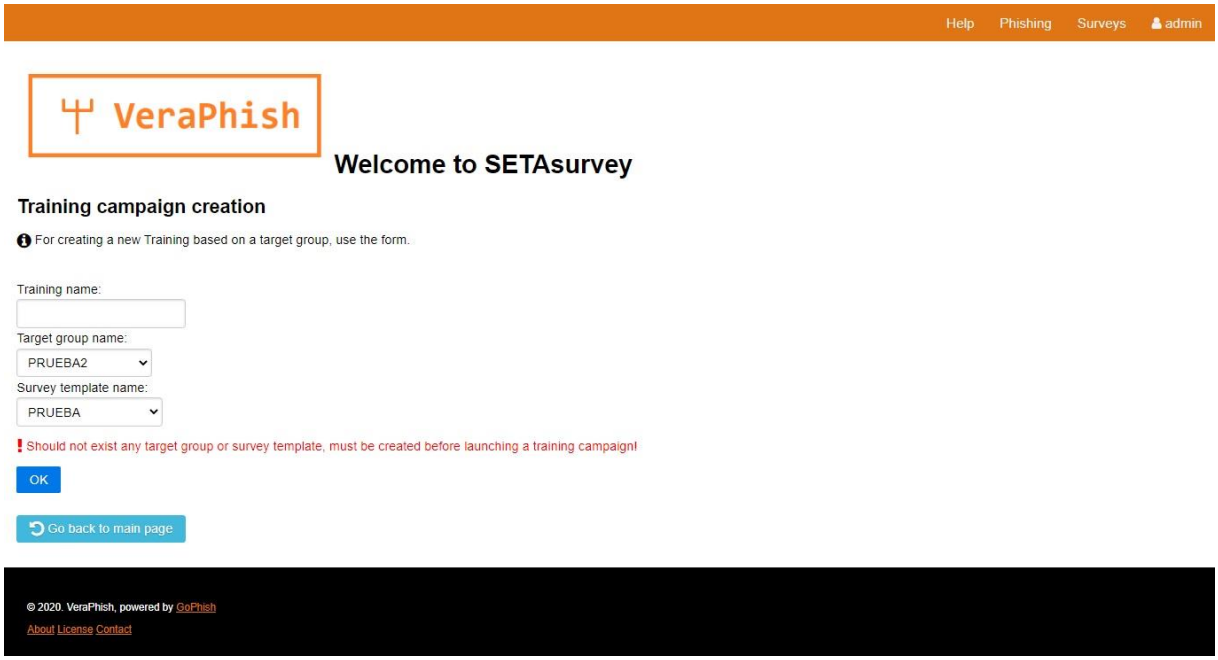

Ilustración 48. Veraphish: creación de Campaña de Formación a partir de Grupo de Usuarios

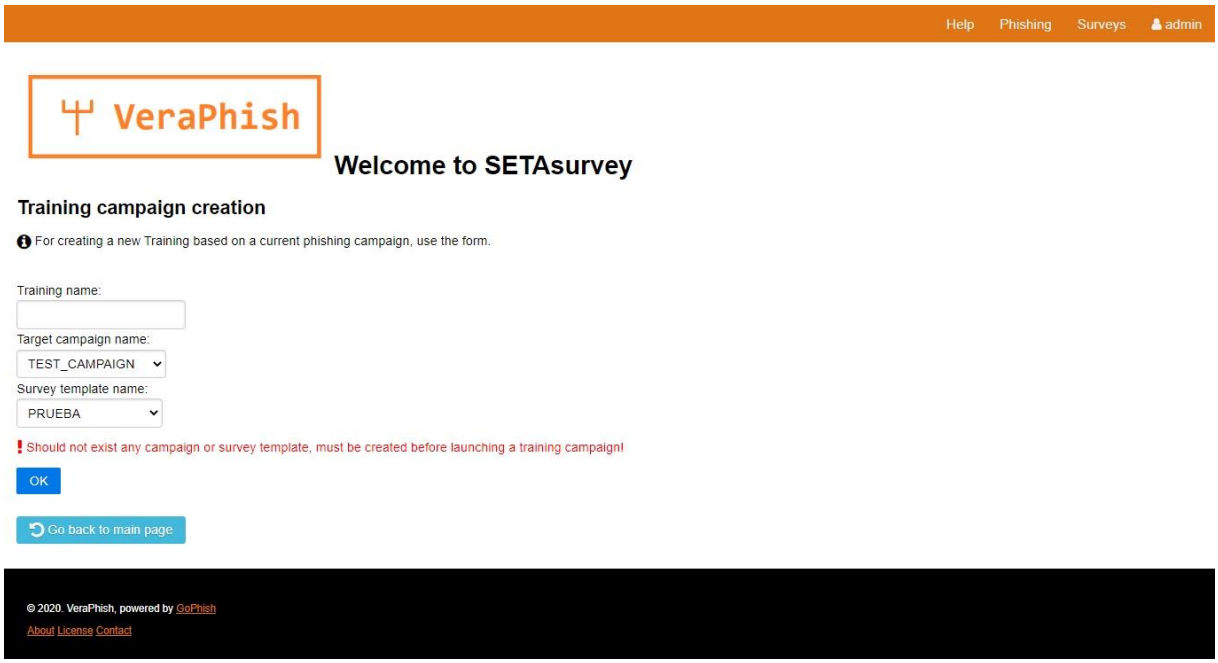

Ilustración 49. Veraphish: creación de Campaña de Formación a partir de Campaña de Phishing

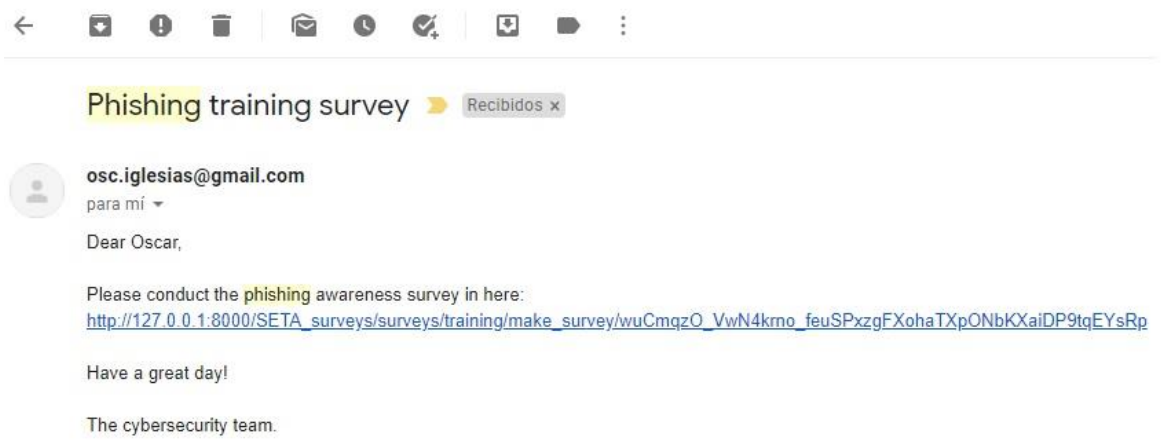

Ilustración 50. Veraphish: email personalizado de notificación de Campaña de Formación

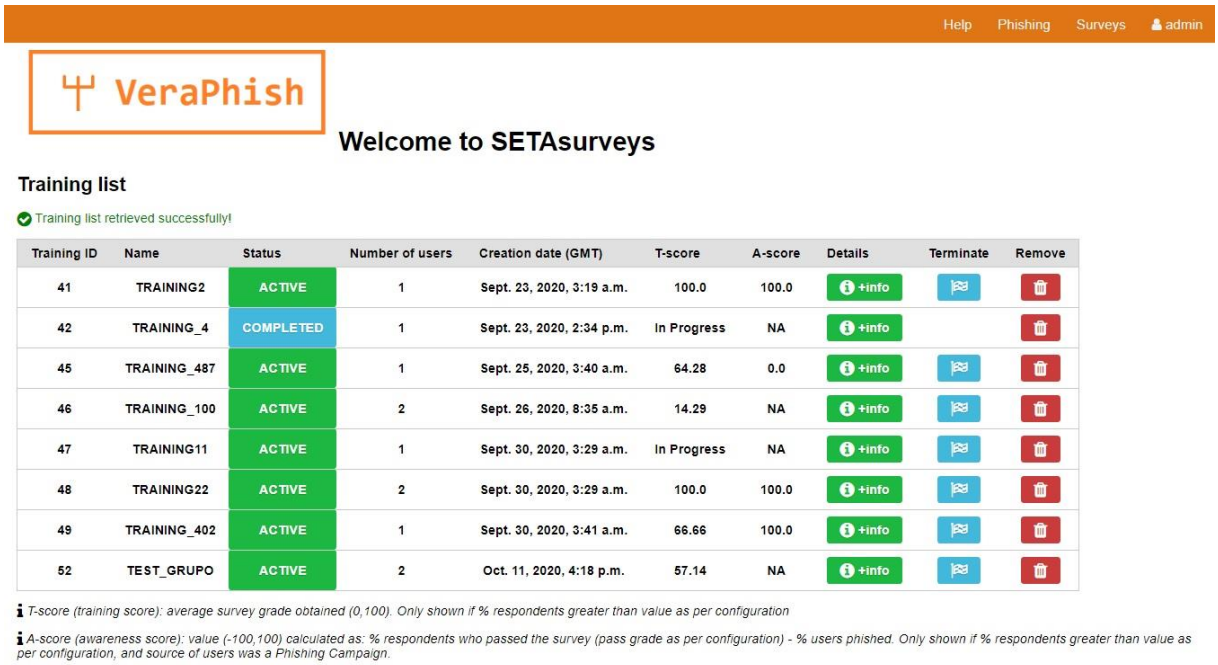

Ilustración 51. Veraphish: listado de Campañas de Formación y KPIs asociados

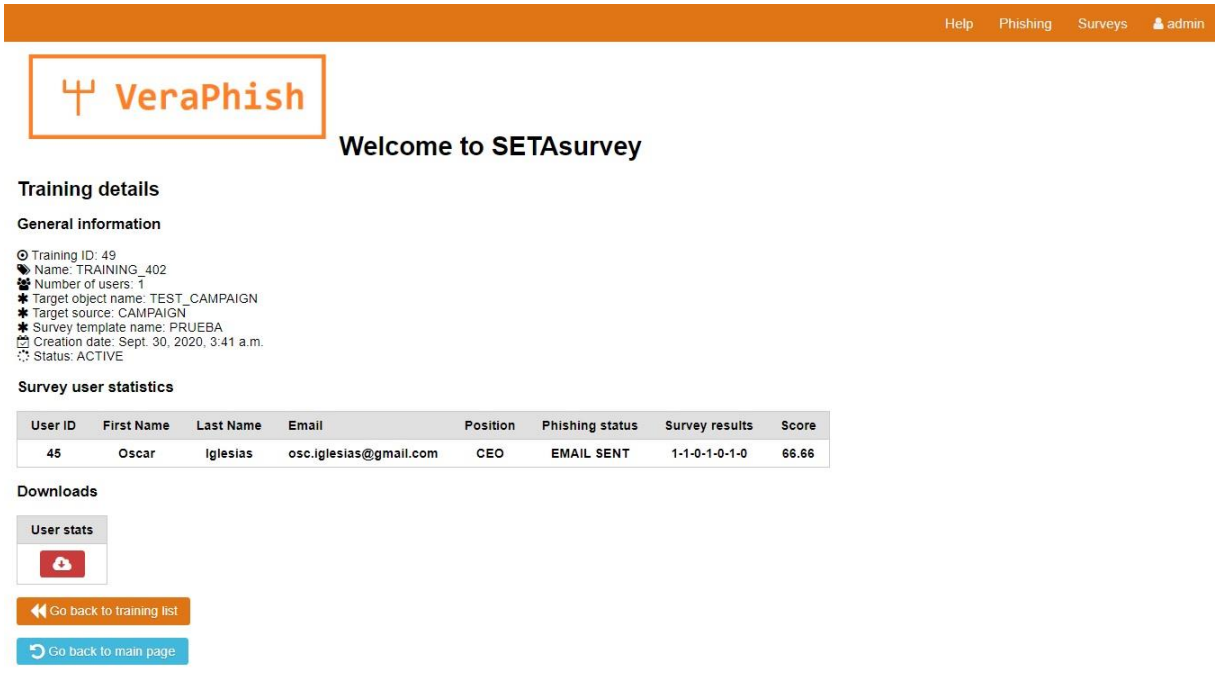

Ilustración 52. Veraphish: detalle de una Campaña de Formación

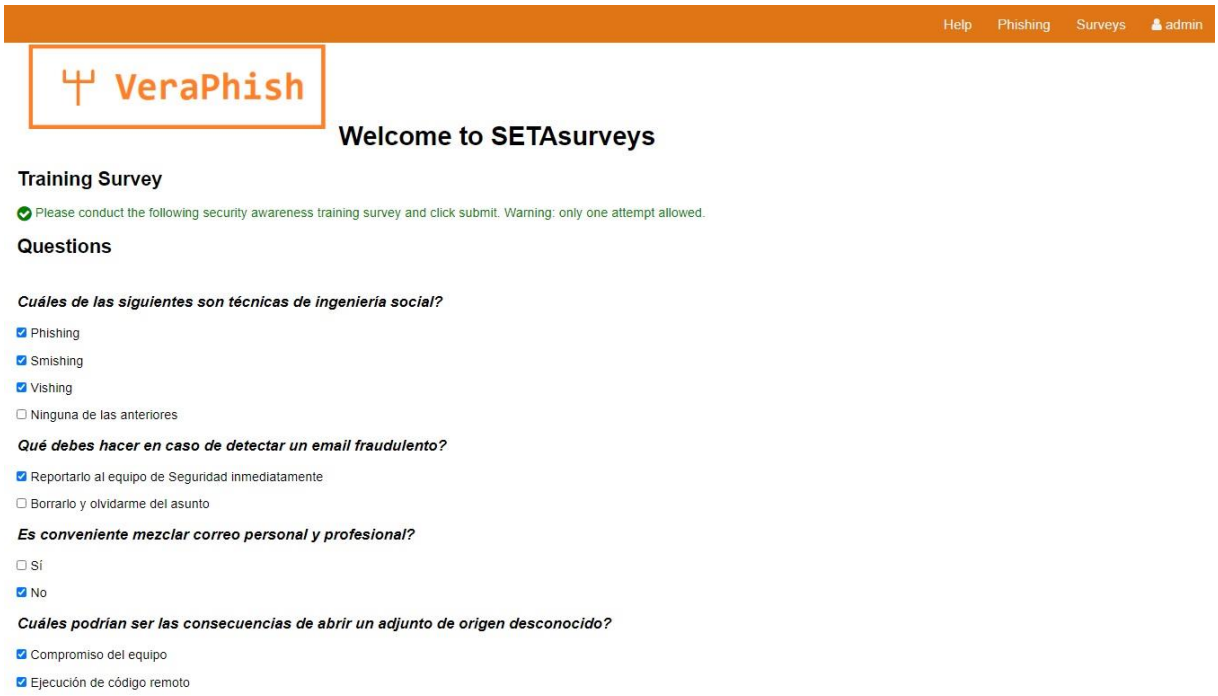

Ilustración 53. Veraphish: ejemplo de Test de Aprendizaje en curso (extracto)

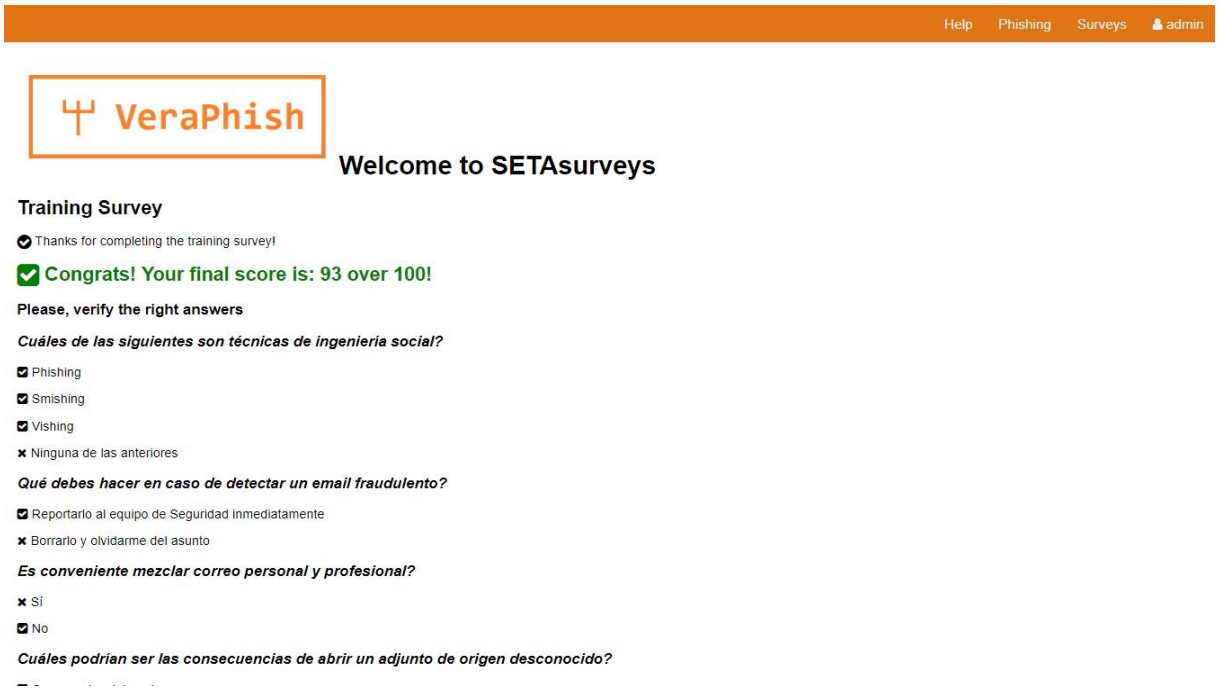

Ilustración 54. Veraphish: ejemplo de resultado de Test de Aprendizaje (extracto)