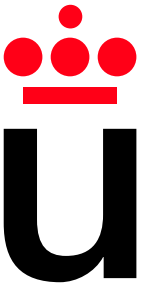

Universidad
Rey Juan Carlos

Escuela Técnica Superior
de Ingeniería Informática

Grado en Ingeniería de la Ciberseguridad

Curso 2023-2024

Trabajo Fin de Grado

GAMIFICACIÓN COMO TÉCNICA DE PREVENCIÓN DE PHISHING

Autor: Alejandro Martín Polo
Tutor: David Concha Gómez

Agradecimientos

A mamá y papá, por creer siempre en mí y enseñarme que con trabajo duro y esfuerzo se pueden lograr grandes cosas. Gracias por vuestro apoyo incondicional que mostráis cada día y por estar ahí cada vez que me he caído para ayudar a levantarme. Juntos habéis logrado que las dificultades del recorrido universitario se sientan más leves y compartir este camino junto a vosotros será un recuerdo para toda la vida.

A mi grupo de amigos, que siempre me han apoyado y me han servido para despejarme en momentos difíciles, han sido mi vía de escape necesaria para volver al trabajo con más fuerzas y energía.

A Sicilia, Jaime, Joselu y al resto de compañeros por formar parte de mi vida universitaria, convirtiendo la pena de los suspensos en momentos divertidos, que seguro recordaremos al hacernos mayores, y las alegrías de los aprobados en momentos de euforia. Gracias por hacer de cada una de las clases algo mejor, por cada uno de los recuerdos y por haberlo hecho todo mucho más llevadero. Cada uno de vosotros habéis dejado vuestra huella en este viaje.

Finalmente, a todas las personas de las que he aprendido algo a lo largo del trayecto, gracias. Cada aprendizaje, consejo y lección me lo llevo para siempre. Gracias de corazón.

Resumen

Este trabajo se centra en la exploración del uso de mecanismos de gamificación para la divulgación de problemas de ciberseguridad centrado en la ingeniería social, se explicarán las diferentes técnicas y métodos que se conocen, con un enfoque particular en la modalidad del phishing, identificada como una de las amenazas más actuales y peligrosas en el ámbito de la ciberseguridad.

La ingeniería social explota las vulnerabilidades humanas, manipulando a las personas para que revelen información confidencial o realicen acciones que comprometan la seguridad de sistemas y datos. El phishing en particular, implica el envío de correos electrónicos fraudulentos que aparentan ser de fuentes confiables, con el objetivo de engañar a los destinatarios para que revelen información sensible o instalen software malicioso.

El objetivo principal de este trabajo es diseñar y desarrollar un juego educativo que permita a los usuarios aprender a identificar y manejar correos electrónicos potencialmente maliciosos. A través del juego, los usuarios se enfrentarán a diferentes escenarios de correos electrónicos, tanto legítimos como de phishing, y deberán tomar decisiones sobre la legitimidad de estos mensajes. Este enfoque lúdico tiene como finalidad mejorar la comprensión y las habilidades de los usuarios para detectar intentos de phishing.

La gamificación en la capacitación sobre ciberseguridad mejora significativamente la participación y la motivación de las personas al integrar elementos de juego como puntos, niveles y recompensas. Esto transforma la formación tradicional, a menudo más aburrida, en una experiencia dinámica y atractiva.[1].

Para la implementación del juego, se utilizaron diversas herramientas y técnicas, incluyendo la biblioteca Dear PyGui para la creación de la interfaz gráfica. Además, se creó un sistema de correos electrónicos que simulan situaciones reales, permitiendo a los usuarios experimentar y aprender de manera interactiva.

En conclusión, este trabajo tarta de hacer uso de un juego educativo como herramienta para mejorar la conciencia y habilidades en ciberseguridad. De cara a un futuro se podría ampliar el juego para incluir otros tipos de amenazas de ingeniería social y mejorar la interfaz gráfica, haciendo así posible analizar los datos obtenidos de los resultados de los distintos jugadores.

Palabras clave:

- Python
- Ciberseguridad
- Ingeniería Social
- Interfaz gráfica
- Phishing
- Ciberataque
- Gamificación

Índice de contenidos

Índice de figuras

1 Introducción

En la actualidad, las nuevas tecnologías han facilitado en gran medida la interacciones sociales en internet, lo cual ha permitido que nos comuniquemos a través de diversos medios como las redes sociales y el correo electrónico sin salir de casa. Sin embargo, esta facilidad también ha permitido que cibercriminales aprovechen la confianza de los usuarios mediante técnicas de ingeniería social, siendo el phishing una de las más comunes.

1.1. Contexto y alcance

En la era digital actual, estamos en constante interacción social, ya que el avance tecnológico nos permite comunicarnos fácilmente de diferentes formas, sin ni siquiera tener que salir de nuestra casa, a través de distintos medios como las redes sociales o el correo electrónico. Sin embargo, no todo es positivo, ya que permite a personas malintencionadas interactuar con la sociedad de forma anónima, con objetivos inmorales que pueden afectarnos a todos, esta omnipresencia lo ha convertido en un blanco atractivo para los cibercriminales que buscan explotar la confianza de los usuarios mediante técnicas de ingeniería social. Entre estas técnicas, el phishing es una de las más utilizadas y dañinas, ya que busca engañar a los destinatarios para que revelen información sensible, como credenciales de acceso, datos financieros, o información personal.

El phishing se presenta en muchas formas, desde correos electrónicos que parecen legítimos provenientes de instituciones financieras hasta mensajes urgentes que requieren una acción inmediata. Estos correos maliciosos están diseñados pa-

ra parecer auténticos y suelen utilizar tácticas psicológicas de manipulación para incitar al usuario a tomar decisiones apresuradas y peligrosas.

La lucha contra el phishing no consiste únicamente en soluciones tecnológicas, sino también la educación y concienciación de los usuarios. Los filtros de correo y las herramientas de seguridad pueden detectar muchos correos de phishing, pero los cibercriminales constantemente desarrollan nuevas estrategias para evadir estas defensas. Por lo tanto, la capacidad de los usuarios para identificar y manejar estos correos es crucial para la protección general contra este tipo de amenazas.

Este Trabajo de Fin de Grado (TFG) se centra en abordar este problema mediante el desarrollo de una interfaz simulada que imita una bandeja de entrada de correo electrónico. El objetivo es proporcionar un entorno interactivo donde los usuarios puedan practicar la identificación de correos de phishing en un entorno seguro. Esta simulación permitirá a los usuarios enfrentarse a una variedad de correos electrónicos, evaluando cada uno para determinar si es legítimo o malicioso.

Al ofrecer esta herramienta, se busca mejorar las habilidades de los usuarios para detectar y reaccionar ante intentos de phishing, reduciendo así la probabilidad de que caigan en estas trampas. Además, esta práctica puede servir como un complemento valioso para programas de formación en ciberseguridad, proporcionando una experiencia práctica que refuerza el aprendizaje teórico.

La relevancia de este proyecto se sustenta en estadísticas alarmantes que muestran un aumento continuo en los ataques de phishing. Según diversos estudios, un gran porcentaje de incidentes de ciberseguridad comienzan con un correo de phishing exitoso. Por lo tanto, mejorar la capacidad de los usuarios para reconocer estos correos es una medida preventiva esencial para mitigar los riesgos asociados.

1.2. Ingeniería Social

En términos generales podemos decir que la Ingeniería Social es una técnica que emplean los ciberdelincuentes para ganarse la confianza del usuario y conseguir así que haga algo bajo su manipulación y engaño, como puede ser ejecutar un programa malicioso, facilitar sus claves privadas o comprar en sitios web fraudulentos.[2]

Según la Wikipedia, [3] "La ingeniería social es la práctica ilegítima de obtener información confidencial a través de la manipulación de usuarios legítimos. Es un conjunto de técnicas que pueden usar ciertas personas para obtener información, acceso o permisos en sistemas de información que les permitan realizar daños a la persona u organismo comprometidos y es utilizado en diversas formas de estafas

y suplantación de identidad. El principio que sustenta la ingeniería social es el de que, en cualquier sistema, los usuarios son el eslabón débil."

Con lo cual, podemos decir que la Ingeniería Social es la ciencia de persuadir hábilmente para lograr que las personas realicen alguna acción en algún aspecto de sus vidas con el objetivo de que el atacante pueda extraer información confidencial y usarla para beneficio propio. Aunque es menos sofisticada que otras estrategias de ataque en el mundo ciber, la ingeniería social puede tener consecuencias muy graves y a menudo, puede ser el primer paso de un ataque mayor.

El éxito de estas técnicas depende de la capacidad de los atacantes para manipular a las víctimas para que realicen determinadas acciones o proporcionen información confidencial. Hoy en día, la ingeniería social es reconocida como una de las mayores amenazas a la seguridad que enfrentan las organizaciones. La principal diferencia que encontramos entre la ingeniería social y las técnicas de hacking tradicionales la encontramos en que los ataques de ingeniería social pueden ser no técnicos y no implican necesariamente explotar o comprometer software o sistemas. Cuando tienen éxito, muchos de estos ataques permiten a los atacantes obtener acceso legítimo y autorizado a información confidencial. La ingeniería social se puede utilizar en lugar de, o en combinación con, amenazas y sobornos. El ingeniero social tiene como objetivo no dejar ningún rastro y, en general, dejar la menor huella posible, por lo que las amenazas y los sobornos no son lo más habitual. Así pues, un ataque de ingeniería social se centra principalmente en la vulnerabilidad de las personas. No importa como de seguro sea el sistema en sí mismo, dado que la mayor amenaza serán sus propios usuarios y el uso que hagan de él.

1.3. Phishing

Según IBM, los ataques de phishing son correos electrónicos fraudulentos, mensajes de texto, llamadas telefónicas o sitios web diseñados para engañar a los usuarios para descargar malware, compartir información confidencial o datos personales (p. ej., seguridad social y números de tarjetas de crédito, números de cuenta bancaria, credenciales de inicio de sesión) u otras acciones que expongan a sus organizaciones.[4]

El éxito de los ataques de phishing suele dar lugar a robos de identidad, fraudes con tarjetas de crédito, ataques de ransomware, filtraciones de datos y enormes pérdidas económicas para particulares y empresas[5].

El phishing es el tipo más común de ingeniería social, la práctica de engañar, presionar o manipular a las personas para enviar información o activos a las personas equivocadas. Los ataques de ingeniería social se basan en el error humano y en tácticas de presión para tener éxito. El agresor suele hacerse pasar por una

persona u organización en la que la víctima confía, por ejemplo, un compañero de trabajo, un jefe, una empresa con la que la víctima o su empleador hacen negocios, y crea una sensación de urgencia que lleva a la víctima a actuar precipitadamente. Los hackers y los estafadores utilizan estas tácticas porque es más fácil y menos costoso engañar a la gente que piratear un ordenador o una red.

1.4. Víctimas potenciales

Para lograr el éxito en un ataque de ingeniería social, es crucial seleccionar cuidadosamente a la víctima en función de sus características y vulnerabilidades específicas. Las personas más susceptibles y expuestas suelen ser:

- Personal Administrativo y de Apoyo: Incluyendo auxiliares administrativos, recepcionistas, y guardias de seguridad, que podrían no estar completamente conscientes del valor de la información que manejan.
- Personal con Privilegios Especiales: Como el personal de soporte técnico y administradores de sistemas, que tienen altos niveles de acceso y poder dentro de la organización.
- Fabricantes y Proveedores: Organizaciones que fabrican hardware, software u otros productos de interés para los piratas informáticos.
- Departamentos Específicos: Como los departamentos de contabilidad, recursos humanos u otros que manejan información altamente sensible y valiosa.

En términos generales, los objetivos típicos son aquellos que carecen de altos conocimientos sobre la ciberseguridad, aquellos que prestan ayuda a otros, o aquellos que tienen acceso privilegiado a información o activos valiosos, ya sea por la información que poseen o por el valor económico que tienen. En definitiva, prácticamente cualquier persona con acceso a cualquier parte del sistema puede ser un objetivo potencial para un ataque de ingeniería social.

1.5. Panorama actual

En el panorama actual de las amenazas cibernéticas, la ingeniería social se ha convertido en una herramienta fundamental para los ciberdelincuentes. Su capacidad para manipular el comportamiento humano la convierte en un arma poderosa para obtener información confidencial y acceder a sistemas informáticos.

A continuación, vamos a ver más a detalle distintos puntos que resaltan la relevancia de la ingeniería social en el mundo tecnológico actual, y por qué es

importante tener muy presente este tema, ya que afecta desde a las organizaciones más sofisticadas hasta a el usuario más común que haga uso de las tecnologías de la información.

1. Aumento de los ataques de ingeniería social:

   Los ataques de ingeniería social están en aumento. Según el Informe de Ciberseguridad de IBM de 2023[6], el 95 % de las infracciones de seguridad empresarial implicaron un elemento de ingeniería social. Los ciberdelincuentes se están volviendo más sofisticados en sus técnicas de ingeniería social, utilizando métodos como el phishing, el spear phishing y el smishing para dirigirse a sus víctimas de manera más efectiva. El teletrabajo ha ampliado la superficie de ataque para la ingeniería social, ya que los empleados están más aislados y pueden ser más susceptibles a los engaños.

2. Impacto significativo de los ataques:

   Los ataques de ingeniería social pueden tener un impacto devastador en las empresas y las personas. Las filtraciones de datos causadas por ingeniería social pueden dañar la reputación, interrumpir las operaciones y desencadenar importantes pérdidas financieras que pueden ocasionar un daño irreparable en las organizaciones afectadas. Los ataques de ransomware, a menudo iniciados mediante ingeniería social, pueden paralizar las operaciones y exigir pagos de rescate considerables.

3. Vulnerabilidad del factor humano:

   El factor humano es el eslabón más débil en la cadena de seguridad cibernética. Incluso los sistemas informáticos más seguros pueden ser vulnerables si los empleados son engañados para revelar información confidencial o hacer clic en enlaces maliciosos. La comprensión de las técnicas de ingeniería social es crucial para educar y capacitar a los empleados para que sean más resistentes a estos ataques, es por ello fundamental los proyectos formativos en materia de ciberseguridad que cada vez se producen de forma más frecuente en todas las organizaciones.

4. Ejemplos:

   Ataque de WannaCry en 2017[7]: Este ataque de ransomware, iniciado mediante phishing, afectó a más de 200.000 organizaciones en todo el mundo y causó miles de millones de dólares en daños.

   Robo de datos de Equifax en 2017[8]: Este ataque se llevó a cabo mediante phishing, lo que permitió a los ciberdelincuentes acceder a los datos personales de 147 millones de estadounidenses.

   Ataque al DNC en 2016[9]: Este ataque de spear phishing permitió a los ciberdelincuentes robar correos electrónicos del Comité Nacional Demócrata, lo que influyó en las elecciones presidenciales de Estados Unidos.

2

Estado del arte

En este apartado vamos a tratar de explorar los diversos tipos de atacantes, las etapas de los ataques, clasificaciones relevantes y ejemplos concretos de incidentes reales. Este enfoque proporciona una visión integral de las amenazas contemporáneas en el ámbito de la seguridad cibernética.

2.1. Tipos de atacantes

Hoy en día podemos encontrar diversos tipos de atacantes entre los que destacan[10]:

1. Hackers:

   Hoy en día cada vez el software es más seguro por lo que los hackers buscan formas de atacar que combinen la ingeniería social para aprovecharse del factor humano con sus conocimientos de ciberseguridad.

2. Probadores de seguridad:

   Un probador de seguridad es un profesional que examina las vulnerabilidades o posibles accesos no autorizados a sistemas. Las pruebas de seguridad implican verificar un sistema informático, red, aplicación web o perímetro para identificar debilidades que podrían ser explotadas por atacantes maliciosos. Estos profesionales utilizan diversas estrategias para realizar sus pruebas. Un probador de seguridad replicará los ataques que un ingeniero social malicioso podría usar para intentar comprometer el sistema.

3. Espionaje:

   Es la práctica de obtener de manera encubierta información sobre un gobierno extranjero o una industria competidora, con el propósito de otorgar una ventaja estratégica o financiera a su propio gobierno o corporación. Actualmente, en la ingeniería social, es una de las técnicas que utilizan los espías para intentar obtener información.

4. Ladrones de identidad:

   Un grupo de personas que obtiene datos personales de forma poco ética, como revisando la basura, y los usa para lograr sus objetivos. Esta información puede incluir el nombre, dirección, número de la Seguridad Social, dirección de correo electrónico, entre otros.

5. Empleados descontentos:

   Hay muchas razones que contribuyen al descontento de los empleados en el lugar de trabajo. Los empleados descontentos poseen dos componentes necesarios para causar daños: acceso y motivación. También se pueden agrupar en este apartado los exempleados, dónde se da el caso que un exempleado puede conservar el acceso a las aplicaciones corporativas una vez finalizado su empleo. Este acceso puede convertirse en el talón de Aquiles de la empresa. Si el exempleado se va de malas maneras, existe la motivación de usar ese acceso para orquestar un ataque dañino para la empresa.

6. Corredores de información:

   Empresas que recogen datos, incluyendo información personal de consumidores, de diversas fuentes para revender dicha información a sus clientes con diversos propósitos, como la verificación de identidad, registros, productos de marketing y prevención del fraude financiero.

7. Artistas del timo:

   Los estafadores participan en acciones fraudulentas o engañosas para defraudar a otros. El marketing masivo es un método común que utilizan los estafadores.

8. Cazadores de talento:

   Estos profesionales son una extensión del departamento de contrataciones de una empresa. Suelen ser personas con muchos recursos para satisfacer las necesidades de su empresa consiguiendo la mayor cantidad de información posible de sus distintos candidatos y detectando posibles fallos de estos. Deben ir más allá para obtener información, también deben ser capaces de comprobar si las motivaciones de la persona encajarán con su lugar de trabajo y el potencial de esta.

9. Vendedores:

   El arte de vender es un tipo de trabajo dentro del mundo laboral que hace uso de muchas técnicas que se utilizan en la Ingeniería Social. Estas técnicas suelen ser, por ejemplo: recopilar datos, maniobras de obtención de información, influencia, principios psicológicos, etc. Los vendedores deben usarlas para conseguir que aquello que venden cubra las necesidades de su futuro cliente y lo compren.

10. Gobiernos:

    Los gobiernos emplean métodos de ingeniería social de forma regular en sus esfuerzos por influir en la opinión pública para que apoye las acciones gubernamentales.

2.2. Etapas de un ataque

Las etapas son recopilación de información, establecimiento de relaciones y comunicación, explotación y ejecución[11].

1. Recopilación de la información:

   La probabilidad de éxito de la mayoría de los ataques depende de esta fase. Por esta razón, es común que los hackers dediquen la mayor parte del tiempo y esfuerzo en ella. Con la información adecuada, el atacante puede identificar el vector de ataque, las posibles contraseñas, las respuestas probables de las personas y perfeccionar sus objetivos. En esta etapa, el atacante se familiariza con el objetivo y crea un pretexto sólido.

2. Establecer relaciones y comunicación:

   Esta fase implica establecer una relación de trabajo con el objetivo. Es un punto crucial, ya que la calidad de la relación creada por el atacante determina el nivel de cooperación y hasta qué punto el objetivo ayudará al atacante a alcanzar sus metas. Esta fase puede variar significativamente cada vez. Por ejemplo, puede ser tan breve como un simple contacto visual para que el objetivo mantenga la puerta abierta del lugar que se desea atacar, o puede involucrar una conexión más personal a través de una llamada telefónica. En algunos casos, puede llegar a ser tan íntima que el objetivo muestre fotos familiares y comparta historias personales con el atacante. Otra posibilidad es crear una relación en línea con el objetivo mediante un perfil falso en un sitio de citas o redes sociales.

3. Explotación:

   Esta fase es cuando el atacante usa tanto la información recopilada como las relaciones para infiltrarse activamente en el objetivo. En esta fase, el

atacante se centra en mantener el impulso de cumplimiento que estableció en la segunda fase sin levantar sospechas. La explotación puede tener lugar mediante la divulgación de información aparentemente sin importancia o el acceso otorgado / transferido al atacante. Ejemplos de explotación exitosa que incluyen: El acto de mantener una puerta abierta o permitir que el atacante entre en las instalaciones, revelar la contraseña y el nombre de usuario por teléfono, insertar una unidad flash USB con un software malicioso en un ordenador de la empresa, abrir un archivo adjunto de un correo electrónico infectado, exponer secretos comerciales en una discusión con un supuesto compañero.

4. Ejecución:

   Esta fase se alcanza cuando se cumple el objetivo final del ataque o, por diversas razones, el ataque termina abruptamente para evitar levantar sospechas. Generalmente, un ataque concluye antes de que el objetivo comience a cuestionar lo que realmente está sucediendo. En lugar de eso, el atacante deja al objetivo con la sensación de haber hecho algo bueno por otra persona, asegurando así posibles interacciones futuras. Además, el atacante elimina cualquier rastro digital y se asegura de no dejar elementos o información que puedan delatarlo. Como resultado, se logran dos objetivos cruciales: el objetivo no se percata de que ha sido atacado y el atacante mantiene su identidad en secreto. Una estrategia eficaz para una buena salida sería planificar un ataque bien coordinado y ejecutarlo con fluidez como objetivo principal.

2.3. Clasificación de ataque

Los ataques se pueden clasificar en Human Based o Computer Based[12].

Human Based: En human-based attack el atacante ejecuta el ataque en persona interactuando con el objetivo para recopilar la información deseada. Por lo tanto, pueden influir en un número limitado de víctimas.

Computer Based: En software-based attack los ataques se realizan utilizando dispositivos como computadoras o teléfonos móviles para obtener información de los objetivos. Pueden atacar a muchas víctimas en pocos segundos.

2.4. Tipos de ataque

Ante los diversos tipos de ataques los más destacables son los siguientes[10]:

1. Suplantación de identidad:

   En este tipo de ataque, el ingeniero social se hace pasar por alguien en quien la víctima probablemente confíe u obedezca, de manera lo suficientemente convincente como para engañarla y obtener acceso físico a su oficina o al sistema de información. El ingeniero social recopila pacientemente todos los fragmentos de información encontrados o proporcionados por la víctima. Estas víctimas entregan información inocentemente, pensando que lo que dicen o hacen es completamente inofensivo, pero la combinación de los detalles suministrados le da al atacante lo necesario para alcanzar su objetivo. Cuanta más información tengan, mejor podrán evitar ser detectados. Los atacantes dedican tiempo a investigar su objetivo y obtienen información sobre la víctima o una empresa mediante:

   - Sitios web del mercado negro u otros ingenieros sociales.
   - Sitios web de la empresa.
   - Rebuscando en la basura (dumpster diving), consiste en buscar información de cualquier tipo, la cual pueda ser valiosa para un futuro ataque.
   - Escuchar a escondidas conversaciones de los empleados (eavesdropping), pueden existir diversos tipos, desde escuchar una conversación de forma presencial, por teléfono (wiretapping), o datos a través de internet (network sniffing).
   - Suplantación de identidad (phishing) del correo electrónico, ocurre cuando un atacante envía un mensaje de correo electrónico a un usuario con el objetivo de engañarle para que piense que el remitente es alguien conocido y de confianza. En esos emails pueden añadirse enlaces a sitios web maliciosos o adjuntarse archivos infectados con malware.
   - Pretextos telefónicos, los atacantes pueden hacer que parezca que sus llamadas telefónicas provienen de un número específico. Ya sea uno conocido o confiable para el destinatario, o uno que indique una ubicación geográfica específica. Los atacantes pueden utilizar la ingeniería social, a menudo haciéndose pasar por alguien de un banco o atención al cliente, para convencer a sus objetivos de que proporcionen información confidencial, como contraseñas, información de cuentas, números de seguridad social y más.
   - El acoso a los empleados a través de las redes sociales se ha convertido en el escenario perfecto para cometer delitos de ciberacoso mediante

diversas conductas: contratación de servicios utilizando indebidamente datos personales, intentos persistentes y repetidos de contactar con otra persona, o violaciones de su libertad para obtener información confidencial.

2. Baiting:

   El Baiting es un método de ataque de ingeniería social muy similar al phishing, pero se distingue por la promesa de un regalo que los piratas informáticos utilizan para atraer a las posibles víctimas. Los atacantes que utilizan Baiting, conocidos como baiters, pueden ofrecer a la víctima potencial la posibilidad de descargar películas, música y juegos gratis a cambio de proporcionar credenciales de inicio de sesión en un sitio web creado por el atacante. Otra forma de Baiting es cuando el atacante deja un dispositivo infectado con malware (como una unidad USB, una tarjeta Micro SD, etc.) en un área cercana a empresas, centros comerciales, hospitales, etc. El objetivo es que alguien encuentre el dispositivo y lo conecte a su ordenador sin saber que está instalando malware en su sistema. Una vez instalado el malware, el atacante puede avanzar con el ataque y explotar el sistema.

3. Phishing:

   El phishing es la práctica de enviar correos electrónicos que parecen provenir de fuentes confiables con el fin de influenciar u obtener información personal. Este tipo de ataque puede implicar el envío de correos electrónicos a las víctimas que contienen archivos adjuntos que pueden cargar malware en un ordenador, o enlaces a sitios web ilegítimos. Estos sitios web intentan persuadir a la víctima para que descargue software malicioso o divulgue información y datos personales confidenciales. El objetivo principal de los phishers suele ser obtener dinero o información valiosa. Entre los tipos más comunes se encuentran el spear phishing, whaling, vishing y smishing.

   Un ejemplo de un caso real de phishing es el siguiente:

   Se descubre que está circulando un correo electrónico fraudulento que promete entradas para la Eurocopa 2024. Este correo no proviene de la UEFA (Union of European Football Associations), sino que es un intento de phishing para robar datos personales y bancarios. En este correo se trata de hacer que el receptor participe en un concurso sobre fútbol, al responder esta encuesta, se le asegura que ha ganado las entradas para ver la Eurocopa. Los delincuentes utilizan una página web falsa que solicita información personal y un pago por gastos de envío, inscribiendo a las víctimas en un servicio de suscripción fraudulento.[13]

4. Spear Phishing:

   Es una estafa por correo o comunicaciones electrónicos dirigida a una persona, organización o empresa. Aunque a menudo tienen la intención de robar

datos con fines maliciosos, los ciberdelincuentes también pueden intentar instalar malware en el ordenador del objetivo. Llega un correo electrónico, aparentemente de un remitente confiable, pero en cambio lleva al destinatario desconocido a un sitio web malicioso. Estos correos electrónicos a menudo utilizan tácticas para llamar la atención de las víctimas. Por ejemplo, tener un reembolso de impuestos. Al personalizar sus tácticas de phishing, los spear phishers tienen tasas de éxito más altas para engañar a las víctimas para que otorguen acceso o divulguen información confidencial, como datos financieros o secretos comerciales. Whaling es un tipo de spear phishisng dirigido a perfiles de renombre dentro de una empresa como directivos.

Un ejemplo de caso real es el siguiente:

La metodología del ataque, es la siguiente[14]:

- Primero envían un correo electrónico de spear phishing con un texto sobre Ucrania.
- El email contiene un documento de Word adjunto con un artículo relacionado con la guerra en Ucrania.
- Una función maliciosa dentro del documento deja caer una secuencia de archivos.
- Finalmente, se descarga malware en el dispositivo de la víctima.

En Nicaragua se han descubierto correos electrónicos enviados a organizaciones del sector financiero con un documento de Word adjunto titulado Oscuros planes del régimen neonazi en Ucrania. El mismo lleva la firma de Alexander Khokhólikov, embajador ruso en el país latinoamericano.

5. Vishing:

   Los ataques de vishing se refieren al phishing telefónico que tienen como objetivo manipular a las personas para que proporcionen su información confidencial de tal forma que parezca que se está verificando, un ejemplo de esto es simular una llamada como si fuese del banco.

   Veamos un ejemplo que ha sucedido en la realidad:

   El ataque ocurrió de la siguiente manera, según los investigadores, como nos cuenta el Diario de Burgos[15]:

   El delincuente, probablemente mediante 'phishing' previo o el uso de algún tipo de malware, ha tenido conocimiento de los datos de acceso a la banca online de la víctima. Una vez tiene acceso, realiza una solicitud de extracción de efectivo en cajero mediante un código. Ese código lo recibiría la propia víctima en su teléfono móvil y no el presunto autor, con el presumible objetivo de dificultar la identificación del autor/es y la investigación de la Policía. Posteriormente, para tratar de obtener el código llaman a la

víctima haciéndose pasar por su propio banco (en ocasiones facilitando datos personales y bancarios de la propia víctima para lograr su confianza). Una vez que tienen el código de la verificación, un intermediario o 'mula' lo introduce en cualquier cajero de la red del banco y extrae el dinero que ha sido estafado.

6. Smishing:

   El smishing es una técnica que consiste en el envío de un SMS por parte de un ciberdelincuente a un usuario simulando ser una entidad legítima como una red social, el banco, una institución pública, etc, con el objetivo de robarle información privada o realizarle un cargo económico. Generalmente el mensaje invita a llamar a un número de tarificación especial o acceder a un enlace de una web falsa bajo un pretexto.

   Un ejemplo de este término es el siguiente:

   Como podemos leer en el INCIBE[16], se ha identificado una campaña de mensajes de texto fraudulentos que intentan suplantar a la Dirección General de Tráfico (DGT). En estos mensajes, se informa al destinatario de una presunta multa de tráfico y se le pide que revise el expediente a través del enlace adjunto. Al hacer clic en la URL proporcionada, el usuario es redirigido a un sitio web donde se le sustraerán sus datos personales y bancarios.

   Este es un caso que me pilla muy cercano ya que recibí este sms del que se habla en la noticia, en mi caso particular casi caigo en la trampa, ya que tenía la sospecha de que me podía llegar una multa, pero al ver la url pensé que mejor accedería por mi cuenta y no desde en el enlace, evitando así la estafa. Como vemos solo basta con una coincidencia para que este tipo de ataques se conviertan en algo realmente peligroso.

7. Pretexting:

   El Pretexting es una forma de ingeniería social que a menudo involucra que el atacante se encuentre cara a cara con el objetivo deseado. El atacante típicamente se hace pasar por otra persona, como un técnico, y utiliza accesorios como disfraces, órdenes de trabajo falsas o uniformes para parecer legítimo. La confianza es crucial en la ingeniería social: si el atacante no puede generar confianza entre él y el objetivo, es probable que falle en su intento. El atacante inventa historias ficticias para ganar acceso a datos sensibles u otros objetivos. Los atacantes más sofisticados intentarán manipular a la víctima para que realice acciones que exploren las debilidades estructurales de una organización o empresa.

   Un ejemplo de Pretexting sería un atacante que se presenta como un auditor externo de servicios IT y manipula al personal de seguridad de la empresa para que le permita ingresar al edificio y acceder a la infraestructura de datos.

8. Tailgating:

   También conocido como Piggybacking, es un método de ataque de ingeniería social mediante el cual un atacante busca ingresar a una zona restringida, donde el acceso es controlado por sistemas de control de acceso electrónico simplemente caminando detrás de una persona que tiene derechos de acceso. Un tipo común de ataque es un atacante que se hace pasar por un conductor de reparto y espera fuera de un edificio. Cuando un empleado obtiene la aprobación de seguridad y abre la puerta, el atacante pide que el empleado sostenga la puerta, obteniendo así acceso de alguien que está autorizado a ingresar a la organización. Sin embargo, a veces el tailgating no funciona con organizaciones a gran escala, por lo que todas las personas que ingresan al edificio deben pasar una tarjeta para obtener acceso.

9. Scareware:

   Este tipo de ataque implica bombardear a las víctimas con falsas alarmas y amenazas ficticias. Los usuarios son engañados haciéndoles creer que su sistema está infectado con malware, lo que los lleva a instalar un software que no ofrece ningún beneficio real o que es el malware en sí. Un ejemplo común de Scareware son los pop-ups de aspecto legítimo que aparecen en el navegador mientras se navega por la web, mostrando mensajes como "Su ordenador puede estar infectado con programas dañinos de spyware". Ofrecen instalar una herramienta o redirigen a un sitio malicioso donde la computadora podría infectarse. El Scareware también se distribuye a través de correos electrónicos no deseados que contienen advertencias falsas o promociones para que los usuarios compren servicios inútiles o dañinos.

10. Shoulder Surfing:

    Es una técnica de ingeniería social empleada por los atacantes con el objetivo de conseguir información de un usuario en concreto. Puede parecer mentira, pero es una técnica muy provechosa, que permite robar credenciales, contactos, códigos de desbloqueo (PIN, patrón, etc.), incluso datos bancarios. El éxito reside en la sencillez y paciencia del atacante, y es que ninguno de nosotros llega a ser consciente cuando viajamos en metro, en el autobús o en tren de que, quien se sienta a nuestro lado o se encuentra muy próximo a nosotros, puede estar observando nuestros movimientos en el dispositivo con intenciones maliciosas. Mediante la tecnología de hoy en día, el atacante puede hacer uso de diferentes dispositivos, para ayudar a conseguir su objetivo, desde minicámaras, prismáticos, móviles...

11. Office Snooping:

    Esta técnica parecida a la anterior es aprovechar el momento en que la víctima se ausenta de su lugar de trabajo para revisar y ojear toda la información visible y accesible que has dejado por exceso de confianza.

Estos posibles datos pueden llegar a ser, contraseñas apuntadas en un papel, sesiones abiertas de ordenador, etcétera.

12. Quid Pro Quo:

    El atacante consigue información a cambio de un teórico beneficio. Hay varios ejemplos, uno típico es que el atacante provoca algún tipo de problema en tu ordenador, por ejemplo, interfiere la banda de la red wifi y de pronto no tienes acceso a Internet. Casualmente recibes una llamada, "Hola, soy del departamento de IT, esta mañana estamos teniendo algunos problemas con el acceso a Internet, ¿puedes intentar navegar?", por supuesto no puedes, el atacante irá ganando tu confianza hasta que, para solucionarlo, te pedirá las credenciales de acceso, al dárselas, en un minuto, el problema desaparece, no parece necesario avisar a seguridad.

13. Watering Hole:

    Este tipo de ataques están dirigidos a empresas con altos niveles de seguridad, cuyos empleados frecuentemente visitan sitios web de confianza relacionados con el contenido de la organización. Estos sitios web son previamente estudiados e infectados por los atacantes, quienes primero perfilan a las posibles víctimas mediante un estudio de sus hábitos. Una vez que el empleado objetivo de la empresa visita el sitio web infectado, como suele hacer regularmente, su equipo se infecta con malware que permite a los atacantes tomar el control del mismo, espiar y robar información sensible de la compañía.

    Una vez que el objetivo del ataque está definido, los ciberdelincuentes enfocan sus esfuerzos en monitorear el tráfico de la empresa atacada, prestando especial atención a los sitios web visitados con frecuencia, y recopilando la mayor cantidad de información posible para crear un perfil detallado de la víctima. Cuando el usuario seleccionado visita el sitio web de confianza, los ciberatacantes intentan explotar las vulnerabilidades de su navegador y al mismo tiempo lo redirigen a un servidor malicioso donde pueden instalar malware para tomar el control del equipo.

2.5. Como evitar ser víctima

Existen varias recomendaciones que se pueden seguir para evitar en la medida de los posible ser víctima de este tipo de ataques, podemos destacar las siguientes proporcionadas por organizaciones reconocidas.

1. INCIBE:

    El Instituto Nacional de Ciberseguridad de España (INCIBE), anteriormente Instituto Nacional de Tecnologías de la Comunicación, es una sociedad

dependiente del Ministerio para la Transformación Digital y de la Función Pública a través de la Secretaría de Estado de Digitalización e Inteligencia Artificial y consolidada como entidad de referencia para el desarrollo de la ciberseguridad y de la confianza digital de ciudadanos, red académica y de investigación, profesionales, empresas y especialmente para sectores estratégicos.

Con una actividad basada en la investigación, la prestación de servicios y la coordinación con los agentes con competencias en la materia, INCIBE contribuye a construir ciberseguridad a nivel nacional e internacional.[17]

El INCIBE nos ofrece una serie de recomendaciones para evitar este tipo de ataques[18]:

- No abras correos que no has solicitado o proceden de usuarios desconocidos. Elimínalos y bloquea al remitente.
- No contestes en ningún caso a estos correos, ni envíes información personal como contraseñas, datos personales y bancarios.
- Mantén actualizados todos tus dispositivos y programas.
- Verifica quién te envía un mensaje antes de proporcionar cualquier información confidencial, aunque el mensaje aparentemente proceda de un usuario conocido.
- No descargues ficheros adjuntos que pueda contener el mensaje, ya que podrían ser maliciosos y contener malware.
- Utiliza software de seguridad actualizado, como un antivirus, para proteger tu dispositivo de posibles amenazas de seguridad.
- Activa la autenticación de dos factores siempre que un servicio online lo permita para aumentar la seguridad de tus cuentas.

2. McAfee:

   McAfee, LLC. es una compañía de software especializada en seguridad informática cuya sede se encuentra en Santa Clara, California. Su producto más conocido es su antivirus.[19]

   McAfee también nos ayuda con alguna recomendaciones para evitar el phishing[20]:

   - Incluso las empresas más grandes llegan a cometer pequeños errores en sus comunicaciones. Pero los mensajes de phishing suelen contener errores gramaticales, faltas de ortografía y otros errores flagrantes que las grandes empresas no cometerían. Si ve muchos errores gramaticales muy evidentes en un correo electrónico o SMS donde se le pide información personal, es muy probable que se trate de una estafa de phishing.

- Para reforzar su credibilidad, los estafadores suelen robar los logotipos de las organizaciones por las que se hacen pasar. Sin embargo, muchos de esos logotipos no se ven correctamente ya que, al copiarlos, no han respetado la proporción de la imagen o han rebajado demasiado su resolución. Si tiene que entrecerrar los ojos para distinguir el logotipo del mensaje, lo más probable es que se trate de phishing.
- El phishing siempre se basa en algún vínculo en el que se supone que hay que hacer clic. Para comprobar si el vínculo que ha recibido es auténtico:

  Pase el ratón por encima del vínculo para que se muestre la URL. Las URL falsas suelen estar "mal escritas", es una de las pistas más frecuentes. Si pasa el ratón por encima del vínculo, podrá ver la URL sin hacer clic en él. Si la dirección URL tiene aspecto sospechoso, no interactúe con ella.

  Haga clic en el vínculo con el botón derecho del ratón, cópielo y pegue la URL en un procesador de textos. Así podrá examinar detenidamente el vínculo en busca de errores gramaticales u ortográficos sin que le dirija a la página web potencialmente maliciosa.

  Si está usando un dispositivo móvil, puede comprobar la URL manteniendo pulsado el vínculo con el dedo.

  Si la URL que se ve no coincide con la entidad que supuestamente le ha enviado el mensaje, es probable que el correo electrónico sea falso.
- Los mensajes de phishing pueden variar, pero hay patrones comunes:

  a) Cuentas suspendidas: Notifican sobre la suspensión de una cuenta bancaria por actividad inusual.

  b) Autenticación de dos factores: Solicitan confirmar identidad con códigos de acceso.

  c) Devoluciones de impuestos: Fingen ser de la Agencia Tributaria para robar información.

  d) Confirmaciones de pedidos: Envían recibos falsos para infectar dispositivos con malware.

  e) Phishing en el trabajo: Parecen correos de directivos pidiendo transferencias a clientes cuando en realidad acaba en el bolsillo de los estafadores.

  Es crucial verificar la autenticidad antes de proporcionar información.

3

Objetivos

La ingeniería social explota la vulnerabilidad humana para realizar ciberataques. Este trabajo busca informar sobre los riesgos y tácticas de esta forma de ciberdelincuencia, haciendo especial hincapié en el phishing, tratando de aportar una herramienta que ayude a concienciar de los peligros de estas técnicas desde un punto de vista algo alejado a lo tradicional, haciendo de la enseñanza un juego.

3.1. Objetivo general

En un mundo cada vez más digitalizado, donde las interacciones virtuales ya son una parte fundamental de nuestra vida cotidiana, la ingeniería social se ha convertido en una de las técnicas más utilizadas por los atacantes para explotar la vulnerabilidad humana. Este trabajo no solo busca informar sobre los riesgos inherentes a la ingeniería social, sino también dotar al usuario de un cierto nivel de conocimiento que abarque desde los métodos más rudimentarios hasta alguna de las tácticas sofisticadas empleadas por los ciberdelincuentes.

Uno de los pilares fundamentales de este estudio es la concienciación. A través de casos reales y escenarios hipotéticos, se pretende mostrar cómo la ingeniería social puede infiltrarse en los sistemas de información y las redes personales, causando daños que van desde la pérdida de datos hasta el robo de identidad y el fraude financiero. La concienciación implica no solo reconocer las amenazas, sino también comprender las motivaciones detrás de estos ataques.

El impacto de la ingeniería social puede ser devastador, afectando no solo a individuos, sino también a organizaciones enteras. Por ello, otro objetivo crucial de este trabajo es minimizar dicho impacto a través de la educación y la preparación.

Finalmente, este trabajo busca fomentar una cultura de ciberseguridad proactiva. Al difundir información de manera accesible y comprensible, se espera que más personas tomen conciencia de su papel en la protección de la información y adopten una actitud de vigilancia constante. La colaboración entre individuos, comunidades y organizaciones es vital para construir un entorno digital más seguro, y este trabajo pretende ser un disparador para esa colaboración.

3.2. Objetivos específicos

Disponemos de una serie de objetivos que complementan y mejoran sustancialmente el objetivo principal. Estos le dan un plus de entendimiento sobre la Ingeniería Social y una base de conocimiento que pueda ayudar a entender y comprender mejor esta parte de la seguridad informática.

- Comprender en que consiste la Ingeniería Social para poder estar preparados ante esta categoría de ciberataque.
- Saber qué tipo de atacantes existen en la actualidad.
- Conocer los distintos tipos de ataques que existen y cuales de estos son los más empleados por los ciberdelincuentes para materializar sus ciberataques.
- Aprender de ataques reales producidos los cuales nos aportan conocimientos para que si presenciamos algo similar en un futuro podamos estar preparados y prevenidos.
- Conocer diferentes métodos para detectar estos ataques, desde los métodos más simples hasta los más enrevesados que haya.
- Concienciar sobre los riesgos y motivaciones de los ataques.
- Fomentar una cultura de ciberseguridad proactiva en la que los usuarios tenga curiosidad y se sigan formando para estar prevenidos antes ataques de ingeniería social que podamos encontrarnos en el futuro.

4

Descripción informática

En este capítulo explicaremos aspectos más técnicos del trabajo en relación con la interfaz gráfica, como el lenguaje de programación utilizado y las razones para ello, como las bibliotecas que han ayudado al desarrollo de la aplicación y que función han desempeñado en la misma.

4.1. Desarrollo en Python

Procedemos a ver los motivos por los que he decidido usar Python en lugar de otros lenguajes de programación.

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo[21].

1. Facilidad de uso:

    - Sintaxis clara y sencilla:
      Python es reconocido por su sintaxis clara y fácil de aprender, lo que lo hace una opción ideal tanto para principiantes como para desarrolladores experimentados. Su diseño se enfoca en la legibilidad del código,

utilizando una sintaxis sencilla y una estructura que evita la complejidad innecesaria. Esto permite a los desarrolladores centrarse en la lógica y funcionalidad de la aplicación en lugar de preocuparse por los aspectos técnicos del lenguaje.

Para los principiantes, la simplicidad de Python reduce la curva de aprendizaje, permitiéndoles escribir programas funcionales con un conocimiento previo mínimo de programación. La sintaxis clara y la semántica coherente de Python facilitan la comprensión de los conceptos fundamentales de la programación, como las variables, las estructuras de control de flujo (if, for, while) y las funciones.

Para los desarrolladores experimentados, Python ofrece una plataforma poderosa y flexible para desarrollar aplicaciones complejas. La capacidad de Python para manejar desde tareas simples de scripting hasta el desarrollo de aplicaciones web y análisis de datos avanzados lo convierte en una herramienta versátil en el arsenal de cualquier programador.

La filosofía de diseño de Python, conocida como "The Zen of Python", enfatiza principios como "la simplicidad es mejor que la complejidad" y "la legibilidad cuenta", reflejando su compromiso con un diseño intuitivo y accesible. Estos principios no solo mejoran la experiencia de desarrollo, sino que también fomentan mejores prácticas de programación, como el código limpio y bien documentado[22].

- Amplia comunidad:

  Python cuenta con una comunidad grande y activa que ofrece soporte y recursos abundantes. Esto facilita encontrar ayuda cuando se necesita y aprender de las experiencias de otros desarrolladores. Un ejemplo de esto lo podemos ver en esta página.

2. Productividad:

   - Desarrollo rápido:

     La sintaxis concisa y la escritura dinámica de Python permiten a los desarrolladores crear prototipos e iterar rápidamente. Este rápido ciclo de desarrollo es particularmente valioso en la acelerada industria tecnológica, ya que permite a los equipos dar vida a las ideas de manera eficiente y adaptarse a los requisitos cambiantes [23].

   - Bibliotecas extensas:

     Python ofrece una amplia gama de bibliotecas para diversas tareas, incluyendo manipulación de datos, visualización y análisis. Esto permite a los desarrolladores aprovechar código existente. Podemos ver más información sobre las bibliotecas de python en el siguiente enlace.

3. Flexibilidad:

- Multiplataforma:
  La compatibilidad multiplataforma de Python permite a los desarrolladores escribir código que se ejecute sin problemas en varios sistemas operativos. Esta característica no sólo ahorra tiempo sino que también mejora la portabilidad de las aplicaciones, garantizando una experiencia de usuario consistente en diferentes plataformas[23].
- Escalabilidad:
  Python es un lenguaje escalable que puede usarse para crear aplicaciones pequeñas y simples, así como grandes y complejas.

4. Integración:

   - Se integra con otros lenguajes:
     Python se puede integrar fácilmente con otros lenguajes, como `C++` y `C#`, lo que permite aprovechar las fortalezas de cada lenguaje para crear aplicaciones robustas.
   - Herramientas de desarrollo:
     Existen numerosas herramientas de desarrollo disponibles para Python, como IDEs y depuradores, que mejoran la experiencia de programación.

4.2. Uso de Dear PyGui

En esta sección veremos la herramienta que me ha permitido desarrollar la interfaz gráfica para el juego y los motivos que la han hecho formar parte de mi trabajo.

Dear PyGui es un conjunto de herramientas de interfaz gráfica de usuario (GUI) multiplataforma, acelerado por GPU, dinámico y fácil de usar para Python. Las características incluyen elementos GUI tradicionales, como botones, botones de opción, menús y varios métodos para crear un diseño funcional. Además, tiene una increíble variedad de gráficos dinámicos, tablas, dibujos, depurador y múltiples visores de recursos. Dear PyGui ofrece un marco sólido para desarrollar aplicaciones científicas, de ingeniería, de juegos, de ciencia de datos y otras que requieren interfaces rápidas e interactivas[24].

1. Fácil de Usar y Aprender:

   Dear PyGui ofrece una sintaxis simple e intuitiva que facilita a los desarrolladores el aprendizaje y la utilización de la biblioteca sin una curva de aprendizaje pronunciada. Esto es especialmente útil para todos los desarrolladores que desean implementar interfaces gráficas sin profundizar demasiado en los aspectos técnicos.

En la documentación oficial, se puede ver una herramienta que ofrecen los desarrolladores la cual nos permitirá ver las distintas posibilidades que podemos incluir en nuestras interfaces de forma sencilla.

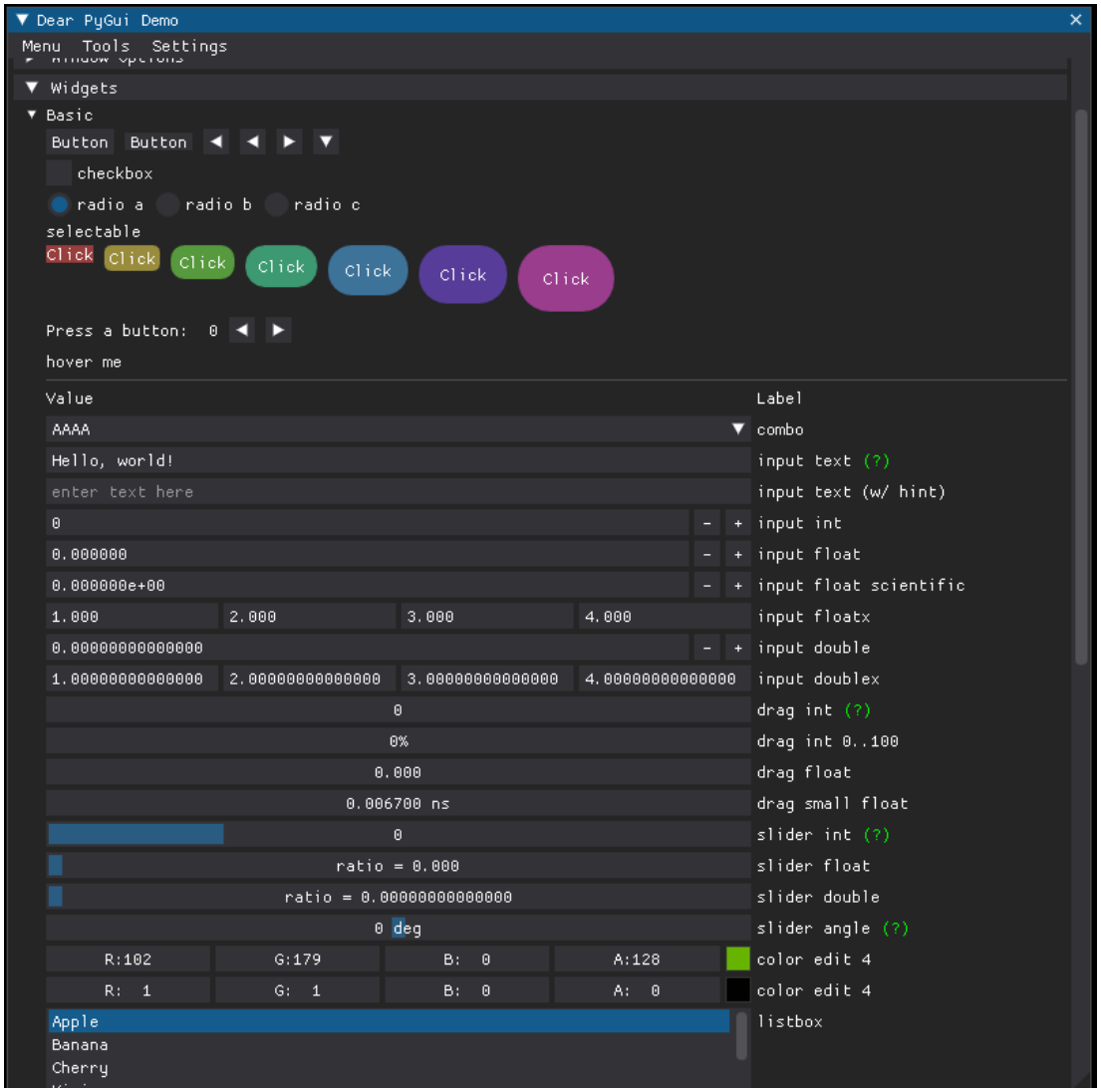

Figura 4.1: Posibilidades interfaz

En la imagen simplemente se muestran las opciones de botones básicos, pero hay muchos más elementos como podemos ver en la **web** oficial.

2. Interactividad y Flexibilidad:

   La biblioteca permite la creación de interfaces altamente interactivas y personalizables. Puedes crear ventanas, botones, cuadros de texto, y otros elementos de la interfaz gráfica con facilidad, ajustándolos a tus gustos y necesidades específicas.

3. Actualizaciones en Tiempo Real:

   Dear PyGui permite la actualización en tiempo real de los elementos de la interfaz, lo cual es bastante importante para aplicaciones dinámicas como la simulación de bandejas de entrada de correos electrónicos. Esto permite reflejar cambios al momento, como mostrar el contenido de un correo o actualizar el estado de un botón.

4. Soporte Multiplataforma:

   La biblioteca es multiplataforma, lo que significa que las aplicaciones desarrolladas con Dear PyGui pueden ejecutarse en diversos sistemas operativos, como Windows, macOS y Linux, sin modificaciones significativas en el código[25].

5. Desarrollo Rápido:

   Gracias a su simplicidad y eficiencia, Dear PyGui permite un desarrollo rápido de aplicaciones. Los usuarios de esta biblioteca pueden crear y desplegar interfaces funcionales en un corto periodo de tiempo, lo cual es ideal para proyectos académicos y de investigación como un Trabajo de Fin de Grado (TFG).

6. Manejo de Eventos y Callbacks:

   La biblioteca permite una gestión eficiente de eventos y callbacks, facilitando la implementación de funcionalidades complejas como la navegación entre correos, respuestas automáticas, y acciones condicionadas.

4.3. Biblioteca JSON

Según Ivan de Souza[26] en la práctica, .json es un archivo que contiene una serie de datos estructurados en formato de texto y se usa para transferir información entre sistemas. Es importante decir que, a pesar de su origen estar en el lenguaje JavaScript, JSON no es un lenguaje de programación.

JSON es una notación para la transferencia de datos que sigue un estándar específico. Por eso, puede emplearse en diferentes lenguajes de programación y de sistemas.

Los datos contenidos en un archivo en formato JSON deben estructurarse por medio de una colección de pares con nombre y valor o deben ser una lista ordenada de valores. Sus elementos tienen que contener:

Clave: corresponde al identificador del contenido. Por eso, debe ser una string delimitada por comillas.

Valor: representa el contenido correspondiente y puede contener los siguientes tipos de datos: string, array, object, number, boolean o null.

4.4. Biblioteca Threading

La biblioteca threading en Python proporciona un medio para crear y gestionar hilos (threads) en una aplicación. Un hilo es la menor unidad de procesamiento que puede ser programada por el sistema operativo. Los hilos permiten que un programa realice múltiples operaciones aparentemente de manera simultánea, lo cual es especialmente útil para mejorar la eficiencia de tareas que requieren operaciones de entrada/salida o que pueden ser paralelizadas.

Según el libro Programming Python, 4th Edition - Mark Lutz[27], el modulo threading usa internamente el modulo _thread para implementar objetos que representan hilos y herramientas comunes de sincronización. Está vagamente basado en un subconjunto del modelo de hilos del lenguaje Java, pero difiere en aspectos que solo los programadores de Java notarían.

4.5. El juego

El código desarrollado que se explicará posteriormente, tiene la función de proporcionar una interfaz gráfica sencilla e intuitiva que permita al usuario interactuar con esta sin grandes dificultades de comprensión sobre lo que tiene que hacer. El funcionamiento es sencillo, la interfaz simula una bandeja de entrada de correo electrónico, en esta bandeja la persona que este jugando al juego podrá ver los distintos correos que irán llegando progresivamente, teniendo que tomar dos decisiones simples, si el jugador considera que el correo que esta visualizando es legítimo entonces hará click en la opción que se alinea con este criterio, en caso contrario elegirá la opción contraria. De esta manera se irá progresando entre los diferentes correos que quieren poner a prueba los conocimientos del jugador sobre el tema que nos ocupa, la ingeniería social. Dependiendo de las decisiones que tome el usuario, visualizará unos correos u otros, el objetivo es que se le muestre al jugador los mensajes que más pueden retar a su conocimiento para que realmente aprenda algo de valor. Alguno de los correos están relacionados entre si, por lo que si el usuario que esta jugando toma uno de estos caminos es posible que para algún correo se generen respuestas automáticas, como si el jugador estuviera respondiendo al correo y hablando directamente al remitente, en los momentos en los que esto sucede se pueden ver las respuestas generadas automáticamente en otra sección de la interfaz, esta zona está reservada exclusivamente para visualizar estas respuestas.

4.5.1. Datos del programa

1. Correos:

   Para llevar a cabo el juego descrito anteriormente es evidente que será necesario disponer de los correos que el usuario ha de ir respondiendo, para la creación de estos hemos empleado un fichero a parte del propio fichero .py que contiene el código del juego, este fichero recibe el nombre de correos.json, como vemos en su extensión sabemos que es un fichero de tipo .json, este contiende distintos campos con información que utilizará el programa para que la interfaz pueda hacer visibles estos correos y se permita trabajar con ellos. Los campos que tenemos en este fichero por cada uno de los correos son:

   - Id:

     En este campo aparecerá un valor numérico único que representa a un determinado correo electrónico, cada correo tiene asociado un Id el cual no se podrá repetir para no confundir correos.

   - Asunto:

     Este campo es un string, es decir una cadena de caracteres o una frase que describe el asunto del que trata el correo al que pertenece, este campo corresponde con la parte que veremos del correo en la bandeja de entrada, antes de hacerle click para mostrar el contenido.

   - Contenido:

     Este campo contiene el mensaje del correo electrónico al que está asociado, los jugadores deberán leer el contenido de los correos y discernir si lo que leen les convence y piensan que es legítimo o si por el contrario notan algo sospechoso y deciden que no les parece de fiar. Es la parte más importante del correo.

   - Puntuación:

     Este campo contiene los puntos que el jugador obtendrá si pasa por ese camino de correos siguiendo sus elecciones, cuantos menos fallos más puntos obtendrá al final del recorrido y significará que posee más conocimiento del tema.

2. Respuestas:

   Como se ha comentado anteriormente en algunos de los caminos vamos a encontrar correos que simula una conversación entre el emisor y el receptor, es decir la persona que esta jugando al juego, como es complicado que esta respuesta la genere el jugador, ya que cada uno respondería una cosa, se generan respuestas preparadas según la opción que haya elegido el usuario en el juego, para introducir estas respuestas en el código del juego se ha creado un fichero a parte llamado respuestas.json, este fichero contendrá los siguientes campos:

- Id:

  En este campo tenemos un valor numérico que representa a un determinado correo electrónico, cada correo tiene asociado un Id el cual no se podrá repetir para no confundir correos.

- Asunto:

  Este campo es una cadena de caracteres o una frase que describe el asunto del que trata el correo al que pertenece y sobre el que se está generando una respuesta. Este campo corresponde con la parte que podremos ver en el botón de la sección diseñada específicamente para ver las respuestas.

- Contenido:

  Este campo contiene el mensaje de respuesta que se genera al seleccionar una opción determinada sobre el correo con el que tiene relación. Este es el texto que veremos al seleccionar uno de los botones en la sección de respuestas.

- Correo al que pertenece:

  Este campo esta formado por un número el cual hace referencia al correo sobre el cual estamos generando una respuesta, en el caso de que para un mismo correo se hagan varios intercambios de mensajes entre el emisor y receptor, las respuestas estarán catalogadas como que pertenecen al primero de los correos.

3. Árboles:

   Para llevar a cabo la estructura de decisiones y que dependiendo de las respuestas de los usuarios los siguientes correos a visualizar sean unos u otros hemos utilizado una estructura conocida como árbol, un árbol es una estructura que nos permite modelar las relaciones que existen entre diferentes objetos, estos están formados por un conjunto de nodos y las aristas, estas ultimas son las encargadas de representar las conexiones entre los distintos nodos.

   Para entenderlo y aplicándolo a nuestro contexto del videojuego, un nodo viene a representar un correo electrónico, mientras que una arista representa cual es el siguiente correo que verá el jugador dependiendo de la decisión que haya tomado.

   Para representar esto en el código hemos utilizado un archivo externo con el nombre de grafo.json, este archivo contiene los distintos correos electrónicos que lo forman, cada uno representado por un valor numérico único que no se puede repetir, y en cada número, es decir, en cada correo, tenemos cuál será el siguiente en el caso de que el usuario considere y elija que el correo con ese número es legítimo, o por el contrario, el correo que sigue si el jugador decide que no es legítimo.

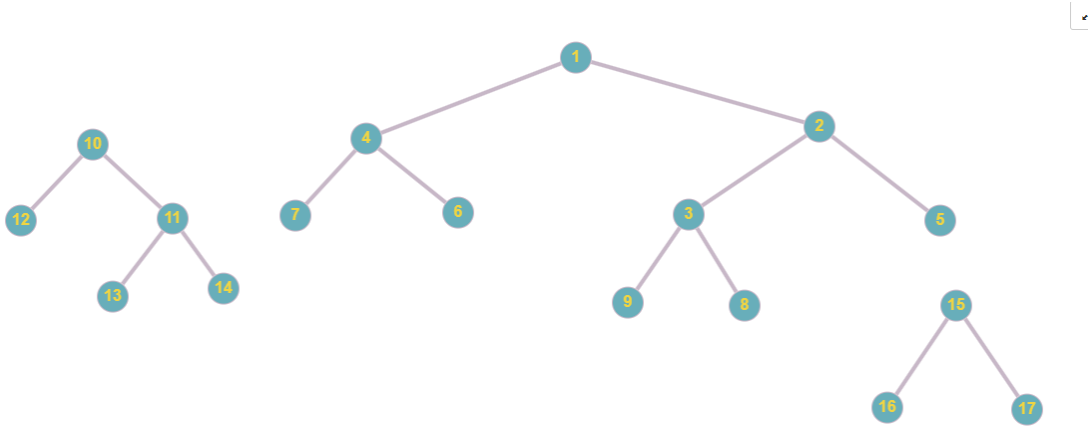

Figura 4.2: Árbol

Así es como se ve un ejemplo de árbol de correos, cada número es un correo y están unidos mediante aristas para conocer el siguiente correo que se mostrará dependiendo de las decisiones que se tomen.

5

Resultados Experimentales

A continuación se va a explicar como se ha utilizado la interfaz gráfica para dar vida al juego, como se trabaja con los distintos tipos de datos y estructuras y el funcionamiento.

5.1. Ingesta de datos

Una vez que tenemos los distintos datos necesarios para el correcto funcionamiento del código, debemos de realizar un tratamiento para después poder manejarlos con mayor facilidad y poder tener acceso directo a la información que necesitemos.

El código carga datos de varios archivos JSON y genera una estructura de datos combinada que representa correos electrónicos, un árbol y respuestas. Vamos a seguir el flujo del código paso a paso para entenderlo mejor.

1. Carga de correos y respuestas:

   El código primero carga un conjunto de correos desde un archivo JSON llamado correos.json. Para cada correo en este archivo, crea un diccionario donde las claves son los IDs de los correos y los valores son los correos completos. Esto se hace con la función cargar_correos.

```
1         def cargar_correos(filename):
2         with open(filename, 'r') as file:
```

```
            correos = json.load(file)
        return {correo['id']: correo for correo in correos
            }
```

El archivo de correos.json sigue el formato que se muestra en la siguiente imagen:

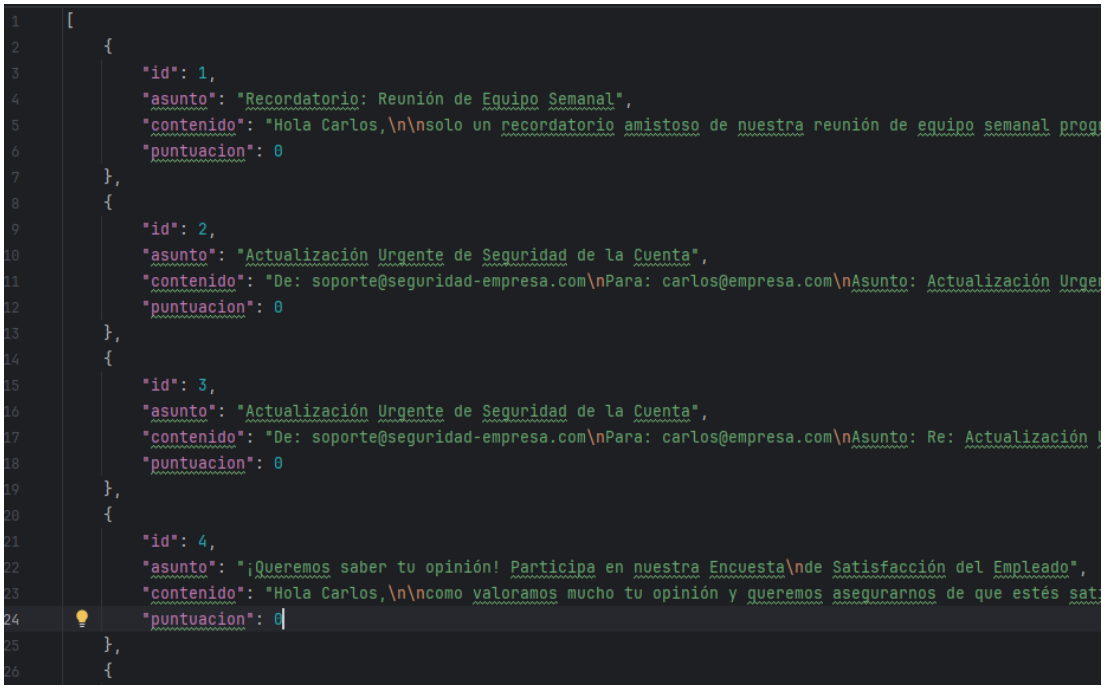

Figura 5.1: correos.json

Luego, carga un conjunto de respuestas desde un archivo JSON llamado respuestas.json, de manera similar a cómo se cargaron los correos. La función cargar_respuestas crea un diccionario donde las claves son los IDs de las respuestas y los valores son las respuestas completas.

```
        def cargar_respuestas(filename):
        with open(filename, 'r') as file:
            respuestas = json.load(file)
        return {respuesta['id']: respuesta for respuesta
            in respuestas}
```

El fichero de respuestas tiene el formato que se muestra en la siguiente imagen:

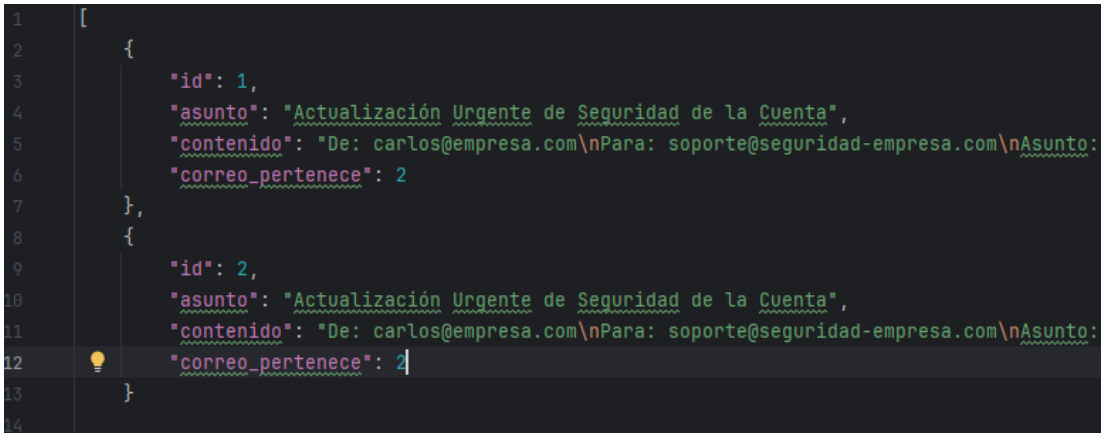

Figura 5.2: respuestas.json

2. Carga de árboles:

   El código carga tres árboles diferentes desde tres archivos JSON distintos: grafo.json, grafo2.json, y grafo3.json. Cada árbol se almacena en una variable separada (grafo, grafo2, grafo3). La función cargar_grafo se utiliza para abrir y cargar el contenido de cada archivo JSON en un diccionario. Cada uno de estos árboles representa la estructura de control que siguen los correos para saber cual será el próximo correo dependiendo de las respuestas del usuario

```
        def cargar_grafo(filename):
        with open(filename, 'r') as file:
            grafo = json.load(file)
        return grafo
```

   Estos ficheros siguen el siguiente formato:

Figura 5.3: Fichero árboles

3. Árbol manejable:

   Para poder trabajar con todos los datos de una manera más óptima se combinan los parámetros de entrada en un array que nos facilitará el acceso a la información deseada, funciona de la siguiente forma: Se inicializa un array vacío llamado array_grafo que será utilizado para almacenar los nodos y las decisiones de cada árbol. La función generar_grafo toma los tres árboles y el array vacío como parámetros. Itera a través de cada árbol (tres componentes conexas en total) y para cada nodo en el árbol, extrae las decisiones de "legítimo" y "phishing". Estas decisiones se añaden al array llamado array_grafo.

   Básicamente, esta función recopila las decisiones de cada nodo en los árboles y las organiza en un array.

```
        def generar_grafo(grafo1, grafo2, grafo3, array):
        numCompConexas = 3
```

```
        for i in range(numCompConexas):
            if i == 1:
                grafo = grafo2
            elif i == 2:
                grafo = grafo3
            inicio = grafo['inicio']
            fin = grafo['fin']
            while inicio <= fin:
                nodo = grafo['nodos'][str(inicio)]
                array.append([])
                array[inicio-1].append(nodo['decisiones'][
                    'legitimo'])
                array[inicio-1].append(nodo['decisiones'][
                    'phishing'])
                inicio += 1
```

La función poner_respuestas toma el array llamado array_grafo y el diccionario de respuestas. Para cada respuesta, añade el ID de la respuesta al nodo correspondiente en el array. Esto se hace basándose en el ID del correo al que pertenece la respuesta, permitiendo que cada nodo del árbol tenga asociadas las respuestas relevantes. Es destacable comentar que si una conversación de correos comienza en el correo con ID 2 por ejemplo, y se intercambian diferentes respuestas, todas ellas se adjuntaran como si perteneciesen al correo 2 para facilitar la programación.

5.2. Funcionamiento

A continuación vamos a explicar el funcionamiento del programa. No se va a seguir un orden secuencial de elementos en el código ya que considero que se va a entender mejor siguiendo un orden lógico de funcionamiento de la interfaz.

Para comenzar hacemos el import de las bibliotecas explicadas anteriormente.

Como hemos comentado antes para el desarrollo de esta interfaz hemos usado Dear PyGui el cual requiere hace una inicialización creando el contexto de la siguiente forma:

```
    dpg.create_context()
```

Al final del código, deberemos escribir unas líneas de esta biblioteca para definir el tamaño de la ventana de la interfaz y hacer comenzar la visualización de la misma.

```
    dpg.create_viewport(title='Juego IngSoc', width=1225,
        height=1050)
```

```
dpg.setup_dearpygui()
dpg.show_viewport()
dpg.start_dearpygui()
dpg.destroy_context()
```

1. Variables globales:

   Una vez realizado el proceso anterior, se definen una serie de variables globales que van a permitir facilitarnos la programación de distintas funcionalidades más adelante.

   - correo_seleccionado:

     Esta variable nos va a permitir saber que correo se está mostrando en la interfaz en todo momento, siendo de las más utilizadas a lo largo de todo el programa.

   - respuestas_votos:

     Esta variable corresponde con un array con tantas posiciones como correos tengamos, nos permite guardar las respuestas que el usuario va seleccionando en su recorrido por el árbol de correos.

   - nivel:

     Esta variable permite conocer el nivel del árbol en el que nos encontramos y se utiliza para la muestra por pantalla de correos de forma organizada por nivel.

   - guardar_correo:

     Es un array el cual permite ir almacenando los correos que vamos a mostrar posteriormente, el objetivo de esto es que los correos no se muestren según el usuario vaya respondiendo, si no que salgan de forma ordenada por tandas.

   - correos_json:

     Variable global que contiene todos los correos con sus distintos campos.

   - array_grafo:

     Variable que permite organizar la estructura de correos como un árbol en forma de array de arrays.

   - respuestas:

     Variable que contiene los distintos campos del archivo respuestas.json para su manejo.

2. Ventanas de la interfaz:

   La interfaz tiene tres ventanas, en la parte superior izquierda se puede ver la bandeja de entrada, en esta parte aparecerán todos los botones con el asunto del correo que deberemos ir pulsando para visualizar los mismos.

En la parte inferior izquierda tenemos una ventana que nos permitirá ver las respuestas que se generan como si contestase el jugador al emisor del mensaje. Finalmente en la parte derecha tenemos la ventana que usaremos para visualizar los distintos mensajes de texto como el contenido de los correos o el de las respuestas.

En la creación de las ventanas se establecen parámetros como su posición, su tamaño y un "mote" con el que hacerlas referencias para poder mostrar contenido o añadir objetos sobre ellas.

Para una mejor comprensión de la forma de la interfaz la podemos ver en la siguiente imagen:

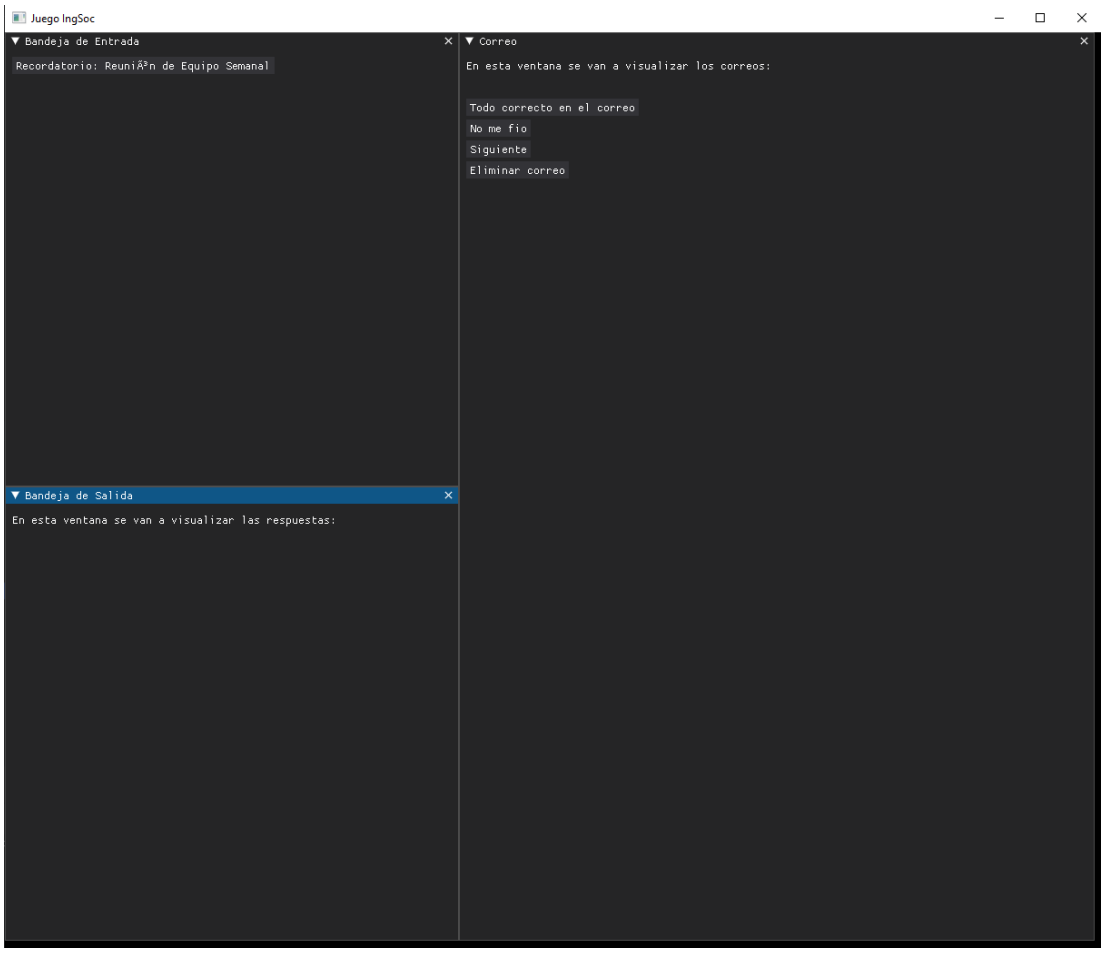

Figura 5.4: Interfaz gráfica

- Bandeja de entrada:
  Esta ventana se crea y en ella se introduce el primer correo directamente, desde este partirán todos los jugadores y después según sus decisiones irán por unos caminos u otros. Para añadir este primer correo, se hará de la misma forma para todos los demás, se crea un botón

al cual se le pasa un asunto, que es lo que se muestra en el botón, una función callback, esta es la función del código a la que se le llama una vez lo pulsamos, en este caso será mostrar_correo y por último un "mote" con el que podremos hacer referencia a este botón.

- Bandeja de salida:

  Esta se crea simplemente con un mensaje de texto informativo.

- Ventana de visualización:

  En esta ventana se crea un grupo de la biblioteca Dear PyGui el cual nos permitirá ir modificando su valor con el texto que queremos mostrar. También se añaden los botones que permiten al usuario votar que un correo es legítimo, votar que no lo es, un botón que permite sacar la siguiente tanda de correos y una opción que permite borrar un correo.

3. Mostrar correo:

   Esta función es llamada cada vez que el usuario decide pulsar uno de los botones de correo, una vez que pasa esto la función comprueba cual es el "mote" del botón que le está llamando para saber a que correo pertenece y poder mostrarlo. Una vez que sabemos que correo queremos mostrar consultamos en la estructura de correos y sacamos su contenido, para mostrarlo modificamos el contenido de la ventana de visualización. Finalmente se establece la variable global correo_seleccionado con el valor del "mote" del botón para saber que correo estamos mostrando.

4. Toma de decisiones:

   Una vez que el jugador haya leído el correo tendrá que decidir si considera que es legítimo o por el contrario si piensa que no lo es, para hacer esto tenemos dos funciones que se llaman votar_si y votar_no, como se puede intuir por el nombre, la primera será llamada cundo el usuario considere que el correo es legítimo y la otra al contrario.

   Estas funciones son casi iguales, lo primero que hacemos es sacar de la variable global que nos indica que correo se está mostrando, el número de correo que nos interesa, una vez que lo tenemos, en el caso de que el usuario vote que el correo es legítimo, en la posición del array respuestas_votos correspondiente al número de correo, se pone un 1 y se llama a la función siguiente_correo con el valor True, en caso contrario se pondrá un 0 y se llamará a la función con el valor False.

5. Siguiente correo:

   Cada vez que se toma una decisión sobre uno de los correo debemos de saber cual es el siguiente que tenemos que crear para que el jugador lo pueda ver y seguir con el juego, para llevar esto a cabo tenemos la función de siguiente_correo. Lo primero que hace esta función es obtener cual es el correo en el que está el usuario actualmente, este lo podemos hacer gracias

a la variable global correo_seleccionado, de ella obtendremos el número del correo actual. También debemos recordar que se nos pasa a la función un booleano, ya sea True si el jugador a votado que es legítimo o False si votó lo contrario, con esto y el número de correo en el que estamos podemos consultar nuestra estructura de datos almacenada en la variable array_grafo y ver que correo es el siguiente.

Una vez que sabemos cual es el siguiente correo nos guardamos el asunto y el contenido del mismo.

Como en la estructura nos pueden ir llegando diferentes correos que se suman al hilo de correos principal dependiendo el nivel en el que nos encontramos en el árbol de inicio, se comprueba esta condición y si se cumple se llama a otra función llamada siguiente_correo2, esta nos devolverá el asunto y contenido de un nuevo correo que tendrá que crearse.

También en esta función, en los casos en los que encontremos algún correo que tienen respuestas generadas, se manejan y crean con la ayuda de la función crear_boton_salida, esta simplemente crea el botón en la ventana inferior izquierda, que al hacerle click nos mostrará la respuesta por pantalla.

Cada vez que se tome una decisión sobre un correo, el siguiente correo correspondiente se guardará en la variable guardar_correo para ser imprimido correctamente en otra función.

6. Visualizar correos correctamente

   En esta función que recibe el nombre de vaciar_mochila se trata de dar realismo y no mostrar los correos siguientes de golpe, por lo que a través de la biblioteca threading se crean y se manejan unos hilos de ejecución de forma que podamos controlar cuando se van a crear los botones. Para hacer esto tenemos la función crear_boton_correo la cual será llamada por los distintos hilos y creará los botones de correo necesarios. La creación de hilos se hace de la siguiente forma:

```
1         hilo1 = threading.Timer(2, function=
              crear_boton_correo, args=(aux, aux1))
2         hilo1.start()
```

   Se le pone un nombre al hilo, se establece un tiempo de espera antes de llamar a la función, se proporciona la función a la que queremos llamar y se le pasan dos argumentos que son el asunto y la etiqueta que identifica al correo.

   La función crear_boton_correo simplemente crea un botón con los parámetros de asunto y etiqueta proporcionados.

   Cuando se responda al último de los correos y hayamos terminado, se mostrará un mensaje que indica el fin del juego y unas recomendaciones sobre como evitar el phishing.

7. Mostrar respuesta:

   Para esta tarea se llama a la función mostrar_respuesta la cual se encarga exclusivamente de ver que respuesta le llama, para añadir el contendido de esta en la ventana correspondiente consultando la estructura de respuestas.

8. Ventana emergente de borrado:

   Uno de los botones que nos aparece en la ventana de visualización es el de borrar correo, por lo que tendremos una función que lo maneja, esta función lo que hace es mostrar una ventana emergente que te solicita que confirmes si de verdad quieres borrar el correo, esto se hace para evitar posibles errores del usuario y que no borre un correo sin querer, una vez que el usuario da confirmación el correo desaparece.

9. Recuento de puntos y de aciertos:

   Una vez que el jugador haya terminado de responder todos los correos querrá saber cuantos a acertado y la puntuación que ha obtenido. Como ya hemos comentado anteriormente cada vez que el usuario da una respuesta se almacena en una variable que es un array, esta variable se llama respuestas_votos, en ella cada posición representa un correo y podemos encontrar tres tipos de respuestas, veremos un 1 si el usuario respondió que el correo es legítimo, veremos un 0 si respondió que no lo era y veremos el valor "None" si no se ha respondido a ese correo en concreto.

   Para contar los aciertos que ha tenido el usuario he creado un array con las respuestas correctas para cada correo y se comparan con el array de respuestas del usuario, cada coincidencia es un acierto y se van sumando.

   Para dar una puntuación lo haremos según los nodos hoja por los que pase, ya que para llegar allí habrá tenido una serie de fallos y aciertos que podemos medir y puntuar. Un nodo hoja es aquel que se encuentra en la parte mas baja del árbol y no tiene nodos hijos, es decir, es el final de un camino de nuestro árbol.

6

Conclusiones y trabajos futuros

Para finalizar, se comentará el trabajo que se ha realizado y una serie de mejoras que se podrían implementar y que por falta de tiempo no es posible, pero que pueden ser interesantes para desarrollarlas en un futuro.

6.1. Conclusiones

El presente trabajo ha abordado un tema crítico y de creciente relevancia en el ámbito de la ciberseguridad: la ingeniería social, con un enfoque particular en el phishing. A lo largo del proyecto, se ha realizado una exploración de las diversas técnicas de ingeniería social, analizando tanto sus métodos de ejecución como los perfiles típicos de sus víctimas. El phishing, como una de las formas más prevalentes de ingeniería social, ha sido desglosado en sus componentes esenciales.

Uno de los retos principales que se ha perseguido en este TFG ha sido tratar de dar importancia a la educación y la concienciación de los usuarios como una medida preventiva fundamental a través de la gamificación. Está demostrado que, aunque las tecnologías de seguridad avanzan continuamente, el factor humano sigue siendo el eslabón más débil de la cadena. Por lo tanto, la formación adecuada de los usuarios para reconocer y reaccionar ante intentos de phishing es crucial para minimizar los riesgos y crear un mundo más seguro.

La implementación de una herramienta interactiva, que simula una bandeja de entrada de correo electrónico, ha sido una pieza central de este proyecto. Esta

herramienta permite a los usuarios practicar en un entorno controlado y seguro, enfrentándose a correos electrónicos diseñados para imitar tanto mensajes legítimos como fraudulentos. A través de esta práctica, los usuarios pueden mejorar sus habilidades de detección, aumentando su capacidad para distinguir entre correos electrónicos genuinos y aquellos que presentan amenazas de phishing.

6.2. Trabajos futuros

Para continuar avanzando en la lucha contra la ingeniería social y mejorar la efectividad de las herramientas educativas, se proponen varias líneas de trabajo futuras:

1. Ampliación de escenarios de simulación:

   Una de las primeras expansiones que se podrían implementar es la creación de una variedad más amplia de escenarios y tipos de ataques de phishing. Actualmente, la herramienta se centra en un conjunto limitado de situaciones. Ampliar estos escenarios permitiría a los usuarios enfrentarse a una gama más diversa de tácticas de ingeniería social, proporcionando una formación más completa y realista.

   También se podrían añadir otro tipo de funcionalidades, como la de detectar fraudes en mensajes sms (smishing) o incluso adentrarse en el mundo de las redes sociales, ya que en estas plataformas también son habituales los engaños y los ciberataques de ingeniería social.

2. Evaluación y retroalimentación en tiempo real:

   Implementar un sistema de evaluación y retroalimentación en tiempo real proporcionaría a los usuarios una comprensión inmediata de sus acciones y decisiones. Este sistema podría ofrecer comentarios detallados sobre cada correo electrónico de phishing con el que el usuario interactúe, explicando por qué una determinada decisión fue correcta o incorrecta. Además, proporcionar sugerencias sobre cómo mejorar la identificación de correos fraudulentos en los casos específicos que se encuentre.

3. Investigación sobre nuevas técnicas de Ingeniería Social:

   La ingeniería social es un campo en constante evolución. Los cibercriminales desarrollan continuamente nuevas tácticas y estrategias para engañar a los usuarios. Por lo tanto, es fundamental continuar investigando y actualizando el contenido de la herramienta para incluir las últimas técnicas de este área de la ciberseguridad. Mantenerse al día con las tendencias emergentes permitirá que la herramienta siga siendo relevante y efectiva, asegurando que los usuarios estén preparados para enfrentar las amenazas más recientes.

4. Análisis de resultados y ajuste de estrategias de formación:

   Sería muy útil analizar los resultados que van obteniendo los distintos usuarios. Este análisis no solo proporciona una visión clara del desempeño individual, sino que también ofrece una comprensión detallada de las tendencias y patrones a nivel de grupo. Podríamos identificar casos como por ejemplo cuales son los tipos de correo que más daño hacen o que menos se aciertan y así poder hacer más énfasis en ellos o mejorar las explicaciones y ayudas para enfrentarlos.

Bibliografía

[1] MoldStud, "The role of gamification in cybersecurity education and training," 2024, accedido: 27-06-2024. [Online]. Available: https://moldstud.com/articles/p-the-role-of-gamification-in-cybersecurity-education-and-training

[2] INCIBE, "Ingeniería social," accedido: 10-06-2024. [Online]. Available: https://www.incibe.es/aprendeciberseguridad/ingenieria-social

[3] W. Contributors, "Ingeniería social (seguridad informática)," 2024, accedido: 10-06-2024. [Online]. Available: https://es.wikipedia.org/wiki/Ingenier%C3%ADa_social_(seguridad_inform%C3%A1tica)

[4] M. Kosinski, "¿qué es un ataque de phishing?" 2024, accedido: 15-06-2024. [Online]. Available: https://www.ibm.com/es-es/topics/phishing

[5] G. del Perú, "Alerta integrada de seguridad digital 118-2024-cnsd," 2024, accedido: 27-06-2024. [Online]. Available: https://cdn.www.gob.pe/uploads/document/file/6384364/5597931-alerta-integrada-de-seguridad-digital-118-2024-cnsd.pdf

[6] IBM, "Ibm report: Half of breached organizations unwilling to increase security spend despite soaring breach costs," 2023, accedido: 15-06-2024. [Online]. Available: https://newsroom.ibm.com/2023-07-24-IBM-Report-Half-of-Breached-Organizations-Unwilling-to-Increase-Security-Spend-Despite-Soaring-B

[7] W. contributors, "Ataques ransomware wannacry," 2024, accedido: 15-06-2024. [Online]. Available: https://es.wikipedia.org/wiki/Ataques_ransomware_WannaCry

[8] ——, "2017 equifax data breach," 2024, accedido: 15-06-2024. [Online]. Available: https://en.wikipedia.org/wiki/2017_Equifax_data_breach

[9] ——, "Ciberataques al comité nacional demócrata," 2024, accedido: 15-06-2024. [Online]. Available: https://es.wikipedia.org/wiki/Ciberataques_al_Comit%C3%A9_Nacional_Dem%C3%B3crata

[10] G. Lluís and L. Alberto, "Estudio de los ataques y su defensa en la ingeniería social," 2022, accedido: 01-06-2024.

[11] E. T. A.-S, "The attack cycle," 2022, accedido: 01-06-2024. [Online]. Available: https://www.social-engineer.org/framework/attack-vectors/attack-cycle/

[12] F. Salahdine and N. Kaabouch, "Social engineering attacks: A survey," *Future internet*, vol. 11, no. 4, p. 89, 2019.

[13] Maldita.es, "Cuidado con este supuesto sorteo para ganar dos entradas para la eurocopa: es 'phishing' · maldita.es - periodismo para que no te la cuelen," 2024, accedido: 17-06-2024. [Online]. Available: https://maldita.es/timo/bulo/20240617/entradas-eurocopa-uefa-phishing/

[14] Bernardo.valades, "Spear phishing basado en la guerra de ucrania afecta a latinoamérica," 2022, accedido: 01-06-2024. [Online]. Available: https://www.segurilatam.com/actualidad/spear-phishing-basado-en-la-guerra-de-ucrania-afecta-a-latinoamerica_20220406.html

[15] D. de Burgos, "Estafan 250 euros a un burgalés con el método del 'vishing'," 2024, accedido: 01-06-2024. [Online]. Available: https://www.diariodeburgos.es/noticia/zcf0472f1-07b7-16bc-ed65f8bb0ab13fc7/202404/estafan-250-euros-a-un-burgales-con-el-metodo-del-vishing

[16] INCIBE, "Detectadas campañas que suplantan la identidad de varias entidades bancarias a través de smishing," 2024, accedido: 27-06-2024. [Online]. Available: https://www.incibe.es/ciudadania/avisos/detectadas-campanas-que-suplantan-la-identidad-de-varias-entidades-bancarias

[17] ——, "Qué es incibe," 2024, accedido: 27-06-2024. [Online]. Available: https://www.incibe.es/incibe/informacion-corporativa/que-es-incibe

[18] ——, "Phishing," 2024, accedido: 18-06-2024. [Online]. Available: https://www.incibe.es/ciudadania/tematicas/ingenieria-social-fraudes-online/phishing

[19] Wikipedia, "Mcafee," 2024, accedido: 27-06-2024. [Online]. Available: https://es.wikipedia.org/wiki/McAfee

[20] McAfee, "Ejemplos de phishing: cómo detectar un correo de phishing," 2024, accedido: 18-06-2024. [Online]. Available: https://www.mcafee.com/blogs/es-es/internet-security/ejemplos-de-phishing-como-detectar-un-correo-de-phishing//

[21] aws, "What is python? - python programming language explained - aws," 2024, accedido: 28-06-2024. [Online]. Available: https://aws.amazon.com/es/what-is/python/

[22] Tuxskar, "Pep 20 - el zen de python," 2020, accedido: 17-06-2024. [Online]. Available: https://elpythonista.com/zen-de-python

[23] S. C. I. Services, "Why choose python over other programming languages?" 2024, accedido: 17-06-2024. [Online]. Available: https://www.linkedin.com/pulse/why-choose-python-over-other-programming-languages-superiorcodelabs-oh69c

[24] D. PyGui, "Dear pygui's documentation," 2021, accedido: 28-06-2024. [Online]. Available: https://dearpygui.readthedocs.io/en/latest/

[25] Juanweb, "Desarrollo de interfaces de usuario con dearpygui en python," 2023, accedido: 14-06-2024. [Online]. Available: https://codigospython.com/desarrollo-de-interfaces-de-usuario-con-dearpygui-en-python/

[26] I. de Souza, "Archivo json: descubre qué es y para qué sirve," 2021, accedido: 16-06-2024. [Online]. Available: https://rockcontent.com/es/blog/archivo-json/

[27] M. Lutz, *Programming Python*, 4th ed. O'Reilly Media, 2010, página de interés 241.