Simulazione di email di phishing

Introduzione

L’obiettivo di questo esercizio è simulare un attacco di phishing, utilizzando ChatGPT come strumento di supporto alla creazione dell’e-mail. Lo scopo non è quello di ingannare realmente un utente, ma di comprendere come vengono strutturate le e-mail di phishing, quali tecniche psicologiche vengono utilizzate e quali elementi dovrebbero far scattare un campanello d’allarme nel destinatario.

Scenario scelto

Lo scenario che ho scelto è quello di una presunta comunicazione da parte di una banca. Questo contesto è particolarmente realistico perché le banche sono spesso utilizzate come pretesto nei tentativi di phishing sfruttando la fiducia degli utenti e la paura di perdere l’accesso al proprio conto. L’obiettivo dell’attacco è quello di portare la vittima a cliccare su un link e inserire le proprie credenziali di accesso, fingendo la necessità di una verifica urgente dell’account.

Il messaggio è organizzato in modo da creare urgenza, facendo credere al destinatario che il conto possa essere sospeso se non viene effettuata immediatamente /un’azione correttiva/la verifica dell’account/

Strumenti Utilizzati

Per la realizzazione dell'esercizio sono stati selezionati strumenti standard del settore Penetration Testing:

Kali Linux: Sistema operativo utilizzato come piattaforma di attacco. Kali Linux è stato impiegato come sistema operativo per l’ambiente di test,

poiché offre un contesto sicuro e isolato ed è ampiamente utilizzato nel settore della cyber security.

Gophish: Framework open-source per il phishing. A differenza di tool manuali, Gophish permette di gestire intere campagne, tracciare l'apertura delle email, i click sui link e l'inserimento dati in un database centralizzato, simulando un vero scenario aziendale.

Ideazione dello Scenario

Per massimizzare il tasso di successo ho scelto uno scenario bancario basato sui principi di persuasione, urgenza e paura.

- Vettore d'attacco: Email spoofing.
- Target: Clienti di un istituto bancario generico.
- Pretesto: Rilevamento di una transazione fraudolenta e conseguente blocco preventivo dei fondi.
- Obiettivo: Indurre la vittima a visitare una pagina clone e inserire Codice Cliente e PIN

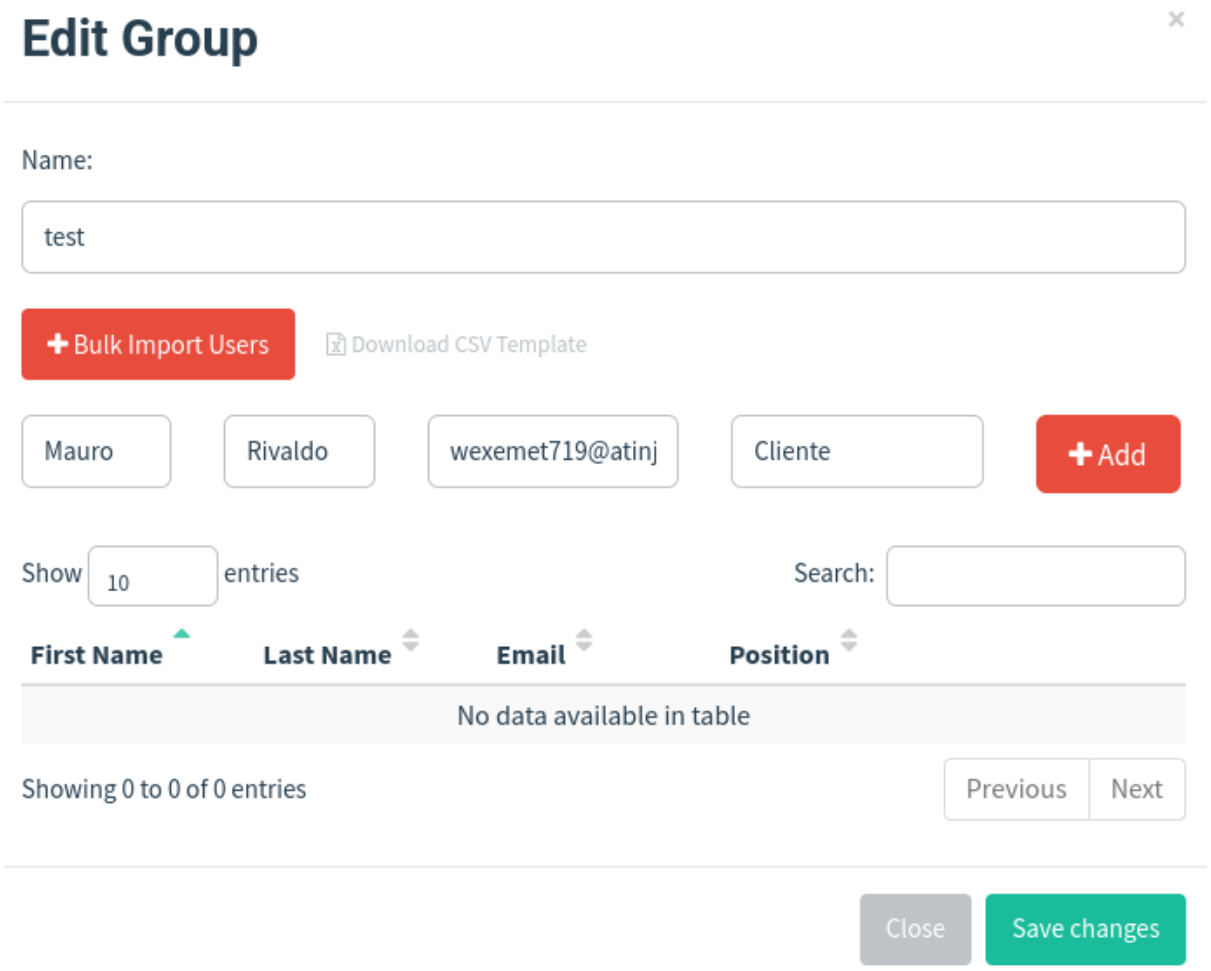
Configurazione della Campagna

Implementazione Tecnica

Definizione del Target (Users & Groups)

Ho creato un gruppo di destinatari denominato `test`.

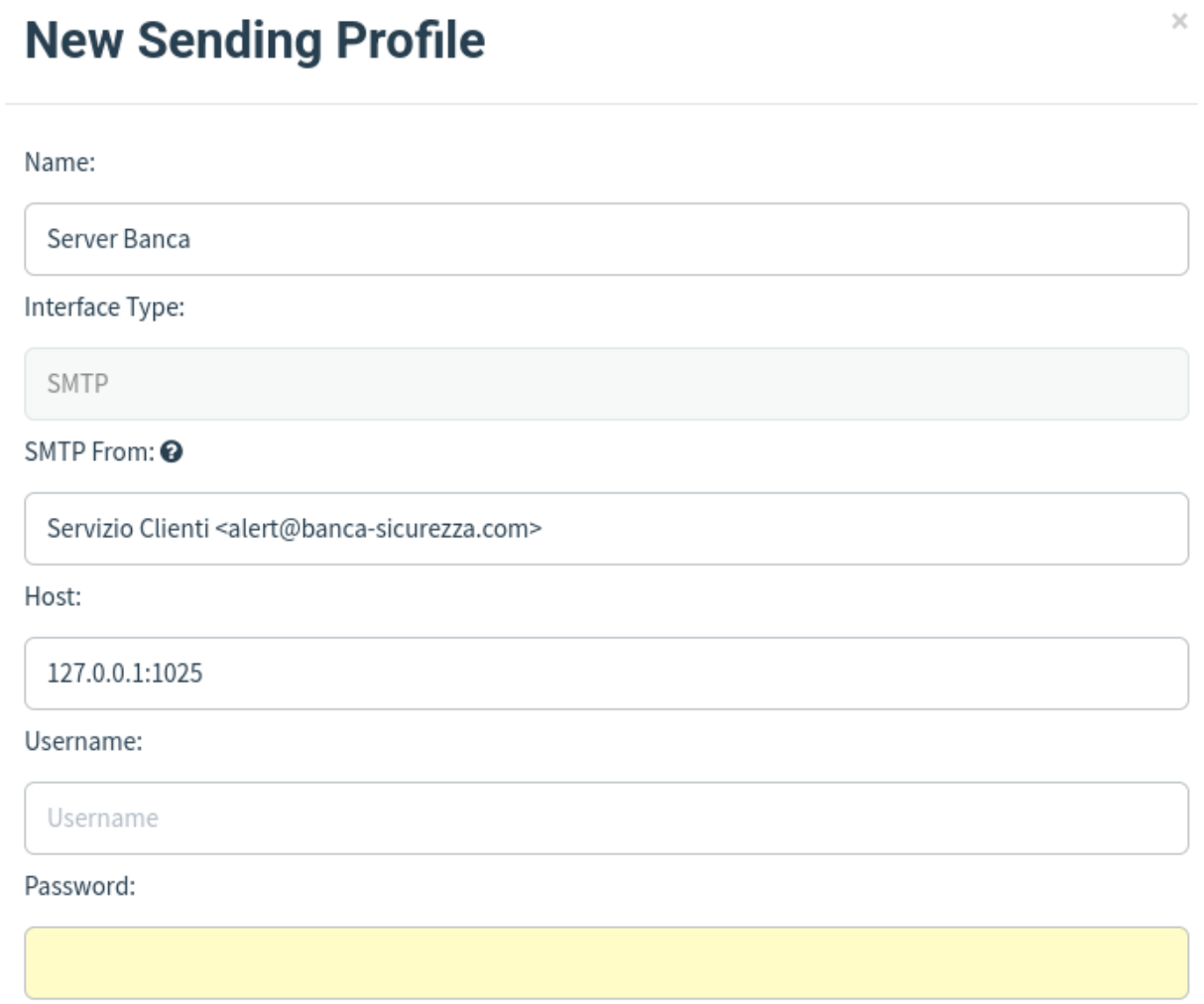
- Per la simulazione, ho inserito un utente fittizio associato ad un'e-mail temporanea generata con tempmail.
- Questa fase è critica per definire il perimetro dell'attacco e limitare l'invio esclusivamente ai soggetti autorizzati al test.



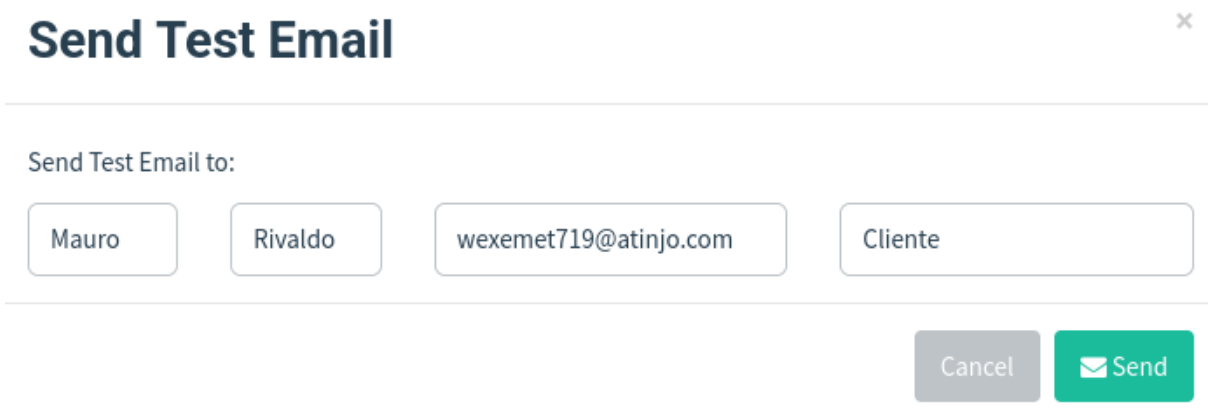
Configurazione del Mittente (Sending Profiles)

Per simulare l'invio delle email, ho configurato un profilo SMTP denominato `Server Banca`.

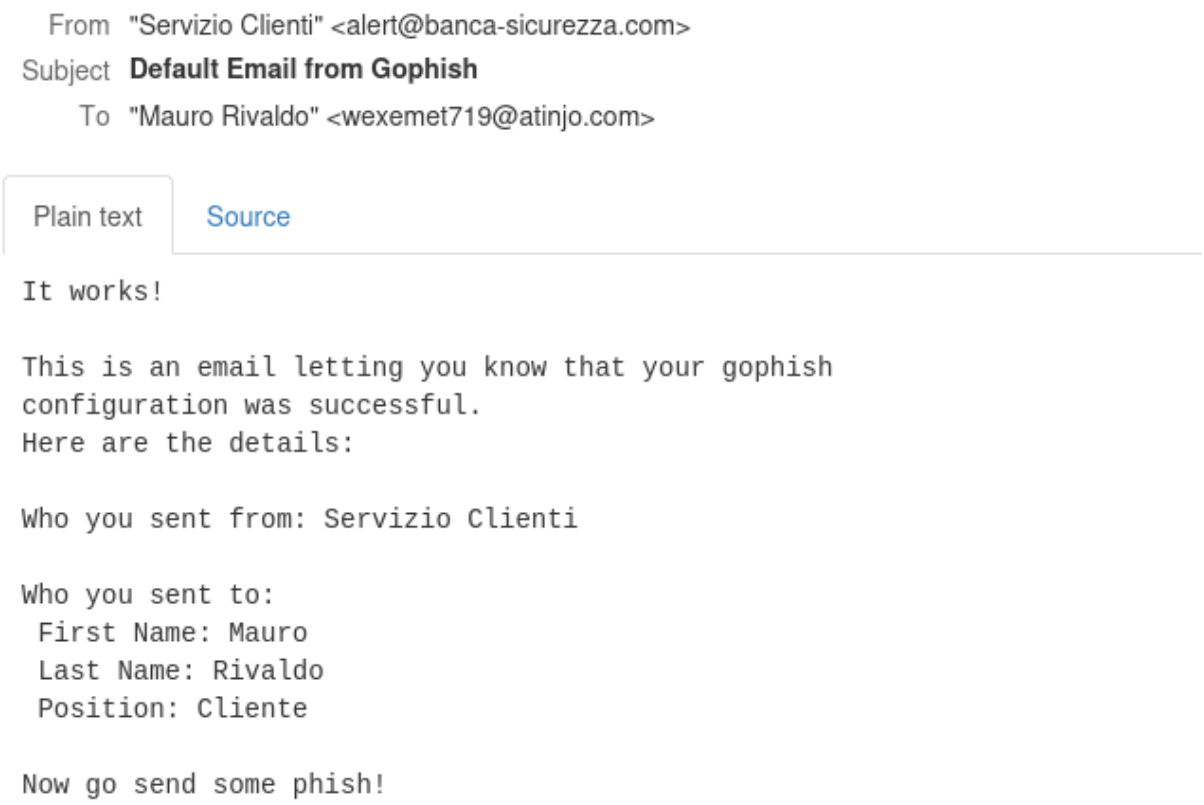
- Mittente (From): È stato impostato lo spoofing dell'identità tramite la stringa: `Servizio Clienti <alert@banca-sicurezza.com>`.
- Host: È stato utilizzato l'indirizzo di loopback `127.0.0.1:1025` per simulare l'invio in ambiente locale senza coinvolgere server di posta reali esterni.



In seguito per verificare che l'email venisse inviata ho fatto il Send Test Mail inserendo le infomazioni corrette.



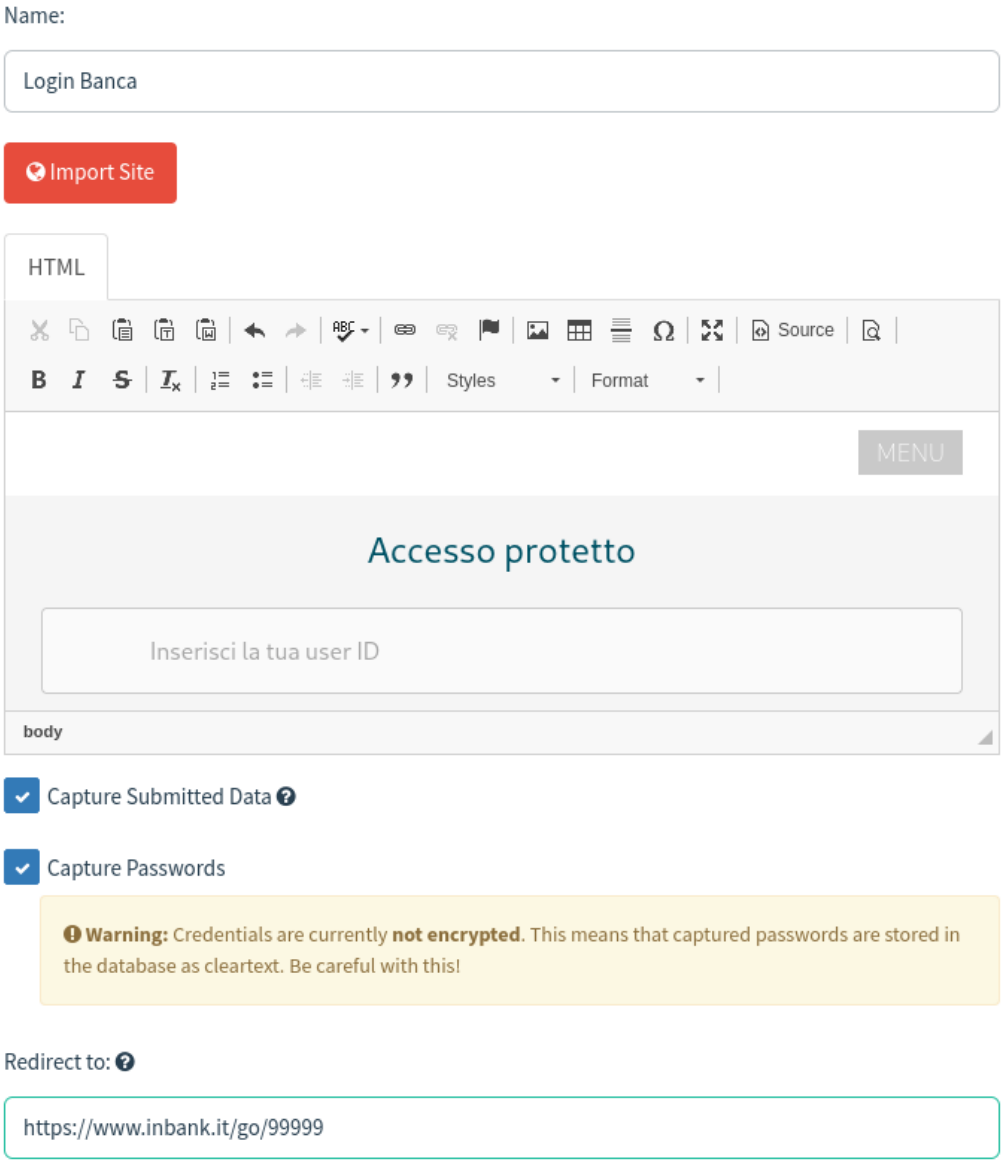
E ho verificato che fosse arrivata



Creazione della Trappola (Landing Pages)

La pagina fraudolenta è stata generata clonando una reale pagina di login bancaria.

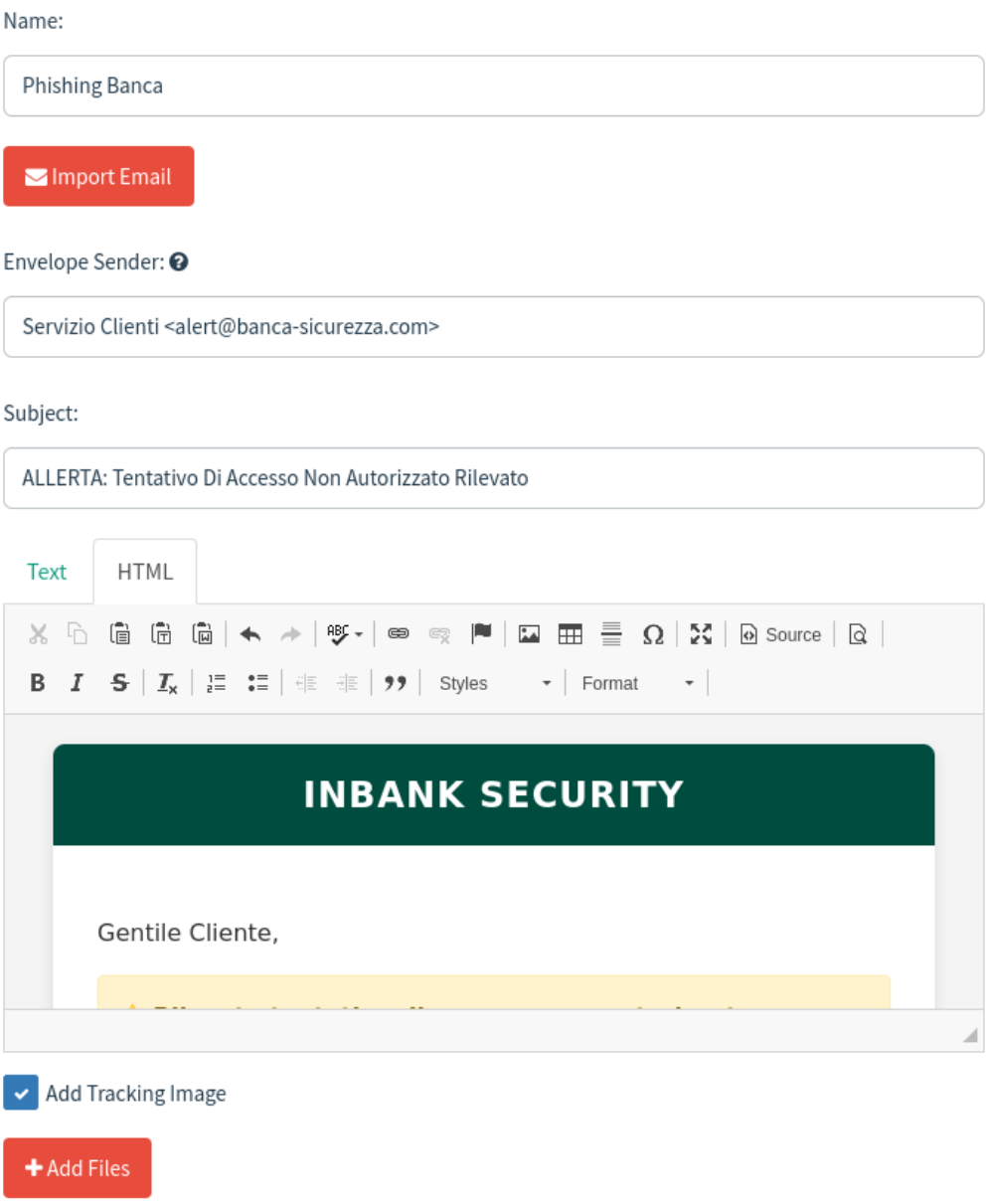
- Tecnica di Clonazione: Utilizzando la funzione Import Site di Gophish, è stato estratto il codice HTML/CSS da un URL pubblico.
- Credential Harvesting: Sono state abilitate le opzioni critiche "Capture Submitted Data" e "Capture Passwords". Questo istruisce Gophish a intercettare e salvare nel database locale i dati inviati tramite metodo POST dal form di login.
- Evasione: È stato configurato un Redirect post-login verso il sito ufficiale della banca. In questo modo, dopo aver sottratto le credenziali, la vittima viene reindirizzata sulla pagina vera, riducendo la percezione dell'attacco.



Sviluppo del Payload (Email Templates)

È stato redatto un template email HTML con oggetto "ALLERTA: Tentativo Di Accesso Non Autorizzato Rilevato".

- All'interno del corpo dell'email, il collegamento ipertestuale ("Clicca qui per accedere").



Orchestrazione e Lancio (Campaigns)

Tutti gli asset precedenti sono stati aggregati in una nuova campagna (`test 1`).

- URL Config: Ho impostato l'indirizzo `http://127.0.0.1` (o l'IP della macchina Kali) come base per i link di phishing.
- L'avvio della campagna ha attivato il listener sulla porta 80, rendendo la landing page accessibile alle vittime simulate.

New Campaign

Name:

test 1

Email Template:

Phishing Banca

Landing Page:

Login Banca

URL:

http://127.0.0.1

Launch Date

January 9th 2026, 8:09 am

Send Emails By (Optional)

Sending Profile:

Server Banca

Send Test Email

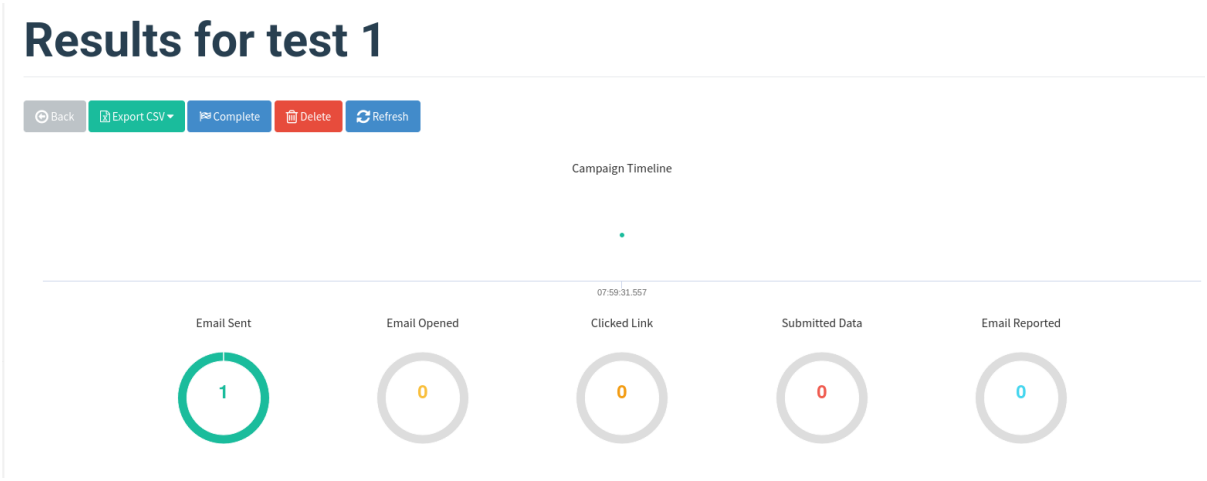
Groups:

× test

Close

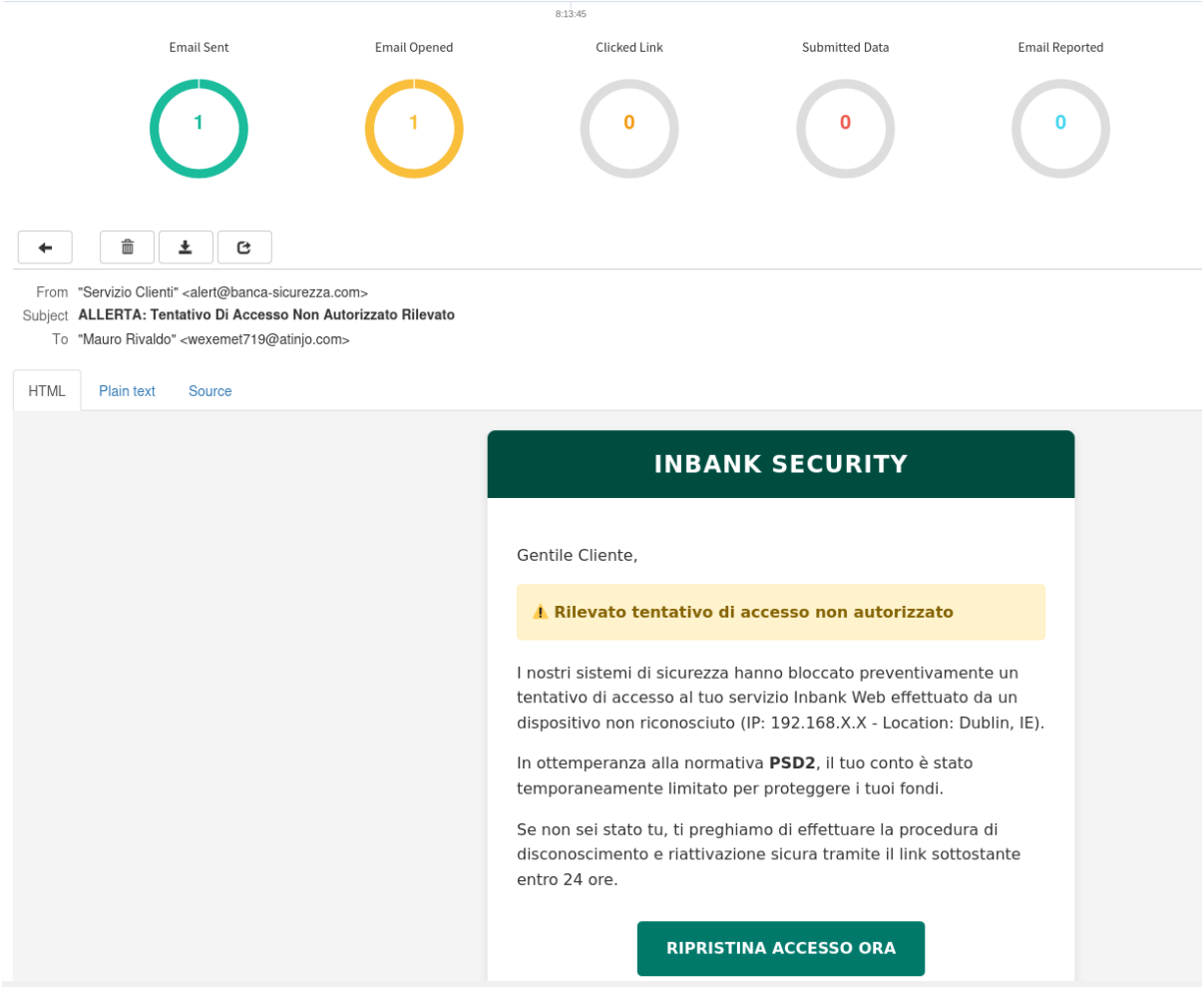
Launch Campaign

Esecuzione e Analisi dei Risultati (Dashboard)



Durante la fase di esecuzione, ho simulato il comportamento della vittima accedendo al link generato.

1. La vittima (io) visualizza una copia perfetta della pagina di login bancaria.



2. All'inserimento di Username e Password fittizi, il sistema ha reindirizzato l'utente al sito legittimo.
3. Sulla Dashboard di Gophish, lo stato dell'utente è passato a "Submitted Data". Espandendo i dettagli della riga (Row Details), è stato possibile visualizzare in chiaro le credenziali appena inserite (`Username: Pippo`, `Password: SuperSegreta`).

Analisi Difensiva e Red Flags

Nonostante l'efficacia tecnica un'analisi attenta dell'email avrebbe potuto sventare l'attacco identificando i seguenti indicatori di compromissione:

1. Passando il cursore sul link, l'URL di destinazione (indirizzo IP o dominio non ufficiale) non corrispondeva al dominio della banca.
2. Un controllo degli header avrebbe rivelato che il dominio `banca-sicurezza.com` non possiede record SPF/DKIM validi per l'IP di invio.
3. La richiesta di agire "immediatamente" è una tecnica classica di pressione psicologica non utilizzata nei canali ufficiali per comunicazioni di blocco conto.

Conclusioni

L'esercizio ha dimostrato come la combinazione di **Kali Linux** e **Gophish** permetta di automatizzare l'intero ciclo di vita di un attacco di phishing, dalla creazione dello scenario all'esfiltrazione dei dati.

Per mitigare tali rischi, le organizzazioni non possono affidarsi solo ai filtri antispam, ma devono implementare:

- **MFA (Multi-Factor Authentication):** Anche se l'attaccante ottiene la password (come dimostrato nel report), l'accesso viene negato senza il secondo fattore.
- **Formazione Continua:** Simulazioni periodiche come quella appena svolta sono quasi obbligatorie per "patchare" il fattore umano.