Progetto Finale: Phishing - Creazione di un Allegato Word Malevolo

Autori: Rocco Carpi, Riccardo Rossi

Questo repository contiene il progetto finale del corso di **Cybersecurity e Data Protection**, tenuto dalla **Prof.ssa Federica Paci** presso l'**Università di Verona**.

Panoramica del Progetto

L'obiettivo di questo progetto è dimostrare un attacco di phishing simulato tramite la creazione di un documento Word malevolo. Il documento include una **macro VBA**, progettata per collegarsi a Dropbox, scaricare un'immagine contenente l'indirizzo IP del server C2 e, una volta ottenuto l'IP, connettersi a esso per scaricare un **information stealer**. Quest'ultimo estrae file contenenti informazioni sensibili dal computer della vittima e li invia al server dell'attaccante per la consultazione.

Fasi dell'Attacco

1. Creazione dell'Immagine

La prima fase del progetto consiste nella codifica dell'indirizzo IP del server C2 all'interno di un'immagine utilizzando tecniche di steganografia.

Processo di Codifica

Abbiamo utilizzato il modulo Python `stegano` per nascondere l'indirizzo IP nell'immagine. Lo script `stegano_image.py` si occupa di:

1. Caricare un'immagine base (ad esempio, un file PNG). In questo caso, è stata utilizzata l'immagine di un gattino per renderla apparentemente innocua.
2. Nascondere l'indirizzo IP del server C2 utilizzando il metodo **LSB** (Least Significant Bit).
3. Salvare l'immagine modificata come `encoded_image.png`.

Questo approccio garantisce che l'indirizzo IP sia impercettibile visivamente, rendendo l'immagine apparentemente innocua.

Processo di Decodifica (per Debug)

Per estrarre l'indirizzo IP dall'immagine, lo stesso script utilizza il metodo `lsb.reveal` del modulo `stegano`, che legge i bit modificati per ricostruire il messaggio nascosto.

Vantaggi

- La steganografia con LSB è semplice da implementare e difficilmente rilevabile senza strumenti specifici.

---

2. Macro VBA

La seconda fase prevede la creazione di una macro VBA che automatizza il recupero dell'immagine, la decodifica dell'indirizzo IP, il download dello stealer e la sua esecuzione.

Macro Prima dell'Offuscamento

Il codice VBA è suddiviso nelle seguenti fasi:

1. **Scaricamento dell'immagine:** La macro utilizza `MSXML2.XMLHTTP` per effettuare una richiesta HTTP al link Dropbox contenente l'immagine codificata.
2. **Decodifica IP:** La macro richiama uno script Python che estrae l'indirizzo IP dal file immagine.
3. **Download dello Stealer:** L'indirizzo IP del server C2 viene utilizzato per scaricare un eseguibile (`stealer.exe`) dalla macchina dell'attaccante.
4. **Esecuzione dello Stealer:** La macro avvia lo stealer utilizzando il comando `Shell`, eseguendolo in modalità nascosta.

Tecniche di Offuscamento

Per rendere il codice della macro meno leggibile e più difficile da analizzare, sono state adottate le seguenti tecniche:

- **Rinominazione delle variabili e delle funzioni** con nomi casuali e privi di significato per nascondere il loro scopo.
- **Inserimento di istruzioni inutili e commenti fuorvianti** per rendere più complessa l'analisi statica del codice.
- **Frammentazione delle stringhe e ricostruzione dinamica** per evitare il rilevamento tramite scansione automatica.
- **Utilizzo di chiamate indirette a funzioni** per ostacolare la comprensione del flusso di esecuzione.

L'offuscamento del codice rende più difficile l'analisi da parte di software antivirus e ricercatori di sicurezza.

---

3. Infrastruttura d'Attacco

Per simulare l'intero processo di attacco, sono state configurate due macchine virtuali.

3.1 Macchina dell'Attaccante

- **Sistema Operativo:** Ubuntu 22 (Virtualizzato con Parallels Desktop)
- **Strumenti Utilizzati:**
  - Python 3.9 per lo script di steganografia.
  - Flask per implementare il server C2.
- **Configurazione del Server C2:**
  Il server ascolta sulla porta 5000 e gestisce richieste HTTP POST per ricevere i file rubati. I file vengono salvati nella directory `Version_3.0/uploads`. Inoltre, il server ospita il file `stealer.exe`, che la macro scarica e avvia.

3.2 Macchina Target

- **Sistema Operativo:** Windows 11 (Virtualizzato con Parallels Desktop)
- **Configurazione:**
  - Installazione di Microsoft Office per l'esecuzione della macro.
  - Modifica delle policy di sicurezza per consentire l'esecuzione delle macro (per scopi dimostrativi).

Descrizione dei File

c2_server_2.0.py

- Configura un server C2 utilizzando Flask.
- Accetta richieste POST per ricevere e salvare file rubati.
- Ospita il malware scaricato dalla macro.

information_stealer_2.0.py

- Scansiona una directory specificata alla ricerca di file sensibili.
- Invia i file rubati al server C2 tramite richieste HTTP POST.
- Gestisce eventuali errori di rete con tentativi di ritrasmissione.

stegano_image.py

- Codifica l'indirizzo IP del server C2 in un'immagine utilizzando LSB.
- Decodifica l'indirizzo IP per consentire alla macro di recuperarlo.

macro_vba.bas

- Scarica un'immagine da una URL specificata.
- Decodifica l'indirizzo IP del server C2.
- Scarica ed esegue lo stealer di informazioni.

Dimostrazione

Durante l'esame orale, la dimostrazione seguirà questi passaggi:

1. Avvio del server sulla macchina Ubuntu.
2. Apertura del documento Word malevolo sulla macchina Windows (target).
3. Esecuzione automatica della macro che:
    - Scarica l'immagine.
    - Decodifica l'indirizzo IP del server C2.
    - Scarica e avvia lo stealer.
4. Verifica della trasmissione dei file sensibili al server C2.

Conclusioni

Il progetto dimostra come un attacco basato su documenti Word malevoli possa essere strutturato utilizzando tecniche di steganografia, offuscamento e automazione. La configurazione è stata realizzata rispettando i requisiti richiesti ed è pronta per la dimostrazione pratica.

> ***Nota:*** *Questo progetto è stato sviluppato esclusivamente per scopi accademici ed educativi. L'uso non autorizzato di queste tecniche è illegale e punibile dalla legge.*