Progetto 2: Creazione di un malware basato su un server C2, capace di scaricare ed eseguire un malware reale

Francesco Cecconello VR457796

17 febbraio 2023

1 Introduzione

Fra i vari progetti proposti durante il corso di quest'anno, quello riguardante la creazione di un malware ha attirato fin da subito la mia attenzione, poiché ho trovato particolarmente stimolante mettermi dalla parte dell'attaccante, provando ad approfittare delle debolezze di un sistema operativo diffisussimo come Windows10. Nonostante le condizioni dell'ambiente virtuale non corrispondano esattamente alla realtà, è stato sicuramente istruttivo capire come possano lavorare a grandi linee i malware più pericolosi presenti in rete.

2 Cyber Kill Chain

Il progetto si pone a metà della Cyber Kill Chain, in corrispondenza della fase di Installation, durante la quale il malware si annida nel PC e guadagna la propria persistenza sul sistema modificandone file e registri; in seguito, dopo aver aperto una reverse shell che permetta di passare alla fase C2, consente all'attaccante di passare all'ultimo step, ovvero alla fase Actions&Objectives, durante la quale il malware potrà prendere il sopravvento sulla macchina della vittima.

3 Strumenti

Per portare a termine il lavoro sono stati utilizzati i seguenti strumenti.

3.1 VirtualBox

La versione di VirtualBox utilizzata, nonché la più recente disponibile, è la 7.0.6. Tutti i file *.iso* e i file riguardanti le macchine virtuali sono stati posti sullo stesso hard disk della macchina reale, poiché sia in ambiente Linux che in ambiente Windows si sono verificate alcune problematiche salvandoli su un hard disk esterno.

3.2 Macchina Virtuale KaliLinux

L'immagine di KaliLinux utilizzata per creare la macchina server Linux è quella scaricata dal sito ufficiale, ovvero KaliLinux 2022.4 (pagina per il download).
L'impostazione di rete di default è stata modificata, impostandola su "Scheda con bridge"; in questo modo non solo le due macchine potranno vedersi a vicenda, ma sarà possibile anche collegarsi ad internet per scaricare i file necessari. Sarebbe stato possibile creare una rete fittizia fra le due VM per isolare il sistema ma, poiché effettivamente non viene eseguito alcun malware reale (anche se sarebbe teoricamente possibile), non si corre il rischio che un eventuale virus possa attaccare la macchina host o espandersi sulla rete.

3.3 Macchina Virtuale Windows10

L'immagine di Windows utilizzata per creare la macchina Windows non è la versione originale, ma un'alternativa presente qui; si tratta di una versione molto leggera di Windows10 Home, adatta quindi all'esecuzione su una macchina virtuale, nella quale non è presente Windows Defender. Si è optato per questa soluzione sia per la ridotta quantità di memoria assegnabile ad ogni VM, sia per l'assoluta mancanza del sistema di sicurezza, vantaggio non da poco per quanto riguarda il download e l'esecuzione di file senza l'intromissione dell'antivirus di Windows.

3.4 Dropbox

Gli unici due servizi di hosting gratuiti trovati che fornissero hotlink per i file caricati sono stati Dropbox e GitHub. Dato che il periodo di disponibilità dei file doveva essere limitato, si è scelto di usare Dropbox, iscrivendosi con un'e-mail temporanea e caricando i file eseguibili *helper.exe* e *payload.exe* sul folder principale.

3.5 Moduli Python

Per creare e offuscare i file eseguibili è stato necessario scaricare, nell'ordine, il modulo *PyInstaller* (`pip install pyinstaller`) e il modulo Pyarmor (`pip install pyarmor`), mentre per crittografare la conversazione fra client e server C2 si è utilizzato *Pycryptodome* (`pip install pycryptodome`). La versione di Python utilizzata è la 3.10.9.

3.5.1 Pyarmor e PyInstaller

A partire dal sorgente *file.py*, utilizzando Pyarmor è possibile non solo renderlo un eseguibile standalone, ma anche offuscarne il contenuto per renderlo innocuo agli occhi degli antivirus.
Per creare un file eseguibile in una determinata architettura e in un certo sistema operativo, è necessario generarlo in un ambiente compatibile; in questo caso, tutti gli eseguibili sono stati creati sulla macchina virtuale Windows per ottenere la massima compatibilità.
Aprendo il prompt di Windows e spostandosi nella directory contenente il file Python da convertire in *.exe*, è possibile generare il corrispondente esegui-

bile eseguendo il comando `pyarmor pack -e " -noconsole -onefile" file.py`, in cui

- `pack` indica che non basta offuscarne il contenuto, ma a partire dal file va generato anche l'eseguibile
- `" -noconsole"` indica che durante l'esecuzione del file non va mostrata alcuna shell
- `" -onefile"` indica che il file eseguibile deve essere standalone, ovvero deve comprendere tutti i moduli importati dal sorgente

3.5.2 Pycryptodome

Fra le tante possibilità, il modulo Pycryptodome fornisce anche la crittografia a chiave simmetrica tramite AES, dove la chiave è presente (seppur offuscata in seguito con pyarmor) sui file *server.py* e *client.py*. Nel mondo reale sarebbe stato meglio utilizzare un protocollo di crittografia asimmetrica, poiché scrivere una chiave crittografica su un file non è mai una buona idea; in questo caso, però, si è optato per questa soluzione in modo da semplificare le operazioni.

4 Scelte implementative

Nonostante KaliLinux mettesse a disposizione i vari strumenti visti durante il laboratorio del corso, si è proceduto alla creazione riga per riga dei file del client e del server e del contenuto del payload. A grandi linee, il malware sviluppato si compone di due macro passaggi:

- Sulla macchina Linux, la quale impersona il server C2, viene lanciato lo script *server.py*, il quale apre una connessione in attesa dell'avvio del file *client.exe* sulla macchina della vittima. Una volta stabilita la connessione, avvia lo script *exploit_server.py* in attesa della creazione della connessione alla reverse shell sulla macchina della vittima.
- Sulla macchina Windows, ovvero il PC della vittima, si esegue il file *client.exe*, il quale svolge tre compiti:
  1. Crea la cartella *helper_folder* e scarica al suo interno il file *helper.exe*
  2. Crea la cartella *payload_folder* e scarica al suo interno il file *payload.exe*
  3. Modifica il registro *HKCU\Software\Microsoft\Windows\CurrentVersion\Run*, in modo da consentire l'esecuzione automatica del file ad ogni accesso dell'utente
  4. Avvia il payload e crea una reverse shell invisibile, dialogando con il server C2.

4.1 SERVER

Il file è presente sulla macchina Linux ed è utile all'attaccante per capire se il file *client.exe* sia stato eseguito sulla macchina della vittima. Essendo in costante ascolto alla porta 5555, una volta ricevuta la richiesta di connessione da parte di *client.exe* capisce che *payload.exe* e *helper.exe* sono stati scaricati sulla macchina

infetta e quindi può lanciare il vero server C2, ovvero *exploit_server*, in ascolto alla porta 4444.

4.2 CLIENT

È il file che effettivamente, una volta eseguito, apre le porte all'attaccante. Si dà per scontato che sia stato scaricato in qualche modo dall'utente e che quest'ultimo si fidi della sua genuinità al punto da lanciarlo. In realtà, anche se a prima vista il file non fa nulla, poiché non apre alcuna finestra e in generale non mostra segni di attività, nel frattempo sta spargendo i file legati al malware nelle cartelle sul disco C: , in particolare verifica che i folder *helper_folder*, *payload_folder* ed i file in essi contenuti siano presenti, altrimenti genera le cartelle e scarica i file di utility al loro interno; inoltre, scrive nella chiave di registro *HKCU\Software\Microsoft\Windows\CurrentVersion\Run* il valore il valore *C:\Users\vittima\Desktop\helper\folder\helper.exe*, in modo da abilitarne l'esecuzione automatica ad ogni riavvio. In seguito, una volta effettuati gli accertamenti, viene lanciato il payload, ovvero uno script che si connette al server C2 alla porta 5555 e attende i comandi da eseguire via shell, nel caso in cui l'attaccante voglia compiere alcune azioni sulla macchina della vittima prima di avviare il malware vero e proprio.

4.3 PAYLOAD

Presente sulla macchina della vittima; apre silenziosamente una connessione alla porta 5555 del server C2 e rimane in attesa dei comandi da eseguire; lo scambio di messaggi con il server è completamente crittografato con AES. Viene avviato automaticamente dal file *helper.exe* ad ogni accesso della vittima al proprio utente.

4.4 HELPER

Ad ogni riavvio controlla che il file di payolad e il rispettivo folder siano ancora presenti sul PC della vittima; in caso contrario, provvede prima alla creazione della cartella e, in seguito, al download e all'esecuzione di *payload.exe*.

4.5 EXPLOIT_SERVER

È il vero e proprio server C2, ovvero lo script preposto a ricevere i comandi dell'attaccante e ad inoltrarli alla shell sul PC della vittima, apparendo come una shell di Windows. Se l'attaccante invia il comando "run malware", questo viene prima scaricato e poi avviato sulla macchina infetta, danneggiandone il sistema.

4.6 MALWARE

Simula il comportamento di WannaCry, crittografando i file presenti sul desktop e aprendo una finestra pop-up, in cui viene chiesto un riscatto per ripristinarli, come nel più classico dei ransomware.

5 Attacco

Per portare a termine l'attacco subito basta seguire i seguenti step:

1. Eseguire il file *server.py* sulla macchina Linux.
2. Eseguire il file *client.py* sulla macchina Windows e attendere il completamento delle operazioni.
3. Utilizzare la reverse shell creata sulla macchina Linux per inviare comandi. Per chiudere la connessione inviare il comando "exit", mentre per scaricare e avviare il malware inviare "run malware".
4. Verificare che i file vengono crittografati dal malware e decrittografati inviando il falso pagamento di 1$.

Se invece si vuole aspettare il riavvio del PC per verificare la persistenza del malware, è possibile rimuovere dalla macchina Windows sia il file *client.exe* (ormai inutile), sia la cartella *payload_folder* con il relativo contenuto. Grazie a *helper.exe*, infatti, il payload verrà scaricato ed eseguito ad ogni riavvio della macchina.

6 Risultati

Tutti i file sono stati sottoposti ad un controllo su Virus Total: mediamente, solo 10 antivirus sui 71 utilizzati hanno rilevato un comportamento malevolo da parte degli eseguibili offuscati. In particolare, il controllo effettuato su una macchina Windows provvista di Windows Defender e Norton Antivirus ha rilevato solo a runtime la presenza dei file malevoli, bloccandone l'esecuzione.