Keylogger per Windows

Progetto di Fondamenti di Sicurezza e Privacy, A.A. 2021/2022

Simone Ferrari VR479912
Università degli Studi di Verona, Verona, Italia
simone.ferrari_03@studenti.univr.it

***Abstract* — Il keylogger rappresenta una famosa tipologia di malware nato attorno agli anni '70. Permette di ascoltare e raccogliere tutte le pressioni dei tasti digitate sulla tastiera del computer su cui tale programma è eseguito. Nel corso di questo progetto, quindi, si andrà a sviluppare un Keylogger per Windows con il linguaggio di programmazione Python. Si analizzerà l'intero processo di sviluppo e di messa in esecuzione sulla macchina vittima, seguendo la struttura della Cyber Kill Chain. Verranno quindi dettagliate le 7 fasi che la compongono focalizzandosi sul progetto in questione.**

Keywords — Cyber Kill Chain, Keylogger for Windows, Python, Socket, AES Encryption, Malware, Social Engineering Toolkit, Kali Linux.

I. Introduzione

Nel 2013, nonostante i sistemi informatici ormai si fossero ben affermati nella vita quotidiana, i servizi segreti russi utilizzavano ancora, per la stesura di documenti altamente sensibili e classificati, le macchine da scrivere. Ciò fu strettamente legato a motivi di sicurezza, in quanto la scrittura su macchine digitali non era considerata realmente sicura. Infatti, i primi keylogger, ovvero strumenti software o hardware il cui scopo è il logging dei caratteri digitati, vennero alla luce attorno al 1970; negli anni successivi furono poi numerose le segnalazioni di malware di tale tipo. Il primo keylogger noto risale al periodo della guerra fredda e permise all'intelligence sovietica di spiare i diplomatici americani. I russi, infatti, sfruttarono un bug presente all'interno delle macchine da scrivere IBM Selectric, di uso americano, ottenendo quindi informazioni classificate. Tali macchine erano caratterizzate da una testina di scrittura che, in fase di digitazione di un tasto, veniva ruotata in una determinata direzione. Questa era unica per ogni carattere, e il movimento necessitava di un'energia magnetica adeguata e precisa. In modo relativamente semplice, quindi, i sovietici riuscirono a convertire tali campi magnetici in segnali elettrici, i quali poi venivano trasmessi e catturati, permettendo quindi la ricostruzione dei testi americani.

Come anticipato, un keylogger può essere di tipo hardware o software. In entrambi i casi, l'obiettivo primario consiste nel catturare e registrare l'intero flusso di tasti digitati su di una tastiera. Essi, poi, vengono solitamente loggati e inviati ad un server remoto. Nel caso di keylogger software, solitamente, si tratta di un piccolo file eseguibile che entra in esecuzione confondendosi tra i vari processi attualmente in esecuzione all'interno di una macchina. Durante la sua esecuzione rimane in ascolto, attendendo che vengano digitati i tasti della tastiera. All'attivazione dell'evento, il keylogger può comportarsi in diversi modi: alcuni registrano un certo numero di caratteri in un file e, raggiunta una determinata quota, inviano l'intero contenuto al server dell'attaccante. Altri, invece, creano da subito una connessione con il server e, carattere per carattere, inviano il log direttamente al server.

In questo paper, si andrà ad analizzare la procedura di creazione di un keylogger per Windows e quali sono le componenti principali per renderlo realmente funzionale. Verrà analizzato il processo Cyber Kill Chain, analizzando le 7 fasi che lo compongono e descrivendo i vari passaggi e i vari requisiti necessari al compimento dell'obiettivo.

Per la parte tecnica, si userà il linguaggio di programmazione Python, e si avrà bisogno di due macchine Windows (una per simulare l'attaccante e una per la vittima) e di una macchina Kali Linux per l'utilizzo di alcuni strumenti di Social Engineering. Per semplicità si userà l'indirizzo di localhost per simulare la connessione client-server e creeremo un server single thread, limitando quindi ad 1 il numero di client massimi connessi contemporaneamente.

II. Reconnaissance

La prima fase della Cyber Kill Chain è la Reconnaissance, ovvero la ricognizione. L'obiettivo è il raccoglimento di informazioni riguardo la vittima e l'individuazione di vulnerabilità. Solitamente, questa fase è di notevole importanza, in quanto permette di ottenere informazioni cruciali per la buona riuscita dell'attacco. Tra gli elementi utili che si possono ricavare vi è l'indirizzo IP della vittima, l'architettura del sistema della macchina vittima, meccanismi di autenticazione, VPN, protocolli, nomi di dominio e molto altro.

Il progetto in questione mira alla creazione di un Keylogger per il sistema operativo Windows, pertanto l'unica informazione che si dovrebbe raccogliere su di una specifica vittima è il sistema operativo installato sulla macchina obiettivo. Tale azione, invece, non è indispensabile nei casi in cui si preveda la diffusione del malware senza mirare ad una specifica vittima, in quanto, semplicemente, nel momento in cui si entra a contatto con il malware, quest'ultimo verrà installato sulla macchina solamente se il sistema operativo è Windows, altrimenti non avrà effetto. Se la macchina risulta compatibile, quindi, il software malevolo si installerà, prenderà persistenza e inizierà lo scambio di informazioni con il server.

Nel seguito si assume di voler colpire una specifica vittima, la quale opera sul sistema operativo Windows.

III. Weaponization

La seconda fase della Cyber Kill Chain è la Weaponization, in italiano armamento. In questa fase, viene individuato o creato l'effettivo malware sulla base delle informazioni raccolte al passo precedente.

Di seguito quindi si descrivono le procedure necessarie per la creazione del programma malevolo e del programma server, ovvero il processo che si occupa di ricevere le informazioni provenienti dalla macchina vittima. Viene inoltre descritto come allegare tale malware ad un file Word: quest'ultimo, quando aperto in una macchina Windows, eseguirà uno script che provocherà l'installazione furtiva del software dannoso.

A. Server (file server.py)

Si descrivono innanzitutto i requisiti del programma server.

Il processo server, una volta avviato, dovrà aprire un socket e mettersi in attesa di connessione da parte di un client. Per semplicità di studio, il programma deve accettare una singola connessione alla volta, ma permette la connessione da client distinti. Una volta che il client si sarà connesso, dovrà iniziare a ricevere il flusso di caratteri inviati dalla macchina infetta. Per ogni carattere ricevuto, quindi, dovrà occuparsi di decifrarlo (in quanto verrà utilizzata una connessione crittografata) e dovrà scrivere su un file di log. Quando il client si disconnetterà, ad esempio a causa dello spegnimento della macchina infetta, il server dovrà chiudere la connessione e rimettersi in attesa sul socket: in questo modo sarà pronto a ricevere nuovamente il flusso quando il sistema vittima verrà riacceso. Per comodità di lettura, sia alla connessione che alla disconnessione del client, il server inserisce una nota all'interno del file di log, così da agevolare poi la lettura da parte dell'attaccante. I file di log vengono creati all'interno della directory di lavoro del programma server e deve essere creato un file di log per ogni client che si connette al server: in questo modo, i log relativi a diversi client resteranno separati. Inoltre, per comodità, i file di log contengono solamente i log della giornata corrente. Al cambio della data, quindi, viene creato un nuovo file.

Si analizza quindi il codice sorgente (riferimento al file server.py allegato)

Righe [1-5]: import dei moduli utilizzati. Per maggiori dettagli, si vedano i commenti.

Righe [8-11]: definizione della funzione per decrittografare il carattere proveniente al server. Per maggiori dettagli, si faccia riferimento al paragrafo 7 (Command and Control).

Righe [14-15]: avvio del ciclo while che mantiene il programma server in esecuzione. Utilizza una variabile booleana per poter valutare quando il socket è chiuso e quindi iterare il ciclo di apertura e ascolto.

Righe [16-23]: definizione del socket e binding di esso sull'indirizzo IP locale e porta numero 9090. Il programma si mette quindi in attesa di connessione da parte del client sul socket appena aperto.

Righe [25-27]: avvenuta connessione da parte del client. Viene quindi salvato l'IP della macchina connessa su una variabile.

Righe [29-31]: Apertura/Creazione del file di log così nominato: "DATE_giorno.mese.anno-IP_clientIP".

Righe [35]: apertura ciclo che resta in esecuzione finché il client è connesso al socket.

Righe [36-38]: Ricezione del carattere crittografato dal client, decrittazione del carattere e sistemazione del formato.

Righe [40-50]: Se il carattere ricevuto è diverso dal carattere vuoto (che per diverse ragioni potrebbe arrivare dal client), allora esso viene inserito all'interno del file di log. Altrimenti, dopo il 100esimo carattere vuoto consecutivo, si assume sia avvenuto un errore di connessione, e pertanto viene chiuso il socket e riavviata la procedura. Notare che 100 caratteri vuoti, in caso di errore, arrivano in un tempo pressoché istantaneo.

Righe [52-56]: Catch di una possibile eccezione proveniente dal metodo .recv() che viene sollevata nel caso in cui il socket sia chiuso da parte del client. Si riavvia quindi la procedura.

B. Client (file client.py)

Di seguito i requisiti del programma client (programma malevolo).

Il programma client dovrà innanzitutto essere eseguito sulla macchina vittima, pertanto dovrà essere in formato di file eseguibile (.exe). Esso, una volta avviato, dovrà innanzitutto prendere persistenza all'interno della macchina della vittima, e quindi dovrà inserire, in una directory di raro accesso, una copia di sé stesso e dovrà aggiungere tale path all'interno dei registri di sistema Windows, in particolare nel registro RUN. In questo modo, anche successivamente al riavvio della macchina, il sistema operativo provvederà a rimettere in esecuzione il programma malevolo. Il processo inoltre dovrà connettersi al socket aperto dal server, e dovrà avviare un thread listener dei tasti digitati sulla tastiera. Ad ogni evento, quindi, dovrà crittografare il carattere e inviarlo al server mediante il socket. Infine, il programma dovrà essere resistente a possibili disconnessioni da parte del server, e, nel caso non riuscisse a stabilire una connessione, dovrà ritentare ogni 15 secondi.

Si analizza quindi il codice sorgente (riferimento al file client.py allegato)

Righe [1-9]: Import dei moduli utilizzati. Per maggiori dettagli, si vedano i commenti.

Righe [12-13]: Recupero dell'username attualmente collegato e calcolo di un path per l'installazione del file malevolo. Si noti che la cartella AppData risulta "nascosta" di default nel sistema, pertanto l'individuazione del file malevolo da parte della vittima risulta molto difficile. Inoltre, il nome del file eseguibile simula un servizio di sicurezza, pertanto non dovrebbe destare sospetti agli occhi di una vittima poco esperta.

Righe [14-17]: Esegue la copia del file .exe nella directory specificata. Questo avviene solo al primo avvio del programma; le volte successive solleverà un'eccezione in quanto il file sarà già contenuto in quella esatta cartella.

Righe [19-24]: Il programma prende persistenza sulla macchina, modificando i registri di sistema. Si veda il paragrafo 6 (Installation) per maggiori dettagli.

Righe [27-30]: Funzione che si occupa di crittografare il carattere da inviare. Per maggiori dettagli, si faccia riferimento al paragrafo 7 (Command and Control).

Righe [33-34]: Ciclo while che mantiene in esecuzione il programma fino ad esplicita terminazione.

Righe [35-39]: Definizione del socket a cui collegarsi.

Righe [41-45]: Tentativo di connessione al socket aperto dal server. Se non va a buon fine, viene sollevata un'eccezione.

Righe [48-50]: Funzione che si occupa di inviare il carattere digitato al server dopo averlo accuratamente crittografato.

Righe [54-60]: Creo il thread listener che si mette in ascolto della tastiera e, ad ogni tasto premuto, esegue la callback definita alle righe precedenti, inviando così il carattere al processo server. In caso di problemi con l'invio del carattere, viene sollevata un'eccezione e il processo si rimette in attesa di connessione con il server.

Righe [62-64]: Se la connessione con il server fallisce, viene sollevata un'eccezione. Il programma quindi attende 15 secondi e riavvia l'intero processo di connessione.

C. *Creazione file exe e documento malevolo*

Per far sì che il programma malevolo appena creato possa essere eseguito su qualsiasi macchina Windows, è innanzitutto necessario rendere eseguibile il file Python. In particolare, si deve creare un file .exe che svincoli l'esecuzione dello script dalle librerie e dalle varie dipendenze, creando quindi un file autosufficiente. A tale scopo, quindi, si utilizza pyinstaller, un software che permette la creazione di file .exe da file .py. Per la creazione di questo file .exe si utilizzi il seguente comando:

```
pyinstaller .\client.py --onefile --noconsole
 -i='.\icon.ico' --name "SecurityService"
```

Una volta creato il file eseguibile, è necessario creare il documento Word .docm che, mediante l'attivazione di alcune Macro, possa eseguire l'installazione del file .exe appena creato. Per fare ciò, è necessario l'uso di uno strumento contenuto di default all'interno di una macchina Kali Linux. In particolare, le macro del pacchetto Office sono scritte in VBA, e pertanto si deve convertire il file .exe in un formato compatibile per la sua esecuzione.

Scaricata ed avviata la macchina Kali Linux, si importi sul Desktop (per comodità) il file .exe generato con pyinstaller. Dopodiché, si apra una shell e, sfruttando il tool exe2vba.rb, si va a creare un file .vba a partire dal file .exe:

```
usr/share/metasploit-
framework/tools/exploit/exe2vba.rb
Desktop/SecurityService.exe Desktop/SecurityService.vba
```

Si ottiene così un file SecurityService.vba. Questo file è suddiviso in due parti: la prima, riguarda la macro di esecuzione, la seconda, invece, contiene il payload del codice .exe.

Tornando sulla macchina Windows, quindi, si crei un nuovo documento Word, e si incolli all'interno tutto il contenuto della seconda parte del documento .vba appena creato. Una volta incollato il tutto, lo si imposti a font 1px (dimensione dei caratteri) e di colore bianco: in questo modo, il file sembrerà vuoto. Una volta fatto ciò, è necessario cliccare su "visualizza" e poi su "macro". Si scriva quindi il nome che si vuole dare alla macro e si clicchi su "crea". Si incolli nell'editor aperto tutto il codice contenuto nella prima parte del file .vba. Una volta fatto, si salvi la macro, si chiuda la finestra, si salvi il file Word con estensione .docm e si chiuda anche quest'ultimo. Arrivati a questo punto, si è a disposizione di un file Word con macro malevolo. Alla sua apertura, verrà richiesta l'attivazione delle macro. Se l'utente conferma, la macro verrà avviata e lo script entrerà in esecuzione, installando il keylogger e rendendolo persistente sulla macchina vittima.

IV. Delivery

La fase di Delivery, o consegna, è il terzo passaggio della Cyber Kill Chain. Si occupa di selezionare il canale di trasmissione per il software malevolo. I mezzi principali sono le Email, i supporti fisici come le chiavette USB e i siti web compromessi. Nel nostro esempio, sfrutteremo l'invio di una finta mail a cui allegata vi sarà il file Word infetto creato al passo precedente. Ancora una volta, si fa uso di uno strumento presente all'interno della macchina Kali Linux: il Social Engineering Toolkit. Mediante questo strumento di ingegneria sociale, è possibile attuare diverse tecniche di persuasione della persona, a scopo di infettare la vittima. Si proceda quindi avviando tale tool con i permessi root e si selezioni l'opzione per eseguire phishing e spearphishing. Successivamente, si seguano le istruzioni per la creazione di una mail con file allegato, e si alleghi il file creato nella fase precedente. In quanto i principali servizi di mail (ad esempio gmail, outlook, …) bloccano l'invio di mail mediante questi tools, non è stato eseguito un vero e proprio invio della mail, pertanto si assume che la vittima abbia ricevuto tale allegato via email.

La sfida quindi consiste nel creare tale mail in modo da invogliare la vittima a scaricare il file allegato e ad aprirlo all'interno del proprio pc. Una volta aperto, l'utente dovrà anche confermare l'attivazione delle macro, consentendo quindi l'esecuzione dello script. Si potrebbe, ad esempio, creare una mail di phishing appositamente creata per simulare il Ministero della Salute. Infatti, nel periodo particolare che si sta vivendo, le persone risultano particolarmente suscettibili riguardo il tema della salute, soprattutto se strettamente legato alla pandemia di Covid-19. Si potrebbe quindi invogliare la possibile vittima nello scaricare il file allegato, ad esempio per recuperare una nuova forma di Green Pass dovuta alle nuove norme emanate dal Governo; si potrebbe richiedere l'attivazione delle macro ad esempio per effettuare l'autenticazione con Codice Fiscale per la visualizzazione del documento personale. Si potrebbe inoltre inserire una finta scadenza per il recupero di tale documento, in tal modo l'utente sarà indotto ad eseguire le azioni richieste rapidamente, senza porsi troppe domande.

V. Exploitation

L'esecuzione del malware avviene qui, nella quarta fase della Cyber Kill Chain: l'exploitation. Il programma malevolo entra quindi in esecuzione ed inizia a svolgere le azioni per il quale è stato programmato.

Una volta ricevuto il documento infetto, se l'utente aprirà il file e acconsentirà l'uso delle macro, verrà eseguita una successione di attività il cui scopo sarà quello di eseguire il keylogger sul pc vittima. In particolare, verranno eseguite le macro inserite nel documento word, che si occuperanno di leggere il payload precedentemente inserito all'intero del file. Questa azione provocherà successivamente l'esecuzione del programma malevolo, il quale creerà ed avvierà un processo per il programma SecurityService.exe. Il programma, una volta connesso al server, creerà un secondo thread: il listener per l'ascolto dei tasti digitati. A questo punto, il malware è attivo sulla macchina vittima e può iniziare a raccogliere le pressioni dei tasti, inviandole quindi al server; il documento word non è più di utilità, pertanto anche se tale file dovesse essere eliminato non vi sarebbero conseguenze: il programma malevolo ormai si è duplicato in una cartella remota del pc e, come si vedrà nel capitolo successivo, sarà avviato ad ogni accensione della macchina, ad insaputa della vittima.

Si noti che, a scopo didattico, non è stata utilizzata alcuna tecnica di offuscamento del codice sorgente, pertanto i principali sistemi di protezione come Windows Defender, ma soprattutto i software antivirus più sofisticati, rileveranno con molta probabilità il malware, impedendone l'esecuzione.

VI. Installation

Come anticipato, la fase di Installazione, ovvero il quinto passaggio della Cyber Kill Chain, ha l'obiettivo di conquistare persistenza sulla macchina vittima. In altri termini, il software malevolo deve poter persistere sulla macchina, anche nei casi in cui essa venga spenta e/o riavviata. In questo modo, una volta andata a buon fine la fase di Exploitation, non sarà più

necessario eseguire alcuna azione di infezione, in quanto il malware risulterà attivo e sarà autosufficiente.

In ambiente Windows si possono sfruttare i registri di sistema a tale scopo. Cosa sono i registri di sistema? Essi rappresentano un database in cui sono mantenute e gestite le varie opzioni e impostazioni del sistema operativo. Sono inoltre salvate varie informazioni e opzioni delle applicazioni installate sulla macchina e dei drivers. Questa base di dati è quindi dinamica: ad ogni azione sul sistema, come ad esempio l'installazione di un programma, tale database viene interrogato per estrarre e/o inserire diverse informazioni. Il registro si basa sul concetto di chiave e valore: le chiavi, in linea generale, rappresentano le variabili di configurazione, mentre i valori rappresentano l'informazione da memorizzare. Tra la moltitudine di tali chiavi, vi compare anche RUN, la quale consente l'esecuzione di un programma (specificato nel valore) ogni volta che l'utente accede al proprio computer. Sfruttando questa funzionalità, quindi, è possibile inserire una nuova entry per la chiave RUN, specificando come valore il path del programma malevolo.

Facendo quindi riferimento al codice sorgente del file client.py allegato, si analizzino le righe n° 19-24. La variabile reghive contiene la specifica della tipologia di chiave: nel caso in oggetto, si utilizzano le chiavi associate all'utente attivo (evitando quindi la necessità di permessi da amministratore). Dopodiché, la variabile regpath contiene il percorso in cui è presente la chiave RUN all'interno del registro di sistema. In tutti i sistemi Windows, essa è collocata in

```
SOFTWARE\Microsoft\Windows\CurrentVersion\Run.
```

Successivamente, tramite alcuni metodi importati dal modulo winreg, si crea una connessione alla base di dati, si apre in scrittura la chiave RUN e si inserisce come value il path del programma malevolo, che nell'esempio è previsto nel path

```
C:\Users\user\AppData\Local\Microsoft\
Windows\Safety\SecurityService.exe
```

In questo percorso, "user" rappresenta il nome utente della persona collegata, mentre si fa notare che AppData è una cartella sempre presente ed impostata come "nascosta" di default dal sistema: ciò quindi aumenta la difficoltà nell'individuare il file eseguibile. Si noti che si è deciso di installare il software in tale path, ma qualsiasi altro percorso può essere utilizzato; l'uso di un path non comune semplicemente diminuisce la probabilità che la vittima possa individuare il programma malevolo.

Terminati questi passaggi, quindi, è stata inserita una nuova entry nella chiave RUN, e in tal modo il programma verrà avviato ogniqualvolta l'utente accederà al proprio sistema.

VII. Command and Control

Ora che il malware è stato avviato ed installato all'interno della macchina vittima, deve mettersi in comunicazione con il server dell'attaccante. Si analizza quindi la fase di Command and Control, la sesta della Cyber Kill Chain.

Lo scambio di informazioni con la macchina dell'attaccante è di fondamentale importanza, in quanto permette al creatore del malware di ricevere le informazioni d'interesse. Questo canale di comunicazione può essere creato in diversi modi, e può essere di tipo unidirezionale o bidirezionale: ad esempio, si può sfruttare un canale sul web client-server oppure un canale tramite mail. Inoltre, il canale può prevedere o meno una connessione crittografata, inviando e ricevendo quindi informazioni opportunamente crittografate mediante diversi algoritmi.

Nel codice in oggetto, l'obiettivo della fase Command and Control consiste nell'invio al server dei caratteri digitati sulla macchina vittima. Si è quindi creata una connessione crittografata mediante socket web diretta alla porta 9090 della macchina dell'attaccante. La connessione è di tipo bidirezionale in quanto i socket Python supportano questa modalità di default, ma si sfrutta solo la direzione con sorgente nella macchina vittima e destinazione nella macchina del programmatore. La connessione inoltre è crittografata con l'algoritmo a chiave simmetrica AES: si è scelto questo algoritmo in quanto molto semplice e performante. La chiave è simmetrica, ovvero sia il programma malevolo che il server la condividono ed essa è hardcoded all'interno del programma: non vi è quindi necessità di instaurare un meccanismo di scambio delle chiavi come accadrebbe invece nel caso si usasse un algoritmo a chiave pubblica, come l'RSA, o se si dovesse generare periodicamente la chiave da inviare poi al client.

Di seguito si analizza più nel dettaglio l'implementazione della fase Command and Control, facendo riferimento al programma client e al programma server.

A. *Client (file client.py)*

Il programma malevolo necessita di collegarsi al socket aperto dal server, di crittografare i caratteri da inviare e di inviarli quindi al destinatario. Deve inoltre poter gestire eventuali eccezioni generate in caso di problemi di connessione con il server.

Si faccia quindi riferimento alle righe n° 27-30: questa funzione prende come parametro un oggetto di tipo Stringa e ritorna in output un byte array crittografato. La procedura di crittografia AES è contenuta nel modulo Crypto, importato a riga n° 5 e viene invocata passando come argomento la chiave e il testo da crittografare.

Le righe n° 35-45 si occupano invece di creare e gestire la connessione con il processo server. Viene definito quindi il socket e poi si esegue un tentativo di connessione: se essa va a buon fine, il programma è correttamente collegato al server e procede che le azioni, altrimenti viene sollevata un'eccezione, mettendo in attesa per 15 secondi il processo per poi riprovare. Tale azione viene poi iterata fino a connessione eseguita con successo.

A riga n° 50 avviene l'effettivo invio del carattere: esso, però, viene prima passato come argomento alla funzione di crittografia descritta in precedenza, che ne ritornerà una versione crittografata.

B. *Server (file server.py)*

Il processo server, al contrario del client, si occupa dell'apertura del socket e dell'ascolto su di esso: all'arrivo di un messaggio, esso dovrà essere decrittografato per consentire poi l'uso corretto.

Alle righe n° 8-11, quindi, è definita la funzione per decrittografare l'input: ancora una volta viene utilizzata una procedura AES contenuta nel modulo Crypto, denominata decrypt. La funzione quindi ritorna il testo in chiaro,

corrispondente all'originale inviato dal client. Tale testo poi sarà scritto all'interno del file di log.

VIII. Actions on Objectives

L'ultima fase della Cyber Kill Chain è denominata Actions on Objectives, ovvero Azioni sugli Obiettivi, e consiste nel vero e proprio attacco nei confronti della vittima, portando a termine l'obiettivo inziale. Solitamente, si vuole raccogliere dati sensibili e classificati sull'utente coinvolto, ma a volte alcuni attacchi mirano invece a rendere tali informazioni indisponibili, chiedendo poi un riscatto alla vittima per lo sblocco: è il caso dei ransomware. Un'altra possibile tipologia di obiettivo è la modifica e l'alterazione dei dati per poter causare poi un effetto collaterale d'interesse.

Nel caso dei Keylogger, l'obiettivo primario è la raccolta dell'intero flusso di digitazione proveniente dalla macchina vittima. In tal modo, è possibile catturare email, password, numeri di carte di credito e qualsiasi altra informazione sensibile digitata sulla tastiera del computer infetto.

Il malware sviluppato e descritto in precedenza produce, giornalmente, dei file di testo al cui interno vengono loggati i tasti digitati sulla macchina della vittima. L'attaccante, quindi, può successivamente analizzare tali file, che per comodità sono catalogati per Client IP e per Data, entrando in possesso di tutte le informazioni classificati digitate manualmente dalla persona obiettivo; il tutto, a completa insaputa dell'utente.

IX. Risultati e Conclusioni

Nonostante il Keylogger sia una tipologia di malware ormai presente da molti anni, se non opportunamente individuato può tutt'ora compromettere la riservatezza di dati ed informazioni personali di una ipotetica vittima. Si è infatti visto come, con un programma relativamente semplice e compatto, si possa sviluppare un keylogger funzionante e che permetta ad un attaccante di spiare l'intera attività di scrittura svolta su una macchina infetta. I sistemi anti virus più attuali sono ad oggi molto resistenti a tali tecnologie, ma un'accurata attività di offuscamento del codice potrebbe eludere i controlli. Pertanto, è utile formare e mettere in guardia le persone circa queste tipologie di minacce. In particolare, ad oggi, il mezzo principale di delivery sono le Email di phishing. Esse, infatti, sfruttano l'ingenuità della vittima spingendole a cliccare su link malevoli e/o a scaricare file infetti. Nel caso in oggetto, si è inviata una mail di phishing contenente un documento compromesso, invitando poi l'utente ad aprilo e ad abilitare le macro. Una buona formazione della persona, probabilmente, lo farebbe dubitare nell'abilitare le macro sul file, o ancor prima lo fermerebbe nel cliccare sul pulsante di apertura dell'allegato. Ad oggi, non esistono strumenti e metodologie precise per individuare chiaramente una mail di phishing, ma ci sono diversi fattori che, se accuratamente analizzati, possono fornire chiari segnali di pericolo: ad esempio, nomi di dominio errati, link mismatch, errori grammaticali, richieste insolite di dati personali, richieste con scadenze vicine, e molti altri.

In conclusione, si descrive brevemente la procedura consigliata per l'esecuzione dei programmi sviluppati in modalità development, in maniera tale da poter apportare modifiche al codice sorgente e/o vedere in esecuzione il keylogger in modo più chiaro e completo.

Una volta scaricate, installate, configurate ed avviate due macchine virtuali Windows, installare in entrambe le macchine l'interprete Python. In entrambe le macchine, poi, aprire un terminale e recarsi nella cartella del progetto. Qui, creare un ambiente virtuale (vedere documentazione python) ed installare le varie dipendenze richieste. Una volta completati questi passaggi, su una macchina eseguire il comando python3 server.py mentre sull'altra eseguire python3 client.py. In questo modo, i due processi saranno attivi. Ora, recarsi sulla macchina virtuale vittima ed iniziare a digitare qualche carattere in qualsiasi finestra/file aperto nel sistema. Si noterà che nei due terminali aperti, client e server, verranno visualizzati i messaggi inviati e ricevuti. Nella macchina server, inoltre, si sarà creato il file contenente l'intero log. Si noti che, perché le due macchine possano comunicare, dovrà essere configurata una rete NAT tra di esse, e il traffico dovrà essere rediretto al'IP 127.0.0.1 e porta 9090.

Nel caso in cui la macchina server fosse un ambiente Windows non virtuale, invece, è sufficiente avviare una singola macchina virtuale simulante il sistema vittima. Una volta quindi aperto il server sulla macchina host, si deve avviare lo script python del client con collegamento al'IP 10.0.2.2 e porta 9090. In tal modo, il flusso sarà rediretto all'IP 127.0.0.1 porta 9090 della macchina host. Procedendo quindi con la digitazione di alcuni caratteri all'interno della macchina infetta, il server li riceverà e loggherà il tutto sia a console che su file.

X. References

[1] Socket, Low-level networking interface. Python 3.10.2 documentation. https://docs.python.org/3/library/socket.html

[2] AES — PyCryptodome 3.14.1 documentation. https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html

[3] Pynput Package Documentation — pynput 1.7.6 documentation. https://pynput.readthedocs.io/en/latest/

[4] Winreg Windows registry access — Python 3.10.2 documentation. https://docs.python.org/3/library/winreg.html

[5] Keylogger - Wikipedia. Wikipedia, l'enciclopedia libera. https://it.wikipedia.org/wiki/Keylogger

[6] Cyber Kill Chain – Materiale del corso Fondamenti di Sicurezza e Privacy tenuto dalla Prof.ssa Federica Maria Francesca Paci