Facoltà di Scienze e Tecnologie

Corso di Laurea Magistrale in Sicurezza informatica

Relazione progetto Advanced Computer Programming

Ransomware

Anno Accademico 2016–2017

Indice

1 Descrizione

Il progetto svolto consiste nella creazione di un ransomware.

Il ransomware è un malware che “tiene in ostaggio” dei sistemi operativi o dei dati in essi contenuti richiedendo una ricompensa:

- disabilitano un servizio di sistema essenziale o bloccano il sistema all’avvio;
- criptano alcuni dati dell’utente.

Le vittime devono solitamente inserire un codice ottenuto solo dopo aver pagato l’attaccante, o comprare un tool di remove che è comunque venduto dall’attaccante.

Figura 1: 5 FASI DI UN RANSOMWARE

1.1 Obiettivo

L’obiettivo di tale progetto è stato quello di creare un ransomware, a scopo didattico, che simula il comportamento reale del malware.

1.2 Sistema Operativo target

Il progetto è stato sviluppato per il sistema operativo Microsoft Windows come obiettivo dell'attacco, poichè il ransomware prende in considerazione la radice (ad esempio C:) del SO stesso.

Inoltre, attualmente, Microsoft Windows risulta essere il sistema operativo più diffuso, permettendo dunque al malware di raggiungere più dispositivi possibili.

Un altro aspetto considerato nella scelta del sistema operativo target è la possibile diffusione del malware, in quanto ad esempio uno scenario tipico in Windows è l'installazione di crack per ottenere gratuitamente dei servizi che sarebbero invece a pagamento.

Essendo stato sviluppato per Windows si sono utilizzate API che permettono di interagire a basso livello con il sistema.

1.3 Linguaggio di programmazione utilizzato

Per la realizzazione del progetto sono stati utilizzati due linguaggi di programmazione:

- per lo sviluppo del client è stato utilizzato il C++;
- per lo sviluppo del server è stato utilizzato il Java.

L'utilizzo del linguaggio C++ è dettato dal fatto che, essendo un linguaggio compilato, viene eseguito direttamente dal sistema operativo e non da un interprete come nel caso dei linguaggi interpretati, dato che tali interpreti talvolta introducono ulteriori livelli di sicurezza e potrebbero non risultare presenti su un eventuale macchina target, riducendo quindi, in modo considerevole, il numero di macchine infettabili.

Sono state utilizzate funzioni e classi fornite dalle seguenti librerie:

- C++ Standard Library: stdio.h, stdlib.h, conio.h, ctime, string, …
- Microsoft: windows.h, winable.h, winsock2.h, ws2tcpip.h, …

L'utilizzo delle librerie fornite da Microsoft per Windows ha permesso un integrazione a basso livello e completa con il sistema operativo, permettendo di sviluppare funzionalità che rendono il malware più completo e versatile.

Il server è stato sviluppato in Java, linguaggio di programmazione ad aggetti, specificatamente progettato per essere il più possibile indipendente dalla piattaforma di esecuzione.

L'IDE utilizzato per lo sviluppo di entrambe le parti è NetBeans versione 8.2, e la versione di JDK (Java Development Kit) che è stata utilizzata è la 1.8.

É stato inoltre utilizzato un DataBase per memorizzare delle informazioni utili. La piattaforma utilizzata è XAMPP, una piattaforma software gratuita costituita da Apache HTPP Server, il databaseMySQL e tutti gli strumenti necessari per utilizzare i linguaggi di programmazione PHP e Perl. Il nome è un acronimo dei programmi sopra citati: la X sta per cross-platform, la A sta per Apache HTTP Server, la M sta per Mysql , la P sta per PHP e l'ultima P sta per Perl. XAMPP è rilasciato sotto la GNU General Public License e, oltre ad essere gratuito, è caratterizzato da un approccio user friendly. Il DBMS che è stato utilizzato è MySQL. Uno dei programmi più popolari per amministrare i database MySQL e che è stato anche utilizzato è phpMyAdmin, utilizzabile facilmente tramite un qualsiasi browser. Il DataBase può essere modellato utilizzando il linguaggio SQL (Structured Query Language).

2 Database

Il Database è costituito da un'unica tabella.

Questa tabella contiene tutte le informazioni utili che riguardano le macchine infettate dal ransomware.

I campi della tabella sono:

- IdHost: chiave primaria della tabella di tipo varchar che serve a identificare univocamente all'interno della tabella le macchine infettate;
- KeyHost: variabile di tipo text che serve a indicare la chiave di cifratura con cui sono stati cifrati i file sulla macchina infettata identificata da IdHost;
- Payment: variabile di tipo boolean che serve a tener conto se il pagamento è stato effettuato o meno.

3 Funzionalità

Le funzionalità del ransomware verranno elencate e illustrate in base al nome dei metodi della classe, raggruppate per tipologia.

3.1 Client

3.1.1 Rete

3.1.1.1 RawRequest

Funzione che permette al malware di inviare richieste su una porta di un server, ritornando una stringa contenente la risposta.

Questa funzione è particolarmente versatile, dato che permette di inviare su una qualsiasi porta una quantità di dati, ad esempio una richiesta HTTP GET:"GET=HTTP=1:1nrnnnrnn".

I parametri della funzione RawRequest sono:

- serverAddress: nome Host o indirizzo IP a cui inviare i dati;
- port: porta a cui inviare i dati;
- request: stringa di dati da inviare;
- bufferLength: dimensione del buffer.

3.1.2 Sistema

3.1.2.1 SaveValueReg e LoadValueReg

Funzioni che permettono di salvare o recuperare delle informazioni su una chiave del registro di sistema.

Le funzioni si comportano in modo diverso a seconda se si hanno i permessi di amministratore per l'esecuzione del malware, riuscendo a salvare lo stato anche nel caso non si abbiano i permessi, quindi solamente per l'utente in cui è in esecuzione il malware.

I parametri della funzione SaveValueReg sono:

- path: percorso del registro di sistema in cui salvare i dati
- key: nome della chiave del registro di sistema
- value: valore da salvare

I parametri della funzione LoadValueReg sono:

- path: percorso del registro di sistema in cui recuperare i dati
- key: nome della chiave del registro di sistema

3.1.2.2 SaveValueToFile e LoadValueFromFile

Funzioni che permettono di salvare o recuperare delle informazioni su un file nel sistema.

I parametri della funzione SaveValueToFile sono:

- value: valore da salvare
- path: percorso dove salvare il file, di default è la cartella temporanea del sistema.

Il parametro della funzione LoadValueFromFile è path, ovvero il percorso dove recuperare il file, di default è la cartella temporanea del sistema.

3.1.2.3 SaveSession e LoadSession

Funzioni che permettono al malware di salvare o recuperare il suo stato(sessione), all'interno del sistema, in modo tale che al riavvio della macchina possa ripristinare precedenti configurazioni.

Le funzioni utilizzano un attributo della classe, di tipo map, con chiave e valore di tipo string, nel quale tutte le funzioni della classe possono inserire le proprie informazioni, questo per permettere alle singole funzioni di operare indipendentemente e separare i compiti, riguardo il salvataggio dei dati presenti nella variabile e l'inserimento di tali dati in essa.

Le informazioni vengono salvate nel registro di sistema, mediante la funzione SaveValue-Reg e nella cartella temporanea, mediante la funzione SaveValueTo-File, con la logica di una struttura dati di un mappa (chiave-valore) sotto una voce difficilmente individuabile all'occhio "non esperto" e il suo contenuto viene crittato dalle funzioni Encrypt e Decrypt.

3.1.2.4 Persistance

Funzione che si occupa dell'installazione permanente del malware all'interno del sistema operativo. Le fasi sono le seguenti:

1. Rinomina: rinomina il nome dell'eseguibile, in modo da non essere facilmente identificabile, quindi con il nome di un servizio di Windows, ad esempio "winresumer.exe";
2. Propagazione eseguibili: gli eseguibili rinominati, vengono dislocati, sia nelle cartelle di sistema, sia nelle cartelle temporanee.
3. Esecuzione all'avvio: si salva nel registro, mediante la funzione SaveValueReg, le voci riguardanti i file, facendo in modo che all'avvio vengano eseguiti i file del malware.

3.1.2.5 ShowWindow

Funzione che permette di rendere non visibile una finestra scelta nel sistema, pensata principalmente per la finestra del malware, limitando molto la possibilità di essere individuata da un utente.

Per motivi di versatilità si è reso possibile rendere non visibile una qualsiasi finestra, aumentando i possibili impieghi della funzione.

I parametri della funzione ShowWindow sono:

- la finestra che si desidera nascondere;
- un int che mostra per quanto tempo sarà visibile la finestra indicata precedentemente.

3.1.3 Scansione FileSystem

3.1.3.1 GetFileNamesInDirectory

Funzione che permette di ottenere il nome di tutti gli elementi presenti all'interno di una directory.

Il parametro della funzione GetFileNamesInDirectory è directory, ovvero il nome della directory che bisogna scansionare.

3.1.3.2 controllo

Funzione che permette di verificare se l'elemento scansionato sia un file o una directory.

Il parametro della funzione controllo è path, ovvero il nome del file o della directory da controllare.

3.1.3.3 ricerca

Funzione che permette di richiamare la funzione precedente (GetFileNamesInDirectory) per poter scansionare il contenuto delle varie directory ricorsivamente.

I parametri della funzione ricerca sono:

- path: il path che bisogna controllare per verificare che si tratti di un file o di una directory;
- psw: la password utilizzata per poi crittare o decrittare i file;
- esiste: un booleano che indica se bisogna effettuare una crittazione o una decrittazione di quel file.

3.1.4 Scansione File

3.1.4.1 leggiCritta e leggiDecritta

Funzioni che permettono di:

1. scansionare il contenuto di un file, letto in binario;
2. memorizzazione del contenuto;
3. crittazione del contenuto (nel caso di leggiCritta), decrittazione del contenuto (nel caso di leggiDecritta);
4. memorizzazione del contenuto crittato o decrittato in un nuovo file;
5. eliminazione del file originale.

I parametri delle funzioni leggiCritta e leggiDecritta sono:

- path: nome del file da crittare o decrittare;
- psw: chiave utile per crittare o decrittare il file.

3.1.5 Crittografia

3.1.5.1 Encrypt e Decrypt

Funzione che effettua la crittografia simmetrica(a chiave privata), eseguendo lo XOR logico di ogni carattere della stringa da crittografare ed un numero dato in input alla funzione. Per decrittare la stringa crittata, applica lo stesso procedimento alla stringa crittata ottenendo la stringa in chiaro.

L'algoritmo di crittazione e decrittazione coincidono per proprietà dell'operazione logica XOR.

I parametri della funzione EncryptDecrypt sono:

- input: stringa da crittare o decrittare;
- key: chiave privata.

3.1.6 Richieste al Server

3.1.6.1 createId

Funzione che effettua una richiesta al server per ottenere un identificatore univoco della macchina infettata.

Il parametro della funzione è una chiave di cifratura dei file generata randomicamente che verrà associata a quell'identificatore.

3.1.6.2 payment

Funzione che effettua una richiesta al server per comunicare che il pagamento è avvenuto.

Il parametro della funzione è l'identificatore della macchina.

3.1.6.3 checkPayment

Funzione che effettua una richiesta al server per controllare se il pagamento è avvenuto o meno.

Il parametro della funzione è l'identificatore.

3.1.6.4 deleteData

Funzione che effettua una richiesta al server per eliminare qualsiasi dato riguardante quella macchina.

Il parametro della funzione è l'identificatore.

3.1.7 Varie

3.1.7.1 SelfDelete

Funzione che consente al programma di rimuovere se stesso definitivamente se l'esecuzione è terminata sotto determinate condizioni.

3.1.7.2 checkUsersPath

Funzione che permette al programma di determinare quello che è il path dell'utente all'interno della macchina.

3.2 Server

3.2.1 MySQL

È una classe che semplifica la creazione e la gestione di una connessione al database per effettuare le query con cui ottenere dati o inserirne di nuovi. Infatti, in questa classe, troviamo dei metodi per la gestione degli statement e ResulstSet (Initialize, Close), e la possibilità di effettuare diversi tipi di query, semplificando quelli che possono essere gli stati sulla corretta transazione dei dati, e quindi sapere se i dati sono stati inseriti correttamente o meno.

3.2.2 Crypto

Classe che contiene le funzioni che effettuano la crittografia simmetrica(a chiave privata), eseguendo lo XOR logico di ogni carattere della stringa da crittografare ed un numero dato in input alla funzione. Per decrittare la stringa crittata, applica lo stesso procedimento alla stringa crittata ottenendo la stringa in chiaro.

L'algoritmo di crittazione e decrittazione coincidono per proprietà dell'operazione logica XOR.

I parametri della funzione EncryptDecrypt sono:

- input: stringa da crittare o decrittare;
- key: chiave privata.

3.2.3 RandomGenerator

Classe che permette di generare l'identificatore univoco da inviare al client.

3.2.4 Servlet

3.2.4.1 createIdKey

Presa una chiave di cifratura in ingresso inviata dal client e generato un identificatore reso univoco tramite un confronto con quelli già memorizzati nel DB, attraverso una query inserisce all'interno del DB queste informazioni associate tra loro.

Successivamente invia una risposta al client contente l'identificatore di quella macchina.

3.2.4.2 checkPayment

Preso un identificatore in ingresso inviato dal client, attraverso una query verifica se il campo payment del DB è settato a "1" o a "0", e lo invia come risposta al client.

3.2.4.3 payment

Preso un identificatore in ingresso inviato dal client, attraverso una query aggiorna il campo payment del DB da "0" a "1", inviando al client una risposta contente se l'aggiornamento è avvenuto correttamente o meno.

3.2.4.4 delete

Preso un identificatore in ingresso inviato dal client, attraverso una query elimina dal DB tutti i campi associati ad esso, inviando al client una risposta contente se l'eliminazione è avvenuta correttamente o meno.

4 Conclusioni e sviluppi futuri

In questa relazione si è illustrato il funzionamento di un ransomware. Sono stati spiegati i metodi del progetto e la simulazione di tali metodi utilizzati per effettuare attacchi. I possibili sviluppi futuri sono:
- implementazione di una schermata che blocchi l'utilizzo dell'intero sistema.
- implementazione di un algoritmo che gestisca il tempo disponibile per effettuare il pagamento.