I processi e la memoria RAM

Una guida al mondo dei processi e di come la memoria ram viene gestita

I processi sono la parte più importante nel mondo dei controlli e della forensica digitale. I processi sono le istanze di un programma (.exe per windows) in esecuzione. I processi vengono gestiti da vari servizi, quello più noto è il PCB (Process Control Block) oppure il TCB (Thread control block). Questi gestiscono rispettivamente processi e thread, hanno informazioni sulla loro memoria, il loro ultimo avvio, le loro risorse etc. Una applicazione come "task manager" ci indica tutti i processi che sono in esecuzione ma si limita ad indicare i processi e il loro impatto sull'hardware (percentuale di ram usata, cpu etc..). Per avere una overview più dettagliata possiamo usare programmi come Process Explorer dalla sysinternal suite etc.. anche se il più utilizzato in ambito controlli è proprio Process Hacker (ora System Informer). Una informazione da notare e che parlando di windows, stiamo interagendo con un sistema proprietario e non saremo mai in possesso del codice sorgente di tale sistema operativo, quindi le nostre capacità di indagine si limitano a questi programmi offerti da queste persone. Non si riesce ad ottenere raw data se non quello che fornisce la windows API, che a volte da problemi di permessi, ma questo lo vedremo parlando di vari processi più avanti. Quando un programma viene avviato, viene creato un processo, che contiene le seguenti informazioni (P.S. non sono tutte ma sono la maggior parte):

- Il file eseguibile da cui deriva il processo
- Il path di quel file
- Il processo parente
- La console parente
- Gli argomenti con cui è stato avviato
- La directory a cui sta accedendo
- I threads
- I moduli
- La memoria stessa del processo che gli viene allocata
- Il programma stesso viene salvato nella memoria ram
- Gli handles

Cerchiamo di spiegare ora queste varie informazioni riguardo al processo. Piccola premessa, tutte queste informazioni sono riguardo hai processi che sono in esecuzione, quando un processo viene terminato, le informazioni elencate li sopra non sono salvate, almeno non raw. Sistemi come il prefetch salvano alcune di quelle informazioni con obbiettivo di ottimizzazione ad un successivo

avvio del medesimo programma, ma le informazioni raw così non sono salvate da nessuna parte, anche perché la ram è appunto una memoria volatile, oltre che incredibilmente limitata rispetto alla dimensione dei programmi.

Processo parente

Il processo parente, è il processo che ha avviato quel programma. Se noi facciamo doppio click su un eseguibile di qualsiasi tipo, il processo parente di quel processo sarà “explorer.exe”. Una cosa molto importante è il fatto che processi particolari come i servizi, hanno sempre dei processi parente noti, e quindi un ipotetico processo che prova a fingersi un servizio di sistema è facile da notare quando esso non ha lo stesso processo parente del servizio emulato e/o di qualsiasi servizio comune. Il 90% dei servizi hanno come processo parente “services.exe”, oppure “svchost.exe”. Servizi come il “csrss.exe” e “wininit.exe” non hanno un processo noto come parente, perché essi sono avviati direttamente dal sistema operativo.

Argomenti di avvio

Gli argomenti con cui un processo è stato avviato, sono le opzioni diciamo, che si possono dare ad un programma per fare varie cose. Alcuni processi hanno lo stesso eseguibile, ma si distinguino per il fatto che sono stati avviati con diversi argomenti e quindi compiono task differenti.

I threads

Nei sistemi moderni, ogni processo ha uno o più thread, ed è un modo di suddividere il lavoro della CPU. Il processo di minecraft ha tanti thread, come alla fine ogni processo di un programma moderno.

I moduli

I file eseguibili non hanno spesso tutto il codice al loro interno. Facendo un esempio semplice, chiunque abbia provato a vedere la cartella di gioco di fortnite, ha visto che ci sono “FortniteWinShipping64.exe”, che è l’eseguibile responsabile per l’avvio di fortnite, ma pesa relativamente poco, una cosa come 200 e qualcosa megabytes. Non sembra normale considerando che il gioco pesa 30 gigabytes e più. Questo perché fa utilizzo delle DLL. Le DLL o meglio conosciute come dynamic link libraries, sono librerie che possono essere caricate da degli eseguibile, e contengono codice ordinario. La comodità è che sono dinamiche, quindi possono essere caricate da un programma più volte e a piacimento. Questo permette un ottimizzazione del programma molto buona, oltre che una suddivisione più accurata delle varie parti del codice. I moduli sono importanti spesso per i cheater. Un esempio noto può essere appunto minecraft. I cheat maker, sia che si tratta di injection o external client, non vanno a mirare il processo “javaw.exe” in se per se, bensì un modulo particolare, il “jvm.dll”. Questo modulo contiene oltre alle istruzioni della jvm creata, quindi il codice del gioco, anche tutta la memoria di esso. Quando un injection client si collega a minecraft, crea un collegamento in particolare con questo modulo, e tramite librerie come “JNI.h” viene a contatto con il codice arbitrario del gioco. Gli external client invece per modificare i valori specifici della memoria, vanno a cercare gli offsets che puntano appunto al modulo “jvm.dll”, perché è dove effettivamente quei valori si trovano. I moduli di un processo sono fondamentali per capire cosa fa.

Gli handles

Meglio conosciuti come risorse in italiano, gli handles sono ogni singolo file aperto da un determinato processo. Quando noi andiamo a fare il self destruct di una mod, istantaneamente noi

non possiamo eliminarla o spostarla, questo perché l'handle che il processo di minecraft ha su quel file è ancora attivo. Quindi da li viene poi il metodo bypass di unloaddare una mod, andando manualmente con process hacker a chiudere l'handle che la java virtual machine ha sulla mod, rendendola così spostabile ed eliminabile. Gli handle riguardano mod come tutte le risorse. Qualsiasi cosa aperta dal processo si trova negli handles.
Alcune di queste informazioni vengono salvate nel prefetch, come i nomi degli eseguibili, le risorse e le directory aperte, ma i prefetch sono manipolabili in vari modi, e di conseguenza possono risultare di poco aiuto a volte. Un artifatto importante nelle indagini / controlli, è andare a controllare la memoria di alcuni processi. All'interno ci potrebbero essere segnali dell'esecuzione avvenuta di determinati programmi che noi valutiamo poi come cheats. Ora facciamo una breve lista di processi / servizi.

"Explorer.exe"

Forse il più noto, l'explorer è la shell che ci permette di interagire graficamente con il sistema operativo, e ha tante informazioni importanti. Osservando come la memoria del processo si comporta, si è notato che ogni file visualizzato nell'explorer appare nella sua memoria, come altre cose tra cui il PCA Client, utile per vedere gli ultimi 10 processi eseguiti (metodo deprecato ma comunque utile da controllare)

"csrss.exe"

Il "client server runtime process" è forse l'artefatto più importante di tutti. Ogni eseguibile, quindi .exe .dll etc.. viene trovato nella memoria di questo processo. Sembrerebbè quindi il metodo finale, purtroppo per noi però la windows API, come accennato prima, può dare problemi di permessi quando si tratta di accedere alla memoria, e questo è uno di quegli esempi. Molto spesso neanche process hacker può accedere a tale memoria e i programmi come eseguibili per ss tool si possono scordare di avere l'accesso a questa parte di memoria. Inoltre possiamo provare che un programma è stato eseguito, ma non quando, e in questi casi nei controlli è importante capire se un cheat era in istanza oppure no. Apparte tutto comunque rimane un processo utilissimo per i nostri scopi, nonstante la sua difficoltà nell'accederci. Metodi in sviluppo ci permettono di dumpare tutta la memoria RAM e isolare quella di determinati processi. Chissà in un prossimo futuro potremmo automatizzare il controllo del csrss e approfondire la nostra conoscenza di questi processi low level.

"msmpeng.exe"

Questo è il processo del servizio anti-malware del windows defender. Un processo poco conosciuto e poco esplorato dai methods finder, ma pieno di potenzialità, perché facendo parte della suite di antivirus di windows, ha informazioni su tutti i programmi eseguiti e quindi possibili flag di cheats in particolare.

Conclusione

Anche se questi sono 3 processi e ce ne sono tantissimi altri che vengono controllati, ho scelto quelli base perché almeno secondo me lo studio dei processi serve fino ad un certo punto. Il mondo delle "stringhe" magiche di process hacker è un mondo superficiale e che si basa su tanti false flag e metodi volatili, che un giorno vanno e altri no. Quando si ha a che fare con la memoria RAM, bisogna realizzare che volatile non è un attributo messo li a caso. Le stringhe sono si utili magari nel loro piccolo, ma non sono niente in confronto allo studio accurato di artefatti più importanti e che ci danno un overview generale su cosa è successo nel computer nel tempo. Lo studio della memoria RAM in modo avanzato sta andando avanti e si stanno scoprendo nuove cose. È un mondo

complicato e difficile da tradurre in informazioni significative per noi, ma nonostante ciò è comunque un argomento fondamentale nelle indagini forensi che noi chiamiamo superficialmente controlli. Detto questo rimanete aggiornati e curiosi. Non vi limitate alle stringhe private e robe del genere. Cercate di capire come un computer funziona, per quindi poi ricavare informazioni 10 volte quelle che una banale stringa su process hacker ti può dare.