

**1.**

Immagine che contiene testo, Carattere, numero, schermata

Descrizione generata automaticamente

La funzione **FindResourceA()** è una funzione fornita dalle API di Windows.

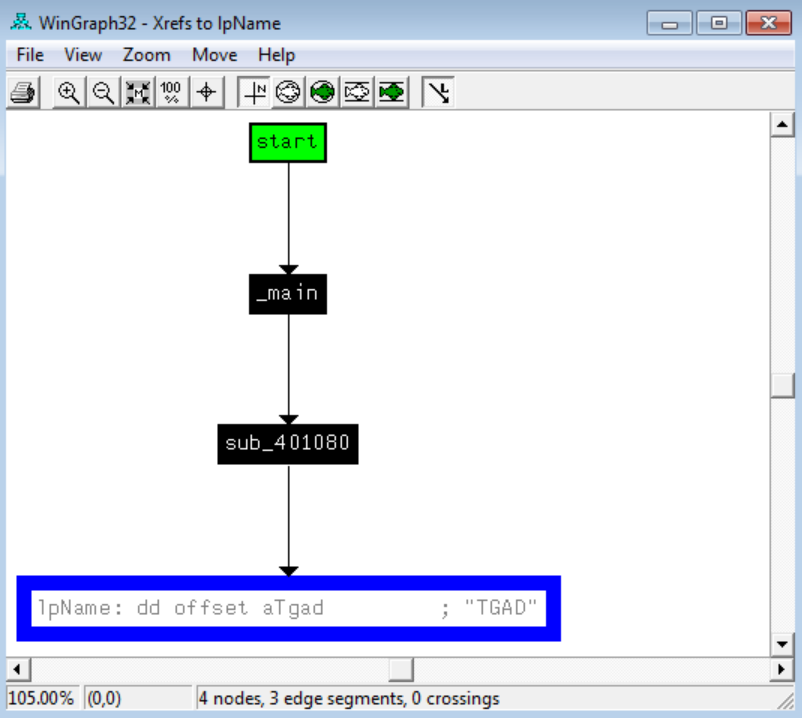
Immagine che contiene testo, Carattere, schermata, numero

Descrizione generata automaticamente

Il parametro ResourceName di nostro interesse è rappresentato dalla variabile **lpName**, un tipo stringa (LPCSTR).

Abbiamo due modi per ottenere il suo valore: attraverso l’analisi statica o dinamica.

Nel caso dell’analisi statica utilizziamo **IDA**; una volta trovata la variabile lpName nel codice facciamo clic con il tasto destro e clicchiamo su **Chart of xrefs to** (la funzione che mostra tutte le referenze a un determinato indirizzo o valore nel codice assembly; in pratica permette di vedere tutte le istruzioni o i luoghi nel codice in cui un particolare indirizzo o valore è utilizzato o referenziato).



Osservando che lpName è correlata alla variabile aTgad, ci spostiamo nella parte del codice che riguarda aTgad.

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

L'istruzione **lpName dd offset aTgad** indica che lpName è un puntatore all'offset della variabile aTgad; quindi lpName contiene l'offset della variabile aTgad e può essere utilizzato per accedere al valore di aTgad nel codice, poiché punta all'indirizzo di memoria dove si trova la variabile aTgad.

L'istruzione **aTgad db "TGAD", 0** indica la definizione di una variabile stringa di nome aTgad che contiene i caratteri "TGAD".

Pertanto deduciamo che il valore di lpName (ResourceName) è “TGAD”.

Nel caso dell’analisi dinamica lanciamo **OllyDbg** e impostiamo un breakpoint alla chiamata della funzione FindResourceA() e facciamo partire il **debugger**. Esso interromperà l’esecuzione del codice automaticamente una volta arrivati al breakpoint settato e da qui possiamo vedere i valori dei parametri passati alla funzione. Troviamo che il valore di lpName (ResourceName) è “TGAD”.

Immagine che contiene testo, ricevuta, schermata, Carattere

Descrizione generata automaticamente

**2.**

Il susseguirsi delle chiamate di funzione **FindResourceA**, **LoadResource**, **LockResource** e **SizeofResource** rappresenta il comportamento tipico di un malware di tipo **dropper**.

Un dropper è un programma malevolo che contiene al suo interno un malware. Nel momento in cui viene eseguito, un dropper inizia la sua esecuzione ed estrae il malware che contiene per salvarlo sul disco. Generalmente, il malware incluso nel dropper è contenuto nella sezione **.rss** dell’eseguibile, ovvero nella sezione risorse (talvolta anche identificata con .rsc).

I dropper hanno delle caratteristiche distintive piuttosto singolari; per estrarre il malware contenuto nella sezione delle risorse utilizzano delle API come: FindResourceA(), LoadResource(), LockResource() e SizeOfResource(), esattamente quelle contenute nel file analizzato. Queste API permettono di localizzare all’interno della sezione risorse il malware da estrarre, e successivamente da caricare in memoria per l’esecuzione o da salvare sul disco per esecuzione futura.

**3-4.**

Come rispondo a questa domanda? Si, effettivamente eravamo riusciti a capirlo utilizzando l’analisi statica basica, come si può dedurre leggendo il Report del giorno 1 con tutte le evidenze a supporto.

**5.**

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

Funzioni relative all’ottenimento della persistenza:

* RegSetValueExA()
* RegCreateKeyExA()

Funzioni relative al drop del primo eseguibile:

* LoadResource()
* FindResource()
* LockResource()
* SizeofResource()