# Breve storia di un'analisi malware

**Author :** Giuseppe Turano

**Date :** 14 Febbraio 2019

Un software che causa danni al computer o alla rete è considerato software maligno, detto anche **malware**. L'analisi di un malware consiste nello **studiare il suo comportamento**, con l'obiettivo di:

1. Identificare
2. Difendere
3. Eliminare

Durante questo tipo di analisi è bene tenere a mente che si sta analizzando un **software dannoso** e che, quindi, è necessario (se non fondamentale) prendere le dovute **precauzioni**.

Esistono **due metodi principali** per l'analisi del malware:

1. Analisi statica: studio del malware a riposo.
2. Analisi dinamica: studio del malware e del suo comportamento durante il suo ciclo di attività.

Queste due metodologie differiscono solo per lo stato (livello) di esecuzione del malware.

Come si può intuire, le due metodologie di analisi presentano vantaggi e svantaggi in virtù dello stato in cui intervengono.

## TIPI DI MALWARE

Esistono diverse tipologie di malware. Quelle di mio interesse - e che analizzeremo in questo articolo - sono raggruppabili in queste **categorie:**

1. Backdoor: questa tipologia di malware crea degli accessi nascosti in un PC che possono essere utilizzati da terzi al fine di controllare/monitorare il dispositivo infetto.
2. Botnet: simile alla precedente categoria, una botnet è una rete di dispositivi infetti che

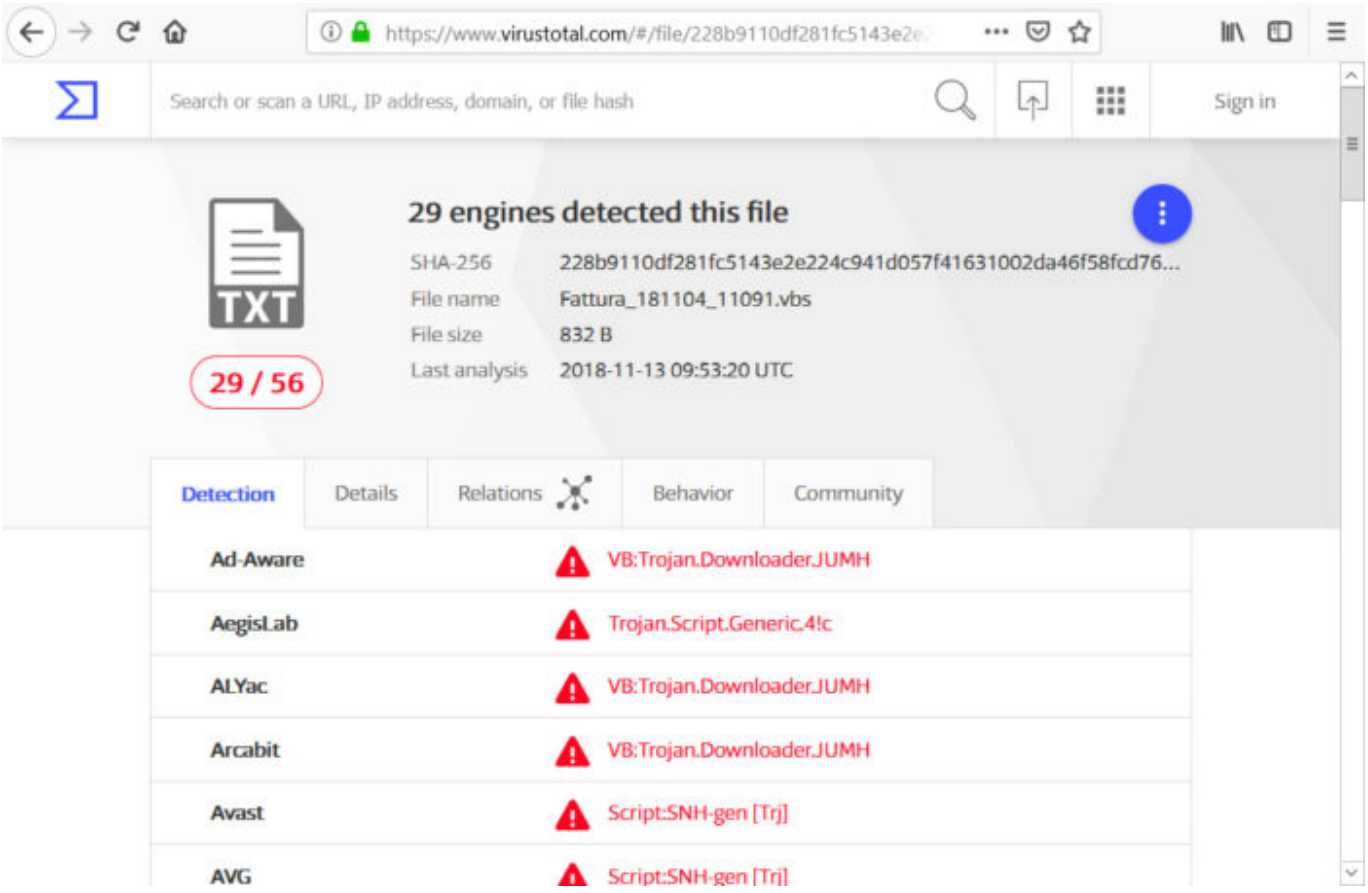
tramite una backdoor riceve istruzioni dall'esterno.
3. Spyware: malware utilizzati per raccogliere informazioni dal sistema su cui sono installati.
4. Ransomware: virus che cripta tutti i dati di un dispositivo.

A questo link un elenco dettagliato e completo di tutti i malware ad oggi noti.

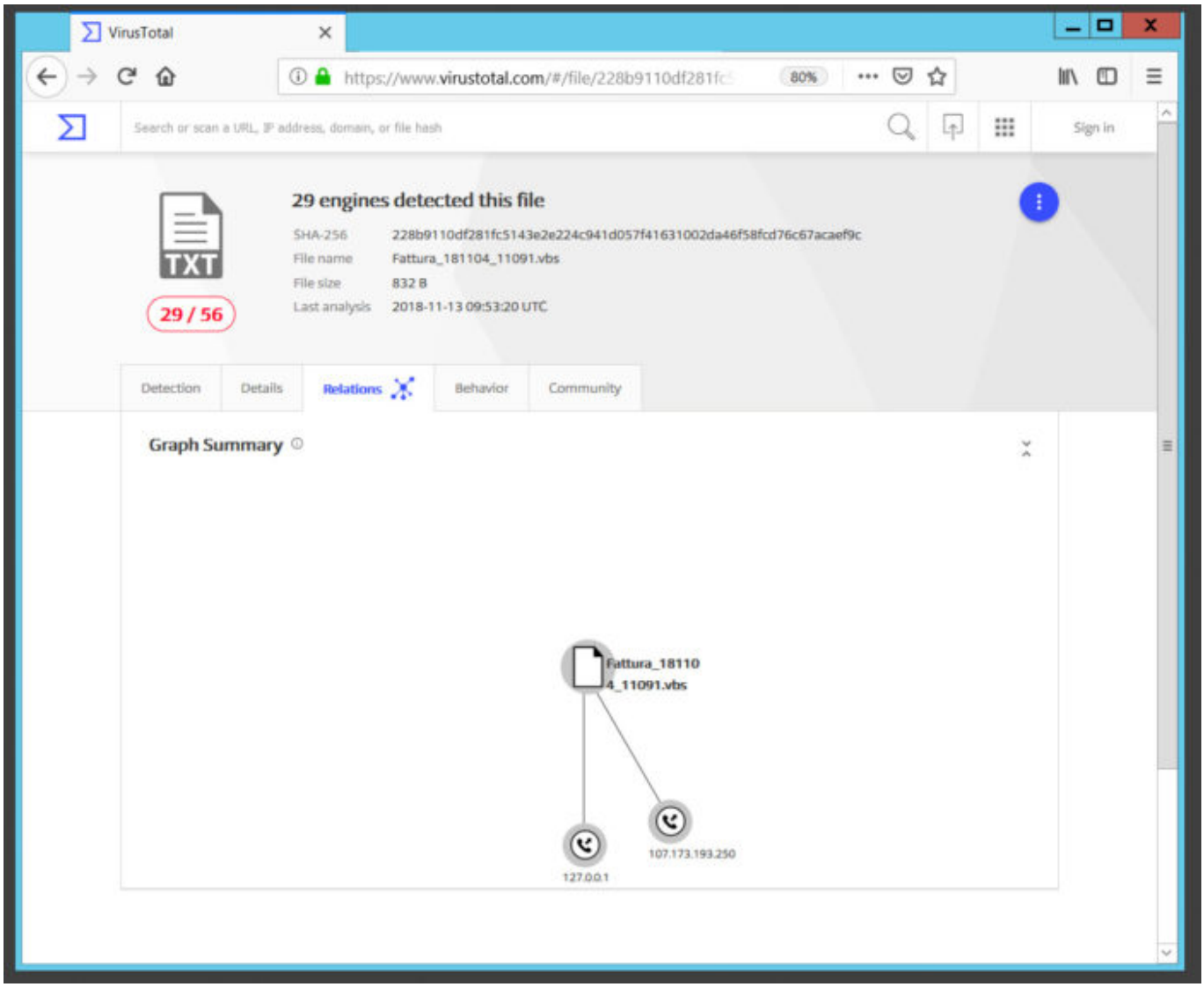
## METODOLOGIA STATICA

Come abbiamo anticipato, l'analisi statica cerca di studiare il comportamento del **malware a riposo**. Essa si suddivide in una analisi di base e una avanzata.

Nell'**analisi di base** si cerca di studiare l'eseguibile[1] senza prendere in considerazione le istruzioni in esso contenute. In questo modo possiamo confermare che il file sia effettivamente nocivo o meno e confermare alcune sue funzionalità.

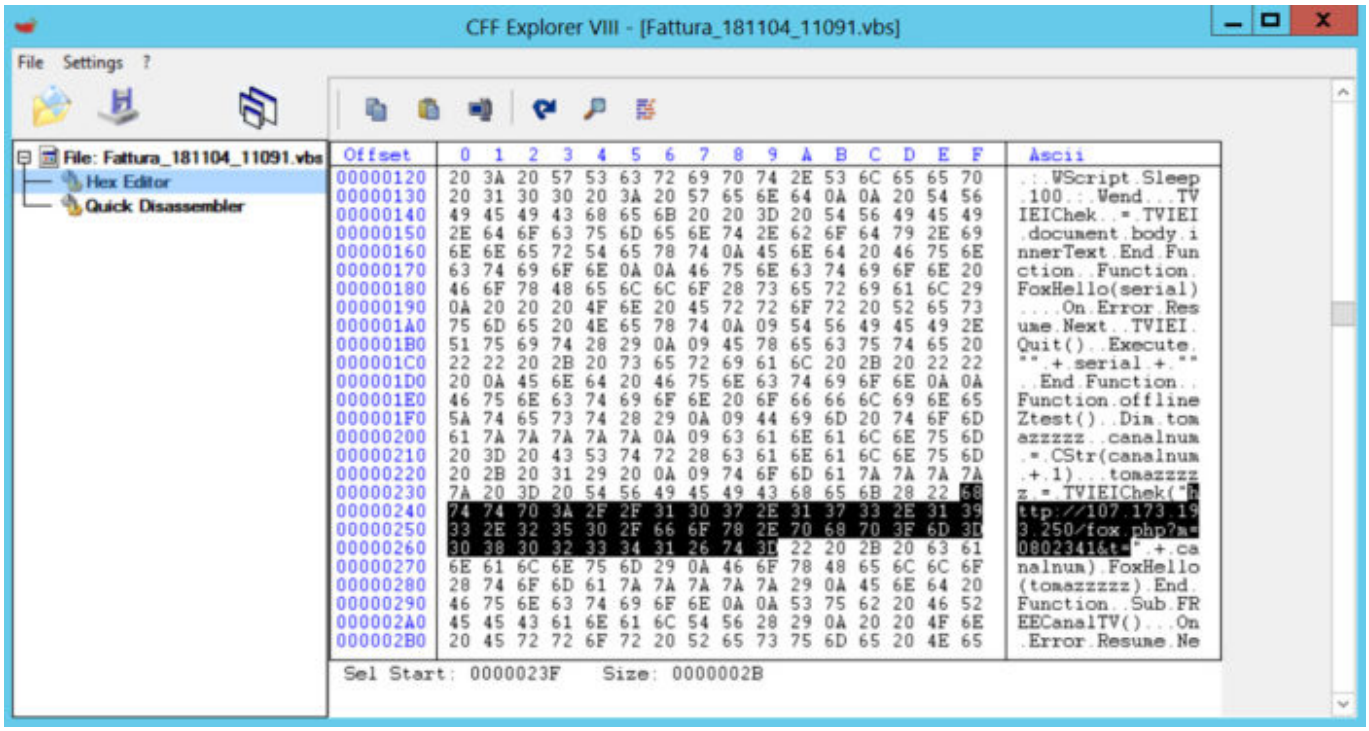



Analisi sicuramente veloce e semplice, ma largamente **inefficace** per stabilire delle contromisure adeguate e incompleta.




L’**analisi avanzata** consiste nell’effettuare il “*reverse engineering*[2]” del malware. Utilizzando dei software chiamati *“disassembler”*[3] si riescono a ottenere e studiare le istruzioni contenute nell’eseguibile. Questo ci permette di capire esattamente come si comporterà il malware all’interno di un PC una volta avviato.

L’analisi è molto lunga e complessa ed è facile cadere in errore, specie se non si conosce bene il linguaggio macchina e non si è in grado di intuire il possibile comportamento del malware.




## METODOLOGIA DINAMICA

L'analisi dinamica, suddivisibile anch'essa in base e avanzata, consiste nell'eseguire e osservare il comportamento del malware nel sistema. È bene precisare che il sistema che stiamo infettando sarà potenzialmente inutilizzabile: prima di effettuare l'analisi non sappiamo nulla sul malware, quindi il suo comportamento e i potenziali danni sono al momento sconosciuti. Potrebbe trattarsi di un Ransomware che inizierebbe a criptare i nostri dati o di un Backdoor che esporrebbe immediatamente il nostro sistema al vero e proprio attacco.

Quindi, **prima di avviare l'analisi dinamica è meglio creare un ambiente di laboratorio** (anche utilizzando delle macchine virtuali[4]) ben isolato e monitorato e che non contenga dati utili.

Nella sua parte **base**, si esegue il potenziale malware e si cerca di capirne il suo comportamento dallo stato del dispositivo infettato.

Nell'analisi dinamica **avanzata** viene utilizzato un debugger[5] per esaminare lo stato interno del malware mentre è avviato. Questo ci permette di leggere le singole istruzioni e di comprenderne, passo passo, il comportamento.

## CONCLUSIONI

Quando si vuole comprendere la natura di un malware, identificarlo, eliminarlo e proteggersi da

esso non sempre è possibile fermarsi alla semplice costatazione della sua mera natura di software malevolo.

Bisogna **indagare a fondo** e studiare con attenzione ogni riga di codice. Solo così sarà possibile comprendere la vera natura dell'eseguibile e il suo vero scopo.

## BIBLIOGRAFIA


https://it.wikipedia.org/wiki/Malware#Categorie_di_Malware

http://www.ollydbg.de/

https://www.hex-rays.com/products/ida/support/tutorials/debugging.shtml

https://www.packtpub.com/networking-and-servers/learning-malware-analysis

https://www.oreilly.com/library/view/practical-malware-analysis/9781593272906/

https://resources.flexera.com/web/pdf/Research-SVM-Vulnerability-Review-2018.pdf

https://www.blackhat.com/presentations/bh-dc-07/Kendall_McMillan/Presentation/bh-dc-07-Kendall_McMillan.pdf


## NOTE

[1] In informatica un file eseguibile, o semplicemente un eseguibile, è un file che contiene un programma eseguibile per un computer, ovvero un programma scritto in linguaggio macchina nel formato adatto ad essere caricato dal sistema operativo, quindi pronto per l'esecuzione e adatto all'architettura hardware del processore che lo esegue.

[2] Il processo di *reverse engineering* (chiamato in italiano anche ingegneria inversa) consiste nell'analisi dettagliata del funzionamento, progettazione e sviluppo di un oggetto (dispositivo, componente elettrico, meccanismo, software, ecc.) al fine di produrre un nuovo dispositivo o programma che abbia un funzionamento analogo, magari migliorando o aumentando l'efficienza dello stesso, senza in realtà copiare niente dall'originale; inoltre, si può tentare di realizzare un secondo oggetto in grado di interfacciarsi con il primo.

[3] Un disassemblatore o *disassembler* è un programma che traduce dal linguaggio macchina al linguaggio *assembly*. Effettua l'operazione inversa di un *assembler*.

[4] Il termine macchina virtuale (VM) indica un software che, attraverso un processo di virtualizzazione, crea un ambiente virtuale che emula tipicamente il comportamento di una macchina fisica (PC client o server) grazie all'assegnazione di risorse hardware (porzioni di disco rigido, RAM e risorse di processamento) e in cui alcune applicazioni possono essere eseguite come se interagissero con tale macchina; infatti, se dovesse andare fuori uso il

sistema operativo che gira sulla macchina virtuale, il sistema di base non ne risentirebbe affatto.

[5] Un debugger in informatica è un programma/software specificatamente progettato per l'analisi e l'eliminazione dei bug (debugging), ovvero errori di programmazione interni al codice di altri programmi.

Articolo a cura di **Giuseppe Turano**