

Malware Analysis 2 novembre 2023

	Basic Static Analysis	Advanced Static Analysis	Basic Dynamic Analysis	Advanced Dynamic Analysis
White box	V	V		
Grey box				V (più potente)
Black box			V	

Quando parliamo di **Analisi statica**, abbiamo due categorie:

- **Advanced:** uso del disassembler interattivo, come Ghidra, in grado di analizzare il codice mediante interazione con utente.
- **Basic:** Si limita a guardare cosa contiene il file per rapportarsi al mondo circostante. Usa dei tool per capire cosa c'è dentro il programma, senza scendere al livello dell'assembler. Posso vedere API, system call, .dll usate, ma non istruzioni macchina. A volte basta questo. Cosa ci posso fare?
 - **hash:** firme digitali, usata in ambito forense, per dimostrare che un certo file o documento non è stato alterato.
Possiamo calcolare con `sha256sum w03.exe`
 - Esistono siti che raccolgono *malware già noti*, quindi non ho bisogno di rifare il lavoro da 0. Tuttavia compaiono numerosi malware ogni giorno, quindi è difficile avere un database aggiornato. Gli *anti-virus*, per superare questo limite, identificano dei pattern particolari (*euristiche*) per l'identificazione. Il sito **Virustotal** contiene euristiche conosciute e ci dice se, un certo file, fornito da input, le contiene. Fornire un file a tale sito però, comporta aggiungere la firma del file nel sito, e quindi, chi ha creato il malware, può vedere se è stato analizzato su questo sito.
 - **stringhe:** cercare le stringhe dentro un eseguibile, mediante comando `strings`, oppure `strings -n 2 nomefile` se cerco stringhe di due caratteri. Il malware può offuscare le stringhe!
 - **PE header:** dentro troviamo numerose informazioni, come le API. Ci sono vari software, in Windows `cffexplorer`, a cui passo un eseguibile con *drag&drop* e ne decodifica la testata. Oppure c'è `PEview`, `peBear`, `peTools`, `dependency worker` (prende un eseguibile e ricostruisce API che individua, con tutti i collegamenti, anche ricorsive. Non più supportato.). Infine `resourceHacker`, in cui possiamo vedere risorse incluse, come icone, manifest, ... tutte sostituibili!

Parlando di **analisi dinamica**, si ha:

- **Basic:** esegue monitoraggio, come WireShark. Viene visto anche ciò che viene scritto o letto dalla nostra applicazione. `peID` è un tool per analisi statica, ma usa plugin che potrebbero eseguire il codice, quindi non è proprio di analisi statica.

- **Advanced:** eseguita con Debugger, strumento più efficiente. Monitor che associa ciò che fa il programma con codice ad alto livello. Lenta e costosa, ma potente, il malware lo teme e cerca di proteggersi.