Libro bianco

Crea una sandbox migliore

Una strategia operativa per la protezione completa antimalware

Indice dei contenuti

La co-evoluzione del malware e dell'analisi del rilevamento

I criminali informatici e gli strateghi della sicurezza IT sono coinvolti in una corsa agli armamenti in ascesa. Man mano che il malware sfruttato per infiltrare gli ambienti IT diventa sempre più sofisticato ed elusivo, emergono nuove tecnologie destinate a trovare il proverbiale ago nel pagliaio, a prescindere dall'efficacia con cui si camuffa.

Uno degli sviluppi più promettenti sul versante difensivo si verifica nell'area del rilevamento tramite analisi dinamica, comunemente noto come sandboxing. Numerosi prodotti sono già disponibili e altri si trovano in fasi diverse del processo di commercializzazione. Non desta sorpresa il fatto che gli approcci architetturali adottati dai principali sviluppatori variano profondamente tra loro, come pure i metodi che questi nuovi prodotti hanno di adattarsi a strategie di sicurezza più ampie. Al momento non è facile effettuare con fiducia confronti tra tecnologie o persino essere certi che la terminologia di base venga applicata in modo uniforme.

Questo documento propone una strategia di progettazione logica per un'analisi dinamica del malware che ottimizza efficacia, efficienza ed economia del rilevamento. Ci adopreremo per identificare i limiti del rilevamento dinamico e i metodi supplementari necessari per garantire una sicurezza solida. Infine, suggeriremo alcune distinzioni importanti che vengono solitamente eclissate da una terminologia imprecisa.

Analisi dinamica e statica a confronto

Forse il più importante elemento contestuale di qualsiasi discussione sulle soluzioni di sandbox malware avanzate è la distinzione tra analisi dinamica e analisi statica. L'analisi dinamica cerca di identificare i file eseguibili dannosi caricandoli in un ambiente di runtime sicuro, generalmente virtualizzato, e osservandone il comportamento per un intervallo di tempo prestabilito.

Per contrastare adeguatamente questa tattica con l'analisi statica è necessario innanzi tutto risolvere una discrepanza comune nell'uso di tale termine. L'analisi statica effettiva (definita a volte analisi statica del codice) prevede il probabile comportamento di un eseguibile in base a una valutazione dettagliata del relativo codice. Il termine "analisi statica" viene spesso applicato erroneamente a tecniche più semplici e meno rivelatrici (a volte definite analisi statica dei file) che possono esaminare solo una porzione dell'intestazione di un file o accedere soltanto al contenuto di un file non occultato. La relativa utilità volta a identificare il malware avanzato è limitata e qualsiasi impiego dell'analisi statica in questo documento fa riferimento a tecniche in grado di estrarre, esaminare e analizzare il codice completo di un file.

Tanto le tecniche dinamiche quanto le tecniche statiche effettive hanno pro e contro. L'analisi dinamica è in grado di identificare il malware con un livello di fiducia estremamente elevato in base all'osservazione diretta del relativo comportamento. È il metodo più affidabile per identificare accuratamente minacce nascoste in eseguibili complessi, ma può essere facilmente ostacolato con vari stratagemmi. Un file può semplicemente attendere oltre il periodo di osservazione, ritardando l'inizio del comportamento rivelatore per un intervallo di tempo prestabilito che può essere più lungo rispetto a un'ispezione sandbox economicamente fattibile. È anche possibile programmare un file affinché riconosca un ambiente sicuro dall'assenza (o dalla presenza) di determinate risorse e perché esegua soltanto un insieme limitato di operazioni apparentemente innocue.

L'ispezione statica individua il codice nocivo con un minore livello di fiducia rispetto all'analisi dinamica perché si affida alla deduzione anziché all'osservazione, e tuttavia assicura una finestra sulla natura del codice latente (non eseguibile) che risulta totalmente invisibile all'analisi dinamica. Ad esempio, l'analisi del codice statico identifica analogie strutturali tra il codice latente ed esemplari di malware conosciuti. Quantifica la percentuale di codice che esegue durante una valutazione sandbox ed esegue anche il mapping dei percorsi logici di esecuzione di un file complesso senza doverne di fatto eseguire il codice.

Ciò che sorprende è il grado di complementarietà dei punti di forza e dei punti deboli dell'analisi statica e dinamica. Mentre le tecniche comunemente associate al sandbox malware sono dinamiche, è improbabile che si possa ottenere un rilevamento affidabile e accurato a meno che entrambe le analisi del codice dinamica e statica effettiva vengano applicate nell'ambito di un processo ben integrato. Una sandbox efficace deve essere al contempo dinamica e statica.

L'imperativo delle prestazioni

Una caratteristica condivisa dall'analisi dinamica e dall'analisi statica del codice è che entrambe hanno un'elevata intensità di calcolo, tanto che nessuna delle due può essere applicata ai flussi del traffico di rete in tempo reale ed entrambe devono essere applicate selettivamente per evitare di degradare le prestazioni della rete e delle applicazioni. Per completare questo tipo di analisi potrebbero occorrere pochi minuti o anche diverse ore, a seconda del tipo di prodotto e fornitore. Un approccio razionale consiste nell'inserire un front-end con tecnologie più efficienti a livello di risorse, capaci di eliminare in modo rapido e conveniente le minacce identificate facilmente.

Riteniamo che sia possibile creare una sandbox malware altamente efficiente ed efficace sovrapponendo tipi diversi di motori di analisi in una sequenza impilata ad intensità di calcolo crescente. Tutti i file sconosciuti intercettati dai sensori di sicurezza di rete possono essere segnalati a questo servizio per la valutazione. Ciascun file passa attraverso i motori di ispezione impilati a partire dal più rapido e a minore intensità di uso di risorse. I file identificati come dannosi a qualsiasi livello vengono immediatamente bloccati e rimossi dal flusso di ispezione, con conseguente riduzione del carico sull'analisi a valle a maggiore intensità di calcolo.

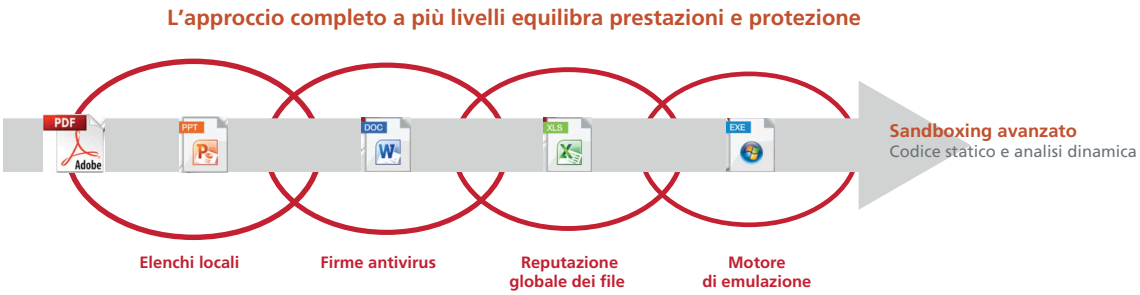


Figura 1. Un processo di circoscrizione multimotore per l'ispezione del malware.

Mentre la composizione dello stack per l'ispezione deve essere flessibile per poter incorporare nuove tecniche di rilevamento man mano che emergono, l'attuale stato dell'arte può venire ben rappresentato da una sequenza che inizia con rilevamento attacchi noti (firme e servizi di reputazione), seguito da rilevamento del comportamento in tempo reale (euristica ed emulazione), analisi dinamica e analisi statica del codice. Definiamo questa un'architettura di ispezione tramite circoscrizione. Esaminiamo la sequenza stratificata del processo livello per livello.

Primo livello: rilevamento attacchi noti

Le due tecniche di rilevamento malware utilizzate nell'ispezione di primo livello sono le tecniche più vecchie e più estesamente implementate. Sono anche le tecniche maggiormente in tempo reale e più leggere dal punto di vista della potenza di calcolo richiesta. L'ispezione basata sulle firme, la tecnologia di base di tutti i prodotti antivirus, assicura un'identificazione rapida e positiva basata sulle corrispondenza di schemi a fronte di una libreria di esemplari di codice nocivo noto. I servizi di reputazione raccolgono informazioni sulle fonti note di attacchi precedenti, compresi hash del malware effettivo, posizioni geografiche, domini, URL e indirizzi IP, per garantire una base per l'identificazione di attacchi noti, sconosciuti e zero-day provenienti da noti vettori dannosi.

Entrambe queste tecniche sono rapide, frugali dal punto di vista della potenza di calcolo richiesta e forniscono un'identificazione delle minacce estremamente affidabile in tempo reale. I relativi attributi essenziali sono (1) una libreria completa di firme e fonti note delle minacce e (2) un'infrastruttura rapida e affidabile per l'acquisizione di nuove informazioni sulle minacce a livello globale con conseguente distribuzione ai sensori locali.

Poiché entrambe queste tecnologie e, in misura minore, le tecnologie incluse nel secondo livello illustrato di seguito, sono ampiamente implementate nei prodotti di sicurezza esistenti, è estremamente utile quando le nostre funzionalità di gestione sandbox comprendono la capacità di definire separatamente i motori di ispezione delle sandbox da applicare ai file a cui fa riferimento ciascun tipo di sensore. Di conseguenza, l'ispezione basata sulle firme che è stata eseguita da un servizio IPS, ad esempio, non verrà ripetuta in corrispondenza della sandbox.

Secondo livello: difese comportamentali in tempo reale

Vengono anche applicati due diversi tipi di rilevamento nel secondo livello di ispezione: euristica ed emulazione. L'identificazione euristica utilizza regole e analisi dei modelli di comportamento per creare costrutti generici e individuare analogie tra un file sospetto e gruppi o famiglie di minacce note correlate. L'emulazione simula l'esecuzione dei file su un ambiente host semplificato e registra i comportamenti risultanti. L'ambiente di emulazione può includere un sottoinsieme di CPU, memoria e di risorse API del sistema operativo. L'emulazione viene talvolta descritta come analisi dinamica da intermedia a completa o come sandboxing leggero, ma fa un uso sufficientemente meno intenso delle risorse e assicura risultati in tempo reale.

Euristica ed emulazione assicurano l'identificazione in tempo reale delle minacce non osservate in precedenza e il loro livello di affidabilità è solo leggermente inferiore alle tecniche basate sulle firme. Esse implicano alcune operazioni di decompilazione e decifratura del codice, ma poiché si tratta di un processo in tempo reale, sono stati implementati pochi strumenti per decifrare o decompilare i file occultati.

Occorrono emulatori diversi e specifici per lingua per diversi tipi di contesto (eseguibili, codice shell, JavaScript, HTML e Java). L'affidabilità e l'efficacia di un motore di emulazione sono direttamente correlate alla completezza delle sue funzionalità.

Terzo livello: analisi dinamica

Il terzo livello nella nostra architettura sandbox modello segna la linea divisoria tra le analisi che si svolgono effettivamente in tempo reale e quelle che utilizzano tecniche con un maggiore uso di risorse e che inevitabilmente impongono una latenza lievemente superiore. È in questa circostanza che ai file che non sono stati identificati inequivocabilmente come dannosi durante le ispezioni precedenti viene consentita l'esecuzione in un ambiente virtuale isolato in modo sicuro. L'analisi dinamica effettiva differisce dall'emulazione perché esemplifica un ambiente di runtime totalmente operativo che è virtualizzato e isolato per consentire un'esecuzione sicura del codice potenzialmente nocivo, quindi registra o classifica tutti i comportamenti osservati.

Sandboxing specifico per obiettivo

Esistono due approcci comuni alla configurazione degli ambienti virtuali utilizzati nel sandbox di malware. Le differenze sono importanti perché la maggior parte degli ambienti IT comprende una vasta gamma di piattaforme hardware e software e gran parte degli esemplari di malware prende di mira un ambiente operativo o un'applicazione specifici. Il primo approccio consiste nel virtualizzare un unico ambiente generico e nell'utilizzarlo in tutte le analisi degli esemplari. Questo approccio rischia tuttavia di non cogliere i comportamenti dannosi che dipendono da specifiche serie di risorse o parametri di configurazione, che potrebbero non essere disponibili nell'immagine generica, ma è efficiente dal punto di vista delle risorse poiché richiede un unico passaggio di analisi.

Il secondo approccio consiste nel virtualizzare più ambienti (varie piattaforme e configurazioni Windows server più una selezione di immagini di PC e piattaforme mobili). Gli esemplari sospetti vengono eseguiti a fronte di ciascuno di questi ambienti. Questa strategia, tuttavia, rischia comunque di non cogliere l'effettivo ambiente di destinazione, può produrre altri falsi positivi ed è anche assai più costosa dal punto di vista della potenza di calcolo.

Una strategia estremamente più efficace ed efficiente consiste nell'eseguire un eseguibile sospetto in un ambiente virtuale che corrisponde esattamente al sistema per cui il file era stato preso di mira. Questo approccio richiede la disponibilità di una vasta gamma di sistemi operativi diversi o la possibilità di importare immagini gold di tutte le piattaforme endpoint nell'ambiente. La sandbox deve essere in grado di identificare l'ambiente host di destinazione al volo e di lanciare rapidamente una macchina virtuale corrispondente; non si tratta di un'attività di rete ma richiede comunque l'integrazione con sistemi endpoint. Se è possibile soddisfare queste condizioni, è molto più probabile che la gamma completa dei comportamenti potenziali di un file sospetto possa essere ottenuta e osservata, e che possa essere effettuata una valutazione accurata del suo intento.

L'importanza dell'interazione

Affinché l'ispezione della sandbox possa valutare completamente l'intento di un eseguibile, l'ambiente virtuale deve rispondere in modo interattivo al suo comportamento come farebbe un normale sistema host. In particolare, la sandbox deve emulare automaticamente la normale risposta dell'host alle richieste delle connessioni di rete L'assenza di tali risposte previste potrebbe informare il malware che è in corso un'analisi nella sandbox e potrebbe consentirgli di intraprendere azioni evasive. Anche i servizi di reputazione devono essere disponibili alla sandbox, affinché le richieste ad alto rischio per l'accesso a noti indirizzi IP, URL e file dannosi possano venire immediatamente identificati come indicatori di minacce ad elevata probabilità.

Oltre all'interattività, che deve essere disponibile per avviare automaticamente le ispezioni dei file inline, deve essere disponibile agli analisti della sicurezza anche una modalità completamente interattiva per l'analisi dei manuali offline. In questo modo, un analista dovrebbe essere in grado di avviare manualmente una macchina virtuale e caricare un esemplare di eseguibile con funzionalità KVM completa. Spesso è solo con la possibilità di avviare sessioni browser e altre normali applicazioni di lavoro che è possibile attivare e osservare comportamenti specifici delle minacce.

Interpretazione delle analisi dinamiche

I primi due livelli del nostro stack per l'ispezione producono output semplici in tempo reale. Un file sconosciuto viene rapidamente identificato come minaccia nota tramite una corrispondenza di firma o viene considerato sufficientemente dannoso negli ambienti di emulazione in tempo reale. In entrambi i casi, viene rapidamente presa la decisione di bloccare o lasciar passare.

Viceversa, l'output iniziale di un'analisi dinamica è un file di registro che assume significato unicamente con la correlazione. Eventi comportamentali specifici devono venire identificati e aggregati e il loro significato valutato nel contesto di altri eventi. L'output di una sandbox enterprise-ready non è una lettura di registro lunga e complicata e non è nemmeno una decisione per lasciare passare/bloccare particolarmente semplice. Un utile strumento di sandboxing per l'azienda deve fornire un rapporto aggregato e organizzato che definisce e classifica comportamenti rilevanti e assegna un punteggio complessivo. Tale punteggio può essere sufficiente per attivare una decisione di blocco o può richiedere informazioni di supporto da analisi statiche successive. Quale che sia la scelta effettuata, vengono fornite informazioni attuabili che gli operatori di sicurezza possono usare.

Nasconde il file modificandone gli attributi.	Manipolato con contenuto attivo nella directory temporanea di amministrazione.
Rilevato contenuto eseguibile lasciato cadere dall'esemplare.	Ottenuta e utilizzata icona di applicazione di sistema legittima.
Creato contenuto eseguibile nella directory temporanea dell'amministratore.	Creato contenuto eseguibile nella directory Windows.
Da Microsoft: CreateURLMoniker può produrre risultati che non sono equivalenti all'input e il suo utilizzo può comportare problemi di sicurezza.	Eseguito contenuto attivo dalla cartella di sistema di Windows.
Impegna una regione di memoria all'interno dello spazio dell'indirizzo virtuale di un processo estraneo.	Impostare la funzione di callback per controllare il sistema e gli eventi hardware del computer.
Tentativo di connettersi a un fornitore di servizi specifico.	Scaricati dati da un server web.
Creato contenuto nella directory Windows System.	Registrato (annullata registrazione) del nome del servizio nei supporti del server Dynamic Data Exchange (DDE).

Figura 2. Un riepilogo comportamentale e un rapporto di classificazione dell'analisi dinamica.

Quarto livello: analisi del codice statico

La circoscrizione finale nella nostra sandbox espansa multifase è l'analisi del codice statico effettivo, che potremmo definire in modo più descrittivo analisi di disassemblaggio del codice elenco. Questo processo viene avviato con l'analisi dinamica di fase tre e integra alcuni output di ispezione dinamica non appena diventano disponibili.

Abbiamo rilevato in precedenza che le nostre proposte di tecnica per l'ispezione di fase due, euristica ed emulazione, si affidano all'accesso al codice sorgente di un file, ma abbiamo anche detto che verrebbero forniti pochi strumenti nell'analisi in tempo reale per estrarre il codice occultato o cifrato. Questo tipo di esame forense approfondito è precisamente il nostro obiettivo qui nella fase quattro.

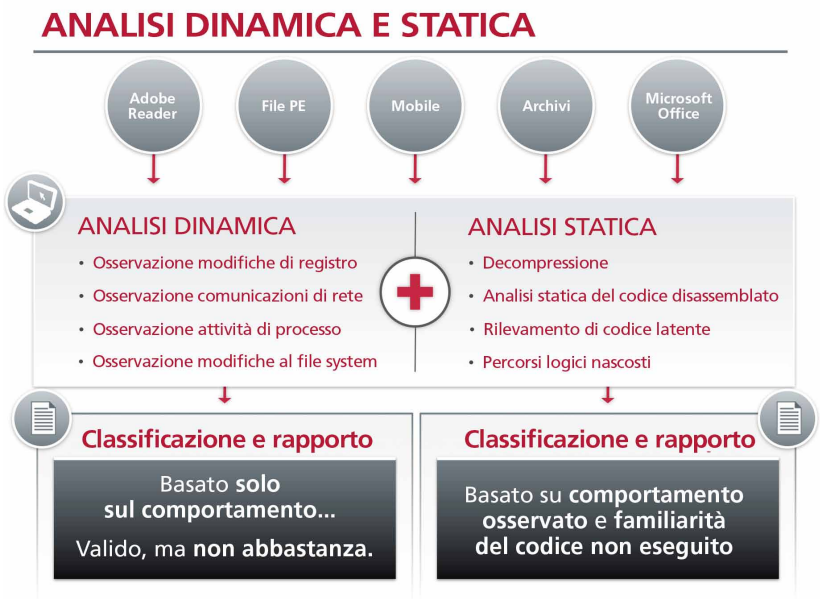


Figura 3. La sola analisi dinamica è un'analisi incompleta.

Cifratura, decifratura e decompilazione

Esistono dei motivi perfettamente legittimi per nascondere o per occultare il codice eseguibile compilato di un programma, il più ovvio dei quali è la protezione della proprietà intellettuale. Comprensibilmente, gli sviluppatori software desiderano impedire alla concorrenza di decompilare i loro prodotti lavorando a ritroso dal codice assembly distribuito alla sorgente. Ed evidentemente, altri intraprendenti sviluppatori hanno creato una vasta scelta di strumenti commerciali per rendere questa operazione particolarmente difficile. Questi speciali programmi di compressione (ad es. Themida, Armadillo) facilitano l'applicazione di un intero arsenale di tecniche di mascheratura e sequenza casuale al codice del programma compilato, rendendo estremamente difficile la ricostruzione del codice assembly e quindi l'accesso alla sorgente. Gli autori e sviluppatori di malware hanno semplicemente adottato le stesse tecniche dell'industria software, rendendo assai più difficile distinguere i loro attacchi occultati dai file legittimi.

Difficile forse, ma non impossibile.

Figura 4. Il menu di occultamento in Themida, un potente programma di compressione per le applicazioni Windows.

Nella quarta fase della nostra sandbox malware, i file compressi e occultati vengono sottoposti al reverse engineering per recuperare versioni integre del codice assembly compilato. Queste vengono quindi analizzate e sottoposte a un'analisi statistica, che fornisce:

- Una valutazione delle analogie con le famiglie di malware note.
- Una misura del codice latente che non è stato eseguito durante l'analisi dinamica.
- Una mappa logica del percorso completo di esecuzione del file.

Le moderne minacce avanzate persistenti sono generalmente adattamenti di codice malware noto ed efficace. Sono sufficienti modifiche marginali per eludere l'ispezione delle firme, che per avere un esito positivo richiede una corrispondenza esatta. Tuttavia, l'ispezione dell'intero codice, rispetto a una libreria di riferimenti di famiglie di malware note, finisce spesso per individuare malware clandestino.

Ad esempio, un file sospetto con alcuni indicatori minori di compromissione che non sono tuttavia abbastanza gravi da causare il blocco del file, ma che hanno una somiglianza superiore al 70% a una famiglia di malware nota (ad es. conficker e voter_1), è un file che deve essere bloccato. Senza l'analisi statica del codice, questo malware sarebbe penetrano nella rete.

Figura 5. La somiglianza di famiglia è una prova convincente del potenziale dannoso.

Analogamente le minacce avanzate persistenti sono sempre più alla ricerca di consapevolezza ambientale o richiedono una sequenza specifica di interazione per stimolare azione. Ciò significa che gran parte del codice del malware clandestino può restare latente durante l'analisi dinamica. Anche se non è possibile estrarre il comportamento del codice latente, il fatto che una porzione consistente del codice di un file sospetto resti latente dovrebbe costituire una parte importante dell'indagine di un analista di sicurezza.

Consideriamo un file sospetto che non mostra alcun comportamento dannoso. La sola analisi dinamica arriverebbe alla conclusione che il file è sicuro. E se il file avesse una somiglianza superiore al 70% a una famiglia di malware nota e se più del 40% del codice del file non fosse attivo o analizzato nella sandbox? Questi due elementi di prova indiziari sono sufficienti a bloccare la consegna del file, almeno fino a quando un operatore della sicurezza non sarà in grado di condurre un'indagine.

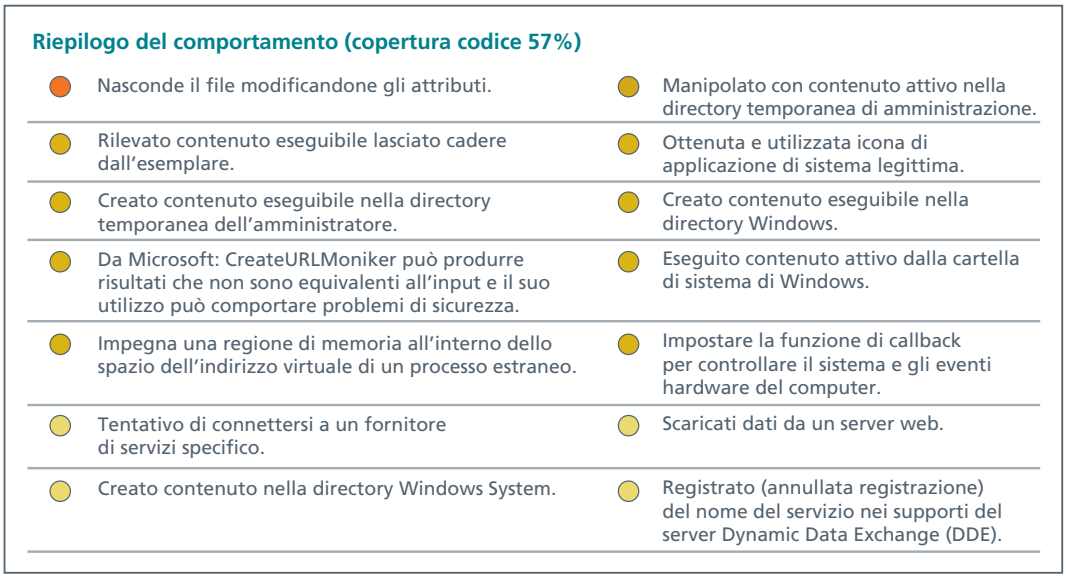
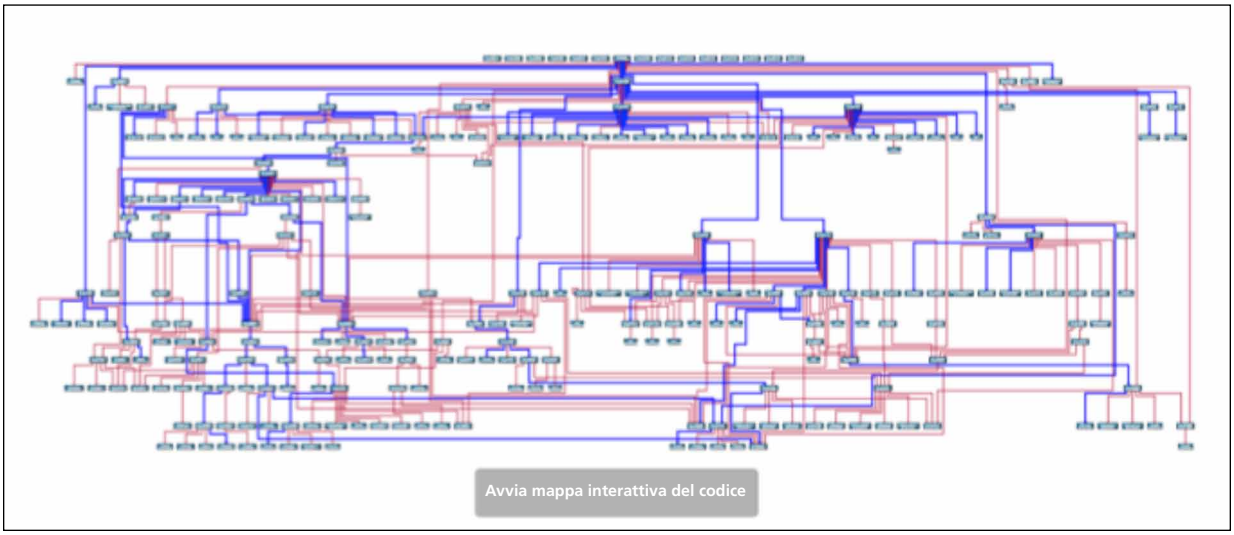


Figura 6. Porzioni significative del codice latente indicano l'esistenza di falle nell'analisi esclusivamente dinamica.

Quando si deve ricorrere all'indagine umana, un diagramma delle operazioni del file può rivelarsi uno strumento forense estremamente utile. A differenza dei file di registro o delle osservazioni del comportamento (analisi dinamica), un diagramma consente agli operatori della sicurezza di investigare le aree nascoste e l'interazione del codice. Questa è spesso la chiave per individuare codice illecito altrimenti latente, specie se utilizzato in abbinamento a una sandbox che consente l'interazione manuale con il file sospetto all'interno della macchina virtuale.



Le linee blu mostrano del codice che è stato eseguito in modo dinamico.
Le linee rosse mostrano del codice che è stato eseguito in modo statico.

Figura 7. La visualizzazione dei processi consente a un operatore di stimolare il codice latente.

Questi risultati vengono quindi integrati alle osservazioni provenienti dall'analisi dinamica di fase tre per fornire un punteggio complessivo che indica con che grado di certezza un file esemplare o un eseguibile sono dannosi.

Una strategia di circoscrizione per la superiorità della sandbox

Un servizio avanzato di rilevamento malware progettato e configurato seguendo la strategia di circoscrizione impilata descritta sopra assicurerà una soluzione significativamente più sofisticata, affidabile e conveniente rispetto a quanto è attualmente disponibile sul mercato. Impedirà il sovraccarico delle sandbox con la schermatura di minacce note e facilmente identificabili tramite servizi di ispezione e informazioni sulla reputazione ad elevata percentuale e bassi costi di esercizio basati sulle firme. Farà aumentare drasticamente l'efficienza e l'accuratezza dell'analisi dinamica con il sandboxing dell'esecuzione specifica per obiettivo. Infine, l'analisi effettiva statica del codice renderà trasparente l'occultamento per rivelare la natura nascosta del codice latente ed elusivo.

Per una volta nella storia contorta della co-evoluzione della sicurezza del malware, la sicurezza sta per compiere un proattivo salto in avanti.

Ulteriori informazioni all'indirizzo www.mcafee.com/it/products/advanced-threat-defense.aspx.

McAfee
An Intel Company

McAfee Srl
via Fantoli, 7
20138 Milano
Italia
(+39) 02 554171
www.mcafee.com/it

60837wp_build-sandbox_0214_fnl_ETMG