Università degli Studi di Messina

Università degli Studi di Messina
Dipartimento di Scienze Matematiche e Informatiche, Scienze Fisiche e Scienze della Terra

Corso di Laurea Triennale in Scienze e Tecnologie Informatiche

Virtualizzazione e Sicurezza: Attacchi e Contromisure Esplorazione di attacchi Tecniche di hardening e isolamento

Studente:
Vitaliy Lyaskovskiy

Insegnamento:
Sistemi Operativi mod. B

Anno Accademico 2024-2025

Indice

Capitolo 1

Introduzione

Negli ultimi anni la virtualizzazione ha rivoluzionato il modo in cui concepiamo e utilizziamo i sistemi operativi. Non si tratta soltanto di un'ottimizzazione delle risorse, resa possibile dalla capacità di eseguire più sistemi su un unico hardware, ma di un vero cambiamento radicale che permette oggi di astrarre completamente le risorse fisiche e costruire ambienti isolati facilmente replicabili. Questa tecnica ha trovato applicazione in numerosi contesti: dal cloud computing, che permette alle aziende di affittare risorse da provider come AWS o Google Cloud, allo sviluppo software in ambienti virtualizzati, rendendo possibile la creazione di applicazioni portabili su sistemi operativi diversi.

Accanto a tutti questi vantaggi, tuttavia, emergono anche aspetti negativi da considerare che riguardano la sicurezza. L'uso crescente di macchine virtuali e container ha aumentato i punti vulnerabili che un attaccante potrebbe sfruttare; un singolo errore di configurazione di una macchina virtuale, infatti, potrebbe compromettere l'intero sistema che ospita i servizi di un'azienda.

In questa tesina ho scelto di approfondire il tema della sicurezza nella virtualizzazione, trattandosi di una tecnologia ormai centrale nel mondo IT. Dopo una discussione dei concetti fondamentali riguardanti la virtualizzazione e la sicurezza in cui saranno presentate le principali tecniche di hardening e le contromisure adottate per prevenire la violazione dei sistemi, sarà esaminato un caso reale di attacco informatico utile per comprendere le vulnerabilità e gli errori più comuni nella gestione delle infrastrutture virtuali.

Capitolo 2

Le basi della virtualizzazione

2.1 Cos'è la virtualizzazione

La virtualizzazione è una tecnologia che permette di creare macchine virtuali, ovvero rappresentazioni astratte di risorse hardware come processore, memoria, rete e storage, con l'obiettivo di creare ambienti isolati eseguibili su un singolo sistema fisico noto come host. Una macchina virtuale, configurata in un host, possiede un proprio sistema operativo che prende il nome di guest.

2.2 Tipi di virtualizzazione

Quando parliamo di virtualizzazione è importante distinguere tra diverse tecniche che, pur condividendo l'obiettivo di astrarre le risorse fisiche, lo implementano in modi differenti.

2.2.1 Virtualizzazione a livello di hypervisor

Nella virtualizzazione a livello di hypervisor, quest'ultimo è un componente software chiave in quanto gestisce l'esecuzione delle macchine virtuali sullo stesso host, garantendo che ogni macchina virtuale abbia accesso alle risorse fisiche di cui ha bisogno e impedendo interferenze reciproche. Ulteriori sue funzionalità riguardano il controllo dell'allocazione dinamica della memoria, la gestione delle interfacce di rete virtuali e la possibilità di eseguire snapshot e backup delle macchine virtuali. In base alle loro implementazioni, gli hypervisor possono essere classificati in due tipi differenti:

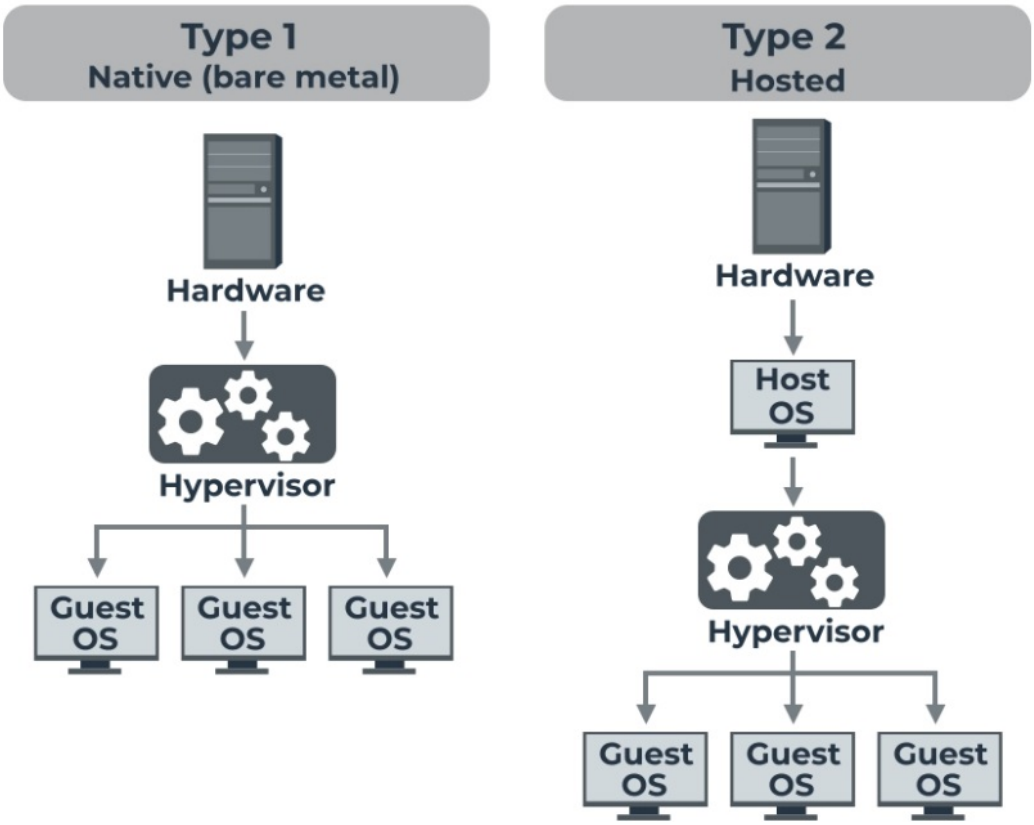

Figura 2.1: Confronto tra Hypervisor di Tipo 1 e di Tipo 2. Fonte

Tipo 1: bare-metal

L'hypervisor di tipo 1 viene installato direttamente sull'hardware, senza un sistema operativo sottostante. Questa architettura, definita bare-metal, lo rende estremamente efficiente in quanto può comunicare direttamente con le risorse hardware eliminando lo strato intermedio. Alcuni esempi di hypervisor di tipo 1 sono VMware ESXi e Microsoft Hyper-V.

Tipo 2: hosted

L'hypervisor di tipo 2 è invece un'applicazione che viene eseguita sopra un sistema operativo host già presente. È facile da installare e configurare poiché è il sistema operativo dell'host a gestire la comunicazione con l'hardware. Tuttavia, risultano generalmente meno efficienti rispetto agli hypervisor di tipo 1 proprio a causa dello strato aggiuntivo tra se stesso e le risorse fisiche. Alcuni esempi di hypervisor di tipo 2 sono VMware Workstation e VirtualBox.

2.2.2 Virtualizzazione a livello di sistema operativo

La virtualizzazione a livello di sistema operativo, nota anche come containerizzazione, consente l'esecuzione di più ambienti isolati che condividono il kernel del sistema operativo host. Un container è un'immagine software formata da diversi strati: un'applicazione, le sue librerie e tutte le altre dipendenze necessarie per il funzionamento. Questo approccio rende l'applicazione portabile e isolata dal sistema operativo sottostante. La gestione dei container è affidata a un componente chiamato container engine, come Docker Engine, che si occupa della loro creazione ed esecuzione. Il vantaggio principale dei container è il notevole risparmio di risorse rispetto alle macchine virtuali, poiché eliminano la necessità di un hypervisor.

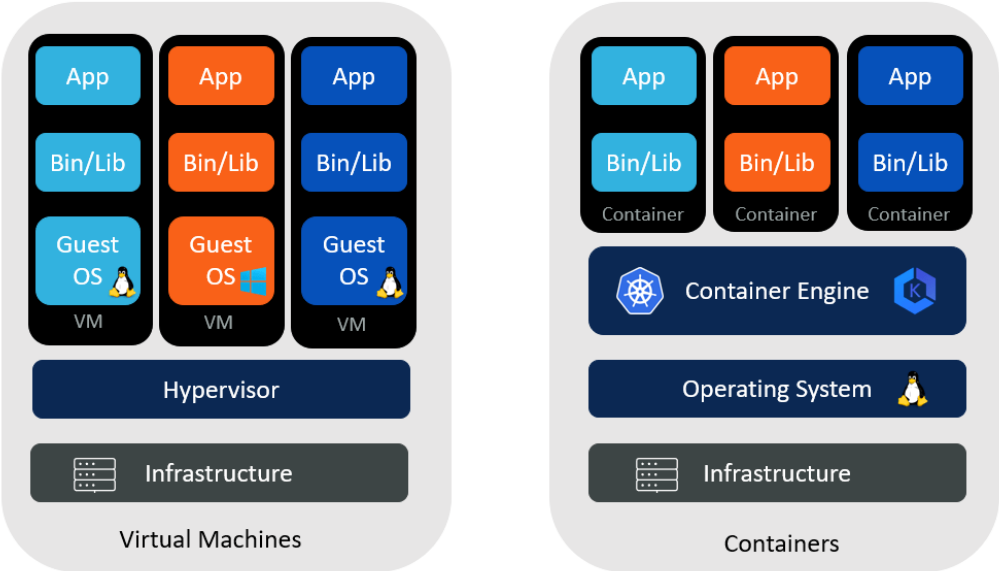

Figura 2.2: Confronto tra Macchine Virtuali e Containers. Fonte

2.3 Sviluppo della virtualizzazione

Negli ultimi anni la virtualizzazione ha subito una rapida crescita favorita dalla diffusione dei servizi cloud. Grazie a queste tecnologie, infatti, le aziende possono gestire le proprie infrastrutture in maniera più scalabile ed efficiente. Secondo uno studio di Credence Research:[1]

[1]Fonte: www.credenceresearch.com

- Si prevede che il mercato globale delle macchine virtuali crescerà da 39,5 miliardi di dollari nel 2024, a oltre 171 miliardi entro il 2032, con una crescita annua del 20,1%.
- Nello stesso periodo di tempo il mercato della virtualizzazione nei data center, si prevede passerà da circa 10,1 a oltre 33 miliardi di dollari.

Un'altra ricerca condotta da G2[2] mostra che:

- Il 66% delle aziende afferma di aver aumentato le proprie prestazioni di business grazie alla virtualizzazione.
- Il 50% dichiara che la virtualizzazione ha migliorato l'efficienza generale delle proprie attività professionali.

2.4 Principali piattaforme di virtualizzazione

Nel panorama attuale della virtualizzazione esistono diverse piattaforme, ognuna con le proprie caratteristiche che la rende adatta per un contesto specifico. Di seguito vengono riportate le soluzioni maggiormente utilizzate:

- **VMware:** rappresenta la principale piattaforma del mercato con una presenza pari al 44,5%.[3]
- **Microsoft Hyper-V:** è una piattaforma Microsoft integrata in Windows Server, è presente con una quota di mercato pari all'11%.[4]
- **KVM (Kernel-based Virtual Machine):** si tratta di una piattaforma integrata nel kernel Linux.
- **Docker:** è la principale piattaforma per quanto riguarda la containerizzazione di applicazioni.

[2]Fonte: learn.g2.com
[3]Fonte: growrk.com
[4]Fonte: www.maximizemarketresearch.com

Capitolo 3

Sicurezza e vulnerabilità nei sistemi virtualizzati

3.1 Introduzione alle vulnerabilità legate alla virtualizzazione

L'evoluzione delle soluzioni informatiche ha portato numerosi vantaggi nella gestione delle risorse, ma ha introdotto anche nuove superfici d'attacco che prima non esistevano; un attaccante che riesce a violare un sistema, infatti, potrebbe compromettere contemporaneamente diverse macchine virtuali in esecuzione sullo stesso host. Le minacce e i rischi non si limitano alle sole macchine virtuali, ma riguardano sempre più anche i container. Nonostante le differenze strutturali, entrambe le tecnologie condividono molte delle problematiche legate alla sicurezza.

Secondo un rapporto di IMARC Group, il mercato globale della container security, stimato in 2,4 miliardi di dollari nel 2024, è destinato a raggiungere i 16,6 miliardi entro il 2033, con un tasso di crescita annuale del 24,14%.[1].

Di seguito vengono descritte le principali vulnerabilità che riguardano i sistemi virtuali.

Espansione incontrollata delle macchine virtuali

L'espansione incontrollata delle macchine virtuali rappresenta una vulnerabilità nota come VM Sprawl, si verifica quando vengono create nuove macchine virtuali senza eliminare quelle non più necessarie in quanto vi è il rischio che alcuni sistemi contenenti informazioni sensibili, diventino vulnerabili a causa di configurazioni non aggiornate. Per evitare che questo accada, è essenziale

[1]Fonte: www.zerounoweb.it

monitorare regolarmente le macchine virtuali esistenti e terminare quelle non più utilizzate.

Autenticazione per l'accesso ai sistemi virtuali

Un'altra vulnerabilità riguarda l'utilizzo di credenziali poco sicure in quanto potrebbe permettere l'accesso non autorizzato ai sitemi virtuali, per impedirlo, è fondamentale implementare robusti meccanismi di sicurezza come l'utilizzo di password complesse e autenticazione multifattore per accedere ai sistemi critici, come per la gestione dell'hypervisor.

Configurazioni di rete inadeguate

Un'ulteriore vulnerabilità è dovuta a configurazioni di rete inadeguate come porte firewall lasciate aperte oppure un routing configurato in modo errato, in quanto rappresentano punti d'ingresso sfruttabili dagli attaccanti.

Copie di backup di ripristino

La sicurezza delle macchine virtuali riguarda anche le copie di backup utilizzate per il disaster recovery che, se ripristinate con configurazioni obselete, possono introdurre ulteriori gravi vulnerabilità.

Utilizzo di API in ambienti ibridi

Infine, nelle architetture ibride che combinano infrastruttura locale e servizi cloud, la comunicazione tra i due ambienti avviene attraverso le Application Programming Interface. Poiché queste interfacce consentono lo scambio di dati e la gestione delle risorse tra i due ambienti, se non protette adeguatamente, rappresentano un punto vulnerabile che può essere sfruttato dagli attaccanti.

3.2 Dati attuali sulle vulnerabilità nei sistemi informatici

La rapida crescita degli ambienti virtuali ha comportato un aumento delle vulnerabilità che possono essere sfruttate per compromettere l'integrità dei sistemi informatici.

Secondo il Verizon Data Breach Investigations Report[2] del 2024, lo sfruttamento di vulnerabilità ha rappresentato il punto d'ingresso nel 14% di tutte le violazioni dei dati analizzate.

Secondo il rapporto di Kiteworks[3] dello stesso anno, il costo medio globale di una violazione dei dati ha raggiunto i 4,88 milioni di dollari, con un aumento del 10% rispetto all'anno precedente.

[2]Fonte: www.verizon.com
[3]Fonte: www.kiteworks.com

Capitolo 4

Attacchi mirati ai sistemi virtuali

4.1 Introduzione alle tipologie d'attacco

I moderni sistemi di virtualizzazione hanno introdotto nuove superfici d'attacco sfruttabili per violare i dati. In questo capitolo vengono analizzati i meccanismi d'attacco indirizzati alle macchine virtuali, container e piattaforme cloud.

4.2 Attacchi all'hypervisor e alle macchine virtuali

L'attacco all'hypervisor ha come obiettivo sfruttare una vulnerabilità nell'isolamento tra le macchine virtuali e il loro ambiente ospitante per ottenere l'accesso a tutte le macchine virtuali in esecuzione sullo stesso host.

Virtual Machine Escape

La fuga dalla macchina virtuale è uno degli attacchi più critici, consiste nel rompere l'isolamento tra la macchina virtuale ed il sistema host, ottenendo pertanto accesso diretto al sistema operativo sottostante. Per poter uscire dall'ambiente virtuale, un processo malevolo sfrutta generalmente una vulnerabilità nota, come per esempio un buffer overflow in un driver virtualizzato.

Hyperjacking

L'attacco di tipo hyperjacking consiste nel prendere il controllo dell'hypervisor sfruttando una vulnerabilità nel sistema host. In alcuni casi l'aggressore riesce a installare un proprio hypervisor malevolo, noto come rootkit hypervisor, al di sotto del sistema operativo host, ottenendo il controllo completo sia del sistema host che delle macchine virtuali in esecuzione.

Attacco Side-Channel

Se un attaccante riesce a violare una macchina virtuale, può effettuare un attacco di tipo Side-Channel per cercare di violare le altre macchine in esecuzione sullo stesso host. Nello specifico, questa tipologia di attacco sfrutta informazioni indirette, come i tempi di accesso alla cache oppure il consumo di risorse condivise, per dedurre le operazioni eseguite da un'altra macchina virtuale in esecuzione sullo stesso host, aggirando di fatto l'isolamento software.

Virtual Machine Hopping

Il Virtual Machine Hopping è una tecnica che permette a un attaccante, dopo aver compromesso una macchina virtuale, di spostarsi lateralmente verso altri sistemi sullo stesso host o nella stessa rete virtuale. Questo attacco non si basa su un singolo exploit, ma sfrutta tipicamente configurazioni errate oppure vulnerabilità presenti nella gestione di risorse condivise.

Man in the Middle durante la Live Migration

Gli attacchi Man in the Middle durante la Live Migration sfruttano il trasferimento a caldo di una macchina virtuale tra host fisici. Se i dati di migrazione, come il contenuto della memoria RAM o del disco virtuale, non sono adeguatamente protetti utilizzando la crittografia, un aggressore li può intercettare e manipolare per installare una backdoor nel sistema trasferito.

Denial of Service dell'hypervisor

Il Denial of Service dell'hypervisor è una tipologia di attacco che ha come obiettivo quello di rendere indisponibile l'hypervisor e, di conseguenza, tutte le macchine virtuali che esso gestisce. Un aggressore, tramite un sistema già compromesso, sovraccarica l'hypervisor di richieste anomale, causando un consumo eccessivo di risorse fino a generare un blocco dell'intero sistema vir-

tualizzato. Anche il sistema host, pertanto, può diventare instabile durante l'attacco.

4.3 Attacchi ai container

Per quanto riguarda gli ambienti containerizzati, come Docker, gli attaccanti possono tentare di compromettere l'isolamento fornito dai container oppure sfruttare componenti dell'ecosistema, come i registri di immagini. Sebbene i container condividano il kernel dell'host e siano isolati tramite meccanismi come namespace e cgroup, vulnerabilità nel kernel o configurazioni errate possono essere sfruttate per superare questi meccanismi di sicurezza. In altri casi è sufficiente una gestione impropria dei privilegi, come quando gli amministratori li abilitano per eseguire software legacy all'interno del container, per permettere a un attaccante di violarne l'isolamento. Un classico esempio è rappresentato dai container a cui vengono assegnate capability Linux eccessive, come la SYS_ADMIN. Se un container con questi privilegi dovesse essere compromesso, l'attaccante potrebbe montare file system e modificare impostazioni del kernel, ottenendo un controllo simile ad un utente root sull'host.

Alcuni attacchi ai container sono simili a quelli già descritti per le macchine virtuali, pertanto, in questa sezione, ci focalizzeremo sulle tecniche non ancora spiegate.

Attacco tramite Docker.sock

L'Attacco tramite Docker.sock sfrutta il socket Unix /var/run/docker.sock, l'interfaccia principale tramite cui il demone Docker accetta comandi API. Se il socket viene accidentalmente condiviso all'interno di un container o esposto in rete senza adeguata autenticazione, un aggressore può sfruttarlo per assumere il controllo completo del sistema host. Pertanto, l'accesso a questo socket equivale a tutti gli effetti a una backdoor di root, poiché permette di eseguire comandi arbitrari, creare container privilegiati o montare volumi di sistema sull'host, compromettendo così l'intera infrastruttura.

Supply Chain Attack

Il Supply Chain Attack è una nuova tecnica che consiste nella diffusione di immagini Docker modificate contenenti malware o backdoor, tali immagini vengono pubblicate su registri pubblici con nomi ingannevoli in modo tale che quando un utente le scarica e avvia, compromette il proprio sistema. Il codice malevolo può svolgere attività come mining di criptovalute o furto di

credenziali. In alcuni casi, se il container ha privilegi elevati, l'attacco può infettare anche l'host.

Attacco all'orchestratore

Gli orchestratori di container, come Kubernetes, rappresentano un obiettivo privilegiato per gli attaccanti poiché gestiscono l'intero ciclo di vita dei container. Una compromissione del control plane, ovvero il cuore dell'orchestratore, consente all'aggressore di pianificare l'esecuzione di container malevoli, accedere a dati sensibili e muoversi lateralmente nel cluster, fino a compromettere l'intero ambiente. Gli attacchi all'orchestratore, pertanto, sfruttano solitamente vulnerabilità nei componenti centrali, come il server API o il database etcd, oppure configurazioni errate.

4.4 Attacchi negli ambienti cloud

L'ultima categoria di attacchi che verranno trattati riguardano gli ambienti cloud, dove più tenant condividono risorse gestite da provider come AWS o Azure. In questo ambiente emergono attacchi specifici legati alla multi tenancy, alla gestione remota e alla configurazione dei servizi.

Oltre alle minacce già trattate per macchine virtuali e container, il cloud introduce altre nuove superfici di attacco come le API di gestione, lo storage condiviso e i servizi PaaS e SaaS.

Man in the Cloud

Nei servizi cloud Software as a Service, come Google Drive e OneDrive, gli attaccanti possono mirare direttamente agli account utente sfruttando la fiducia tra client e servizio cloud. Un esempio noto è l'attacco Man in the Cloud che consente di ottenere l'accesso a un account di cloud storage senza dover utilizzare malware o violare le credenziali dell'utente. I servizi SaaS usano token di sincronizzazione salvati sul dispositivo per mantenere attiva la connessione all'account. Se un attaccante riesce a intercettare questo token, ad esempio tramite uno script eseguito sfruttando social engineering, può inserirlo su un proprio dispositivo ed impersonificare il client legittimo, ottenendo così la capacità di visualizzare o scaricare tutti i file sincronizzati e persino di caricare file dannosi che verranno successivamente sincronizzati in maniera automatica sui dispositivi della vittima. Essendo un attacco che non compromette il server e non richiede le credenziali dell'utente, risulta difficile da rilevare.

Attacco Server Side Request Forgery al Metadata Service

Infine, un attacco Server Side Request Forgery si verifica quando un'applicazione web vulnerabile viene indotta a effettuare richieste HTTP verso indirizzi scelti dall'attaccante. Nei sistemi cloud, ogni macchina ha accesso a un servizio interno chiamato metadata service, questo servizio fornisce credenziali temporanee per l'accesso alle risorse cloud. Se l'attaccante riesce a sfruttare l'SSRF per contattare questo servizio, ad esempio http://169.254.169.254 su AWS, può ottenere le credenziali valide per accedere alle risorse del cloud.

Capitolo 5

Hardening, Contromisure e Isolamento in ambienti virtuali

5.1 Introduzione

In questo capitolo, a complemento del precedente sugli attacchi, vengono analizzate le tecniche di hardening, contromisure difensive e le pratiche di isolamento in ambienti formati da macchine virtuali, container e in ambienti cloud come AWS e Azure.

5.2 Tecniche di hardening e contromisure per macchine virtuali

La sicurezza delle macchine virtuali è un processo che comporta sia il rafforzamento interno dell'ambiente virtuale, sia la protezione dell'hypervisor e del sistema host.

Configurazione e aggiornamento delle macchine virtuali

Una delle tecniche fondamentali di hardening per ridurre la superficie d'attacco delle macchine virtuali, consiste nell'adottare una configurazione sicura fin dalla fase di creazione. È importante disabilitare tutti i servizi non necessari, chiudere le porte di rete non utilizzate e limitare le funzionalità esposte all'esterno. Inoltre, per aumentare la robustezza, è fondamentale definire e applicare una rigida policy di sicurezza, mantenere aggiornati sia il sistema operativo guest e le applicazioni installate, sia l'hypervisor e i driver di virtualizzazione, applicando le patch di sicurezza appena disponibili.

Contromisure per il VM Escape

Le contromisure per contrastare gli attacchi più comuni verso gli ambienti virtualizzati, hanno tutte come punto di partenza l'impiego delle tecniche di hardening già discusse. Nel caso di attacchi di tipo VM Escape, l'obiettivo è rinforzare il perimetro di isolamento che separa le macchine virtuali dall'hypervisor e dal sistema host. Questo può essere fatto eseguendo le macchine virtuali ed i suoi processi con il minor livello di privilegi possibile oppure utilizzando moduli di sicurezza come sVirt in ambienti KVM. sVirt consiste nell'assegnare etichette distinte, tramite SELinux, a ogni macchina virtuale, in modo tale che, se una di queste dovesse essere violata, le policy di controllo degli accessi le impedirebbero di uscire al di fuori del proprio ambiente.

Contromisure per l'Hyperjacking

L'attacco di tipo Hyperjacking consiste nel prendere il controllo dell'hypervisor, spesso tramite l'installazione di un bootkit malevolo. Per contrastare questa minaccia vengono utilizzate tecnologie come Secure Boot e Trusted Platform Module. Secure Boot è una funzionalità del firmware UEFI che garantisce che, durante il processo di avvio, vengano eseguiti solo componenti autorizzati e firmati digitalmente. Il TPM, invece, è un chip che memorizza i valori hash delle componenti critiche del sistema in modo tale che, durante l'avvio, questi hash vengano confrontati con quelli calcolati in tempo reale per rilevare le manomissioni come la presenza di un bootkit.

Contromisure per il Side-Channel

Gli attacchi side-channel hanno come obiettivo ottenere informazioni sensibili osservando il comportamento delle risorse hardware, come memorie e processore, condivise tra macchine virtuali. Per contrastare questi attacchi vengono utilizzate diverse contromisure sia a livello software che hardware. Dal lato software, i sistemi operativi e gli hypervisor implementano tecniche di isolamento della memoria e svuotamento delle cache durante il cambio di contesto delle VM, riducendo così il rischio che dati residui possano essere letti da processi non autorizzati. Dal lato hardware viene implementata la virtualizzazione confidenziale, ovvero una tecnica presente nelle CPU moderne con il nome di AMD SEV e Intel TDX, che consente la cifratura della memoria in modo tale da garantire che ogni macchina virtuale possa accedere solamente al proprio spazio di indirizzamento, anche nel caso in cui l'hypervisor sia stato compromesso.

Contromisure al Denial of Service ed effetto Noisy-Neighbor

In ambienti virtualizzati, una macchina virtuale mal configurata oppure compromessa che esegue un attacco Denial of Service, potrebbe monopolizzare l'hardware del sistema host a causa del consumo eccessivo di risorse fisiche, rallentando le prestazioni degli altri sistemi e dello stesso host. Questo fenomeno è noto come effetto noisy neighbor. Per evitare che questo accada, negli ambienti Linux, si utilizzano i cgroups, una funzionalità del kernel Linux che permette di organizzare un insieme di processi in gruppi gerarchici, consentendo di monitorarli e limitarne l'utilizzo delle risorse hardware. Infine è importante prevenire il fenomeno del VM sprawl, ovvero la proliferazione di macchine virtuali inutilizzate, in quanto potrebbero essere utilizzate per effettuare degli attacchi di tipo flooding di rete.

Monitoraggio contro le intrusioni

Infine, per aumentare la sicurezza dell'ambiente virtualizzato, oltre alle tecniche di hardening, vengono utilizzati strumenti di monitoraggio e rilevamento delle intrusioni a livello di hypervisor, come per esempio VMware vShield e Trend Micro Deep Security. Entrambi gli strumenti possono funzionare sia in modalità Intrusion Detection System, rilevando attività sospette all'interno delle macchine virtuali e generando allarmi, sia in modalità Intrusion Prevention System, in cui oltre a rilevare le minacce, possono anche intervenire per bloccarle, ad esempio terminando una connessione o mettendo in quarantena un indirizzo IP sospetto. Infine è importante consultare periodicamente i file di log dell'hypervisor dove vengono registrati eventi critici come accessi in memoria non autorizzati, eccezioni ed altre forme di anomalie.

5.3 Tecniche di hardening e contromisure per container

In questa sezione vengono spiegate le tecniche di hardening per container non ancora affrontate nella sezione precedente.

Sicurezza delle immagini

La sicurezza dei container è un processo che inizia dalla scelta dell'immagine e si estende per tutto il ciclo di vita in cui è in esecuzione. Le immagini scelte, per ridurre la superficie d'attacco, devono contenere solo le componenti necesarie all'applicazione, come per esempio le immagini Distroless e Alpine. Le immagini Alpine contengono la parte minima indispensabile del sistema

operativo Linux a livello di spazio utente, sono quindi formate da una shell leggera, un gestore di pacchetti .apk e le librerie essenziali come musl libc, ovvero una implementazione della libreria C standard per il sistema operativo Linux. Le immagini Distroless sono ancora più estreme, contengono solamente l'applicazione e le sue dipendenze di runtime, senza alcuna shell, nessun gestore di pacchetti, e nessun componente di un sistema operativo tradizionale a livello di spazio utente. Le immagini scelte per essere utilizzate devono provenire da registri ufficiali come Docker Hub e Google Container Registry, devono essere firmate digitalmente e aggiornate regolarmente. Un'altra pratica utilizzata per rendere sicure le immagini consiste nel automatizzare la scansione delle vulnerabilità durante le fasi di Continuous Integration e Continuous Delivery ovvero le fasi in cui il codice viene aggiornato e rilasciato, in modo da distrubuire sempre immagini sicure.

Gestione dei privilegi e meccanismi di sicurezza

Un'altro aspetto fondamentale per la sicurezza dei container è l'applicazione del principio del minimo privilegio che impone di concedere a ogni processo solo le autorizzazioni strettamente necessarie per la sua funzione, per quanto riguarda i container questo significa evitare di eseguire i processi all'interno del container come utente root. Per poterlo fare viene impiegata la funzionalità user namespace, tale funzionalità consente di mappare l'utente root interno del container a un User ID non privilegiato sul sistema host, in questo modo, anche in caso di evasione dal container, un attaccante non riuscirebbe ad ottenere i privilegi di root sull'host. Un'ulteriore accorgimento consiste nel consentire ai processi in esecuzione nel container, l'accesso di sola lettura al filesystem, al fine di impedire modifiche non autorizzate.

Un altro meccanismo di sicurezza è il Secure Computing Mode, una funzionalità del kernel Linux che consente di limitare le chiamate di sistema che un attaccante potrebbe invocare. Docker applica di default un profilo seccomp restrittivo che blocca molte chiamate non necessarie e potenzialmente dannose, come il caricamento di moduli del kernel oppure il montaggio di filesystem, tale profilo può essere ulteriormente personalizzato analizzando il comportamento reale del container e bloccando tutte le syscall superflue.

Contromisure per il Breakout

In ambito container l'attacco analogo al VM escape prende il nome Breakout, consiste in un processo in esecuzione all'interno dei container che riesce ad ottenere l'accesso al sistema host oppure ad altri container. Poiché i container condividono il kernel, solitamente il breakout avviene sfruttando

vulnerabilità del kernel Linux oppure a causa di configurazioni errate, come per esempio l'esucuzione di un container in modalità privilegiata. Per prevenire il breakout è necessario mantenere il sistema host sempre aggiornato ed evitare di eseguire container con privilegi elevati.

Ulteriori strati di sicurezza vengono aggiunti mediante l'utilizzo di strumenti già visti, come seccomp per limitare le chiamate di sistema accessibili, SELinux che utilizza l'assegnazione di etichette per stabilire quali interazioni tra risorse sono concesse ed infine AppArmor che si basa sul controllo del percorso, in questo caso le policy definiscono cosa un'applicazione può fare in base al suo percorso sul filesystem.

Contromisure per il Supply-Chain Attack

Un'altra tipologia di attacco è il supply-chain attack e riguarda l'inserimento di codice malevolo nelle immagini dei container oppure l'uso di immagini rese volutamente vulnerabili tramite l'inserimento di backdoor. Una contromisura efficace per questo attacco prevede l'impiego di scansioni delle immagini tramite strumenti come Trivy, per rilevare malware, backdoor e altre forme di vulnerabilità. In ambienti orchestrati come Kubernetes, strumenti come Gatekeeper e Kyverno vengono utilizzati per definire policy di ammissione per impedire l'esecuzione di immagini che non rispettano determinati criteri di sicurezza, come per esempio immagini non firmate o provenienti da registri non ufficiali. Infine, in contesti ad alta sicurezza come aziende o in presenza infrastrutture critiche, prima di eseguire un'immagine in produzione, è possibile testarla in ambienti isolati chiamati sandbox, in modo tale da poterne monitorare il comportamento alla ricerca di attività sospette causate dalla presenza di malware dormienti. Questi malware solitamente sfuggono alle scansioni tradizionali.

5.3.1 Sandbox e isolamento

In ambienti in cui vengono utilizzati i container, spesso vengono impiegati degli strumenti per aumentarne l'isolamento e monitorarne la sicurezza. Un esempio è gVisor, un container sandbox sviluppato da Google che agisce come un kernel user space. Il suo compito è intercettare le chiamate di sistema dai container ed eseguirle in un kernel applicativo separato ed isolato dal kernel del sistema host, tuttavia gVisor non supporta tutte le chiamate di sistema, ma offre comunque una soluzione isolata senza dover ricorrere all'uso di macchine virtuali.

Un altro esempio è Kata Containers, un progetto open source che integra container con microVM. Il vantaggio di Kata è quello di fornire un isolamento

pressoché totale, in quanto, invece di eseguire un container direttamente sul kernel host, viene avviata una versione leggera di macchina virtuale con un proprio kernel per ciascun container, fornendo un isolamento equivalente a quello di una VM e, contemporaneamnte, cercando di mantenere la velocità e praticità di un container grazie al ridotto consumo di risorse. Questa tecnologia viene attualmente utilizzata in Kubernetes, dove ogni pod viene eseguito dentro una macchina virtuale KVM dedicata, isolandola completamente dal host e dagli altri container.

5.4 Tecniche di hardening e contromisure negli ambienti cloud

Negli ambienti cloud pubblici, la virtualizzazione è la tecnica che viene utilizzata per offrire risorse condivise a molteplici clienti, per questo motivo le misure di sicurezza e hardening riguardano sia gli aspetti che gestisce il cloud provider, sia quelli per cui è responsabile l'utente finale. In particolare, il provider cloud è responsabile della sicurezza dell'infrastruttura fisica sottostante come memoria, processore e rete, mentre il cliente è responsabile della sicurezza delle proprie applicazioni e dei propri dati nel cloud. Per quanto riguarda le tecniche di hardening, vengono implementate direttamente dal provider, che quindi si occupa di tenere aggiornati i propri hypervisor, firmware, garantendo l'isolamento tra gli ambienti. Alcune aziende come AWS, garantiscono la sicurezza dei propri servizi mediante l'utilizzo di una architettura evoluta di virtualizzazione, come l'AWS Nitro System, dove gran parte delle funzionalità, come la gestione delle reti e della memoria, vengono spostate su hardware dedicato chiamato Nitro Card, pertanto l'hypervisor viene semplificato e ridotto ai minimi termini. L'hypervisor di Nitro System è di tipo paravirtuale, ovvero la comunicazione tra gli ambienti virtuali e la parte hardware, rappresentata dalle Nitro Card, avviene mediante l'utilizzo di driver paravirtualizzati installati nei sistemi guest, pertanto l'hypervisor risulta essere più semplice, veloce e con una superficie d'attacco minimizzata, delegando la gestione complessa dell'I/O direttamente all'hardware. Un altro vantaggio di Nitro è l'adozione della politica zero trust interna, ovvero neanche gli operatori AWS hanno accesso ai sistemi dei loro clienti, eliminando ancor più i rischi legati a intrusioni.

Anche altri provider utilizzano architetture simili, Google Cloud utilizza un hypervisor basato su KVM con ottimizzazioni proprie e una componente hardware chiamata Titan chip che garantisce l'integrità del processo di avvio dei propri server.

Anche Azure di Microsoft utilizza un hypervisor evoluto chiamato Hyper-V, impiega anche una suite di funzionalità chiamata Azure Confidential Computing che garantisce la protezione della memoria in uso dalle VM tramite la cifratura, pertanto neanche tramite la violazione dell'hypervisor sarà possibile leggere la memoria.

Infine, anche l'accesso alle API di gestione, come quelle per la creazione di macchine virtuali o la configurazione della rete Virtual Private Cloud, è protetto tramite autenticazione multifattore e protocolli di cifratura come HTTPS e TLS, che garantiscono la riservatezza e l'integrità dei dati scambiati.

5.4.1 Hardening a livello di tenant negli ambienti cloud

L'hardening viene fatto anche a livello di tenant attraverso la gestione dei permessi di accesso alle risorse assegnate, questa configurazione è definita tramite la stesura di documenti, spesso in formato JSON, che prendono il nome di Identity and Access Management policies. Ogni provider cloud offre un proprio sistema IAM per gestire in modo granulare i permessi di utenti, servizi e applicazioni. L'assegnazione di tali permessi, da parte degli amministratori, deve essere fatta seguendo rigorosamente il principio del minimo privilegio, in questo modo, anche se un attaccante riuscisse a compromettere un sistema in cloud e ad estrarne le credenziali IAM, il danno sarebbe limitato.

Nei sistemi AWS, la gestione dei permessi degli utenti viene fatta tramite AWS IAM, mentre attraverso AWS Organizations è possibile gestire centralmente più account AWS e applicare policy di controllo a livello di organizzazione.

Infine, è importante sottolineare che, oltre alle responsabilità del cliente, i provider cloud hanno un forte interesse a prevenire attacchi a livello infrastrutturale. Un singolo incidente su larga scala, infatti, comprometterebbe gravemente la fiducia di molti clienti, pilastro fondamentale del modello cloud.

Capitolo 6

Caso studio: l'attacco ransomware ESXiArgs

6.1 Introduzione

Nei precedenti capitoli sono stati trattati gli attacchi e le contromisure che riguardano i sistemi virtualizzati, lo scopo di questo capitolo è quello di analizzare nel dettaglio l'attacco ESXiArgs, un ransomware che nel mese di febbraio del 2023 ha coinvolto più di 3800 sistemi[1] VMware in tutto il mondo.

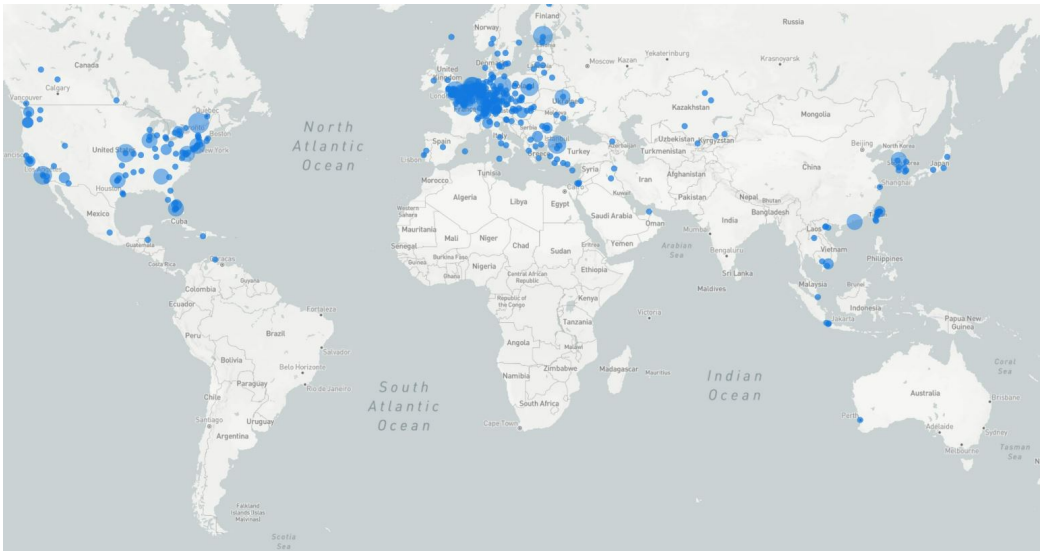

Figura 6.1: Mappa dei paesi colpiti da ESXiArgs. Fonte

[1]Fonte: www.cisa.gov

6.2 Descrizione tecnica dell'attacco

ESXiArgs è un attacco basato su ransomware che ha avuto come obiettivo le macchine virtuali gestite da VMware ESXi, uno degli hypervisor bare-metal più utilizzati nella virtualizzazione. Il ransomware sfruttava due gravi vulnerabilità identificate come CVE-2021-21974[2], e CVE-2020-3992[3]. Entrambe sono state individuate nel modulo openSLP di VMware ESXi, un protocollo che consente a strumenti di gestione come vCenter Server, di rilevare automaticamente gli host VMware ESXi presenti nella stessa rete. Tali vulnerabilità hanno permesso l'invio di pacchetti SLP appositamente modificati alla porta di servizio 427, con lo scopo di generare heap buffer overflow ed eseguire l'installazione del ransomware che ha causato la cifratura dei dati sui server.

Il ransomware, una volta avuto accesso alla macchina bersaglio, ha creato una cartella /tmp con all'interno i seguenti file:

- **Encrypt:** si tratta di un eseguibile binario in formato ELF responsabile della crittografia dei dati, rappresenta il cuore del ransomware.
- **Encrypt.sh:** è uno script bash che prepara l'ambiente per la crittografia, si occupa di arrestare i servizi VMware e di individuare i volumi e i file da crittografare. Vengono cifrati i file con le seguenti estensioni: .vmdk, .vmx, .vmxf, .vmsd, vmsn, .vswp, .vmss, .nvram e .vmem.
- **Public.pem:** è la chiave pubblica RSA utilizzata per la crittografia assimmetrica dei dati.
- **Motd:** è un file di testo contenente la nota di riscatto, inserito nel percorso /etc/motd al posto del file originale. Viene mostrato automaticamente ogni volta che un utente si autentica per accedere al server tramite SSH.
- **Index.html**: Anche in questo caso si tratta della nota di riscatto, il documento sostituisce quello originale in modo da essere visibile nella home page dell'interfaccia web di VMware ESXi.

Per ogni file individuato, lo script encrypt.sh generava un file con estensione .args contenente i parametri necessari per la cifratura, salvandolo nella stessa directory. Successivamente, il file .args veniva utilizzato dall'eseguibile encrypt per avviare il processo di cifratura del file corrispondente.

Il nome del ransomware ESXiArgs deriva proprio dall'utilizzo dei file con estensione .args.

[2]Fonte: nvd.nist.gov
[3]Fonte: nvd.nist.gov

```
for volume in $(IFS='\n' esxcli storage filesystem list | grep "/vmfs/volumes/" | awk -F'  '
'{print $2}'); do
  echo "START VOLUME: $volume"
  IFS=$'\n'
  for file_e in $( find "/vmfs/volumes/$volume/" -type f -name "*.vmdk" -o -name "*.vmx" -o
  -name "*.vmxf" -o -name "*.vmsd" -o -name "*.vmsn" -o -name "*.vswp" -o -name "*.vmss" -o
  -name "*.nvram" -o -name "*.vmem"); do
      if [[ -f "$file_e" ]]; then
        size_kb=$(du -k $file_e | awk '{print $1}')
        if [[ $size_kb -eq 0 ]]; then
          size_kb=1
        fi
        size_step=0
        if [[ $(($size_kb/1024)) -gt 128 ]]; then
          size_step=$((($size_kb/1024/100)-1))
        fi
        echo "START ENCRYPT: $file_e SIZE: $size_kb STEP SIZE: $size_step" "\"$file_e\"
        $size_step 1 $((size_kb*1024))"
        echo $size_step 1 $((size_kb*1024)) > "$file_e.args"
        nohup $CLEAN_DIR/encrypt $CLEAN_DIR/public.pem "$file_e" $size_step 1 $((size_kb*1024))
        >/dev/null 2>&1&
      fi
  done
  IFS=$" "
done
```

Figura 6.2: Script per generare i file .args e cifrare i file sul server. Fonte

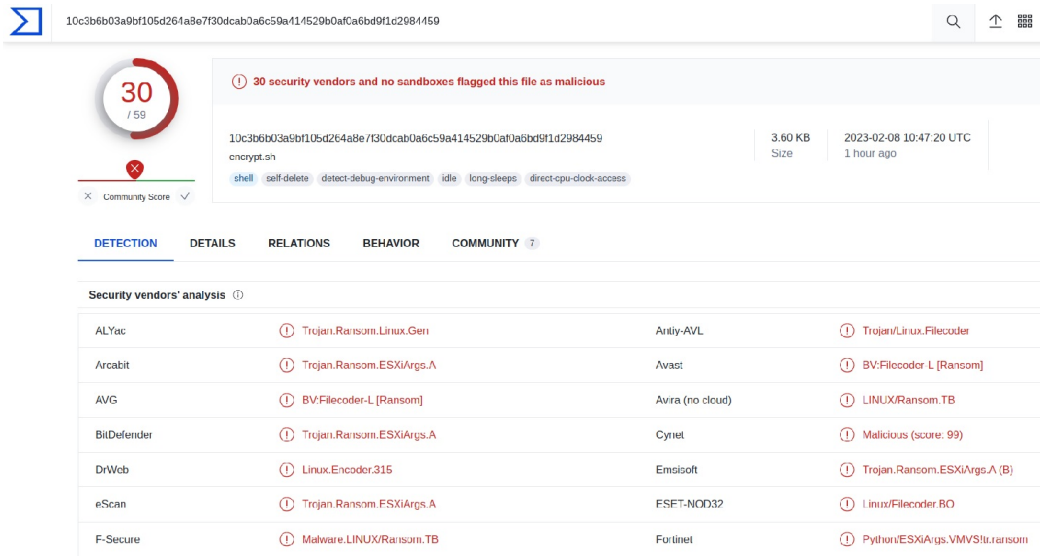

Figura 6.3: Analisi di VirusTotal relativa al file encrypt.sh. Fonte

6.3 Nota di riscatto

Per ogni sistema violato i criminali lasciavano una nota di riscatto contenente le istruzioni su come effettuare il pagamento per riavere accesso ai propri file. In ogni nota era riportato un wallet bitcoin creato appositamente per ogni vittima, e un TOX ID con il quale era possibile comunicare con gli attaccanti

attraverso la chat TOX, un servizio di messaggistica anonima. La somma da pagare variava da vittima a vittima ma si attestava sempre sopra i 40 mila euro in Bitcoin[4]

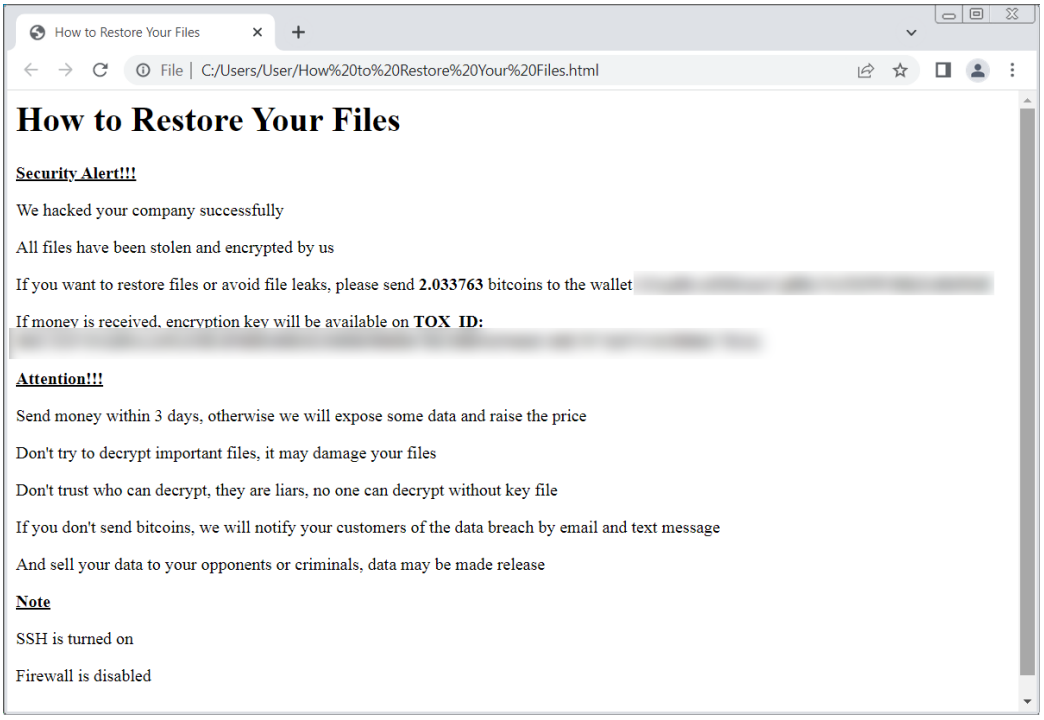

Figura 6.4: Nota di riscatto di ESXiArgs. Fonte

L'elenco degli indirizzi di pagamento[5] e la lista delle macchine coinvolte[6] dal attacco, sono consultabili tramite la piattaforma Github grazie al lavoro svolto da Ransomwhere, un progetto open source che traccia i pagamenti effettuati ai ransomware utilizzando dati pubblici delle blockchain[7].

6.4 Evoluzione di ESXiArgs e strumenti di recupero

Nel corso del tempo il ransomware ESXiArgs è stato raffinato, aumentandone l'efficacia e rendendo sempre più difficile decifrare i dati senza pagare il

[4]Fonte: www.therecord.media
[5]Fonte: github.com
[6]Fonte: github.com
[7]Fonte: www.bankInfoSecurity.com

riscatto. Nella sua prima variante la cifratura dei file di grandi dimensioni era parziale: i file venivano suddivisi in blocchi da 1 MB e la cifratura avveniva a blocchi alterni, seguendo una tecnica chiamata skipping. Questo metodo lasciava porzioni di dati in chiaro che potevano essere utilizzate per tentare il recupero dei file senza possedere la chiave privata.

A tal proposito, la CISA, in collaborazione con l'FBI, pubblicò su GitHub[8] uno script open source chiamato ESXiArgs Recover, progettato per aiutare le vittime a ricostruire i file parzialmente cifrati dalla prima versione dell'attacco, questo strumento si basava proprio sulla presenza dei blocchi di dati non cifrati, residui della tecnica di skipping.

Nelle varianti successive gli attaccanti hanno modificato lo script rimuovendo la tecnica dello skipping, rendendo di fatto quasi impossibile il ripristino dei dati senza l'utilizzo di copie di backup.

6.5 Obiettivi, Autori e Motivazioni

L'attacco ransomware ha coinvolto le organizzazioni che utilizzavano versioni di VMware ESXi precedenti alla 7.0 in quanto prive delle patch di sicurezza rilasciate nei due anni precedenti. Tra i soggetti più colpiti troviamo strutture con grandi database di informazioni riservate come agenzie governative, banche e istituzioni educative[9]. L'attacco ha preso di mira soprattutto territori europei, i primi paesi ad essere stati colpiti sono stati la Francia e la Germania.

Per quanto riguarda l'identità degli attori dietro ESXiArgs, nonostante inizialmente siano state fatte diverse ipotesi, non è mai stata confermata pubblicamente, l'unico aspetto da considerare è una certa somiglianza con il codice di Babuk, un ransomware che nel 2021 ha colpito i sistemi windows e linux di piu di 42 organizzazioni mondiali[10]. Entrambi i ransomware utilizzavano lo stesso cifrario, Sosemanuk[11].

Ad oggi non emergono elementi che facciano pensare ad un coinvolgimento da parte di stati o organizzazioni politiche, al contrario, le motivazioni dell'attacco appaiono prettamente finanziarie; l'obiettivo è stato estorcere denaro alle vittime in cambio dei loro dati, sfruttando una finestra di vulnerabilità su scala globale.

[8]Fonte: github.com
[9]Fonte: www.sisainfosec.com
[10]Fonte: cyble.com
[11]Fonte: www.vmware.com

Capitolo 7

Conclusione e Considerazioni Finali

Nel capitolo precedente è stato trattato uno dei più grandi attacchi ransomware rivolti ai sistemi VMware ESXi, portando all'attenzione globale l'importanza di proteggere adeguatamente le infrastrutture di virtualizzazione.

L'impatto è stato elevato perché ha causato il blocco di numerosi server, decine di migliaia di servizi e applicazioni aziendali sono andati fuori uso. Persino la Corte Suprema della Florida[1] si è trovata con i server ESXi cifrati, a dimostrazione che nessun ambiente esposto resta immune per sempre.

Questo attacco deve essere visto come un campanello d'allarme per tutte le organizzazioni che utilizzano infrastrutture virtuali ma che non implementano un controllo rigoroso della loro sicurezza. Gli amministratori spesso ritardano l'applicazione degli aggiornamenti per non interrompere il flusso di lavoro, ignorando il fatto che attacchi come questo potevano essere evitati se ci fosse stato un'applicazione tempestiva delle patch.

Pertanto, investire in un adeguato processo di gestione degli aggiornamenti di sicurezza è fondamentale, ma non sempre basta, in generale andrebbero disabilitati tutti i servizi non indispensabili per limitare l'esposizione in rete. In tanti casi gli host colpiti erano collegati direttamente alla rete pubblica quando invece è più sicuro che l'interfaccia di controllo sia protetta da VPN, firewall oppure altri sistemi di sicurezza come i jump host. Per maggiore protezione anche i servizi di discovery, come openSLP, andrebbero sempre configurati in modo da non oltrepassare la rete locale, riducendo così il rischio che possano essere sfruttati per eseguire un attacco.

In conclusione, per ridurre la superficie d'attacco bisogna essere sempre vigili, adottare le tecniche di hardening descritte nei precedenti capito-

[1]Fonte: www.reuters.com

li, e assicurarsi sempre di disporre di copie di backup aggiornate per poter ripristinare i sistemi nel caso in cui vengano compromessi.