Sicurezza nel Cloud Computing: Sfide e Prospettive

REPORT by: Fabio Gentili

Andrea Isernio

Dai primi anni del 2000 ad oggi la tecnologia del cloud computing è stata adottata da sempre più aziende e organizzazioni in tutto il mondo e questa tendenza non sembra destinata a invertirsi [Fig. 1]. Nonostante gli evidenti vantaggi pratici ed economici apportati dal cloud, non si possono non sottolineare alcuni aspetti negativi legati alla sicurezza e privacy dei dati e dei processi che eseguono sui server remoti. In questo report descriveremo alcuni di questi problemi ed analizzeremo le motivazioni e gli approcci dei cosiddetti self-hosters, ovvero di quegli utenti che hanno deciso di installare e rendere accessibili i propri software direttamente su macchine di loro proprietà.

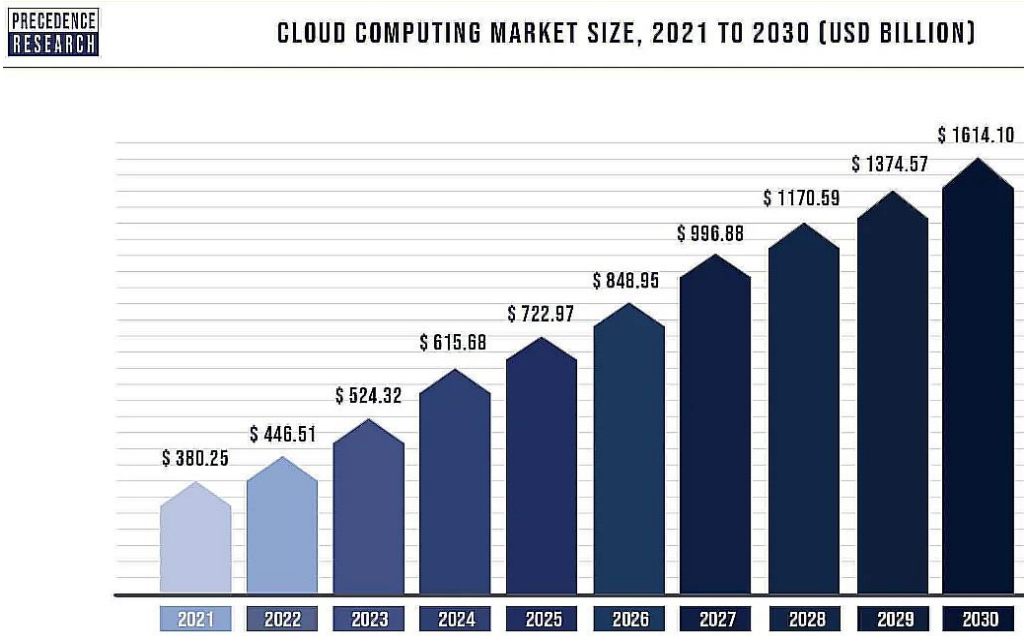

Fig. 1: predizione del market size del cloud computing in miliardi di dollari (https://www.cloudzero.com/blog/cloud-computing-market-size/)

LO STATO DELL' ARTE

"The year is 2023 A.D. The Internet is entirely occupied by commercial cloud services. Well, not entirely... One small minority of indomitable self-hosters still holds out against the invaders."

Per cloud computing si intende la messa a disposizione di risorse di calcolo e di storage on-demand, senza il bisogno di gestione attiva dell'infrastruttura da parte dell'utente. I principali vantaggi dovuti all'adozione del cloud includono:

Riduzione dei costi: dato che l'infrastruttura hardware è completamente fornita dai cloud provider, gli utenti pagano solo per la quantità di risorse di cui hanno bisogno. Ciò implica un abbassamento del costo iniziale di investimento;

Scalabilità e flessibilità: i cloud provider offrono la possibilità di scalare in alto o in basso le risorse a disposizione in base alle esigenze e ai carichi real-time dei servizi ospitati sulle proprie macchine. Ciò può essere svolto in maniera automatica, senza una pre-ingegnerizzazione dei servizi da parte degli utenti per far fronte a eventuali peak loads;

Affidabilità e disaster recovery: I servizi cloud spesso includono capacità di ridondanza e di disaster recovery integrate, fornendo un alto livello di affidabilità e disponibilità;

Monitoraggio continuo: i cloud provider forniscono strumenti e parametri di monitoraggio real-time dei servizi ospitati; quindi, gli utenti non si devono preoccupare di doverli inserire autonomamente all'interno dei propri software;

Distribuzione globale: Il cloud computing consente un facile dispiegamento di applicazioni in molteplici regioni in tutto il mondo, consentendo alle organizzazioni di fornire una migliore esperienza ai loro clienti con una latenza minore e con costi minimi.

Nonostante questi enormi vantaggi, il cloud computing presenta alcuni problemi di sicurezza e privacy dovuti alla sua architettura centralizzata in pochi data center di grosse dimensioni sparsi per il mondo. Infatti, i dati dei clienti che decidono di usufruire dei servizi ospitati su cloud sono interamente conservati nei singoli data center, rendendo questi luoghi i target ideali di criminali informatici. Lo stesso Tim Berners-Lee ha criticato questa centralizzazione di Internet da parte di pochi cloud provider causata dalla creazione dei cosiddetti "data silos".

Inoltre, gli utenti e le aziende che decidono di fornire i propri dati ai servizi cloud devono essere consapevoli del fatto che tali informazioni potrebbero essere sfruttate dai provider per secondi fini non autorizzati, danneggiando gravemente la loro privacy.

Nel prossimo paragrafo discuteremo delle vulnerabilità legate alle distro del sistema operativo Linux, dato che quest'ultimo è utilizzato da circa il 90% dei server messi a disposizione dai cloud provider, focalizzando l'attenzione sull'exploit dei programmi eBPF.

LINUX THREATS

Come analizzato in *"The Linux threat landscape report"*, questo Sistema operativo è tutt'altro che immune ai malware, a differenza di ciò che si pensa comunemente. Le principali tipologie sono:

- **Ransomware**: l'anno scorso i tentativi di attacco di questo tipo sono aumentati del 62%. Un noto esempio è il KillDisk malware. In breve, il suo scopo è quello di sovrascrivere il Master Boot Record dei vari hard disk disponibili nel sistema attaccato e in seguito forzare uno shutdown in modo da non rendere più accessibile il contenuto di tali periferiche;

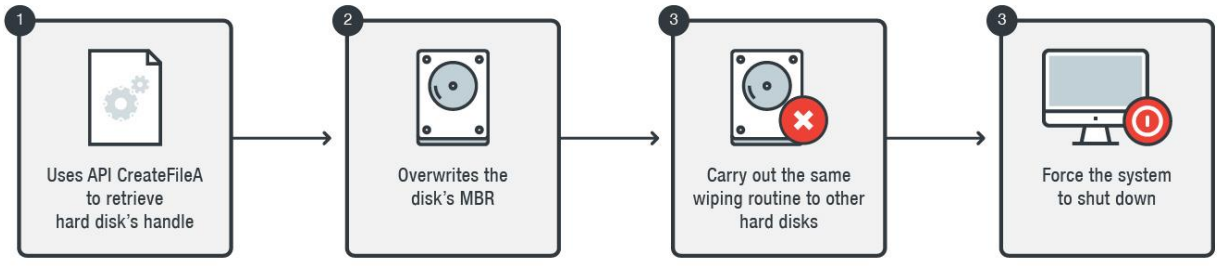

Fig. 2: MBR-wiping routine (https://www.trendmicro.com/en_us/research/18/f/new-killdisk-variant-hits-latin-american-financial-organizations-again.html)

- **Criptocurrency miners**: sfruttano le stesse vulnerabilità dei ransomware. Uno degli esempi più recenti è rappresentato da un XMRig miner che abusa delle difese assenti di alcune applicazioni Linux per infiltrarsi nei sistemi e auto-replicarsi modificando il file */etc/crontab*;
- **Web shells**: sono script malevoli che vengono installati sui server e forniscono un'interfaccia per prenderne controllo. I threat actors hanno usato per anni la web shell China Chopper per accedere a server Linux compromessi. L'Hello ransomware rappresenta un'evoluzione di tale programma e si basa sull'installazione di un beacon sul server vulnerabile che permette, oltre a lanciare comandi bash, anche l'uploading di file, privilege escalation, key logging e tanto altro;

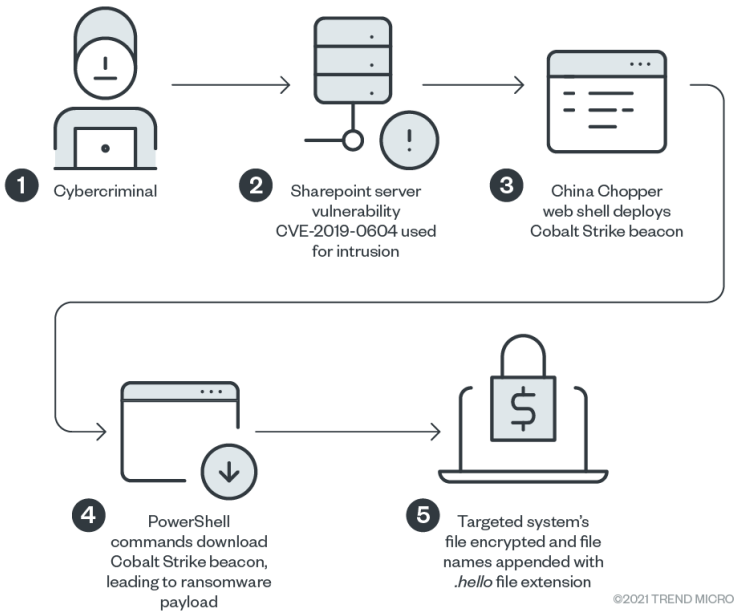

Fig. 3: Infection chain del ransomware Hello (https://www.trendmicro.com/en_us/research/21/d/hello-ransomware-uses-updated-china-chopper-web-shell-sharepoint-vulnerability.html)

- **Rootkits**: malware che spesso richiedono accesso a livello kernel e modificano system call e log di sistema per nascondere sé stessi e altri software malevoli.

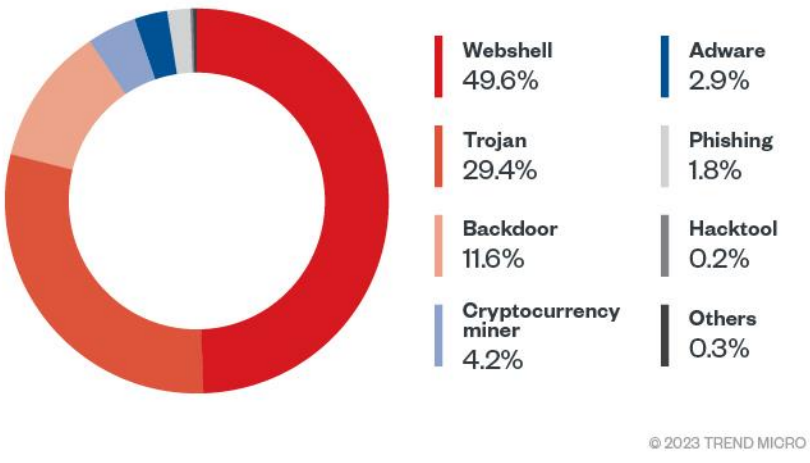

Fig. 4: Principali malware che hanno bersagliato sistemi Linux nel 2022

Spesso questi malware sfruttano vulnerabilità dovute a mancati aggiornamenti di sistema e/o configurazioni poco accurate, oltre che difetti nel codice delle web application. Per mitigare questi problemi è fondamentale seguire le good practice imposte dai principali framework di threat modeling.

Tornando alla questione del cloud computing, oggigiorno il deployment della maggior parte delle applicazioni cloud-native e a microservizi viene effettuato su container Linux-based. Per questo motivo è importante essere a conoscenza dei vari rischi per la sicurezza dei dati e dei processi utilizzati da tali ambienti virtuali. Come evidenziato nel report di Trend Micro, tutte e 10 le immagini ufficiali più scaricate da DockerHub contengono almeno una vulnerabilità considerata critica.

Vulnerability Severity Ratings					
Image	**Critical**	**High**	**Medium**	**Low**	**Total**
python	18	230	223	65	536
node	17	223	218	65	523
golang	10	98	65	11	184
wordpress	8	87	83	16	194
nginx	5	26	13	8	52
httpd	3	23	11	7	44
redis	3	15	4	9	31
mysql	2	18	13	1	34
postgres	1	24	8	8	41
memcached	1	14	2	7	24

Fig. 5: Le 10 immagini ufficiali di DockerHub più vulnerabili

Le principali pratiche di prevenzione delle vulnerabilità dei container sono:

- Usare immagini-base minimali: includendo all'interno dell'immagine solo lo strettamente necessario all'esecuzione delle applicazioni si riduce la superficie di attacco del container;
- Aggiornare spesso le immagini: il mancato aggiornamento delle immagini costituisce la prima causa della presenza di vulnerabilità;
- Seguire il principio di minimo privilegio: a default i container docker eseguono come root, quindi è necessario configurarli in modo che abbiano privilegi minori ove possibile.

Tali accorgimenti potrebbero non essere sufficienti nel caso in cui vengano sfruttate vulnerabilità delle applicazioni essenziali all'operatività dei propri servizi. In questo caso l'unico modo per

mantenere sicuri e isolati i container è quello di creare delle soluzioni ad hoc. Nel prossimo capitolo verrà descritta una tipologia specifica di attacco cross-container e verrà proposta una soluzione per mitigare il problema.

ATTACCHI CROSS-CONTAINER TRAMITE PROGRAMMI eBPF

L'extended Berkeley Packet Filter (eBPF) è una tecnologia che permette di eseguire programmi dello user space in un contesto privilegiato come quello del kernel Linux tramite una macchina virtuale basata su un instruction set general purpose di tipo RISC. Tali istruzioni possono essere compilate a partire da diversi linguaggi (ad esempio C) usando la toolchain LLVM. Ciò permette di estendere le funzionalità del kernel senza modificarne il codice sorgente o installando moduli aggiuntivi. I programmi eBPF sono impiegati per svolgere vari compiti sofisticati grazie all'ampia osservabilità e velocità di esecuzione delle operazioni privilegiate, oltre che a un set di funzioni predefinite. I principali casi d'uso sono: high-performance networking nei moderni data center, tracing di applicazioni e miglioramento della sicurezza a runtime.

Nonostante eBPF sfrutti un'analisi statica del codice per prevenire l'esecuzione di loop infiniti o l'accesso ad aree di memoria non consentiti, l'enorme concessione di privilegi ai programmi eBPF può portare ad alcune gravi vulnerabilità nei sistemi in cui sono impiegati. Infatti, dato che essi richiedono di essere eseguiti come *root* o con la *CAP_SYS_ADMIN* capability abilitata, possono essere sfruttati da eventuali attaccanti per ottenere il controllo del server che contiene il container su cui sono ospitati. In questo modo tutti i container presenti sul nodo attaccato possono essere compromessi e l'attacco può portare a un furto di dati sensibili o a un DoS. Tali attacchi vengono performati tramite l'utilizzo di alcune funzioni, chiamate eBPF helpers, che permettono di leggere/scrivere lo spazio di memoria di un qualsiasi processo utente, modificare i codici di ritorno delle system call e inviare segnali di kill.

ID	Helper Name	Functionality
H1	bpf_probe_write_user	Write any process's user space memory
H2	bpf_probe_read_user	Read any process's user space memory
H3	bpf_override_return	Alter return code of a kernel function
H4	bpf_send_signal	Send signal to kill any process
H5	bpf_map_get_fd_by_id	Obtain eBPF programs' eBPF maps fd

Fig. 6: eBPF helpers

Il tipico workflow di un attacco eBPF è composto da 3 fasi:

1) L'attaccante utilizza le eBPF tracing features per modificare la funzione di dispatch delle system call, facendo in modo, ad esempio, che venga eseguito il suo codice malevolo al termine di ogni syscall;
2) Viene identificato un qualsiasi processo Bash privilegiato in esecuzione sull'host tramite riconoscimento di un certo nome e uid;
3) Una volta trovato il processo bersaglio, l'attaccante può utilizzare la funzione bpf_probe_write_user per iniettare il proprio codice e farlo eseguire dalla shell.

Nel caso in cui la lunghezza del comando iniettato sia maggiore dei bytes letti dalla read, l'attaccante può anche modificare il valore di ritorno di tale system call tramite bpf_override_return.

Analizzando il repository Docker Hub si scopre che il 2,5% dei container sono vulnerabili agli attacchi eBPF appena descritti. Questo percentuale sale al 3% se consideriamo i 300 container più scaricati (riferito a marzo 2023). Altri modi per abilitare la CAP_SYS_ADMIN capability modificando la configurazione di Docker sono:

- Eseguire Docker con il flag *-privileged*;
- Eseguire Docker con il flag *cap-add-SYS_ADMIN*;
- Esporre la *docker.sock* ai container.

Il problema di eBPF è che, nonostante sia necessario al funzionamento di molti tool fondamentali per il monitoraggio, la sicurezza e la gestione del networking dei container, non dispone di una possibilità di configurazione degli accessi fine-grained. Di fatto, può essere solo abilitato o completamente disabilitato. Per questo motivo è stata proposta una soluzione (**CapBits**) basata sull'assegnazione di due campi bitmap alla *task_struct* dei processi. Il primo campo, *cap_bits*, indica quali funzioni helper possono essere usate e quali sono i loro scope. Il secondo campo, *allow_bits*, specifica quali eBPF features possono lavorare su tale processo. I container possono settare tali campi durante la creazione di ogni processo.

SELF HOSTING

Il paper "To Cloud or not to Cloud: A Qualitative Study on Self-Hosters' Motivation, Operation, and Security Mindset" esplora le motivazioni, le operazioni e l'approccio alla sicurezza delle persone e organizzazioni che scelgono di fare hosting autonomo dei propri servizi su macchine di loro proprietà o affittate. Per la scrittura di questo articolo sono stati sottoposti a questionario 994 utenti della piattaforma Nextcloud. Questi "self-hosters" possono essere commerciali, istituzionali o privati e assumono la responsabilità dei loro dati, della sicurezza e dell'affidabilità delle loro operazioni. Il paper rileva che attualmente si sa poco su cosa motiva questi self-hosters, come operano e proteggono i loro servizi, e quali sfide devono affrontare.

Alcune motivazioni possono essere riassunte come segue:

Controllo dei dati: I self-hosters si assumono la responsabilità dei propri dati, garantendo così un maggiore controllo su di essi. Questo può essere particolarmente importante per questioni di privacy e sicurezza, dato che affidarsi ai cloud provider comporta fare affidamento esclusivamente sui loro sistemi di sicurezza;

Flessibilità: Il self-hosting offre una maggiore flessibilità rispetto ai servizi cloud commerciali. Gli utenti possono installare qualsiasi software o servizio desiderato sulle loro macchine, personalizzando l'ambiente in base alle proprie esigenze;

Risparmio economico: I costi dei servizi cloud possono sommarsi nel tempo, soprattutto per le grandi aziende. Il self -hosting può quindi rappresentare una soluzione più economica a lungo termine;

Miglioramento della sicurezza: Alcuni partecipanti allo studio hanno indicato la sicurezza come una motivazione per il self-hosting: credono che in questo modo si possa ottenere una maggiore sicurezza rispetto all'uso di soluzioni cloud commerciali;

Requisiti normativi o interni: In alcuni casi, le normative o le politiche interne possono richiedere che i dati vengano archiviati in un luogo specifico o che si abbia accesso al codice sorgente. In questi casi, il self-hosting può essere l'unica opzione praticabile;

Ragioni ideologiche: Alcuni utenti sono contrari a priori a cedere parte del controllo dei propri dati e servizi a grosse aziende come Amazon o Google per motivi politici e ideologici;

Sfida personale: In alcuni casi gli utenti preferiscono scegliere il self-hosting per mettersi alla prova nella gestione della sicurezza e configurazione dei propri servizi invece di delegare questi compiti ai cloud provider.

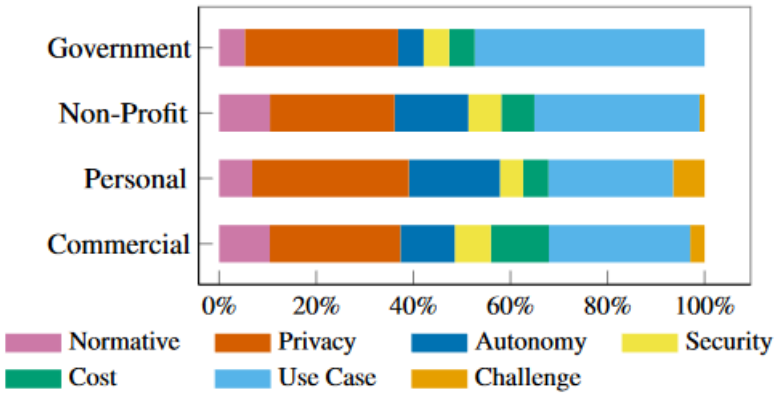

Fig. 7: principali motivazioni delle varie tipologie di self-hoster

È importante notare che, nonostante queste motivazioni, il self-hosting comporta un notevole lavoro aggiuntivo per gli utenti, oltre a richiedere una competenza tecnica non indifferente. I questionari raccolti mostrano come la maggior parte dei self-hoster hanno un approccio poco strutturato alla sicurezza e spesso non si rendono nemmeno conto delle implicazioni pratiche delle loro azioni. Le quattro principali lacune nei loro mindset sono:

1) **Attacker models**: spesso questi modelli non sono chiari o addirittura completamente assenti. Questo perché gli utenti non sono a conoscenza di quali sono i possibili attaccanti e di quali capacità dispongono;
2) **Priorità dei rischi**: spesso gli utenti sono in grado di identificare le potenziali vulnerabilità dei loro servizi, ma non sono in grado di classificarle in base alla gravità secondo un modello di trade-off tra costi e benefici;
3) **Identificazione dei meccanismi di difesa**: molti utenti non sono in grado di mappare i rischi previsti su adeguate contromisure. Questo perché non hanno sufficiente competenza in questo campo o, in alcuni casi, perché sono convinti di aver già adottato i meccanismi corretti;
4) **Manutenzione**: molti utenti non si occupano della manutenzione continua dei loro servizi e non aggiornano regolarmente i software. Questo perché hanno una visione della sicurezza come una one-time action.

PROBLEMI DEL SELF-HOSTING

Nonostante le importanti motivazioni che possono portare aziende e organizzazioni a scegliere la strada del self-hosting, bisogna fare i conti con diversi problemi:

1) **Mancanza di Competenze**: Il self-hosting richiede un certo livello di competenza tecnica per configurare e mantenere i server. Questo può essere una sfida significativa per individui o organizzazioni senza le competenze o risorse necessarie. Lo stesso vale per la gestione autonoma della sicurezza. Il trade-off tra il desiderio del controllo e lo sforzo per ottenerlo potrebbe essere negativo;

2) **Problemi di Scalabilità**: Il self-hosting potrebbe non essere altrettanto scalabile come l'hosting su cloud. Mentre l'hardware necessario ai servizi cloud può essere facilmente dimensionato in base al carico di richieste, scalare orizzontalmente servizi self-hosted richiede più pianificazione e preparazione;

3) **Manutenzione e Supporto**: Il self-hosting richiede manutenzione e supporto continuo, che può essere dispendioso in termini di tempo e denaro. Al contrario, i fornitori di servizi cloud gestiscono tipicamente queste attività come parte del loro servizio;

4) **Costi**: Sebbene il self-hosting possa potenzialmente offrire risparmi in alcune aree, può comportare anche costi iniziali significativi per hardware e costi continui per elettricità, raffreddamento e manutenzione. Al contrario, i servizi cloud operano tipicamente su un modello pay-as-you-go, che può essere più conveniente per alcuni utenti;

5) **Fault tolerance**: La gestione di eventuali fault dei server deve essere eseguita manualmente. Inoltre, bisogna implementare strategie di data recovery per minimizzare i danni dovuti a eventi inaspettati.

CONCLUSIONI

Possiamo concludere affermando che non esiste una scelta migliore a priori tra self-hosting e cloud computing, ma bisogna valutare caso per caso quali siano gli obbiettivi e le risorse a disposizione degli utenti.

Nel caso in cui si abbia la necessità di avere completo controllo sull'infrastruttura che ospita i propri servizi e della relativa politica di sicurezza, e si abbia a disposizione un team altamente formato da dedicare a tali operazioni, la scelta del self-hosting è sicuramente da preferire.

In tutti gli altri casi conviene fare affidamento in uno dei cloud provider presenti sul mercato, optando per quello che in un dato momento offre gli strumenti e le tecnologie che più si adattano alle proprie esigenze, in modo tale da riuscire comunque a garantire certi standard minimi di qualità dei servizi.

BIBLIOGRAFIA

[1] L. Grober et al., “To Cloud or not to Cloud: A Qualitative Study on Self-Hosters’ Motivation, Operation, and Security Mindset”, August 9-11 2023. Available: https://www.usenix.org/ conference/usenixsecurity23/presentation/grober

[2] Y. He and R. Guo et al., “Cross Container Attacks: The Bewildered eBPF on Clouds”, August 9-11 2023. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/he

[3] P. Kinger et al., “The Linux threat landscape report”. Available: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/the-linux-threat-landscape-report

[4] eBPF Documentation. Available: https://ebpf.io/what-is-ebpf/#the-power-of-programmability

[5] R. Bernsen, “Why to self-host your software? Should you go for on-premises or cloud?”, September 21 2021. Available: https://www.openproject.org/blog/why-self-hosting-software/