Risk Assessment

Obiettivo: Identificare e mitigare i rischi di sicurezza di un server LAMP per garantire la conformità al GDPR e la protezione dei dati sensibili presenti sullo stesso.

Ambito*:* Sistema Server Ubuntu 22.04.2 con configurazione base in vista di servizi futuri. Il server si trova al momento in uno stato non idoneo alla produzione, presenta una buona configurazione di base ma manca di protezione verso attacchi comuni e automatizzati esponendolo a potenziali compromissioni che potrebbero intaccare la continuità operativa.

Il presente documento sarà soggetto a revisione ogni tre mesi o successivamente all'implementazione dei servizi sul sistema.

Criteri di valutazione della Probabilità e dell'Impatto:

Probabilità
1 (Rara): Non è mai accaduto, estremamente improbabile che accada nel prossimo anno (probabilità < 5%).
2 (Improbabile): Potrebbe accadere in circostanze eccezionali (5-20% di probabilità annuale).
3 (Possibile): Potrebbe accadere (20-50% di probabilità annuale).
4 (Probabile): Accadrà quasi certamente (50-80% di probabilità annuale), o è accaduto in passato.
5 (Molto Probabile): Accadrà quasi certamente o più volte (80-100% di probabilità annuale), o è un'occorrenza comune.

Impatto:
1 (Insignificante): Nessun impatto rilevabile.
2 (Basso): Interruzione minima, costi finanziari contenuti, nessun danno reputazionale, conformità mantenuta.
3 (Medio): Interruzione locale/temporanea, costi potenzialmente rilevanti, lieve danno reputazionale, potenziale violazione minore della conformità.
4 (Alto): Interruzione estesa/significativa, costi elevati, danno reputazionale significativo, violazione grave della conformità con potenziali sanzioni.
5 (Catastrofico): Interruzione totale del business, perdita di dati critici, costi molto elevati, danno reputazionale irreparabile, gravi sanzioni legali, fallimento potenziale.

Criteri di valutazione del rischio:

Il rischio viene calcolato con una formula Probabilità x Impatto

Basso(1-7):	Rischio minimo, da mitigare entro tre mesi.
Medio(8-12):	Rischio moderato, da mitigare entro un mese.
Alto(13-19):	Rischio alto, da mitigare entro una settimana.
Critico(20-25):	Rischio critico, da mitigare immediatamente.

Da cui la seguente Heat Map:

Probabilità

5	5	10	15	20	25
4	4	8	12	16	20
3	3	6	9	12	15
2	2	4	6	8	10
1	1	2	3	4	5
	1	2	3	4	5

Rischio

Tabella dei rischi

ID Rischio	Asset/Vulnerabilità	Attore Minaccia & Vettore Attacco	Probabilità (1-5)	Impatto (1-5)	Punteggio Rischio	Livello Rischio	Azione raccomandata
RISK-001	Configurazione Firewall	Chiunque può fare uno scan delle porte e attaccare un qualsiasi servizio esposto senza limitazioni	5	5	25	CRITICO	Configurare il firewall per permettere solo le azioni necessarie al funzionamento dei servizi richiesti
RISK-002	Backups	Assenza di backup programmati, nessun piano di recovery	4	5	20	CRITICO	Implementare un sistema di backup robusto e di testing per verificarne l'integrità
RISK-003	Protezione attacchi comuni a servizi web	Attacchi SQL Injection, Cross-Site Scripting (XSS) e altri	4	4	16	ALTO	Installare un WAF e impostare delle configurazioni più restrittive per gli input
RISK-004	Configurazione Password	Chiunque abbia l'accesso può vedere in chiaro i dati all'interno del database o accedere ai servizi web	3	5	15	ALTO	Aggiungere password diverse e sicure per l'accesso ad ogni servizio

RISK-005	Configurazione SSH	Le configurazioni standard aumentano la probabilità di un attaccante di avere successo	3	5	15	ALTO	Disabilitare l'accesso root da SSH. impostare regole per indirizzi permessi, usare una chiave ellittica, cambiare la porta predefinita
RISK-006	Integrità dei files	Le modifiche di un attaccante intenzionato a renderle nascoste potrebbero non essere rilevate	3	5	15	ALTO	Configurare un sistema di controllo hash per rilevare modifiche non autorizzate
RISK-007	Raccolta informazioni	I logs incompleti potrebbero non rilevare attacchi avvenuti con successo o tentati	3	4	12	MEDIO	Configurare il sistema di auditing del kernel e di log
RISK-008	Servizi installati	Possibile sfruttamento di un servizio obsoleto o non utilizzato ma installato come dipendenza o per errore	3	4	12	MEDIO	Fare un censimento dei servizi e delle dipendenze essenziali, disinstallare tutto ciò che non è strettamente necessario per ridurre la superficie d'attacco
RISK-009	Configurazioni accesso	Nessun limite per il brute-force o per la complessità delle password	3	4	12	MEDIO	Aggiungere regole per un limite massimo di try nell'inserimento delle password, aggiungere una regola per la creazione di una password forte e implementare una scadenza per forzare il rinnovo delle password
RISK-010	Protezione da DoS	Un attaccante esterno potrebbe cercare di riempire le cartelle /var e /tmp o inondare di richieste il servizio web causando un fallimento dell'intero sistema	4	3	12	MEDIO	Limitare il numero di pacchetti da un singolo host o contemporanei, valutare l'installazione di software specifici per applicazioni web. Spostare le cartelle /var e /temp su una partizione dedicata
RISK-011	Installazione pacchetti	Un pacchetto considerato affidabile potrebbe essere stato compromesso o avere bug	4	3	12	MEDIO	Installare un sistema di controllo dei pacchetti
RISK-012	GRUB boot loader	Un attaccante potrebbe accedere come root dal boot senza bisogno di inserire nessuna password se l'host è compromesso	2	5	10	MEDIO	Impostare una password per il boot per impedire l'utilizzo del single user mode
RISK-013	Core dump	Il core dump è un file in cui possono esserci dati sensibili visualizzabili in chiaro	3	3	9	MEDIO	Disabilitare il core dump e valutare un sistema di logging esterno su un server dedicato
RISK-014	Porte in ascolto	Possibili attacchi a porte note tramite bot o attacchi mirati	3	3	9	MEDIO	Verificare che siano aperte solo le porte necessarie, e che siano ben protette
RISK-015	Processi attivi	Un processo in ascolto potrebbe essere sfruttato in modo malevolo	3	3	9	MEDIO	Controllare i processi attivi sul server, rimuovere o limitare i processi in ascolto obsoleti o non necessari (come snap)

RISK-016	Pagine accessibili	Files html o php di test possono rivelare informazioni confidenziali che possono essere sfruttate per altri attacchi	3	3	9	MEDIO	Spostare o eliminare qualsiasi pagina sia raggiungibile da web e non sia di dominio pubblico
RISK-017	Logs management	Nessun backup nel caso in cui un attaccante cancellasse i log delle sue azioni malevole	3	3	9	MEDIO	Configurare un sistema di logging in tempo reale ad un server esterno
RISK-018	Utenti	Un numero elevato di utenti con ampi permessi espone la macchina a più rischi	2	4	8	MEDIO	Controllare l'elenco degli utenti, disabilitare quelli non necessari. Gestire i permessi per ogni utente limitando le azioni possibili a quelle necessarie per il funzionamento corretto dei servizi
RISK-019	Permessi compiler	Un utente con permessi limitati potrebbe sfruttare i compiler per elevare i privilegi	2	4	8	MEDIO	Restringere l'accesso ai compiler
RISK-020	Protocolli di rete	Ogni protocollo abilitato ma non utilizzato è un potenziale vettore di attacco	2	4	8	MEDIO	Disabilitare i protocolli non necessari
RISK-021	Permessi nuovi files	La umask è poco restrittiva permettendo la creazione e l'accesso ai nuovi file troppo facilmente	2	3	6	BASSO	Restringere i permessi sui nuovi files
RISK-022	Driver dispositivi esterni	Un attaccante potrebbe montare una USB esterna malevola sul sistema	1	5	5	BASSO	Disattivare driver USB e simili
RISK-023	Accesso fisico	Accesso fisico non autorizzato al server	1	5	5	BASSO	Chiudere a chiave la porta, mettere password al sistema host e al boot
RISK-024	Pulizia vecchi files	Vecchi files e pacchetti non utilizzati potrebbero nascondere vulnerabilità o avere dati sensibili	2	2	4	BASSO	Automatizzare la rimozione di files obsoleti

Piano di Trattamento del Rischio

Risk-001

Livello rischio: Critico
Trattamento: Mitigare

Un server senza configurazioni di firewall è costantemente esposto ad attacchi automatizzati che secondo le ultime stime colpiscono mediamente ogni server internet-facing ogni 39 secondi secondo telemetria 2025 di security vendors. Il tempo medio tra esposizione e primo tentativo di exploitation è diminuito a 14 ore negli ultimi 12 mesi.
Il firewall è già presente sul sistema, la configurazione policy di default suggerisce di bloccare tutto il traffico in entrata e consentire quello in uscita. Prima di fare ciò è necessario aggiungere una regola per ssh per evitare lockout, il mio suggerimento è di permettere connessioni in entrata temporaneamente sulla porta 22 (dato le configurazioni attuali e standard SSH) e su una nuova porta che verrà usata per fare le connessioni SSH (ad esempio la 2222)

_____
Questi comandi permettono connessioni in entrata alla porta 2222 con protocollo tcp e alla porta 22 letta dalle configurazioni SSH attuali:
```
sudo ufw allow 2222/tcp comment 'SSH new port'
sudo ufw allow ssh
```

_____
Queste sono le configurazioni di default del firewall che negano tutte le connessioni in ingresso:
```
sudo ufw default deny incoming
sudo ufw default allow outgoing
```

_____
Permette la connessione al server Apache installato aprendo le porte 80 (http) e 443 (https):
```
sudo ufw allow ‘Apache Full’
```

_____
Verificare le impostazioni modificate prima di applicarle è una buona pratica di sicurezza:
```
sudo ufw show added
```

_____
Applicare le nuove regole del firewall:
```
sudo ufw enable
```

_____
Verificare che siano attive e una volta:
```
sudo ufw status numbered
```

_____
Una volta verificato che sia le porte 22 che 2222 sono aperte allora si può cambiare la configurazione SSH (RISK-005) e riavviare la connessione per connettersi sulla nuova porta.

`sudo systemctl reload ssh`

_____

Dopo aver verificato che la connessione è stata stabilita sulla porta 2222 invece che sulla standard 22, rimuovere la regola che consentiva l'ascolto sulla vecchia porta:
`sudo ufw delete allow 22/tcp`

Risk-002

Livello rischio: Critico
Trattamento: Mitigare

Un server senza un solido sistema di backup viola il GDPR ed è una grave negligenza di sicurezza. L'implementazione del sistema di backup è quindi imperativa per garantire l'operatività del server e la sua conformità normativa.
La soluzione di backup proposta è a 3 livelli, conforme al principio: 3-2-1, ovvero, tre copie dei dati su due dispositivi diversi con 1 copia off-site. Inoltre la strategia include backup automatizzati giornalieri tramite appositi script per garantire consistenza transazionale, inclusione di routine e gestione ottimizzata della memoria tramite compressione.
Il piano di mitigazione di questo rischio prevede l'installazione di alcuni strumenti per semplificare la gestione del backup, garantirne l'integrità e l'automazione. In particolare:

`rsync` essenziale per la sincronizzazione efficiente dei backup remoti garantendo la sincronizzazione e un backup incrementale per risparmiare risorse, da configurare con una connessione SSH.

`mysql-client` che contiene i tool necessari per l'amministrazione di un server sql.

`cron` gestisce l'automazione delle operazioni di backup, minimizzando l'errore umano e velocizzando la procedura.

`logrotate` per la gestione efficiente dei log

`gzip` e `tar` per la compressione di logs e backup

`mailutilis` abilita l'invio di notifiche via email per il monitoraggio

Possono tutti essere installati tramite `apt install`

_____

Il primo passo è quello di creare le directory e sottodirectory dei backups con il comando
`sudo mkdir -p /backup/{database,web,system,logs}/{daily,weekly,monthly}`

Seguito dall’applicazione di permessi più restrittivi
sudo chmod -R 755 /backup && sudo chown -R root:root /backup

Nota: Nel caso in cui si optasse per una soluzione ancora più sicura si potrebbero implementare dei gruppi utenti appositi per la gestione dei backup e per il monitoring tramite script di servizio.

_____
Successivamente si procede con la creazione della chiave di crittografia SSH in una cartella esterna a quelle di backup. Il seguente comando creerà una cartella di storage delle backup-key che potranno essere utilizzate da script di automazione eseguiti con privilegi elevati.
sudo mkdir -p /etc/backup-keys && sudo openssl rand -base64 32 > /etc/backup-keys/backup.key && sudo chmod 400 /etc/backup-keys/backup.key && sudo chown root:root /etc/backup-keys/backup.key

Nota: Per una maggiore sicurezza si valuti la gestione delle chiavi tramite un server dedicato o la suddivisione delle chiavi tramite un sistema che usa tecniche di secret sharing.

_____
Dopo aver verificato i passaggi precedenti si puo’ passare alla gestione dei backups vera e propria che consiglio implementare tramite script con permessi 755, incorporando logiche di error handling, logging dettagliato, verifica di integrità tramite hash-256. In particolare suggerisco:

- Script che crei il backup con una funzione di logging per garantire conformità alle normative, un controllo preventivo della chiave di crittografia, uso di mysql-client per creare i files di backup, crittografia hash SHA-256 per verificare che i backup non siano corrotti e retention automatica per rispettare il principio di minimizzazione dei dati.
- Script separato per il backup delle configurazioni Apache, document root web, configurazioni di sistema, pulizia files temporanei e checksum per ognuno di essi.
- Script per il backup remoto usando rsync over SSH, previa configurazione del server remoto.
- Script di test di integrità e recovery per poter controllare la checksum e provare il decrypt dei backups in una cartella test ed eliminarla successivamente
- Script per il monitoraggio e l’aletring con controllo sulla data di creazione dei backups, alert automatici via mail in caso di fallimento o problemi e spazio su disco occupato dai backups.
- Script di manutenzione e cleanup per automatizzare la retention dei dati in conformità con il GDPR che includa una logica sofisticata che tiene conto delle diverse policy di retention per diversi tipi di dati e crei files di log per registrare l’eliminazione dei vecchi backups.

_____
L’ultimo passo per completare l’automazione del sistema di backups comprende la configurazione di cron tramite sudo crontab -e con scheduling distribuito temporalmente, ad esempio backup database alle 02:00, filesystem alle 03:00, sincronizzazione remota alle 04:00 e cleanup alle 5:30 per garantire che i backup della notte precedente siano completati prima delle operazione di pulizia.

Risk-003

Livello rischio: Alto
Trattamento: Mitigare

Gli attacchi a servizi web rappresentano una delle principali minacce per server LAMP esposti a internet. Secondo i dati del progetto OWASP 2023, l'injection (inclusa SQL injection) rimane al primo posto tra le vulnerabilità più critiche, seguita da cross-site scripting (XSS) e configurazioni di sicurezza errate. Consiglio l'implementazione di un WAF, web application firewall, come modsecurity che è open source e presenta svariate opzioni di configurazione personalizzate oltre ad essere il più diffuso per i server Apache. Installare il modulo con
`sudo apt install libapache2-mod-security2`

_____
Abilitare il modulo appena installato nell'istanza Apache
`sudo a2enmod security2`

_____
Il modulo viene scaricato con delle configurazioni raccomandate che possono essere utilizzate come base per le direttive principali del WAF. Consiglio di utilizzare il template come base e applicare modifiche allo stesso.
`sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf`

____
Una volta copiato il file di configurazioni base, aprirlo con `sudo nano /etc/modsecurity/modsecurity.conf`
e apportare le seguenti modifiche:

SecRuleEngine On: Attiva il motore di regole
SecAuditEngine On: Abilita l'audit logging per tracciare tutti gli eventi
SecRequestBodyLimit 10485760: Limita la dimensione delle richieste a 10MB per prevenire DoS
SecResponseBodyLimit 524288: Limita le risposte a 512KB per ottimizzare le performance

_____
Il prossimo passo consiste nell'implementazione delle regole personalizzate per il corretto funzionamento del WAF, per fare ciò bisogna innanzitutto creare la cartella che le conterrà
`sudo mkdir -p /etc/modsecurity/custom-rules`

_____
E includere la cartella e la configurazione base nel file di configurazione con
`sudo nano /etc/apache2/mods-enabled/security2.conf`

_____
Si passa ora all'installazione di ModEvasive per proteggere il server specificatamente contro attacchi DoS e brute force limitando il numero di richieste per IP in un determinato lasso di tempo. Anche questo modulo va abilitato su Apache.
`sudo apt install -y libapache2-mod-evasive`
`sudo a2enmod evasive`

E successivamente configurato con le seguenti modifiche
`sudo nano /etc/apache2/mods-enabled/evasive.conf`

DOSPageCount/DOSPageInterval: Limite richieste per singola pagina
DOSSiteCount/DOSSiteInterval: Limite richieste per intero sito
DOSBlockingPeriod: Durata del blocco temporaneo dell'IP
DOSWhitelist: IP esclusi dal controllo (reti locali)

_____
Continuando l'hardening del servizio Apache il prossimo passo è quello di configurare la security headers
`sudo nano /etc/apache2/conf-available/security-headers.conf`

Modificare il file con regole per:

- Prevenzione XSS
- Prevenzione clickjacking
- Prevenzione MIME sniffing
- Content Security Policy di base
- HSTS per HTTPS
- Nascondere le informazioni server

E applicarle con
`sudo a2enconf security-headers`
`sudo a2enmod headers`

_____
Lo stesso modulo può essere usato anche per bloccare il movimento laterale modificando le configurazioni in sudo nano /etc/apache2/conf-available/anti-traversal.conf

Con regole per:
- Protezione contro directory trasversal
- Bloccare l'accesso a files sensibili
- Bloccare l'accesso a directory specifiche

che vanno poi attivate con
`sudo a2enconf anti-traversal`

_____
Ulteriori regole personalizzate contro XSS e SQL Injection possono essere create e aggiunte alla cartella `/etc/modsecurity/custom-rules`

Nota: Dopo una valutazione sul tempo d'implementazione e gestione delle esclusioni si è deciso per ora di non applicare le CRS (Core Rule Set), tuttavia l'implementazione futura di questo ruleset open gestito da esperti di cybersecurity mondiali è fortemente consigliata.

_____
L’implementazione di questa soluzione comprende anche un logging completo per monitorare in tempo reale eventuali errori di configurazione o attacchi bloccati e gestibile tramite logrotate in
/var/log/apache2/modsec_audit.log

_____

Risk-004

Livello rischio: Alto
Trattamento: Mitigare

L'assenza di password specifiche per ogni servizio aumenta la probabilità di accessi non autorizzati ed espone tutti i servizi facilmente ad un eventuale attaccante che è riuscito ad ottenere l’accesso al server.L'assenza di password specifiche per ciascun servizio aumenta la probabilità di accessi non autorizzati ed espone tutti i servizi ad un potenziale attaccante che sia riuscito ad ottenere l'accesso al server. Impostare delle password per proteggere MariaDB e Apache è quindi fortemente consigliato.

Per impostare una password al servizio MariaDB eseguire il comando successivo e seguire gli step proposti sul terminale per la creazione della password facendo attenzione a disabilitare il root login da remoto
sudo mysql_secure_installation

_____
Impostare una password anche per Apache utilizzando lo strumento utilis di apache server
sudo apt install apache2-utils

Creare la directory che conterrà il file di autenticazione
sudo mkdir -p /etc/apache2/auth

Creare il file di password (.htpassword) per l’accesso al pannello di controllo
sudo htpasswd -c /etc/apache2/auth/.htpasswd admin

Inserire una password sicura quando richiesto e poi applicare le configurazioni al server Apache creando prima una configurazione per la protezione delle directory interessate con
sudo nano /etc/apache2/conf-available/auth-directories.conf

Abilitare i moduli e la configurazione
sudo a2enmod auth_basic
sudo a2enmod authz_user
sudo a2enconf auth-directories

_____
In generale si consiglia di utilizzare password di almeno 16 caratteri, con 4 classi di caratteri (a-z, A-Z, 0-9, simboli), diverse per ogni servizio ed evitando di usare parole presenti sul dizionario.

Risk-005

Livello rischio: Alto
Trattamento: Mitigare

Le configurazioni SSH standard aumentano significativamente la probabilità di successo per un attaccante. Il server risulta esposto su porta 22 con configurazioni predefinite che facilitano attacchi brute force e compromissioni del sistema.

È buona pratica eseguire un backup delle configurazioni critiche nel caso in cui dovessero esserci errori per poterle ripristinare tempestivamente

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.backup.$(date +%Y%m%d_%H%M%S)
```

_____

Creare una coppia di chiavi di autenticazione ellittiche (Ed25519) per una maggiore sicurezza rispetto a quelle tradizionali ma senza impattare significativamente sulle performances

```
sudo mkdir -p /etc/ssh/hardened_keys
sudo ssh-keygen -t ed25519 -f /etc/ssh/hardened_keys/ssh_host_ed25519_key -N ""
sudo chmod 600 /etc/ssh/hardened_keys/ssh_host_ed25519_key
sudo chmod 644 /etc/ssh/hardened_keys/ssh_host_ed25519_key.pub
```

_____

Creare un banner di avvertimento per informare che il sistema è riservato agli utenti autorizzati, fornisce un avvertimento legale e può avere valore in eventuali processi giudiziari

```
sudo nano /etc/ssh/ssh_banner.txt
```

_____

Aprire il file di configurazione SSH

```
sudo nano /etc/ssh/sshd_config
```

E apportare le seguenti modifiche:

- Cambio porta da 22 a 2222
- Disabilitare il root login
- Restrizioni di accesso solo ad user permessi
- Indicare il path per le chiavi ellittiche
- Impostare soluzioni di hardening generali come un limite ai tentativi di autenticazione, limite alle sessioni SSH contemporanee consentite, tempo tra un try e l'altro per evitare brute force
- Disabilitare funzionalità non necessarie
- Valutare la restrizione ad ip specifici
- Abilitare logging verbose
- Disabilitare password vuote
- Aggiungere path a banner con valore legale

_____

Verifica correttezza della sintassi della configurazione
```
sudo sshd -t
```

_____
Se non si è già fatto, aggiungere una regola al firewall per consentire la nuova porta
```
sudo ufw allow 2222/tcp comment 'SSH hardened port'
```

_____
Riavviare il servizio SSH mantenendo la sessione attiva
```
sudo systemctl reload ssh
sudo systemctl status ssh
```

_____
Dopo aver verificato il funzionamento della porta 2222, rimuovere la regola per la porta 22 non più in uso
```
sudo ufw delete allow 22/tcp
```

_____
Per un'ulteriore protezione consiglio l'installazione di fail2ban per bloccare automaticamente le richieste da ip che effettuano tentativi di accesso falliti ripetuti.
```
sudo apt install fail2ban
```

Modificare le configurazioni per proteggere SSH sulla nuova porta impostando anche un percorso per il salvataggio dei logs
```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
sudo nano /etc/fail2ban/jail.local
```

_____
Riavvio fail2ban dopo aver modificato le configurazioni e verifica.
```
sudo systemctl enable fail2ban
sudo systemctl start fail2ban
sudo fail2ban-client status sshd
```

Risk-006

Livello rischio: Alto
Trattamento: Mitigare
Le modifiche di un attaccante ai files di sistema possono rimanere nascoste senza un sistema di controllo dell'integrità. L'implementazione di un sistema di monitoraggio hash permetterà di rilevare immediatamente modifiche non autorizzate a files critici del sistema.
Una soluzione temporanea è quella di creare due script utilizzando gli strumenti nativi di Linux come sha256sum per creare gli hash dei files critici e il controllo degli stessi.

Creare la work directory
```
sudo mkdir -p /var/security/integrity/{baseline,current,reports}
sudo chmod 750 /var/security/integrity
```

```
sudo chown root:root /var/security/integrity
```

_____

Creare lo script per la baseline, funge da base di controllo con gli hash iniziali

```
sudo nano /usr/local/bin/create-integrity-baseline.sh
```

_____

Aggiungere allo script i permessi necessari ed eseguirlo, ciò creerà un files con gli hash SHA265 di tutti i files critici del sistema

```
sudo chmod +x /usr/local/bin/create-integrity-baseline.sh
sudo /usr/local/bin/create-integrity-baseline.sh
```

_____

Creare un secondo script per il controllo, lo script genera dei nuovi hash basati sui files correnti e li confronterà con quelli creati precedentemente, salvando tutto il processo in un file di log

```
sudo nano /usr/local/bin/check-integrity.sh
sudo chmod +x /usr/local/bin/check-integrity.sh
sudo /usr/local/bin/check-integrity.sh
```

_____

Impostare lo script di check con crontab (precedentemente installato) per automazione.

```
sudo crontab -e
```

Nota: Questa è una soluzione temporanea, consiglio fortemente di passare ad una configurazione che faccia uso di strumenti appositi come AIDE per avere un controllo in tempo reale, configurazioni avanzate ed esclusioni per evitare falsi positivi (dati dalle cartelle di log ad esempio), algoritmi di hash multipli, più velocità e scalabilità.

Risk-007

Livello rischio: Medio

Trattamento: Mitigare

L’assenza di un sistema di auditing del kernel potrebbe prevenire la rilevazione di tentativi di attacco falliti o riusciti, modifiche non autorizzate al sistema, escalation di privilegi, accessi non autorizzati, violazioni delle normative.

Installare auditd, avviarlo e abilitarlo all’ascolto

```
apt install auditd audispd-plugins
systemctl start auditd
systemctl enable auditd
```

_____

Il nucleo funzionale di `auditd` è rappresentato dalle sue regole, le quali ne definiscono gli eventi da registrare. Le regole personalizzate devono essere collocate in file con estensione .rules all'interno della directory /etc/audit/rules.d/. Un insieme di regole, ispirato agli standard di sicurezza del CIS (Center for

Internet Security), include: l'immutabilità della configurazione una volta stabilita; il monitoraggio dei file di configurazione di auditd; l'accesso a file critici per l'identità e la sicurezza; gli eventi di login e logout; i comandi che modificano le informazioni di utenti e gruppi; le alterazioni dei permessi dei file; e i tentativi di accesso non autorizzato ai file.
Per caricare le regole appena create
```
augenrules --load
```

_____
Controllo dello status attuale
```
sudo systemctl status auditd
sudo auditctl -s
```

Nota: auditd crea dei files di log nella cartella var/log/audit/ che possono essere gestiti con logrotate, inoltre è consigliato aggiungere uno script di controllo dei log collegato ad un sistema di alerting con mailutilis e automazione con crontab.

Risk-008

Livello rischio: Medio
Trattamento: Mitigare
Il server presenta numerosi servizi e pacchetti installati (packages.txt negli allegati) che potrebbero non essere necessari per le funzioni richieste. La presenza di software non utilizzato rappresenta una superficie di attacco ampliata, dove un attaccante potrebbe sfruttare vulnerabilità in servizi obsoleti o non mantenuti. Dal censimento dei pacchetti emerge la presenza di centinaia di pacchetti, molti dei quali potrebbero non essere essenziali per il funzionamento di un server LAMP.La riduzione della superficie di attacco è un principio fondamentale della sicurezza informatica: ogni componente software installato rappresenta un potenziale vettore di vulnerabilità.

Analisi approfondita delle dipendenze con
```
apt-cache depends apache2 mariadb-server php php-mysql | grep "Depends:" | awk '{print $2}' | sort | uniq > /tmp/lamp_dependencies.txt
```

_____
Una volta rilevati i pacchetti essenziali al funzionamento del server passare all'identificazione dei pacchetti obsoleti tramite un confronto del file lamp_dependencies.txt con packages.txt.
Utilizzare anche il comando apt per individuare i pacchetti obsoleti
```
sudo apt autoremove --dry-run
```

E per un controllo sui pacchetti installati manualmente
```
apt-mark showmanual | sort > /tmp/manual_packages.txt
```

_____

Disabilitare servizi non essenziali come ad esempio snap, per disattivare il servizio (sostituire il nome del servizio che si desidera rimuovere nel comado)

```
sudo systemctl disable snapd.service
sudo systemctl stop snapd.service
sudo systemctl mask snapd.service
```

_____

Valutare la creazione di uno script per il controllo dei pacchetti orfani e delle dipendenze con logs muniti di timestamp e automazione tramite crontab. Si consiglia inoltre l'installazione di un tool specializzato per l'identificazione di pacchetti orfani come deborphan.

Risk-009

Livello rischio: Medio
Trattamento: Mitigare

Impostare delle password non è sufficiente per garantire la sicurezza, le password devono essere sufficientemente complesse ed essere rinnovate periodicamente. Consiglio di installare il modulo PAM per il creare una policy delle password robusta che non si basi solo sulla segretezza iniziale ma tenga conto anche della sua validità nel tempo.

Installare il modulo necessario

```
apt install libpam-pwquality
```

_____

Modificare le configurazioni per impostare: lunghezza minima della password, presenza di almeno un numero; una lettere minuscola; una lettera maiuscola; un simbolo, impostare numero di caratteri che devono essere diversi dalla vecchia password per scoraggiare l'uso di password incrementali

```
nano /etc/security/pwquality.conf
```

_____

Impostare la scadenza delle password e applicare le regole retroattivamente agli utenti già esistenti. Per fare ciò aprire il file di configurazione delle scadenze ed impostare una durata massima, una minima e dei giorni di preavviso prima della scadenza della password.

```
nano /etc/login.defs
```

Applicare le modifiche retroattivamente agli utenti già esistenti con un ciclo for (variare i parametri a seconda della configurazione scelta)

```
for user in $(awk -F: '$3 >= 1000 {print $1}' /etc/passwd); do
  echo "Applicando la policy di scadenza all'utente: $user"
  chage -M 90 -m 7 -W 14 "$user"
done
```

Risk-010

Livello rischio: Medio
Trattamento: Mitigare
Rappresenta una vulnerabilità del sistema dove un attaccante esterno potrebbe compromettere la disponibilità del servizio attraverso: Saturazione dello spazio disco: Riempiendo le directory /var e /tmp con file temporanei o log voluminosi, Flooding delle richieste: Inondando il server web con richieste simultanee per esaurire le risorse di sistema, Esaurimento delle risorse: Causando un denial of service che renderebbe il sistema inutilizzabile. Per quanto riguarda la protezione da attacchi DoS sul web server vedere le mitigazioni al Risk-003.

Configurare protezioni aggiuntive al livello di sistema usando iptables per limitare le connessioni al livello di kernel per una sicurezza in profondità. Al solito è buona pratica fare un backup delle configurazioni attuali nel caso di errori.

```
sudo iptables-save > /backup/iptables-backup-$(date +%Y%m%d).rules
```

Aggiungere regola per limitare le connessioni HTTP e HTTPS simultanee a 25 per ip

```
sudo iptables -A INPUT -p tcp --dport 80 -m connlimit --connlimit-above 25 -j REJECT --reject-with tcp-reset
sudo iptables -A INPUT -p tcp --dport 443 -m connlimit --connlimit-above 25 -j REJECT --reject-with tcp-reset
```

Prima regola marca nuove connessioni, seconda regola blocca IP che fanno più di 20 connessioni in 60 secondi

```
sudo iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m recent --set --name HTTP_FLOOD
sudo iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 60 --hitcount 20
--name HTTP_FLOOD -j DROP

sudo iptables -A INPUT -p tcp --dport 433 -m state --state NEW -m recent --set --name HTTP_FLOOD
sudo iptables -A INPUT -p tcp --dport 433 -m state --state NEW -m recent --update --seconds 60 --hitcount 20
--name HTTP_FLOOD -j DROP
```

Salvare le nuove regole

```
sudo iptables-save
```

_____
Consiglio di separare le directory /var e /tmp su partizioni dedicate per evitare che un flooding porti ad un denial of service dell'intero sistema. Dopo aver calcolato lo spazio occupato dalle cartelle e lo spazio complessivo disponibile procedere con la creazione delle partizioni (con file di loop se non sono presenti dischi aggiuntivi)

```
sudo dd if=/dev/zero of=/var-partition.img bs=1M count=5120   5GB per /var
sudo dd if=/dev/zero of=/tmp-partition.img bs=1M count=2048   2GB per /tmp
```

Formattare le nuove partizioni con ext4

```
sudo mkfs.ext4 /var-partition.img
sudo mkfs.ext4 /tmp-partition.img
```

Aggiungere le partizioni al File System Table in /etc/fstab previo backup

```
sudo cp /etc/fstab /etc/fstab.backup.$(date +%Y%m%d)
echo '/var-partition.img /var ext4 loop,defaults 0 2' | sudo tee -a /etc/fstab
echo '/tmp-partition.img /tmp ext4 loop,defaults,noexec,nosuid,nodev 0 2' | sudo tee -a /etc/fstab
```

_____

Configurazioni avanzate anti DoS al livello di kernel, creando un file di configurazione per i parametri di rete che contenga: protezione contro SYN flood, riduzione timeout per connessioni, protezione contro ICMP flood, rate limiting per le connessioni

```
sudo nano /etc/sysctl.d/99-anti-dos.conf
```

E applicarlo

```
sudo sysctl -p /etc/sysctl.d/99-anti-dos.conf
```

_____

Aggiungere un sistema di logging specifico che tenga traccia dei tentativi di DoS integrando i logs da iptables e ModEvasive (Risk-003) e integrare Fail2Ban con filtri specifici per attacchi DoS HTTP (riavviare fail2ban per applicare le modifiche)

_____

Valutare la creazione di uno script per il monitoraggio in tempo reale, con timestamp, connessioni attive e bloccate, ultimi eventi e controllo spazio su disco e partizioni per garantire la continuità operativa.

Risk-011

Livello rischio: Medio
Trattamento: Mitigare

Ogni volta che un nuovo software viene installato sul sistema lo espone potenzialmente a vulnerabilità che potrebbero essere sfruttate dagli attaccanti. Come pacchetti compromessi alla fonte (gli attacchi alla supply chain sono sempre più frequenti) o vulnerabilità sconosciute al momento dell'installazione. Consiglio l'implementazione di un sistema di controllo di integrità dei pacchetti e la gestione automatizzata degli aggiornamenti di sicurezza.

Installare gli strumenti di monitoraggio aggiuntivi a quelli già presenti

```
sudo apt install -y debsums apt-listchanges unattended-upgrades needrestart
```

_____

È buona norma tenere le strutture di controllo in cartelle separate (come fatto per gli altri rischi) con permessi restrittivi

```
sudo mkdir -p /var/security/packages/{reports,baselines,logs}
sudo chmod 750 /var/security/packages && sudo chown root:root /var/security/packages
```

_____
Fare un backup delle configurazioni attuali
```
sudo cp -r /etc/apt /etc/apt.backup.$(date +%Y%m%d_%H%M%S)
```

_____
Creare due script, uno per il controllo integrità con l'utilizzo di debsums per il controllo integrità con la versione installata, verifica di firme digitali con apt-key list per verificare che tutti i repository da cui vengono scaricati i pacchetti siano attendibili e con chiavi di firma valide, e controlli se ci sono upgrade di sicurezza su pacchetti già installati. Un secondo script invece che funga come sistema di sorveglianza continua per monitorare attivamente le minacce emergenti con: monitoraggio del sistema ubuntu, controllo CVE sui pacchetti critici, audit di sicurezza con lynis integrato, controllo rootkit e malware con rkhunter, clamav e chrootkit, che restituisca un punteggio di rischio quantificabile.
```
sudo nano /usr/local/bin/check-package-integrity.sh
sudo chmod +x /usr/local/bin/check-package-integrity.sh

sudo nano /usr/local/bin/vulnerability-monitor.sh
sudo chmod +x /usr/local/bin/vulnerability-monitor.sh
```

_____
Configurare aggiornamenti automatici di sicurezza e abilitare gli aggiornamenti periodici modificando i files di configurazione con le regole desiderate
```
sudo nano /etc/apt/apt.conf.d/50unattended-upgrades
sudo nano /etc/apt/apt.conf.d/20auto-upgrades
```

_____
Automazione con crontab per eseguire gli script

Risk-012

Livello rischio: Medio

Trattamento: Accettazione e trasferimento

Un attaccante che riuscisse a compromettere il sistema host su cui vi è il server potrebbe ottenere l'accesso tramite il boot bypassando il login con password. Dato la bassa probabilità che ciò accada suggerisco di accettare il rischio in attesa di trasferirlo migrando verso un provider cloud professionale, trasferendo non solo il rischio specifico ma l'intero layer della sicurezza fisica a specialisti del settore prestando attenzione che il provider scelto sia conforme con le normative vigenti e le certificazioni aggiornate (ISO27001).

Risk-013

Livello rischio: Medio

Trattamento: Mitigare

I core dump sono file di memoria generati automaticamente dal sistema quando un'applicazione termina in modo anomalo. Questi files possono contenere password e credenziali in chiaro, dati sensibili elaborati dall'applicazione, informazioni sulla struttura interna del sistema, chiavi di crittografia temporanee. Nel contesto di un server LAMP i core dump non controllati rappresentano un rischio serio per la protezione dei dati personali.

Il mio suggerimento è di disabilitare completamente la creazione e lo storage di core dump, configurare un server esterno per la collezione dei log di debug e implementare alternative più sicure per il debugging.

Creare un backup delle configurazioni attuali

```
sudo cp /etc/security/limits.conf /etc/security/limits.conf.backup.$(date +%Y%m%d_%H%M%S)
sudo cp /etc/systemd/coredump.conf /etc/systemd/coredump.conf.backup.$(date +%Y%m%d_%H%M%S)
```

_____

Modificare il file limits.conf per disabilitare i core dump per tutti gli utenti. Il file controlla i limiti delle risorse per gli utenti. Impostando il limite dei core dump a 0 sia per i limiti "soft" che "hard", si previene la generazione di core dump per tutti gli utenti del sistema.

```
echo "* hard core 0" | sudo tee -a /etc/security/limits.conf
echo "* soft core 0" | sudo tee -a /etc/security/limits.conf
```

_____

Modificare la configurazione systemd per i core dump.La configurazione Storage=none impedisce a systemd di salvare i core dump, mentre gli altri parametri impostati a 0 assicurano che nessun dump venga creato o conservato.

```
sudo nano /etc/systemd/coredump.conf
```

Applicare le modifiche ricaricando systemd

```
sudo systemctl daemon-reload
```

_____

Creare un file di configurazione sysctl per la sicurezza del kernel, specifico per i core dump che impedisca la creazione di core dump per processi con privilegi elevati, reindirizza qualsiasi tentativo di core dump a /bin/false che non fa nulla, disabiliti l'aggiunta del PID al nome del core dump.

```
sudo nano /etc/sysctl.d/99-disable-coredump.conf
```

Applicare immediatamente le modifiche

```
sudo sysctl -p /etc/sysctl.d/99-disable-coredump.conf
```

_____

Configurazione di sistema per assicurare che ogni nuova sessione utente abbia automaticamente i core dump disabilitati, fornendo un ulteriore livello di protezione

```
echo "ulimit -c 0" | sudo tee -a /etc/profile
```

```
echo "ulimit -c 0" | sudo tee -a /etc/bash.bashrc
echo "ulimit -c 0" | sudo tee -a /root/.bashrc
```

_____

Implementare un sistema di logging esterno tramite l’utilizzo di rsyslog-gnutls per sostituire la funzionalità di debugging persa con la disabilitazione del core dump.

Nota: Il logging esterno è fondamentale perché, disabilitando i core dump, si perdono informazioni preziose per il debugging. Un sistema di logging centralizzato permette di mantenere traccia degli errori senza esporre dati sensibili.

_____

Installare strumenti aggiuntivi per il debugging che non produca core dump ma logs ben strutturati e sicuri, come gdb e strace. Questi strumenti operano in tempo reale e possono essere configurati per filtrare automaticamente informazioni potenzialmente sensibili.

_____

Valutare la creazione di uno script per il monitoraggio delle soluzioni implementate e programmarne l’automazione con crontab.

Risk-014

Livello rischio: Medio
Trattamento: Mitigare

Ogni servizio che rimane in ascolto su una porta di rete crea una superficie di attacco. Gli attaccanti utilizzano scanner automatizzati che testano migliaia di server ogni giorno, cercando porte aperte su servizi vulnerabili. La mitigazione dei rischi 001, 005 e 010 risolvono in parte questa vulnerabilità. Consiglio di implementare un sistema di controllo automatizzato per porte ipv4 e ipv6 che rilevi l’apertura di porte non essenziali integrandolo con un sistema di alerting.

Creazione di uno script che faccia uso di netstat e ss per il controllo delle porte e lo confronti con il file allegato ports.txt per verificare che le porte aperte siano solo quelle necessarie al funzionamento dei servizi, abbia un sistema di alert via mail per nuove porte in ascolto (nel caso in cui il sistema dovesse venire compromesso e delle porte vengano aperte per avanzare nell’attacco), log delle modifiche se ci sono (complementare al sistema implementato nel Risk-007), e automazione con crontab.

Nota: Considerare l'implementazione di controlli di integrità per lo script stesso e, idealmente, l'invio degli alert attraverso canali indipendenti dal server monitorato.

Risk-015

Livello rischio: Medio
Trattamento: Mitigare

La gestione dei processi attivi è un elemento critico nella sicurezza di un server LAMP. Ogni processo in esecuzione rappresenta una potenziale superficie di attacco, specialmente quelli che mantengono connessioni di rete o accedono a risorse sensibili. Un processo compromesso può fungere da punto di ingresso per movimenti laterali all'interno del sistema.

Backup delle configurazioni attuali per rollback di emergenza

```
sudo cp /etc/systemd/system.conf /etc/systemd/system.conf.backup.$(date +%Y%m%d_%H%M%S)
```

_____

Creare una struttura di cartelle dedicata per memorizzare le analisi dei processi.

```
sudo mkdir -p /var/security/processes/{baseline,current,reports}
sudo chmod 750 /var/security/processes
sudo chown root:root /var/security/processes
```

_____

Generare una baseline completa dei processi attivi. Il comando ps aux mostra tutti i processi con informazioni dettagliate (utente, PID, utilizzo CPU/memoria, comando). L'opzione --sort=-%cpu ordina per utilizzo CPU decrescente, mentre --sort=-%mem ordina per utilizzo memoria decrescente.

```
ps aux --sort=-%cpu > /var/security/processes/baseline/processes_cpu_sorted.txt
ps aux --sort=-%mem > /var/security/processes/baseline/processes_mem_sorted.txt
```

_____

Analisi dei servizi systemd attivi systemctl list-units elenca le unità systemd. Il filtro --type=service mostra solo i servizi, mentre --state=active e --state=failed filtrano rispettivamente per servizi attivi e falliti.

```
systemctl list-units --type=service --state=active > /var/security/processes/baseline/active_services.txt
systemctl list-units --type=service --state=failed > /var/security/processes/baseline/failed_services.txt
```

_____

Mappatura delle connessioni di rete dei processi

```
netstat -tulpn > /var/security/processes/baseline/network_connections.txt
ss -tulpn > /var/security/processes/baseline/socket_statistics.txt
```

_____

Creare uno script che usi i files precedentemente creati per fare un’analisi dei processi attivi, che utilizzano la rete e non essenziali.

```
sudo nano /usr/local/bin/analyze-active-processes.sh
sudo chmod +x /usr/local/bin/analyze-active-processes.sh
```

_____

Analisi dei processi all’avvio. Questo comando elenca tutti i servizi systemd configurati per l'avvio automatico. È importante verificare che solo i servizi necessari siano abilitati.

```
systemctl list-unit-files --type=service --state=enabled > /var/security/processes/current/enabled_services.txt
```

_____

Disabilitare servizi non necessari per un server LAMP.Questi comandi disabilitano servizi tipicamente non necessari su un server LAMP di produzione. Bluetooth e CUPS sono per desktop, avahi è per discovery di rete, whoopsie e kerneloops sono per reporting errori.

```
sudo systemctl disable bluetooth.service
sudo systemctl disable cups.service
sudo systemctl disable avahi-daemon.service
sudo systemctl disable whoopsie.service
sudo systemctl disable kerneloops.service
```

_____

Creare configurazione systemd per limitare le risorse con limiti globali per prevenire fork bomb e consumo eccessivo delle risorse, controllo e timeout per i servizi che non rispondono.

```
sudo nano /etc/systemd/system.conf.d/99-resource-limits.conf
```

Applicare le modifiche a systemd

```
sudo systemctl daemon-reload
```

_____

Creare script per monitoraggio continuo dei processi con: funzione per rilevare nuovi processi, alert via email, monitoraggio processi con alto consumo di risorse e automatizzare con crontab. Consiglio l'utilizzo del comando comm per confrontare liste ordinate e awk per filtrare per soglie di utilizzo.

```
sudo nano /usr/local/bin/monitor-processes.sh
```

Risk-016

Livello rischio: Medio

Trattamento: Mitigare

La presenza di pagine di test o history rappresenta una vulnerabilità di information disclosure, dove files HTML o PHP possono rivelare informazioni confidenziali utilizzabili per attacchi successivi. Inoltre quando Apache mostra directory vuote o pagine di errore dettagliate fornisce agli attaccanti preziose informazioni sulla struttura del sistema.

Verificare la presenza di files HTML o PHP di test e rimuoverli dalle cartelle esposte al web. Il comando find esegue una ricerca ricorsiva nella directory /var/www (che include tutte le sottodirectory) per trovare file con estensioni web comuni. L'opzione -type f cerca solo file (non directory), mentre \(-name "*.html" -o -name "*.htm"... \) specifica i pattern di ricerca con OR logico. L'-exec ls -la {} \; esegue ls -la su ogni file trovato per mostrare permessi, proprietario e timestamp.

```
sudo find /var/www -type f \( -name "*.html" -o -name "*.htm" -o -name "*.php" -o -name "*.js" -o -name "*.css" \) -exec ls -la {} \; 2>/dev/null
```

_____

Modificare la configurazione di Apache per nascondere le informazioni del sistema. Questo include la disabilitazione del directory listing, che impedisce agli utenti di vedere il contenuto delle directory vuote.

Consiglio inoltre di creare anche una pagina index generica che sostituisce qualsiasi messaggio predefinito di Apache per una maggiore sicurezza.

```
sudo nano /var/www/html/index.html
sudo nano /etc/apache2/conf-available/web-files-protection.conf
sudo a2enconf web-files-protection
sudo apache2ctl configtest
sudo systemctl reload apache2
```

Risk-017

Livello rischio: Medio
Trattamento: Mitigare
Quando un attaccante compromette un sistema, la sua prima priorità è spesso cancellare le tracce del proprio passaggio. Senza un sistema di logging esterno, tutti i log risiedono sullo stesso server compromesso e possono essere facilmente eliminati.

Consiglio l'installazione di rsyslog-gnutls (se non ancora implementato) e rsyslog-relp perché rappresenta lo standard per il logging in ambiente Linux. La componente rsyslog-gnutls aggiunge la crittografia TLS, che trasforma una comunicazione potenzialmente intercettabile verso un server dedicato remoto per il logging in un canale sicuro. Configurato in modo da utilizzare un protocollo RELP (Reliable Event Logging Protocol) per garantire che ogni messaggio di log arrivi a destinazione.

Assicurarsi che meccanismi automatici per la classificazione e la retention differenziata dei log siano attivi e conformi alle normative vigenti.

È bene inoltre integrare un sistema di buffering locale nel caso in cui la connessione dovesse interrompersi temporaneamente esso garantisce che nessuna evidenza vada perduta anche durante le interruzioni di servizio. Assicurarsi che tale sistema abbia garanzie di ordinamento con timestamp precisi.

Creare uno script di monitoraggio automatico per verificare che la raccolta dei vari log e l'invio sia avvenuto con successo.

Risk-018

Livello rischio: Medio
Trattamento: Mitigare
L'errata gestione degli utenti e dei gruppi utenti è una delle maggiori vulnerabilità su un sistema, spesso utilizzata per compiere movimenti di elevazione dei privilegi. Pertanto è necessario prestare la massima attenzione ai permessi dei singoli utenti, gruppi di utente e account di servizio.

Dato l’allegato etcpassw.txt sono stati rilevati alcuni servizi non in uso che possono essere rimossi, previa verifica, il seguente comando effettua un controllo

```
systemctl status games 2>/dev/null || echo "games servizio non trovato"
systemctl status news 2>/dev/null || echo "news servizio non trovato"
systemctl status uucp 2>/dev/null || echo "uucp servizio non trovato"
systemctl status irc 2>/dev/null || echo "irc servizio non trovato"
```

Se questi servizi risultano non trovati, i relativi account, possono essere rimossi

```
sudo userdel games
sudo userdel news
sudo userdel uucp
sudo userdel irc
```

_____

Altri account potenzialmente obsoleti sono: il servizio Landscape, servizio USB multiplexing

```
sudo userdel landscape
sudo userdel usbmux
```

____

Disabilitare account di servizio non utilizzati, come ad esempio quello per le stampanti

```
sudo usermod -s /usr/sbin/nologin -L lp
```

Risk-019

Livello rischio: Medio

Trattamento: Mitigare

Sul server sono presenti alcuni compiler (allegato compiler.txt) che potenzialmente potrebbero essere sfruttati da un attaccante che sia riuscito ad ottenere l'accesso con un utente avente permessi limitati per elevare i privilegi.

Consiglio di limitare i permessi dei compiler al gruppo utenti amministratore ed aggiungere una policy per prevenire l'installazione accidentale di compiler e valutare l’installazione e configurazione di AppArmor per una protezione aggiuntiva..

Risk-020

Livello rischio: Medio

Trattamento: Mitigare

Ogni protocollo di rete abilitato ma non utilizzato rappresenta un potenziale vettore di attacco. Disattivare i protocolli non essenziali.

Dalle analisi (allegato protocolli.txt) risultano diversi protocolli di rete non essenziali, l’opzione più sicura è disabilitarli.

Creare un backup delle configurazioni attuali (con timestamp)
sudo cp -r /etc/modprobe.d /etc/modprobe.d.backup.$(date +%Y%m%d_%H%M%S)

_____
Creare il file di configurazione per disabilitare i protocolli e aggiungere alla blacklist: dccp, sctp, rds, tipc, atm, can, x25, netrom, rose, ax25, ieee802154
sudo nano /etc/modprobe.d/blacklist-unused-protocols.conf

_____
Valutare l'implementazione di una configurazione firewall per bloccare i protocolli al livello di rete per un'ulteriore protezione.

Risk-021

Livello rischio: Basso
Trattamento: Mitigare
La umask è un valore che determina i permessi assegnati ad un nuovo file. Dei permessi troppo ampi potrebbero essere utilizzati in alcuni exploit molto specifici per permettere un elevazione di privilegi.
La mitigazione prevede l'impostazione di permessi più restrittivi per i nuovi files creati

Cambiare la configurazione umask da 002 attuale a 027 mantenendo rwx per il proprietario ma rimuovendo il permesso di scrittura per il gruppo e tutti i permessi dagli altri utenti.
sudo nano /etc/login.defs
sudo nano /etc/profile
sudo nano /etc/bash.bashrc

_____
Impostare una umask ancora più restrittiva per l'utente root, ad esempio 077
sudo nano /root/.bashrc

_____
Configurare una umask meno restrittiva (UMask=0022) per apache che necessita di permessi per poter servire correttamente le pagine web
sudo mkdir -p /etc/systemd/system/apache2.service.d/
sudo nano /etc/systemd/system/apache2.service.d/umask.conf

E una più restrittiva per MariaDB (UMask=0077)
sudo mkdir -p /etc/systemd/system/mariadb.service.d/
sudo nano /etc/systemd/system/mariadb.service.d/umask.conf

Ricaricare systemd per applicare le modifiche
sudo systemctl daemon-reload

Risk-022

Livello rischio: Basso
Trattamento: Mitigare
Il driver usb-storage permette al sistema di riconoscere e montare automaticamente dispositivi di memorizzazione USB. Disabilitandolo, si elimina la possibilità di attacchi tramite chiavette USB malevole.

Creare un file che impedisce al kernel di caricare il driver USB (blacklist usb-storage)
```
sudo nano /etc/modprobe.d/blacklist-usb-storage.conf
```

_____
Rimuovere il modulo attivo
```
sudo modprobe -r usb-storage
```

_____
Aggiornare l'intramfs per rendere la blacklist attiva all'avvio del sistema
```
sudo update-initramfs -u
```

_____
Verifica che il driver non sia più caricato
```
lsmod | grep usb_storage
```

Risk-023

Livello rischio: Basso
Trattamento: Accettare
Il rischio dell'accesso fisico al sistema non è mai del tutto eliminabile. Al momento è accettabile nella situazione corrente ma si valuti un trasferimento del rischio a terzi con una migrazione del server su un servizio cloud dedicato.

Delle mitigazioni temporanee che possono essere implementate sono
- Impostare una password per il boot GRUB
- Crittografia delle partizioni sensibili
- Chiudere a chiave la porta della stanza contenente il server
- Implementare un sistema di monitoraggio con alert in caso di anomalie

---

Risk-024

Livello rischio: Basso
Trattamento: Mitigare
La presenza di file obsoleti nel sistema costituisce una vulnerabilità, poiché possono contenere pacchetti datati con vulnerabilità note, file con dati sensibili, causare problemi di gestione dello spazio e, a lungo termine, rendere difficoltosa l'identificazione dei file legittimi rispetto a quelli obsoleti. Questo rischio è stato mitigato come parte di mitigazioni dei rischi precedenti nello stesso documento.

Consiglio l’implementazione di uno script da automatizzare con crontab per l’eliminazione dei files obsoleti con l’identificazione di: files temporanei più vecchi di 30 giorni, cache files di sistema, files di backup più vecchi di un anno

_____

Identificazione di pacchetti obsoleti con
apt autoremove

_____

Controllare la configurazione di deborphan (già suggerito RISK-009) e logrotate (RISK-002).

Rischio Residuo

L'analisi del rischio residuo relativa al server LAMP Ubuntu 22.04.2 evidenzia un significativo miglioramento del profilo di sicurezza a seguito dell'implementazione delle ventiquattro mitigazioni proposte. Il punteggio complessivo del rischio residuo si attesta su un livello basso-medio, denotando una sostanziale riduzione rispetto alla condizione iniziale, caratterizzata da rischi critici con punteggi fino a 25. Gli elementi principali di rischio che permangono sono correlati all'accesso fisico al sistema e alle vulnerabilità zero-day non ancora identificate, fattori intrinsecamente difficili da eliminare completamente mediante controlli tecnici. Le mitigazioni adottate, che spaziano dalla configurazione del firewall all'hardening SSH, dal sistema di backup crittografato al Web Application Firewall, hanno convertito rischi critici in vulnerabilità gestibili. Il rischio residuo necessita di un approccio di gestione continua basato su monitoraggio proattivo, aggiornamenti tempestivi e procedure di incident response ben definite. La strategia di trasferimento del rischio verso provider cloud professionali, raccomandata per i rischi inerenti alla sicurezza fisica, rappresenta l'evoluzione naturale per il raggiungimento di un livello di sicurezza di tipo enterprise. L'accettazione formale di questo livello di rischio residuo da parte del management è imprescindibile, considerando che esso rappresenta un equilibrio ottimale tra protezione dei dati, conformità al GDPR e operatività del sistema. La revisione trimestrale delle metriche di sicurezza e l'aggiornamento continuo delle baseline di controllo garantiranno il mantenimento di questo profilo di rischio nel tempo. In sintesi, il server risulta ora adeguatamente protetto per un ambiente di produzione, con un rischio residuo gestibile e documentato.

Considerazioni ulteriori e best practice

Per rafforzare la sicurezza di un server LAMP, oltre alle mitigazioni tecniche, è cruciale adottare diverse best practice. È fondamentale la formazione e consapevolezza del personale, per mitigare i rischi da errore umano e negligenza. Bisogna applicare il principio del privilegio minimo, concedendo solo i permessi strettamente necessari a utenti e processi per ridurre la superficie d'attacco. La segmentazione della rete, tramite VLAN o firewall, limita il movimento laterale degli attaccanti in caso di compromissione.

Il monitoraggio continuo con sistemi SIEM e un piano di risposta agli incidenti ben definito sono essenziali per rilevare proattivamente le minacce e gestire efficacemente le violazioni. Test di penetrazione e vulnerability assessment regolari permettono di identificare e correggere le debolezze prima che vengano sfruttate. È imperativo crittografare i dati sensibili a riposo (data at rest) e gestire le configurazioni in modo rigoroso, utilizzando strumenti di automazione per garantire coerenza e conformità.

Integrare la sicurezza nel ciclo di vita dello sviluppo del software (DevSecOps) è cruciale per le applicazioni interne. Infine, un robusto processo di gestione delle patch e degli aggiornamenti è indispensabile per proteggere il server dalle vulnerabilità note. L'applicazione congiunta di queste pratiche crea una difesa a più strati, migliorando la resilienza e la conformità al GDPR.

Data: 28/07/2025

_____ Andrea Emanuele Peluso