Engineer

Посилення захисту

усунення того, чим міг би скористатися нападник

Що це таке

Посилення захисту — це налаштування системи так, щоб вона відкривала й дозволяла лише те, чого вимагає її робота: менше запущених служб, менше портів, що слухають, менше прав для кожного процесу і суворіші усталені значення для всього, що лишається. Програми до і після ті самі. Змінюється інше: скільки з них досяжно для зловмисника і як далеко його заведе одна точка опори. В англомовній документації це hardening, тож інженери часто кажуть просто «харденінг» або уточнюють шар: посилення захисту ОС, сервера чи SSH.

Типовий прохід посилення захисту на Linux-сервері зачіпає кілька шарів одночасно. Віддалений вхід приймає ключі й відхиляє паролі. Обліковий запис root не може входити напряму. Пакети, якими ніхто не користується, видаляють, а не просто зупиняють. Кожна довготривала служба дістає власного користувача без привілеїв, файлову систему лише для читання і перелік системних викликів, які їй дозволено робити. Брандмауер пропускає тільки ті кілька портів, заради яких хост існує. Екзотики тут немає. Уся складність у тому, щоб зробити все це на кожному хості й тримати таким і надалі.

Посилення захисту часто плутають із суміжними практиками. Оновлення виправляє конкретну ваду в коді, коли виправлення вже вийшло. Моніторинг і виявлення вторгнень помічають атаку, що триває. Брандмауер вирішує, хто може дістатися порту. Посилення захисту вирішує, що програмі за цим портом дозволено робити після того, як до неї дісталися, і чи варто цьому портові взагалі бути відкритим. Опубліковані базові конфігурації, як-от CIS Benchmarks, перетворюють це на контрольні списки для кожної ОС і продукту. Як нижня планка вони корисні, як фінішна риса — оманливі.

Застосуйте поняття до рішення

Застосуйте «Що таке посилення захисту?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Облікові записи, паролі й права налаштовані так, щоб доступ відповідав роботі, а звільнені втрачали його того ж дня.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Написала 1 000 правил RBAC для Linux-серверів — 1,000 правил доступу написано, щоб усунути вразливість

Написала 1 000 правил RBAC для Linux-серверів Зв'язатися з нами →

Чому це важливо

Справжнє рішення стосується того, куди спрямувати обмежений бюджет безпеки, і посилення захисту має рідкісну властивість: воно спрацьовує проти вразливостей, які знайдуть лише наступного року. Оновлення захищає від однієї відомої вразливості. Служба з посиленим захистом обмежує можливості процесу, вразливістю якого скористався зловмисник. Наприклад, заборона запису у виконувану пам’ять, відкриття сирих сокетів і доступу до домашніх каталогів має на меті обмежити наслідки вади віддаленого виконання коду. Оновлення закривають діри, посилення захисту знецінює кожну з них. Потрібні обидва, і саме посилення захисту виграє час у ті дні, що минають між оприлюдненням вразливості й виходом виправлення.

Ціна тут — переважно інженерний час і ризик щось зламати, а не ліцензії. Кожне обмеження — це ставка на те, що службі ніколи не знадобиться відібране, і частина ставок програє: нічна задача резервного копіювання, ротація журналів, що надсилає сигнал, плагін, який запускає зовнішню команду. Тому ручне посилення захисту на живому сервері зазвичай буксує. Людина, яка ним займається, одного разу обпікається на обмеженні, що поклало службу о третій ночі, і далі вже не затягує гайки. Коли ж ту саму роботу записано як код, переглянуто, версіоновано й застосовано однаково на кожному перезбиранні, її дешево повторити і безпечно відкотити.

Для замовника практичні питання такі: коли це робити і як переконатися, що зроблено. Відповідь на перше — під час створення хоста, як частину автоматизації, що його будує, бо контрольний список, одноразово застосований до живої машини, за кілька місяців розповзається. Відповідь на друге — докази з працюючої системи, а не з репозиторію конфігурацій. Аудитори дедалі частіше просять обидва: задокументовану базову конфігурацію і підтвердження, що продакшн їй відповідає.

Як це виглядає на практиці

На власному робочому хості компанії SSH — єдиний інтерактивний шлях усередину, а його налаштування пише автоматизація. У вересні 2026 року звіт про доступ зчитав фактичні параметри, які застосовує демон й показав, що дві із задуманих директив перекрито: усталені значення операційної системи завантажувалися раніше за файл-вставку компанії, а для цих параметрів перемагає перше значення. Порядок виправив префікс імені файлу, що сортується раніше. Тривалою зміною стала триетапна перевірка 24 директив SSH під час кожного застосування конфігурації: файл-кандидат перевіряється на синтаксис до встановлення, зібране налаштування — після, а фактичні налаштування демона зчитуються й звіряються з усіма 24. Ще дві директиви навмисно не включено, бо демон їх більше не реалізує.

На тому самому хості працюють 13 юніт-файлів, які встановлює його власна автоматизація, і вони виконувалися з усталеними правами менеджера служб. Кожен шаблон юніта отримав стелю привілеїв, фільтр системних викликів, обмежені родини адрес і пам’ять, яка може бути записуваною або виконуваною, але ніколи водночас. За шкалою самого менеджера служб робота з пісочницями знизила оцінку служби dead man’s switch з 9,6 до 1,5, оцінку доступного з інтернету git-сервера з 8,3 до 1,5, а двох юнітів резервного копіювання — з 9,6 до 2,3 і 2,5. Під час цієї роботи з’ясувалося, що одну директиву написано з помилкою у трьох шаблонах: менеджер служб записує в журнал невідомий ключ і все одно запускає юніт. Тепер під час кожного застосування конфігурації всі юніти перевіряє власний валідатор менеджера служб.

Огляд захищеності в серпні 2026 року перевіряв, чи працює захист SSH від перебору паролів, і для цього читав систему в роботі. Усі три засоби захисту були бездіяльними від створення хоста: налаштований jail стежив за портом 22, тоді як демон слухав 1986, обмеження частоти на брандмауері стояло нижче ширшого правила, що спрацьовувало першим, а дія блокування викликала програму фільтра пакетів, яку система пакунків лише рекомендує і якої на хості бракувало. Огляд залишив два правила, що й досі керують репозиторієм: засіб безпеки отримує виконувану перевірку, а не коментар, а брандмауер перевіряють за розташуванням правил, а не за самою їхньою наявністю.

Де виникають проблеми

Параметр записано у файл, а служба діє інакше. Багато демонів читають кілька файлів і беруть перше або останнє значення кожного ключа. Директива посилення захисту у файлі-вставці може поступитися усталеному значенню постачальника, яке завантажилося раніше. Запитайте саму програму, що вона робитиме насправді, для SSH через `sshd -T`, і порівнюйте відповідь із задуманими значеннями на кожному прогоні.

Одруківка перетворює засіб контролю на коментар. Менеджери служб і чимало демонів записують невідомий ключ у журнал і працюють далі. Юніт стартує, панель зелена, а обмеження відсутнє. Запускайте власний перевіряч парсера як частину розгортання, щоб невідома директива зупиняла викочування.

```
# Що робитиме демон SSH, коли прочитано всі файли з Include
sshd -T | grep -E '^(passwordauthentication|permitrootlogin) '
# Наскільки відкрита пісочниця юніта, за шкалою від 0,0 до 10,0
systemd-analyze security --no-pager example.service
# Невідомі директиви та інші помилки юніт-файлу
systemd-analyze verify /etc/systemd/system/example.service
```

Захист стереже не ті двері. Порт переносять, а правило блокування далі стежить за старим. Обмеження частоти додають під широке дозвільне правило, яке вже спрацювало. Засіб контролю звітує про справність, бо він запущений, а це ще не означає, що він бодай раз спрацьовує. Перевіряйте кожен засіб за його дією: навмисно активуйте його ззовні й переконайтеся, що пакет відкинуто.

Пісочниця кладе службу вночі. Фільтр системних викликів або шлях лише для читання зачіпає рідкісну гілку коду, як-от щотижневий експорт чи хук резервного копіювання, і збій проявляється через години після зміни, під час виконання, а не під час розбору. Затягуйте обмеження поетапно, проганяйте кожну заплановану задачу після кожного етапу і стежте в журналі за процесами, які вбив фільтр.

Метою стає оцінка за бенчмарком. Хост може пройти кожен рядок загального контрольного списку, поки його єдиний доступний з інтернету застосунок працює від root із правом запису до власного коду. Почніть

із того, що досяжне з мережі, і з того, чого може торкнутися кожен такий процес, а бенчмарк використовуйте, щоб упіймати пропущене під час цього огляду.

Що почитати далі

- Hardening, in the NIST glossary — визначення, яким користується NIST, з посиланням на публікацію, звідки воно походить.
- sshd_config(5) — усі директиви сервера SSH і правило, за яким діє перше отримане значення ключового слова.
- systemd.exec(5) — директиви пісочниці для служб: фільтри системних викликів, родини адрес, пам’ять «запис або виконання».
- systemd-analyze(1) — оцінка відкритості в команді `security` і команда `verify`, що ловить невідомі директиви.
- CIS Benchmarks — узгоджені базові конфігурації для операційних систем, хмарних платформ і серверного програмного забезпечення.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке посилення захисту?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/hardening/