تأمين تطبيق Odoo باستخدام Nginx كـ Reverse Proxy

المفتوح المصدر، والذي يُستخدم كنظام إدارة موارد Odoo في هذا المشروع، تم تنفيذ استراتيجية أمان فعالة لحماية تطبيق عن الوصول Odoo لعزل تطبيق Reverse Proxy كـ Nginx تعتمد هذه الاستراتيجية على استخدام خادم .(ERP) مؤسسية المباشر من الإنترنت، وتوجيه كل الترافيك عبر طبقة وسيطة أكثر أمانًا.

على المنفذ 8069 ويكون قابلًا للوصول من أي عنوان خارجي إذا لم يتم تأمينه. هذا يشكّل Odoo في الوضع الافتراضي، يعمل أو استغلال ثغرات أمنية في الإصدارات القديمة من ،brute-force مخاطرة أمنية، حيث قد يتم استغلال هذا المنفذ في هجمات وبالتالي يمنع أي وصول خارجي ،(localhost) ليعمل فقط على العنوان المحلي Odoo لتقليل هذه المخاطر، تم إعداد .Odoo مباشر.

ومن ثم إعادة ،(HTTPS) أو 443 (HTTP) وتكوينه لاستقبال جميع طلبات المستخدمين على البورت 80 Nginx ثم تم تثبيت عبر `127.0.0.1:8069`. باستخدام هذه الطبقة، يمكن تنفيذ سياسات أمان إضافية مثل تقييد عناوين Odoo توجيهها داخليًا إلى لتشفير الاتصال. Let’s Encrypt باستخدام HTTPS تسجيل الطلبات، وتفعيل بروتوكول IP،

وهي مبدأ أمني يُقصد به تقليل عدد النقاط التي ،(Attack Surface) "الفائدة الأمنية من هذا التكوين هي تقليل "سطح الهجوم محميًا من الاتصال الخارجي Odoo كحاجز أمامي، يكون Nginx يمكن أن يستغلها المهاجمون للدخول إلى النظام. بوجود ويجعل الوصول غير المصرّح به شبه مستحيل ،Nmap أو Shodan المباشر، مما يمنع اكتشافه تلقائيًا من أدوات المسح مثل دون المرور بـ Nginx.

التي تحتوي Odoo هذا النموذج يُعتبر من أفضل ممارسات الأمان عند تشغيل تطبيقات ويب، خصوصًا الأنظمة الحساسة مثل على بيانات الشركات والموارد المالية والإدارية.

الخطوات العملية تم تنفيذها كالتالي:
1. تهيئة Odoo ليستمع فقط على العنوان المحلي 127.0.0.1 من خلال إعداد ملف odoo.conf.
2. تثبيت وتشغيل خادم Nginx.
3. إنشاء إعداد Nginx يعيد توجيه الطلبات من الإنترنت إلى Odoo داخليًا.
4. اختبار إعدادات الأمن باستخدام أدوات مثل netstat. و lsof
5. تفعيل HTTPS باستخدام Certbot لحماية الاتصالات بين الخادم والمستخدم.

النتيجة النهائية أن Odoo أصبح يعمل خلف Nginx عبر HTTPS، فقط، وجميع الاتصالات مؤمنة بالكامل مع مراقبة كاملة، لجميع الطلبات وتصفية للمرور بناءً على السياسات المطبقة في Nginx.