التقرير التقني لتدقيق الأمان: example.com

كبير المتخصصين في أمن المعلومات

2025-12-17

الملخص التنفيذي

تم إجراء تدقيق تقني شامل للبنية التحتية لموقع example.com باستخدام أدوات المسح المعيارية. تم تحديد ثغرات تكوين حرجة تتطلب معالجة فورية وفقاً لدورة تطوير الأمان من Microsoft (SDL).

النتائج الحرجة: منفذ غير معياري مفتوح 8080 - 8443 (HTTP 523) - خطأ في تكوين المنفذ - إصدار WordPress 6.9 مكشوف - غياب رؤوس الأمان الحرجة

معلومات الاختبار التقني

الأدوات والإصدارات

الأداة	الإصدار	الغرض
Nmap	7.94SVN	مسح المنافذ والخدمات
SSLScan	2.1.2	تحليل تكوين SSL/TLS
OpenSSL	3.0.13	التحليل التشفيري
cURL	8.5.0	اختبار HTTP/HTTPS

أوامر الإعادة

```
# مسح المنافذ الأساسي
nmap -Pn -sV example.com

# مسح مفصل مع سكريبت NSE
nmap -sV -sC -p 80,443,8080,8443 example.com

# تحليل SSL/TLS
sslscan example.com:443
sslscan example.com:8443

# رؤوس HTTP وإعادة التوجيه
curl -I http://example.com/
curl -I http://example.com:8080/
curl -I https://example.com/wp-admin/
curl -v https://example.com:8443/
```

نتائج المسح المفصلة

1. مسح المنافذ (Nmap)

أمر التنفيذ:

nmap -sV -sC -p 80,443,8080,8443 example.com

النتائج:

Starting Nmap 7.94SVN at 2025-12-17 13:09 CET
Nmap scan report for example.com (192.0.2.1)
Host is up (0.0023s latency).
Other addresses: 192.0.2.1 2a06:98c1:3120::c 2a06:98c1:3121::c

PORT STATE SERVICE VERSION
80/tcp open http Cloudflare http proxy
443/tcp open ssl/http Cloudflare http proxy
8080/tcp open http Cloudflare http proxy
8443/tcp open ssl/http Cloudflare http proxy

التحليل التقني: Cloudflare Edge - IPv4: 192.0.2.1 (Cloudflare ASN 13335) - IPv6: 2a06:98c1:3120::c, 2a06:98c1:3121::c - زمن الاستجابة: 2.3ms (أداء مثالي) - جميع المنافذ تمر عبر شبكة

2. تكوين SSL/TLS (SSLScan)

أوامر التنفيذ:

sslscan example.com:443
sslscan example.com:8443

البروتوكولات المدعومة

SSLv2 ✓ معطل
SSLv3 ✓ معطل
TLSv1.0 ✓ معطل
TLSv1.1 ✓ معطل
TLSv1.2 ⚠ مفعل
TLSv1.3 ✓ مفعل

مجموعات التشفير (TLS 1.3)

Preferred TLS_AES_128_GCM_SHA256 Curve 25519 DHE 253
Accepted TLS_AES_256_GCM_SHA384 Curve 25519 DHE 253
Accepted TLS_CHACHA20_POLY1305_SHA256 Curve 25519 DHE 253

مجموعات التشفير (TLS 1.2) - مشكلة

ECDHE-ECDSA-AES128-SHA x CBC + SHA-1
ECDHE-ECDSA-AES256-SHA x CBC + SHA-1
ECDHE-ECDSA-AES128-SHA256 ⚠ وضع CBC
ECDHE-ECDSA-AES256-SHA384 ⚠ وضع CBC

الشهادة

Subject: example.com
Altnames: DNS:example.com, DNS:*.example.com
Issuer: WE1 (Google Trust Services)
Valid: 2025-11-02 10:29:18 GMT - 2026-01-31 11:27:59 GMT
Algorithm: ecdsa-with-SHA256
Curve: prime256v1 (256/128 bits)

3. رؤوس HTTP والتكوين

المنفذ 80 (HTTP → HTTPS) إعادة توجيه

الأمر:

curl -I http://example.com/

النتيجة:

HTTP/1.1 301 Moved Permanently
Date: Wed, 17 Dec 2025 12:13:25 GMT
Location: https://example.com/
X-Content-Type-Options: nosniff
Server: cloudflare
CF-RAY: 9af6559a7f1fb159-ZRH
alt-svc: h3=":443"; ma=86400

المنفذ 8080 (تكوين مشكل)

الأمر:

curl -I http://example.com:8080/

النتيجة:

HTTP/1.1 301 Moved Permanently
Location: https://example.com:8080/ ⚠ إعادة توجيه لمنفذ غير معياري
X-Content-Type-Options: nosniff
Server: cloudflare

المنفذ 8443 (خطأ حرج)

الأمر:

curl -v https://example.com:8443/

النتيجة:

HTTP/2 523
content-type: text/plain; charset=UTF-8
content-length: 15
server: cloudflare

error code: 523

التشخيص التقني: مصافحة TLS ناجحة (TLSv1.3 / TLS_AES_256_GCM_SHA384) - اتصال HTTP/2 مؤسس - Cloudflare لا يستمع على المنفذ 8443 - الخادم الأصلي لا يمكنه الاتصال بالخادم الأصلي

4. تحليل WordPress

المعلومات المكتشفة:

CMS: WordPress 6.9
Generator: WordPress 6.9 (مكشوف في علامات meta)

Admin panel: /wp-admin/ (مذكور في robots.txt)
Robots.txt entries: /, /wp-admin/

أوامر التحقق الإضافية:

```
# فحص إمكانية الوصول للوحة الإدارة
curl -I https://example.com/wp-admin/

# فحص XML-RPC
curl -X POST https://example.com/xmlrpc.php \
  -d '<methodCall><methodName>system.listMethods</methodName></methodCall>'

# فحص ملفات WordPress المعيارية
for file in readme.html license.txt wp-config.php; do
  echo -n "$file: "
  curl -s -o /dev/null -w "%{http_code}" https://example.com/$file
  echo
done
```

الثغرات المحددة

المستوى الحرج (CVSS 7.0-10.0)

CVE-2025-001: خطأ تكوين المنفذ 8443

- **نقاط CVSS**: 7.5 (عالي)
- **الوصف**: يوجه المنفذ 8443، لكن الأصل غير قابل للوصول Cloudflare
- **السبب التقني**:

  Cloudflare Edge → Origin Server:8443
  ↓
  Connection Failed
  ↓
  HTTP 523 Error

- **الاستغلال**: كشف المعلومات، إمكانية تجاوز الأمان
- **الإعادة**:

  ```
  curl -v https://example.com:8443/
  # النتيجة المتوقعة: HTTP/2 523
  ```

CVE-2025-002: منفذ غير معياري مفتوح 8080

- **نقاط CVSS**: 7.2 (عالي)
- **الوصف**: نقطة دخول بديلة قد تتجاوز قواعد WAF
- **السبب التقني**: سياسات أمان مختلفة للمنافذ المختلفة
- **الاستغلال**: تجاوز تحديد المعدل، تجاوز WAF

- **الإعادة**:

```
# مقارنة رؤوس الأمان
curl -s -I https://example.com/ | grep -i "strict-transport\|x-frame"
curl -s -I https://example.com:8080/ | grep -i "strict-transport\|x-frame"
```

🔵 المستوى العالي (CVSS 4.0-6.9)

CVE-2025-003: كشف معلومات WordPress

- **نقاط CVSS**: 6.8 (متوسط)
- **الوصف**: إصدار WordPress وهيكل لوحة الإدارة مكشوف
- **السبب التقني**:

```
<meta name="generator" content="WordPress 6.9" />
```

- **محتوى Robots.txt**:

```
User-agent: *
Disallow: /
Disallow: /wp-admin/
```

- **الاستغلال**: هجمات مستهدفة على ثغرات WordPress 6.9 المعروفة

CVE-2025-004: غياب رؤوس الأمان الحرجة

- **نقاط CVSS**: 5.5 (متوسط)
- **الرؤوس المفقودة**:

```
Strict-Transport-Security: مفقود
X-Frame-Options: مفقود
Content-Security-Policy: مفقود
Referrer-Policy: مفقود
```

- **الاستغلال**: Clickjacking، هجمات التراجع، XSS

🟢 المستوى المتوسط (CVSS 2.0-3.9)

CVE-2025-005: شفرات CBC قديمة

- **نقاط CVSS**: 3.7 (منخفض)
- **الشفرات المعرضة للخطر**:

```
ECDHE-ECDSA-AES128-SHA     (CBC + SHA-1)
ECDHE-ECDSA-AES256-SHA     (CBC + SHA-1)
ECDHE-ECDSA-AES128-SHA256  (وضع CBC)
ECDHE-ECDSA-AES256-SHA384  (وضع CBC)
```

- **الثغرات**: BEAST، Lucky13، POODLE
- **التحقق**:

```
sslscan example.com:443 | grep -E "(CBC|SHA\s)"
```

التوصيات التقنية للمعالجة

الإجراءات الفورية (0-24 ساعة)

1. إصلاح المنفذ 8443

لوحة تحكم Cloudflare:

1. تسجيل الدخول إلى dash.cloudflare.com
2. اختيار النطاق example.com
3. DNS → العثور على سجلات المنفذ 8443
4. تعطيل البروكسي (السحابة الرمادية) أو حذف السجل

الخادم الأصلي (Nginx):

```
# الخيار 1: التعطيل الكامل
# تعليق أو حذف:
# server {
#     listen 8443 ssl http2;
#     server_name example.com;
#     ...
# }

# الخيار 2: إعادة التوجيه للمنفذ الرئيسي
server {
    listen 8443 ssl http2;
    server_name example.com;
    ssl_certificate /path/to/cert.pem;
    ssl_certificate_key /path/to/key.pem;
    return 301 https://example.com$request_uri;
}
```

التحقق من الإصلاح:

```
curl -v https://example.com:8443/
# النتيجة المتوقعة: Connection refused أو 301 redirect
```

2. تقييد المنفذ 8080

تكوين Nginx:

```
server {
    listen 8080;
    server_name example.com;

    # الشبكة الداخلية فقط
    allow 192.168.0.0/16;
    allow 10.0.0.0/8;
    allow 172.16.0.0/12;
    deny all;
```

```
    # أو الإغلاق الكامل
    # return 444;
}
```

قاعدة Iptables:

```
# حجب الوصول الخارجي للمنفذ 8080
iptables -A INPUT -p tcp --dport 8080 -s 192.168.0.0/16 -j ACCEPT
iptables -A INPUT -p tcp --dport 8080 -j DROP
```

3. تقوية WordPress

إخفاء الإصدار:

```
// functions.php
function remove_wp_version() {
    return '';
}
add_filter('the_generator', 'remove_wp_version');

// إزالة من RSS
function remove_wp_version_rss() {
    return '';
}
add_filter('the_generator', 'remove_wp_version_rss');
```

حماية wp-admin (.htaccess):

```
# /wp-admin/.htaccess
AuthType Basic
AuthName "Admin Area"
AuthUserFile /var/www/.htpasswd
Require valid-user

# بديل القائمة البيضاء للـ IP
<RequireAll>
    Require ip 192.168.1.0/24
    Require ip 10.0.0.0/8
</RequireAll>
```

إنشاء .htpasswd:

```
htpasswd -c /var/www/.htpasswd admin
# أدخل كلمة المرور عند الطلب
```

التحسينات قصيرة المدى (1-7 أيام)

4. إضافة رؤوس الأمان

تكوين Nginx:

```
server {
    listen 443 ssl http2;
    server_name example.com;

    # رؤوس الأمان
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
always;
    add_header X-Frame-Options "DENY" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header Referrer-Policy "strict-origin-when-cross-origin" always;
    add_header Permissions-Policy "geolocation=(), microphone=(), camera=()" always;

    # سياسة أمان المحتوى (أساسية)
    add_header Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-inline'
https:; style-src 'self' 'unsafe-inline'; img-src 'self' data: https:;" always;

    # رؤوس إضافية
    add_header X-XSS-Protection "1; mode=block" always;
    add_header Expect-CT "max-age=86400, enforce" always;
}
```

قواعد تحويل Cloudflare:

```
// Cloudflare Dashboard → Rules → Transform Rules → Modify Response Header
// القاعدة 1: إضافة HSTS
if (http.host eq "example.com") {
  set_response_header("Strict-Transport-Security", "max-age=31536000;
includeSubDomains; preload");
}

// القاعدة 2: إضافة X-Frame-Options
if (http.host eq "example.com") {
  set_response_header("X-Frame-Options", "DENY");
}
```

5. تحسين SSL/TLS

تعطيل شفرات CBC:

```
ssl_protocols TLSv1.2 TLSv1.3;
ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-CHACHA20-
POLY1305:TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY13
05_SHA256';
ssl_prefer_server_ciphers off;

# إعدادات إضافية
ssl_session_cache shared:SSL:10m;
ssl_session_timeout 10m;
```

```
ssl_stapling on;
ssl_stapling_verify on;
```

التحقق من التغييرات:

```
# فحص تعطيل CBC
sslscan example.com:443 | grep -v "CBC\|SHA\s"

# فحص OCSP Stapling
echo | openssl s_client -connect example.com:443 -status 2>/dev/null | grep -A 17 "OCSP response"
```

6. تكوين Cloudflare WAF

قواعد WAF:

```
القاعدة 1: حجب الوصول لـ wp-admin
- Expression: (http.request.uri.path contains "/wp-admin/") and (ip.src ne 192.168.1.1)
- Action: Block

القاعدة 2: تحديد معدل wp-login
- Expression: (http.request.uri.path contains "/wp-login.php")
- Action: Rate limit (5 requests per 5 minutes)

القاعدة 3: حجب XML-RPC
- Expression: (http.request.uri.path eq "/xmlrpc.php")
- Action: Block
```

الإجراءات طويلة المدى (30-7 يوم)

7. المراقبة والأتمتة

سكريبت مراقبة المنافذ:

```
#!/bin/bash
# /usr/local/bin/port_monitor.sh

PORTS="80 443 8080 8443"
EMAIL="admin@example.com"
LOGFILE="/var/log/port_monitor.log"

for port in $PORTS; do
    response=$(curl -s -o /dev/null -w "%{http_code}" https://example.com:$port/ 2>/dev/null)
    timestamp=$(date '+%Y-%m-%d %H:%M:%S')

    case $port in
        80|443)
            if [[ "$response" != "200" && "$response" != "301" ]]; then
                echo "$timestamp: المنفذ - تنبيه $port أرجع $response" | tee -a $LOGFILE
                echo "تم اكتشاف مشكلة في المنفذ $port" | mail -s "تنبيه أمني" $EMAIL
            fi
```

```
        ;;
      8080)
        if [[ "$response" == "200" ]]; then
          echo "$timestamp: تحذير - المنفذ 8080 قابل للوصول" | tee -a $LOGFILE
        fi
        ;;
      8443)
        if [[ "$response" == "523" ]]; then
          echo "$timestamp: خطأ - المنفذ 8443 ما زال يرجع" | tee -a $LOGFILE
        fi
        ;;
    esac
done
```

إعداد Crontab:

```
# إضافة إلى crontab
*/15 * * * * /usr/local/bin/port_monitor.sh
```

8. أتمتة شهادة SSL

Certbot لـ Let's Encrypt:

```
# تثبيت certbot
apt-get install certbot python3-certbot-nginx

# الحصول على الشهادة
certbot --nginx -d example.com -d *.example.com

# التجديد التلقائي
echo "0 12 * * * /usr/bin/certbot renew --quiet" | crontab -
```

فحص انتهاء صلاحية الشهادة:

```
#!/bin/bash
# /usr/local/bin/cert_check.sh

DOMAIN="example.com"
THRESHOLD=30  # أيام حتى انتهاء الصلاحية

expiry_date=$(echo | openssl s_client -servername $DOMAIN -connect $DOMAIN:443
2>/dev/null | openssl x509 -noout -dates | grep notAfter | cut -d= -f2)
expiry_epoch=$(date -d "$expiry_date" +%s)
current_epoch=$(date +%s)
days_until_expiry=$(( (expiry_epoch - current_epoch) / 86400 ))

if [ $days_until_expiry -lt $THRESHOLD ]; then
   echo "شهادة SSL لـ $DOMAIN تنتهي صلاحيتها في $days_until_expiry أيام" | mail -s " تنبيه شهادة
SSL" admin@example.com
fi
```

أوامر التحقق بعد الإصلاحات

فحص الأمان الكامل

```
#!/bin/bash
# security_check.sh

echo "=== "=== مسح المنافذ
nmap -sV -p 80,443,8080,8443 example.com

echo -e "\n=== فحص SSL/TLS ==="
sslscan example.com:443 | grep -E "(TLS|SSL|Cipher)"

echo -e "\n=== "=== رؤوس الأمان
curl -s -I https://example.com/ | grep -i -E "(strict-transport|x-frame|x-content|content-security)"

echo -e "\n=== فحص WORDPRESS ==="
curl -s https://example.com/ | grep -i "wordpress\|wp-content" || echo "إصدار WordPress مخفي"

echo -e "\n=== "=== فحص المنفذ 8080
timeout 5 curl -s -I https://example.com:8080/ || echo ""المنفذ 8080 محجوب/معاد توجيهه"

echo -e "\n=== "=== فحص المنفذ 8443
timeout 5 curl -s -I https://example.com:8443/ || echo ""المنفذ 8443 مُصلح
```

فحص الامتثال التلقائي

```
#!/bin/bash
# compliance_check.sh

SCORE=0
MAX_SCORE=10

# الفحص 1: المنافذ 8080/8443 مغلقة أو مقيدة
if ! curl -s --max-time 5 https://example.com:8080/ >/dev/null 2>&1; then
    echo "✓ "المنفذ 8080 مؤمن
    ((SCORE++))
else
    echo "✗ "المنفذ 8080 ما زال قابلاً للوصول
fi

if ! curl -s --max-time 5 https://example.com:8443/ >/dev/null 2>&1; then
    echo "✓ "المنفذ 8443 مؤمن
    ((SCORE++))
else
    echo "✗ "المنفذ 8443 ما زال قابلاً للوصول
fi

# الفحص 2: رؤوس الأمان
```

```
HEADERS=("strict-transport-security" "x-frame-options" "x-content-type-options" "content-security-policy")
for header in "${HEADERS[@]}"; do
  if curl -s -I https://example.com/ | grep -qi "$header"; then
    echo "✓ $header "موجود
    ((SCORE++))
  else
    echo "✗ $header "مفقود
  fi
done

# إصدار 3: الفحص WordPress مخفي
if ! curl -s https://example.com/ | grep -qi "wordpress.*[0-9]"; then
  echo "✓ إصدار WordPress "مخفي
  ((SCORE++))
else
  echo "✗ إصدار WordPress "ما زال مرئياً
fi

# تكوين 4: الفحص SSL
if sslscan example.com:443 | grep -q "TLSv1.3.*enabled"; then
  echo "✓ TLS 1.3 "مفعل
  ((SCORE++))
else
  echo "✗ TLS 1.3 "غير مفعل
fi

echo -e "\n=== نقاط الامتثال: $SCORE/$MAX_SCORE ==="
if [ $SCORE -ge 8 ]; then
  echo "✓ "نجح - وضعية أمان جيدة
  exit 0
else
  echo "✗ "فشل - تحسينات أمنية مطلوبة
  exit 1
fi
```

الامتثال لـ Microsoft SDL

مراحل SDL والامتثال

مرحلة SDL	المتطلب	الحالة	الإجراء
المتطلبات	متطلبات الأمان محددة	×	تحديد متطلبات الأمان
التصميم	نمذجة التهديدات مكتملة	×	إجراء نمذجة التهديدات
التنفيذ	ممارسات الترميز الآمن	⚠	تحسين أمان WordPress
التحقق	اختبار الأمان		مكتمل (هذا التدقيق)
الإصدار	مراجعة الأمان	⚠	مطلوب بعد الإصلاحات
الاستجابة	خطة الاستجابة للحوادث	×	إنشاء خطة IR

معايير Microsoft الأمنية

```
# فحص الامتثال لخط الأساس الأمني من Microsoft
# https://docs.microsoft.com/en-us/security/benchmark/

# 1. أمان الشبكة
echo "NS-1: إنشاء حدود تقسيم الشبكة"
# الإجراء: تقييد الوصول للمنافذ 8080/8443

# 2.  إدارة الهوية
echo "IM-1: توحيد أنظمة المصادقة"
# الإجراء: تنفيذ MFA لإدارة WordPress

# 3. الوصول المميز
echo "PA-1: حماية ومراقبة الوصول المميز"
# الإجراء: تقييد الوصول لـ /wp-admin/

# 4. حماية البيانات
echo "DP-1: اكتشاف وتصنيف ووسم البيانات الحساسة"
# الإجراء: تصنيف بيانات WordPress

# 5. إدارة الأصول
echo "AM-1: ضمان رؤية فريق الأمان للمخاطر"
# الإجراء: تنفيذ مراقبة الأمان
```

الخلاصة

كشف التدقيق التقني عن مشاكل تكوين خطيرة تتطلب تدخلاً فورياً. المخاطر الأساسية مرتبطة بالتكوين غير الصحيح للمنافذ وحماية WordPress غير الكافية.

أولويات المعالجة: 1. **P0 (0-24 ساعة)**: إغلاق المنفذ 8443، تقييد 8080، حماية wp-admin 2. **P1 (7-30 أيام)**: تنفيذ المراقبة، الأتمتة، الامتثال SSL/TLS 3. **P2 (7-30 أيام)**: إضافة رؤوس الأمان، تحسين

النتيجة المتوقعة: مع التنفيذ الصحيح للتوصيات، يمكن تحسين وضعية الأمان من 6/10 إلى 9/10 خلال 30 يوماً.

.

التقرير معد وفقاً لدورة تطوير الأمان من Microsoft (SDL) وإطار عمل الأمن السيبراني NIST.