Security Assessment Summary for isagenixhealth.net

Security Assessment Summary for isagenixhealth.net

Overview

This report provides an external assessment of the isagenixhealth.net domain to identify exposed services, outdated components, and potential opportunities for security hardening.
All findings were derived from publicly available information and non-intrusive reconnaissance techniques.

Domain & Registration

Positive Observations:
✅ Private registration in place
✅ Utilizes Cloudflare for DNS

WHOIS Excerpt:

```
Registrar: GoDaddy.com, LLC
Registry Domain ID: 1562731142_DOMAIN_NET-VRSN
Created: 2009-07-17 | Updated: 2023-07-26 | Expires: 2026-07-17
Name Servers: CONRAD.NS.CLOUDFLARE.COM, VAL.NS.CLOUDFLARE.COM
DNSSEC: Unsigned
```

Recommendation:

- Consider enabling DNSSEC for cryptographic validation of DNS responses.

Network Scan Summary

Host Identified: 44.240.61.135 (AWS, us-west-2 region)
Detected Open Ports:

Port	Protocol	Service	Notes
22/tcp	SSH	OpenSSH 8.2p1 (Ubuntu)	Externally exposed
80/tcp	HTTP	Tinyproxy 1.8.3	Proxy service publicly accessible
443/tcp	HTTPS	Nginx	Let's Encrypt TLS certificate (valid until Jan 2026)

Findings by Service

SSH (OpenSSH 8.2p1 – 6 Known CVEs)

Risk: Medium–High

Recommendations:

- Restrict SSH to internal network interfaces only.
- Disable password authentication and enforce key-based access.
- Upgrade to OpenSSH ≥ 9.9p2 to address multiple vulnerabilities.

Notable CVEs:

- CVE-2020-14145: Info leak via algorithm negotiation
- CVE-2020-12062: SCP client overwrite
- CVE-2024-6387 (“RegreSSHion”): Critical remote code execution
- CVE-2025-26466: DoS via resource exhaustion

HTTP / Tinyproxy 1.8.3

Risk: High

Finding: Publicly exposed proxy service may allow misuse or DoS (CVE-2012-3505).

Recommendations:

- Remove Tinyproxy
- Replace using tunnel connector published application route, redirecting HTTP to HTTPS

HTTPS / Nginx

Risk: Low

- Finding: TLS configuration current and valid.

Recommendation:

- Ensure nginx listens on 127.0.0.1 only, proxying all public traffic via Cloudflare eliminating public-facing port-exposure.
- Leverage Cloudflare SSL encryption for published application routes through the tunnel connector.

Future Considerations:

- Eliminate nginx. ***Refer to WordPress Future Considerations for additional details.***

Cloudflare Benefits:

- Full SSL/TLS encryption (Zero Trust)
- DDoS mitigation and WAF protection
- Obfuscates origin server IP

WordPress Instance

Risk: Moderate

Recommendations:

- Harden admin routes: Rename /wp-admin and /wp-login.php using WPS Hide Login or WP Hide & Security Enhancer.
- Add layered security: Implement Wordfence or All-In-One Security (AIOS) for MFA, brute-force prevention, and malware scanning.
- Relocate sensitive configs: Move wp-config.php outside the web root.

Future Considerations:

- Containerize (Docker/Kubernetes) for scalability and isolation.
    - WordPress Docker images contain integrated httpd, eliminating need for nginx/Tinyproxy. ***Reference to Nginx Future Considerations.***
- Integrate into CI/CD pipeline for version control and automated deployments.

Summary of Key Recommendations

Priority	Action	Expected Impact
🔴 **High**	Restrict SSH & remove Tinyproxy	Eliminates remote exploitation vector
🟠 **Medium**	Enable Cloudflare proxying	Masks server IP, adds WAF/DDoS protection
🟠 **Medium**	Local bound Nginx Listener exposed by Cloudflare	Eliminates public-facing port exposure.
🟢 **Low**	Harden WordPress & enable MFA	Reduces CMS exploitation risk
🟢 **Low**	Enable DNSSEC	Improves domain integrity

Conclusion

Site demonstrates good initial use of Cloudflare and standard hardening. However, exposure of administrative services (SSH) and open-ports on public interfaces represents unnecessary attack surface. Implementing the above remediations would immediately improve external security posture and align with best practices for AWS-hosted web infrastructure.