RansomLess
HERE WE ARE

CYBERSECURITY ASSESSMENT REPORT

github.com

REPORT ID	DATE
SA-20260524034107	2026-05-24

PREPARED FOR

GitHub

GENERATED BY MILI

Managed Infrastructure Live Inspection

CONFIDENTIAL

Security Audit Report

github.com | 2026-05-24 | SA-20260524034107

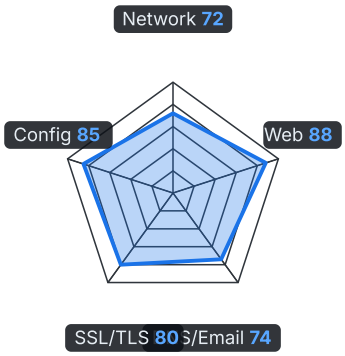
Generated by MILI · RansomLess

28

RISK SCORE

Medium Risk

github.com



CRITICAL	0
HIGH	0
MEDIUM	4
LOW	4

4 Medium | 4 Low | 4 Informational

EXECUTIVE SUMMARY

GitHub.com presents a generally strong security posture consistent with a mature, enterprise-grade web platform. The external scan conducted on May 24, 2026 found only two publicly accessible services — secure web (HTTPS on port 443) and SSH on port 22 — with the vast majority of ports either closed or filtered. Critical web security headers are properly configured, and the site enforces HTTPS with a valid TLS certificate. Email security controls including SPF, DMARC, and DKIM are all in place and well-configured.

The most notable concerns identified are moderate in nature. Port 22 (SSH) is publicly accessible from the internet, which, while operationally necessary for Git operations, represents an attack surface that warrants monitoring and access controls. The SSH service banner reveals a non-standard version string ('SSH-2.0-47e7538') that does not disclose a specific vendor version, which is actually a positive obfuscation practice, though it prevents precise vulnerability assessment.

From a DNS and domain management perspective, the domain is registered through MarkMonitor with strong anti-hijacking protections (clientDeleteProhibited, clientTransferProhibited, clientUpdateProhibited). However, DNSSEC is not enabled, meaning DNS responses are not cryptographically signed and could theoretically be spoofed in a DNS cache poisoning attack. The domain is also approaching its registration expiry in October 2026, which should be renewed promptly to avoid any lapse.

Overall, GitHub.com demonstrates security practices well above industry average. The findings in this report are largely low-to-medium risk observations and configuration refinements rather than critical vulnerabilities. No known CVEs were identified against detected software versions. The primary recommendations focus on

enabling DNSSEC, reviewing SSH exposure, renewing the domain registration, and minor TLS and SPF hardening.

NETWORK EXPOSURE

GitHub.com exposes only two public-facing TCP services: HTTPS (443) for web and API traffic, and SSH (22) for Git operations, with all other ports filtered. The attack surface is appropriately minimal for a platform of this scale, though SSH on port 22 remains a persistent target for brute-force and vulnerability exploitation attempts and should be protected with strong access controls.

SSL / TLS STATUS

GitHub.com presents a valid TLS certificate issued by Sectigo (DV E36) using ECDSA P-256, valid from May 5 to August 2, 2026 (~89 days), with HSTS preload enforced and HTTP-to-HTTPS redirection correctly implemented. The short certificate validity requires a robust automated renewal process to prevent service disruption.

DNS POSTURE

The domain uses a dual-provider DNS architecture (NS One + AWS Route 53) providing high availability, and is protected by MarkMonitor registrar locks against unauthorized transfer or deletion. However, DNSSEC is not enabled, leaving DNS responses unsigned and potentially vulnerable to cache poisoning attacks.

EMAIL SECURITY

Email security is well-configured with SPF covering all major sending services, DKIM keys published for Google and Mailchimp selectors, and DMARC enforced at pct=100 with quarantine policy and forensic reporting enabled. Two improvement opportunities exist: upgrading SPF from soft fail (~all) to hard fail (-all), and elevating the DMARC root domain policy from quarantine to reject to match the existing subdomain reject policy.

DETAILED FINDINGS

F-001 MEDIUM MEDIUM **DNSSEC Not Enabled**

DESCRIPTION

The domain github.com does not have DNSSEC (Domain Name System Security Extensions) enabled. Without DNSSEC, DNS responses are not cryptographically signed, leaving the domain vulnerable to DNS cache poisoning and man-in-the-middle attacks at the DNS layer. An attacker who successfully poisons a DNS resolver cache could redirect users to a malicious server impersonating GitHub, potentially harvesting credentials or distributing malware.

EVIDENCE

```
WHOIS record states: 'DNSSEC: unsigned'. No DS records were observed in the DNS zone data.
```

RECOMMENDATION

Enable DNSSEC on the github.com zone by generating a Zone Signing Key (ZSK) and Key Signing Key (KSK), publishing DS records at the registrar (MarkMonitor), and configuring automatic key rollover. Coordinate with NS One and AWS Route 53 nameserver providers to ensure full chain-of-trust validation. Test using tools such as dnsviz.net after deployment.

F-002 MEDIUM QUICK FIX **Domain Registration Expiry Approaching (October 2026)**

DESCRIPTION

The domain github.com is registered with an expiry date of October 9, 2026 — approximately 4.5 months from the scan date. If the domain is not renewed in time, it could lapse and become available for registration by a malicious third party, resulting in complete loss of the domain and potential for large-scale phishing, credential harvesting, or supply chain attacks affecting millions of developers worldwide.

EVIDENCE

```
WHOIS record: 'Registry Expiry Date: 2026-10-09T18:20:50Z'. Scan date: 2026-05-24.
```

RECOMMENDATION

Immediately verify that auto-renewal is enabled with MarkMonitor for this domain. Confirm that billing and contact information is current. Consider extending the registration period to the maximum available term (typically 10 years) to reduce renewal risk. Set calendar alerts at 90, 60, and 30 days before expiry as a secondary safeguard.

F-003 MEDIUM MEDIUM **SSH (Port 22) Publicly Accessible from the Internet**

DESCRIPTION

TCP port 22 running SSH protocol 2.0 is open and accessible from the public internet. While this is operationally required for Git-over-SSH functionality, it represents a persistent attack surface for brute-force credential attacks, exploitation of SSH implementation vulnerabilities, and reconnaissance. The SSH banner 'SSH-2.0-47e7538' is a custom/obfuscated version string, which is a positive practice, but the underlying implementation version cannot be confirmed for patch status.

EVIDENCE

```
Nmap result: '22/tcp open ssh (protocol 2.0)' with banner 'SSH-2.0-47e7538'. Host IP: 4.228.31.150.
```

RECOMMENDATION

Ensure SSH access is protected by rate limiting and automated blocking of repeated failed authentication attempts (e.g., fail2ban or equivalent). Confirm that password-based SSH authentication is disabled and only public-key authentication is permitted. Implement network-level geo-blocking or allowlisting where operationally feasible. Ensure the underlying SSH daemon is patched to the latest stable version. Consider publishing SSH host key fingerprints via DNS SSHFP records to allow clients to verify host identity.

F-004 MEDIUM QUICK FIX **SPF Record Uses Soft Fail (~all) Instead of Hard Fail (-all)**

DESCRIPTION

The SPF record for github.com ends with '~all' (soft fail) rather than '-all' (hard fail). A soft fail instructs receiving mail servers to accept but mark suspicious emails, whereas a hard fail instructs them to reject emails from unauthorized senders outright. This means that spoofed emails purporting to come from github.com may still be delivered to recipients, increasing phishing risk. Given that GitHub sends security-sensitive notifications (e.g., password resets, 2FA codes), this is a meaningful concern.

EVIDENCE

```
TXT record: 'v=spf1 ip4:192.30.252.0/22 include:spf.protection.outlook.com include:_netblocks.google.com ... ~all'
```

RECOMMENDATION

After thoroughly auditing all legitimate mail-sending sources and confirming they are included in the SPF record, change the SPF policy terminator from '~all' to '-all'. This should be done carefully to avoid blocking legitimate mail flows. Use SPF testing tools (e.g., MXToolbox) to validate before and after the change. The existing DMARC policy (p=quarantine) provides a compensating control, but upgrading SPF to hard fail adds defense-in-depth.

F-005 LOW QUICK FIX **Short-Duration TLS Certificate (90-Day Validity)**

DESCRIPTION

The TLS certificate for github.com has a validity period of approximately 89 days (May 5, 2026 to August 2, 2026). While short-lived certificates are increasingly considered best practice and align with industry trends toward 90-day certificates, they require robust automated renewal processes. A failure in the renewal pipeline could result in certificate expiry, causing browser warnings and service disruption for all GitHub users globally.

EVIDENCE

```
SSL Certificate: 'Not Before: May 5 00:00:00 2026 GMT', 'Not After: Aug 2 23:59:59 2026 GMT'. Issue
r: Sectigo Public Server Authentication CA DV E36. Serial: e7:ce:cc:3b:13:fb:3b:7b:8a:46:ea:8c:d0:a
e:b7:1c.
```

RECOMMENDATION

Verify that automated certificate renewal (e.g., via ACME protocol or Sectigo's automated issuance API) is fully operational and monitored. Implement alerting at 30, 14, and 7 days before certificate expiry. Consider whether an Organization Validated (OV) or Extended Validation (EV) certificate would provide additional trust signals appropriate for a platform of GitHub's scale. Ensure Certificate Transparency (CT) log monitoring is in place to detect unauthorized certificate issuance.

F-006 LOW COMPLEX **No IPv6 (AAAA) DNS Records Present**

DESCRIPTION

GitHub.com has no AAAA DNS records, meaning the service is not accessible over IPv6. While this does not represent a direct security vulnerability, the absence of IPv6 support means that IPv6-only networks cannot reach the service natively and may rely on transition mechanisms that can introduce security and reliability risks. As IPv6 adoption grows globally, this gap may also affect availability for a growing segment of users.

EVIDENCE

```
DNS scan result: '=== AAAA === (no records)'. Only an A record resolving to 4.228.31.150 was found.
```

RECOMMENDATION

Plan and implement IPv6 support for github.com by provisioning AAAA records pointing to IPv6-addressed infrastructure. Ensure that all security controls (WAF, DDoS protection, rate limiting) are equally applied to IPv6 traffic. This is a longer-term infrastructure initiative but should be included in the platform roadmap.

F-007 LOW MEDIUM

Large Number of Third-Party Domain Verification TXT Records

DESCRIPTION

The DNS zone contains 20 TXT records, many of which are third-party service verification tokens (Google, Adobe, Facebook, Atlassian, Shopify, Stripe, Miro, Loom, Calendly, Krisp, Jamf, DocuSign, Tailscale, etc.). While individually benign, this large number of third-party integrations increases the overall attack surface. Stale or orphaned verification records for services no longer in use could be exploited in subdomain takeover or service impersonation scenarios. Additionally, the breadth of integrated third-party services represents a supply chain risk.

EVIDENCE

```
DNS TXT records include verification tokens for: google-site-verification (x2), adobe-idp-site-veri
fication, facebook-domain-verification, atlassian-domain-verification, shopify-verification-code, s
tripe-verification, miro-verification, loom-site-verification, calendly-site-verification, krisp-do
main-verification, jamf-site-verification, docusign, TAILSCALE, and multiple MS= records.
```

RECOMMENDATION

Conduct a periodic audit (at least annually) of all DNS TXT verification records to identify and remove entries for services that are no longer actively used. Maintain an inventory of all third-party SaaS integrations and their associated DNS records. Implement a change management process requiring DNS record cleanup when a third-party service is decommissioned.

F-008 LOW QUICK FIX

DMARC Policy Set to Quarantine Rather Than Reject

DESCRIPTION

The DMARC policy for github.com is set to 'p=quarantine', which instructs receiving mail servers to place non-compliant emails in the spam/junk folder rather than outright rejecting them. While this is significantly better than 'p=none', upgrading to 'p=reject' would provide the strongest protection against email spoofing and phishing attacks impersonating GitHub. Notably, the subdomain policy (sp=reject) is already set to reject, which is inconsistent with the root domain policy.

EVIDENCE

```
DMARC record: 'v=DMARC1; p=quarantine; sp=reject; pct=100; rua=mailto:dmarc@github.com; ruf=mailto:
dmarc@github.com; fo=1'
```

RECOMMENDATION

Review DMARC aggregate reports (rua) to confirm that all legitimate mail flows are passing DMARC alignment checks. Once confident that no legitimate mail will be rejected, upgrade the root domain DMARC policy from 'p=quarantine' to 'p=reject' to align with the existing subdomain policy (sp=reject) and provide maximum email spoofing protection.

F-009 INFORMATIONAL QUICK FIX

SSH Banner Reveals Custom Version String

DESCRIPTION

The SSH service on port 22 returns a non-standard banner string 'SSH-2.0-47e7538' rather than a standard OpenSSH version string. This appears to be an intentional obfuscation of the underlying SSH implementation version. While this is a positive security practice that prevents trivial version-based vulnerability identification, it also means that automated patch verification tools cannot confirm whether the SSH daemon is running a patched version.

EVIDENCE

```
Nmap fingerprint: 'SSH-2.0-47e7538' on port 22/tcp.
```

RECOMMENDATION

No immediate action required. Continue the practice of suppressing detailed version information in SSH banners. Ensure that internal asset management and patch tracking systems maintain accurate records of the actual SSH implementation and version in use, and that patching is performed on a regular schedule independent of banner-based detection.

F-010 INFORMATIONAL QUICK FIX

HTTP to HTTPS Redirection Properly Configured

DESCRIPTION

The server correctly issues HTTP 301 Moved Permanently redirects to HTTPS for all HTTP requests, and the Strict-Transport-Security (HSTS) header is present with a max-age of 31,536,000 seconds (1 year), includeSubDomains, and preload directives. This ensures that browsers will enforce HTTPS connections and prevents SSL stripping attacks. This is a positive finding confirming correct transport security configuration.

EVIDENCE

```
Nmap HTTP response: 'HTTP/1.1 301 Moved Permanently', 'Location: https://github.com/', 'Strict-Tran
sport-Security: max-age=31536000; includeSubDomains; preload'. HTTP headers confirm: 'strict-transp
ort-security: max-age=31536000; includeSubdomains; preload'.
```

RECOMMENDATION

No action required. Continue to maintain HSTS preload list registration. Periodically verify that the domain remains on the HSTS preload list at hstspreload.org.

F-011 INFORMATIONAL QUICK FIX

Security Headers Comprehensively Implemented

DESCRIPTION

GitHub.com implements a comprehensive set of HTTP security response headers including Content-Security-Policy (CSP), X-Frame-Options (deny), X-Content-Type-Options (nosniff), Referrer-Policy, and X-XSS-Protection set to 0 (disabling the legacy XSS auditor in favor of CSP). The CSP policy is detailed and restrictive, explicitly whitelisting trusted sources. This represents a strong defense-in-depth posture against common web attacks including clickjacking, MIME sniffing, and cross-site scripting.

EVIDENCE

```
HTTP headers: 'x-frame-options: deny', 'x-content-type-options: nosniff', 'x-xss-protection: 0', 'r
eferrer-policy: origin-when-cross-origin, strict-origin-when-cross-origin', 'content-security-polic
y: default-src none; base-uri self; ...' (extensive whitelist policy).
```

RECOMMENDATION

No immediate action required. Periodically review the CSP policy to remove any stale or overly permissive source entries as the application evolves. Consider implementing a CSP reporting endpoint (report-uri or report-to directive) if not already in place to capture policy violations in production.

F-012 INFORMATIONAL QUICK FIX

Domain Protected by MarkMonitor with Registrar Lock

DESCRIPTION

The github.com domain is registered through MarkMonitor, a registrar specializing in brand protection for high-value domains. The domain has clientDeleteProhibited, clientTransferProhibited, and clientUpdateProhibited EPP status codes applied, which prevent unauthorized deletion, transfer, or modification of the domain registration. This is a strong domain security posture.

EVIDENCE

```
WHOIS: 'Registrar: MarkMonitor Inc.', 'Domain Status: clientDeleteProhibited', 'Domain Status: clie
ntTransferProhibited', 'Domain Status: clientUpdateProhibited'.
```

RECOMMENDATION

No action required. Continue to maintain registrar lock status. Ensure that MarkMonitor account access is protected with strong multi-factor authentication and that account recovery procedures are documented and tested.

PRIORITY RECOMMENDATIONS

1 **Renew domain registration for github.com before October 9, 2026 and confirm auto-renewal is active with MarkMonitor.**

Domain expiry is the highest-impact single point of failure; loss of the domain would be catastrophic for GitHub's global user base and could enable large-scale phishing and supply chain attacks.

2 **Enable DNSSEC on the github.com zone and publish DS records at the registrar.**

Without DNSSEC, DNS responses are unsigned and vulnerable to cache poisoning attacks that could redirect users to malicious infrastructure, undermining all other security controls.

3 **Upgrade DMARC policy from 'p=quarantine' to 'p=reject' after validating all legitimate mail flows pass DMARC alignment.**

Reject policy provides the strongest protection against email spoofing; the existing subdomain policy already uses reject, making the root domain quarantine policy inconsistent and weaker than necessary.

4 **Change SPF record terminator from '~all' (soft fail) to '-all' (hard fail) after auditing all authorized sending sources.**

Soft fail allows spoofed emails to be delivered; hard fail combined with the existing DMARC quarantine/reject policy provides stronger defense against phishing emails impersonating GitHub.

5 **Implement SSH access controls including rate limiting, public-key-only authentication enforcement, and SSHFP DNS records for host key verification.**

Port 22 is publicly exposed and represents a persistent brute-force and exploitation target; layered controls reduce the risk of unauthorized access to Git infrastructure.

6 **Audit and remove stale third-party DNS TXT verification records for services no longer in active use.**

Orphaned verification records increase the attack surface and may enable service impersonation; a clean DNS zone reduces risk and improves operational clarity.

7 **Verify automated TLS certificate renewal pipeline is operational and implement expiry alerting at 30, 14, and 7 days before the August 2, 2026 expiry.**

The current certificate expires in approximately 70 days; a renewal failure would cause browser security warnings and service disruption for millions of users globally.

8 **Plan and implement IPv6 (AAAA record) support for github.com.**

Absence of IPv6 support limits accessibility for IPv6-only networks and may introduce reliability risks through transition mechanisms; IPv6 adoption continues to grow globally.

LEGAL DISCLAIMER

This security assessment was conducted with explicit authorization from the domain owner. All findings are based on publicly accessible information only. No systems were accessed without permission. This report is confidential and intended solely for the authorized recipient. RansomLess / MILI accepts no liability for actions taken based on this report without proper technical review.