Security Assessment Report

Target: www.itsecgames.com
Date: 17-Sep-2025

Index

Objective

This assessment encompasses both passive and active reconnaissance of the target site, aiming to identify potential vulnerabilities, misconfigurations, and sensitive information exposures that an attacker could exploit. No critical vulnerabilities were found, but some improvements are recommended to harden security.

Scope

What was tested:

- Passive Information Gathering
- Port Scanning
- Directory Enumeration
- Subdomain Enumeration
- SSL/TLS certification assessment
- Manual Info Leakage Analysis
- JavaScript Inspection

Methodology

List tools & techniques used:

- WhatWeb
- WafW00f
- Subfinder / Amass
- Dirb
- Nmap
- SSL Labs
- curl
- Manual Browsing

Finding

1. **Check for robots.txt file**

   Tool Used: Manual
   During the analysis, no robots.txt or sitemap file was found.

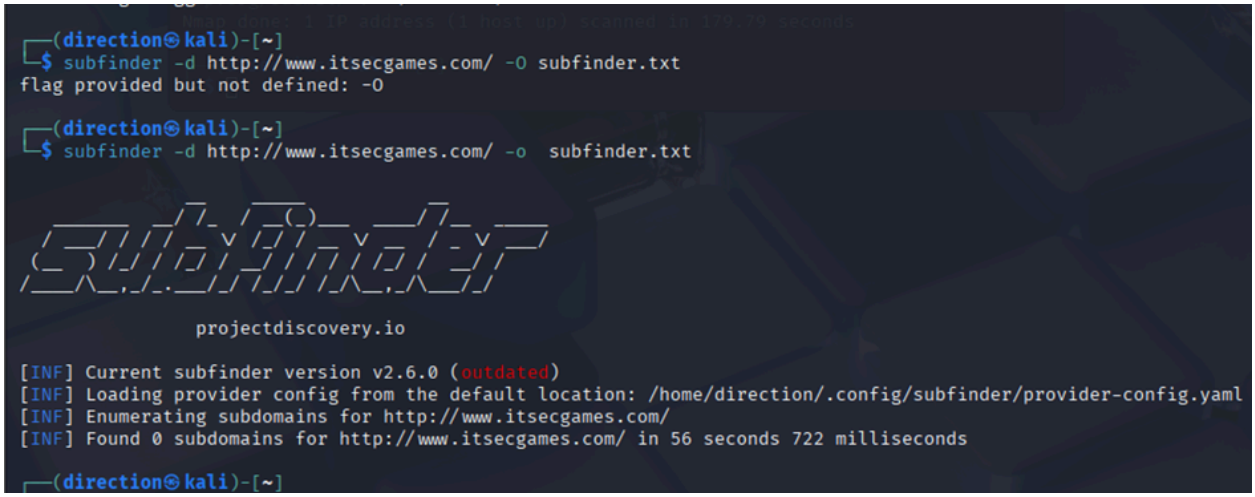
   Mitigation Recommendations

   It is recommended to create a robots.txt file to restrict crawlers from accessing sensitive pages. Note that this is primarily for search engine guidance and not a security control. Example directives include restricting access to administrative or private sections of the website.

2. Subdomain Discovery

Tool Used:subfinder

During testing, no subdomains related to the target website were identified.



Mitigation Recommendations

Regularly monitor DNS records to detect any unauthorized or new subdomains.
Remove or disable unused subdomains to prevent potential takeover risks.
Consider implementing security measures such as wildcard prevention to avoid subdomain exploitation.

3. Firewall Discovery

Tool Used: wafw00f

During testing, no firewall or Web Application Firewall (WAF) was detected protecting the website.

Mitigation Recommendations

Deploy a WAF to filter and block malicious traffic.

Implement rate limiting, bot protection, and IP blacklisting to prevent automated attacks.

Regularly review firewall logs to identify suspicious activity and potential threats.

4. Directories Present

Tool Used: Dirb

```
┌──(direction㉿kali)-[~]
└─$ dirb  http://www.itsecgames.com/

_____________________
DIRB v2.22
By The Dark Raver
_____________________

START_TIME: Tue Sep 16 20:46:17 2025
URL_BASE: http://www.itsecgames.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____________________

GENERATED WORDS: 4612

---- Scanning URL: http://www.itsecgames.com/ ----
==> DIRECTORY: http://www.itsecgames.com/downloads/
==> DIRECTORY: http://www.itsecgames.com/images/
+ http://www.itsecgames.com/index.htm (CODE:200|SIZE:3651)
==> DIRECTORY: http://www.itsecgames.com/javascript/
==> DIRECTORY: http://www.itsecgames.com/js/

(!) FATAL: Too many errors connecting to host
    (Possible cause: COULDNT CONNECT)

_____________________

END_TIME: Tue Sep 16 20:56:43 2025
DOWNLOADED: 2881 - FOUND: 1

┌──(direction㉿kali)-[~]
└─$
```

During directory brute-forcing with **dirb**, the following interesting result was found:

- **URL:** http://www.itsecgames.com/index.htm
- **HTTP Response Code:** 200 (OK)
- **Content Size:** [Include size 3651]

An index.htm file returning 200 means the file is publicly accessible and serves content to visitors.
Depending on its content, this could expose:

- Old or forgotten website versions
- Debug information
- Sensitive comments in source code
- Hardcoded credentials or API endpoints
- Unused functionality or vulnerable scripts

Even if it looks like a normal page, attackers often check these files carefully for hidden info or old scripts that can be exploited.

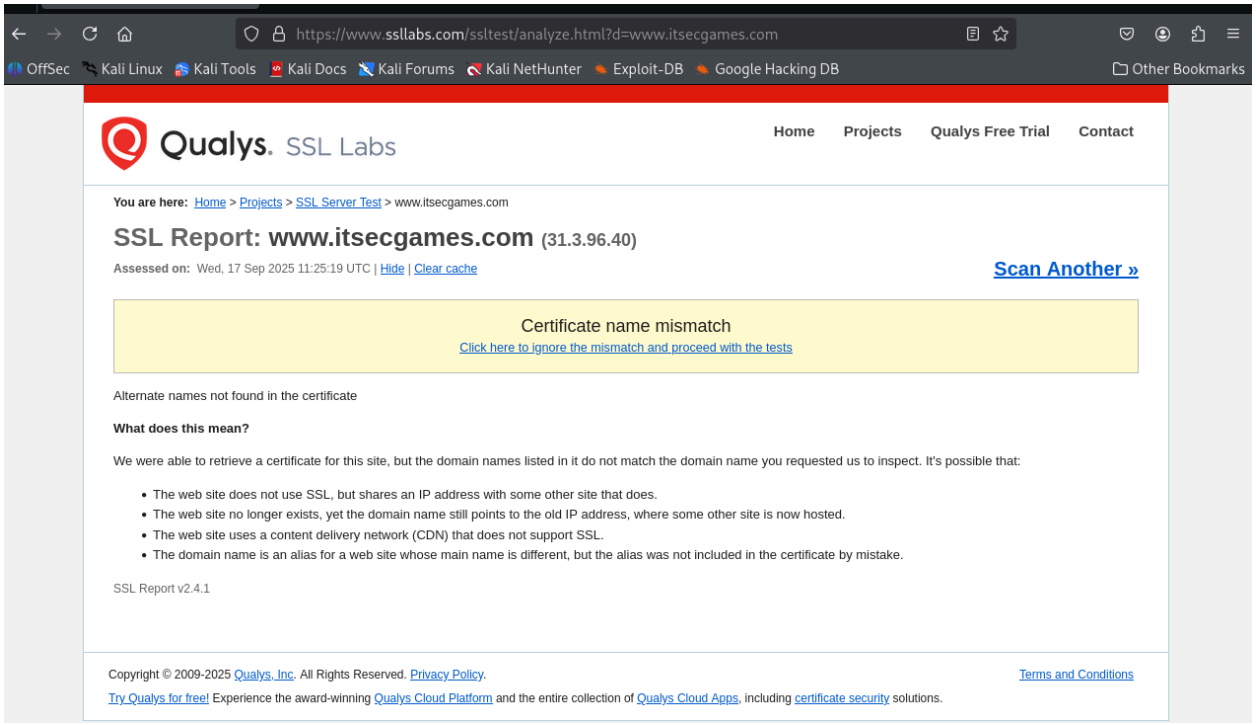
Mitigation Recommendations

1. Manually review the contents of index.htm to ensure there is no sensitive data, comments, hardcoded credentials, or debug information left in the code.
2. If the file is outdated, unused, or serves no functional purpose, it should be removed from the public web directory immediately to reduce the attack surface.
3. If the file is needed for specific purposes (e.g., internal documentation), restrict access using proper web server rules (.htaccess for Apache):

5. SSL/TLS Certification assesment

During the assessment, the website was found accessible over plain HTTP without HTTPS enforcement. This means no SSL/TLS encryption is present, which exposes data in transit to interception or tampering by attackers.

Additionally, server headers such as Server: Apache and old Last-Modified timestamps were visible(in homepage check), indicating potential information leakage about the web server version and configuration.



https://www.ssllabs.com/ssltest

OffSec | Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Other Bookmarks

Qualys. SSL Labs

Home | Projects | Qualys Free T

You are here: Home > Projects > SSL Server Test

SSL Server Test

This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. **Please note th information you submit here is used only to provide you the service. We don't use the domain names or the test results, a will.**

Hostname: www.itsecgames.com Submit

☐ Do not show the results on the boards

Recently Seen	Recent Best		Recent Worst
retailerapp.syngentadigitala ...	pxwebsitelatamprod-frayczghf ...	A+	energietec.eu
www.arthausaffiliates.com	preproduccion.everilion.com	A+	nef.com.tr
bienetresante.shop	aie-ict-enabler-uat.ey.com	A+	api.dpissociety.com
illuminetsolutions.com	epargnants.interepargne.nati ...	A+	controltower.preprod.mypepsi ...
inkafarma.com.pe	acessibilidade.gov.pt	A+	ejit.jyothyit.ac.in
www.berlin-buehnen.de	google-ohttp-relay-safebrows ...	A	reader.i-telligent.com
readylift.com	escalagcmhmg.saobernardo.sp. ...	A	www.nationalpublicseating.co ...
woundpiano28.bravejournal.ne ...	www.docmagic.com	A-	sws.sb.com.ua
olympus-prod-storages3bucket	connect.tagglas.com	A-	telisservices.site

Server Key and Certificate #1

Subject	web.mmebvba.com Fingerprint SHA256: 9e7276cb84903692044a0e1f9b64d1426869813b55b28167913b7e49e778f87e Pin SHA256: moilG7Pck7rm7Q7pJpb+auqA9cuCc0eOAxVrTFBhY0M=
Common names	web.mmebvba.com
Alternative names	- INVALID
Serial Number	00ba5e79e0c2f743cb
Valid from	Mon, 25 May 2015 09:07:54 UTC
Valid until	Thu, 22 May 2025 09:07:54 UTC (expired 3 months and 26 days ago) EXPIRED
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	web.mmebvba.com Self-signed
Signature algorithm	SHA256withRSA
Extended Validation	No
Revocation information	None
DNS CAA	No (more info)
Trusted	No NOT TRUSTED (Why?) Mozilla Apple Android Java Windows

---

Mitigation Recommendations

- Enforce HTTPS across the entire site by configuring HTTP → HTTPS redirects.
- Install a valid SSL/TLS certificate from a trusted certificate authority
- Disable weak or outdated TLS versions
- Implement strong cipher suites.
- Enable HTTP Strict Transport Security (HSTS) to force browsers to use HTTPS.
- Hide unnecessary server headers to prevent information leakage.
- Regularly test the SSL configuration using tools like SSL Labs to ensure no vulnerabilities are present.

6. Code Review

Command →curl -IL http://www.itsecgames.com/ >homepage.txt

--Server responds with HTTP 200 OK, serving the homepage. However, no security-related HTTP headers were found in the response.

ffuf.json | homepage.txt

```
HTTP/1.1 200 OK
Date: Tue, 16 Sep 2025 11:55:30 GMT
Server: Apache
Last-Modified: Wed, 09 Feb 2022 13:14:08 GMT
ETag: "e43-5d7959bd3c800"
Accept-Ranges: bytes
Content-Length: 3651
Vary: Accept-Encoding
Content-Type: text/html
```

Here's the analysis:
Status 200 → Page is up & serving fine.
Server Header Exposed:

Server: Apache → No version info (good, but ideally should hide "Apache" completely for better security).

Missing Security Headers:

- X-Frame-Options → Prevent clickjacking
- X-XSS-Protection → Helps stop some basic XSS attacks
- Content-Security-Policy (CSP) → Limits what content can be loaded/executed

---

Mitigation Recommendations

- Add security headers in Apache config like:

  Header always set X-Frame-Options "DENY"

  Header always set X-XSS-Protection "1; mode=block"

  Header always set Content-Security-Policy "default-src 'self';"

- Hide Server header completely to avoid giving attackers info.

7. Nmap Findings

Open Ports:

- 22/tcp → tcpwrapped (no service version info)
- 80/tcp → tcpwrapped (likely Apache HTTP)
- 443/tcp → tcpwrapped (possibly HTTPS fallback)

Closed Ports: 1117, 1862, etc

```
┌──(direction㉿kali)-[~]
└─$ nmap -sS -sV -Pn www.itsecgames.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-16 20:54 IST
Nmap scan report for www.itsecgames.com (31.3.96.40)
Host is up (0.29s latency).
rDNS record for 31.3.96.40: web.mmebvba.com
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE  SERVICE        VERSION
22/tcp    open   tcpwrapped
80/tcp    open   tcpwrapped
443/tcp   open   tcpwrapped
1117/tcp  closed ardus-mtrns
1862/tcp  closed mysql-cm-agent
2869/tcp  closed icslap
6510/tcp  closed mcer-port
7019/tcp  closed doceri-ctl
49161/tcp closed unknown

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 179.79 seconds
```

Mitigation Recommendations

Only expose necessary ports to the public (block SSH on 22 if not needed externally, use VPN or jump server).

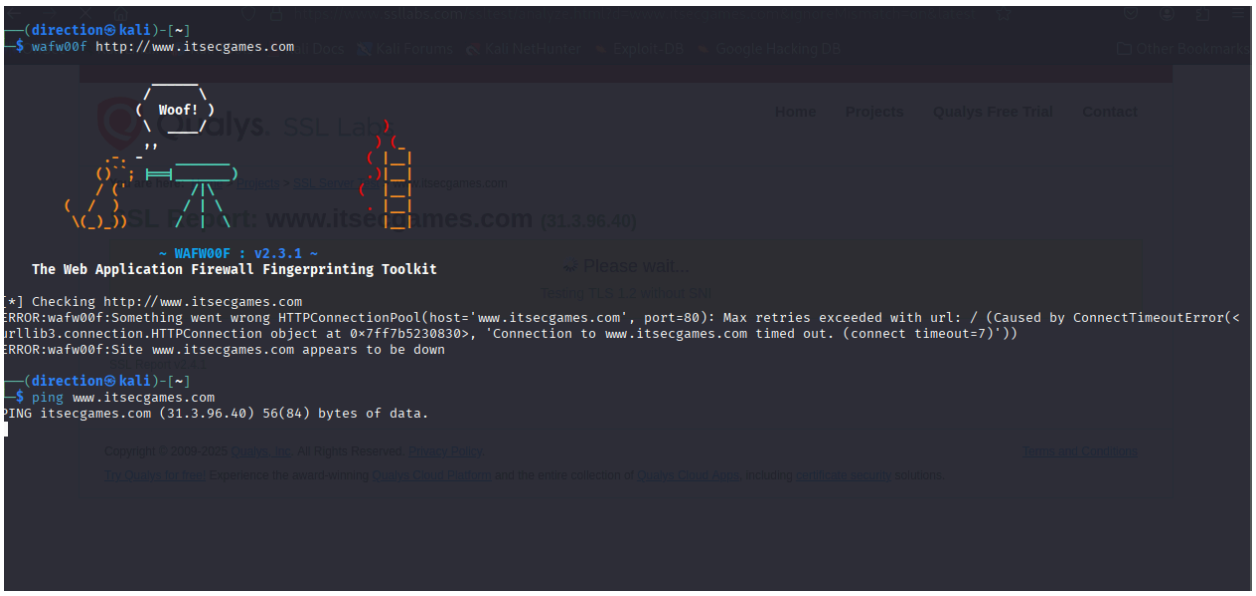
8. Denial of Service

Observation:

During testing, the website was not consistently accessible. It seems multiple requests or a high traffic volume caused temporary unavailability. This prevented further investigation and indicates that the server may not handle concurrent or high-volume requests efficiently.

Risk:

- The website is vulnerable to Denial-of-Service attacks, which could make it unavailable to legitimate users.
- High request rates could overload web and application servers, leading to downtime.



Mitigation Recommendations

1. Limit the number of requests per IP per time interval.
2. Deploy a WAF to block suspicious or automated request patterns.
3. Distribute traffic across multiple servers to handle spikes.
4. Use caching layers (e.g., reverse proxy like Nginx, Cloudflare cache) to reduce load on origin servers.

5. Implement traffic monitoring to detect unusual spikes early.

6. Consider services like Cloudflare, AWS Shield, or Akamai to mitigate large-scale DoS attacks.

9. JavaScript and API findings

During the assessment, references to JavaScript files and a js/ directory were discovered on the target website. These files could expose API endpoints, sensitive information, or implementation details if not properly secured.

Due to the website being temporarily unavailable (likely due to high request volume or a DoS-like condition), further inspection of the JavaScript files and discovery of API endpoints was not possible at the time of testing.

Mitigation Recommendations

- Once the site is accessible, inspect JavaScript files and the js/ directory for hardcoded URLs, credentials, or API endpoints.
- Use browser Developer Tools and proxy tools like Burp Suite or OWASP ZAP to capture and analyze API calls during site interaction.
- Ensure all API endpoints are properly authenticated and validated, and avoid exposing sensitive logic in frontend JS files.

Summary Table

	Area / Test	Observation	Risk Level
1	Passive Info Gathering	Website reveals server info, no subdomains found	**Low**
2	Firewall / WAF	No WAF detected	**Medium**
3	Directory Bruteforce	6 directories found	**Medium**
4	Port Scan / Services	Open ports: 22, 80, 443 (tcpwrapped); closed ports: 1117,1862,2869,6510,7019,49161	**Medium**
5	Subdomain Enumeration	No subdomains found	**Low**
6	SSL/TLS	HTTP accessible, Apache headers exposed, old Last-Modified date	**High**
7	Robots.txt	Not found	**Low**
8	Denial-of-Service	Website temporarily unavailable due to multiple requests	**High**

9	API/	These could potentially contain hardcoded API endpoints or sensitive information	**Medium**

Conclusion

The assessment focused on passive and active reconnaissance of the target website, aiming to identify potential vulnerabilities, misconfigurations, and information exposures. No critical vulnerabilities were found during the testing. However, several areas of improvement were identified, including a lack of SSL/TLS encryption, a missing WAF, exposed JavaScript files, and the absence of robots.txt.

Additionally, availability issues (likely due to high request volume) prevented further in-depth testing of API endpoints and JavaScript files.

Implementing the recommended mitigations—such as enforcing HTTPS, deploying a Web Application Firewall, restricting sensitive directories, and securing API endpoints—will significantly improve the security posture of the website and reduce the risk of potential attacks.