

## Team 7.3

### Report on ten vulnerabilities

Practice Website: <http://www.itsecgames.com/>

Vulnerability Name: **Improper Sanitization of File Paths**

CWE- 284: Improper Access Control

OWASP Category: A01:2021-Broken Access Control

**Description:** The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

**Business Impact:** This vulnerability is significant, as it can lead to unauthorized access to sensitive data or files, potentially causing data breaches, privacy violations, and loss of user trust. Affected sites may also require configuration changes after applying a security release.

**Steps to Reproduce:** <https://www.shodan.io/host/31.3.96.40>

**CVE-2023-  
31250**

The file download facility doesn't sufficiently sanitize file paths in certain situations. This may result in users gaining access to private files that they should not have access to. Some sites may require configuration changes following this security release. Review the release notes for your Drupal version if you have issues accessing private files after updating.

#### Recommendation:

1. **Apply Security Updates:** The primary recommendation is to apply the security update provided by Drupal as soon as it becomes available. Security updates typically include fixes for known vulnerabilities like this one.
2. **Review Configuration:** After applying the update, review your Drupal site's configuration to ensure that private files remain protected as intended. Make any necessary adjustments to your site's file permissions or access controls.
3. **Regular Audits:** Conduct regular security audits and vulnerability assessments to identify and address potential security issues in your Drupal site.

4. **Access Control:** Implement strong access controls to ensure that only authorized users can access private files. Review and restrict file access permissions as needed.
5. **Data Minimization:** Minimize the amount of sensitive data stored on your Drupal site. Only store data that is necessary for your site's functionality.

**Vulnerability Name:** **Invalid Certificate**

**CWE-395:** Improper Certificate Validation

**OWASP Category:** A07:2021 – Identification and Authentication Failures

**Description:** When a certificate is invalid or malicious, it might allow an attacker to spoof a trusted entity by interfering in the communication path between the host and client. The product might connect to a malicious host while believing it is a trusted host, or the product might be deceived into accepting spoofed data that appears to originate from a trusted host.

**Business Impact:** Improper certificate validation can lead to a range of security threats which includes Man-in-the-middle attacks(Attackers can intercept communication between two parties and read or modify the data exchanged between them),Data breaches:(Attackers can gain unauthorized access to sensitive information or sensitive systems, leading to data breaches),Malware distribution(Attackers can use fake digital certificates to distribute malicious software or infect systems with malware)

**Vulnerability Path:**

**Vulnerability Parameter:**

**Steps to Reproduce:**

<https://www.ssllabs.com/ssltest/analyze?d=www.itsecgames.com>

## Certificate #1: RSA 2048 bits (SHA256withRSA)



### Server Key and Certificate #1

Subject	web.mmebvba.com Fingerprint SHA256: 9e7278cb84903692044a0e1f9b64d1426889813b55b28167913b7e49e778f87e Pin SHA256: mollG7Pck7rm7Q7pJpb+auqA8cuCc0eOAxVrTFBhY0M=
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 (expires in 1 year and 7 months)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	web.mmebvba.com Self-signed
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	No
OCSP Must Staple	No
Revocation information	None
DNS CAA	No ( <a href="#">more info</a> )
Trusted	No NOT TRUSTED ( <a href="#">Why?</a> ) Mozilla Apple Android Java Windows



### Additional Certificates (if supplied)

Certificates provided	1 (712 bytes)
Chain issues	None



### Certification Paths

Mozilla Apple Android Java Windows

#### Path #1: Not trusted (path does not chain to a trusted anchor)

1	Sent by server Not in trust store	web.mmebvba.com Self-signed Fingerprint SHA256: 9e7278cb84903692044a0e1f9b64d1426889813b55b28167913b7e49e778f87e Pin SHA256: mollG7Pck7rm7Q7pJpb+auqA8cuCc0eOAxVrTFBhY0M= RSA 2048 bits (e 65537) / SHA256withRSA
---	--------------------------------------	--

## Recommendation:

1. **Use trusted certificate authorities:** Only trust digital certificates issued by well-known and trusted certificate authorities.
2. **Verify certificate chains:** Verify that the certificate presented by the remote party is valid and issued by a trusted certificate authority. Verify the entire certificate chain, including intermediate certificates.
3. **Check certificate revocation status:** Check the revocation status of the certificate presented by the remote party to ensure that it has not been revoked.
4. **Use certificate pinning:** Implement certificate pinning to ensure that the communication only occurs with the exact certificate or certificate authority specified.
5. **Keep software up to date:** Keep software and security protocols up to date, as new vulnerabilities and security patches are regularly released.

## Vulnerability Name: Missing X-Frame-Options Header

**CWE-1021:** Design Improper Restriction of Rendered UI Layers or Frames

**OWASP Category:** A04:2021-Insecure Design

**Description:** A web application is expected to place restrictions on whether it is allowed to be rendered within frames, iframes, objects, embed or applet elements. Without the restrictions, users can be tricked into interacting with the application when they were not intending to.

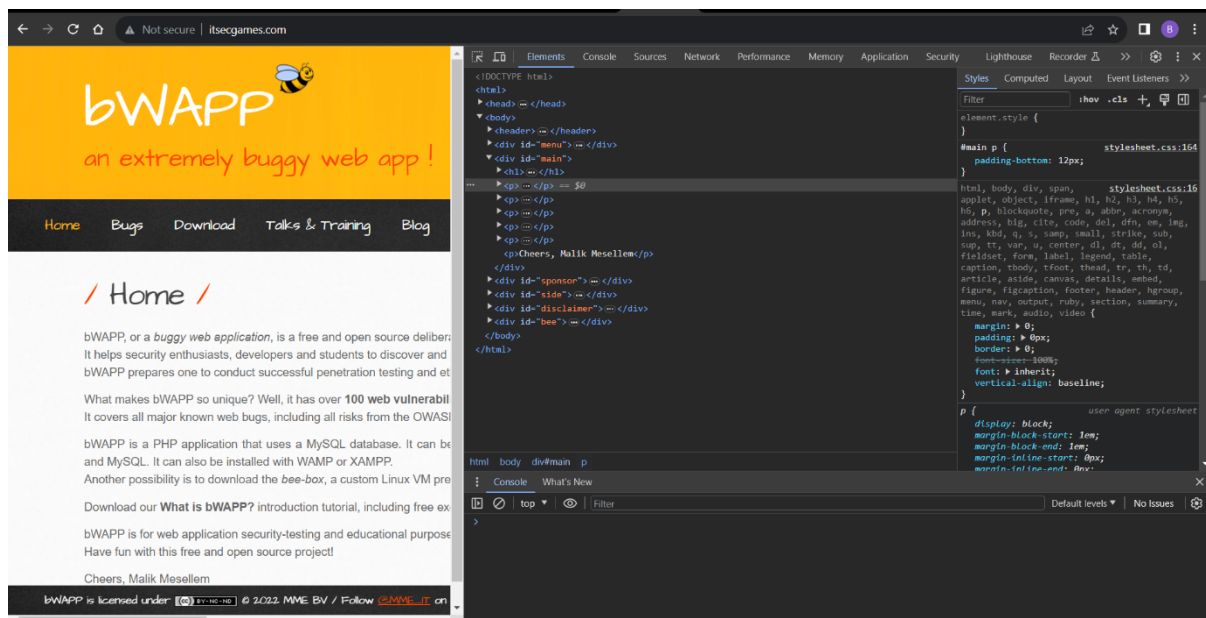
**Business Impact:** This enables path to the attackers to manipulate or gain unauthorized access to user interface elements, potentially leading to deceptive or malicious user interactions. This can result in data leakage, fraud, user distrust, and reputational damage, undermining the integrity and trustworthiness of the application and causing financial losses.

**Vulnerability Path:** <http://www.itsecgames.com/>

**Vulnerability Parameter:**

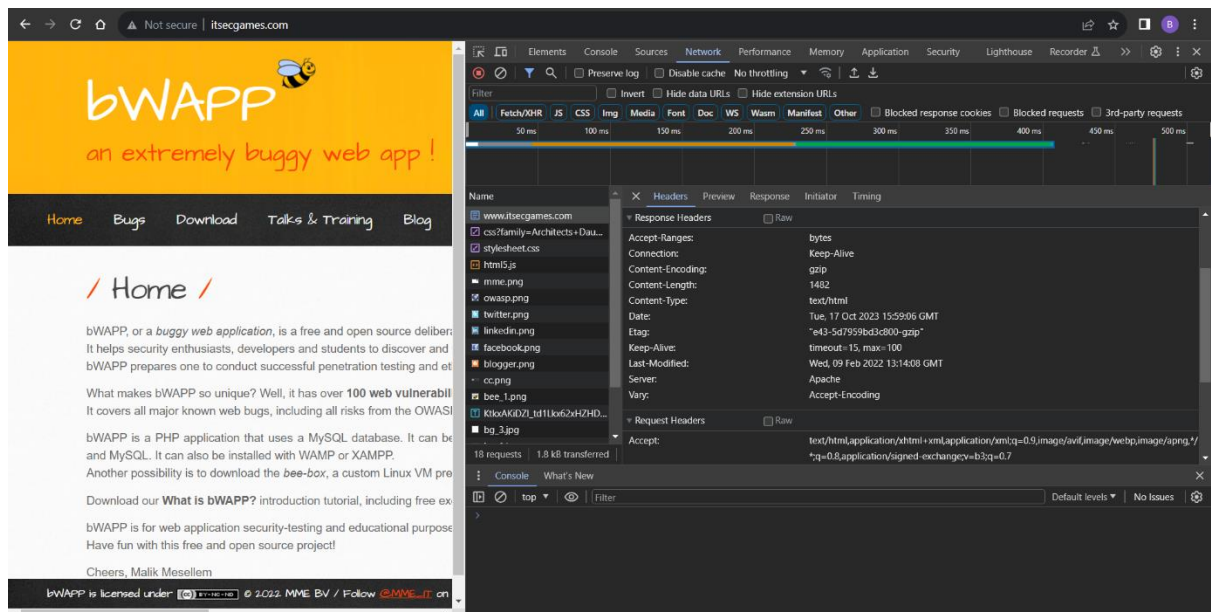
**Steps to Reproduce:**

1. Open the practice website to inspect



2. Go to network and start capturing using F5 or Ctrl+R
3. Go to html document and see if x-frame-options present in response headers or not

4. If not the absence of this header indicates that the website may not have protection against embedding its content within an iframe on other domains, potentially leaving it vulnerable to clickjacking attacks.



### Recommendation:

1. **Implement Proper UI Layer Access Controls:** Enforce access controls to restrict access to UI layers or frames based on user privileges.
2. **Use Session and Authentication Management:** Ensure robust session management and user authentication mechanisms to prevent unauthorized access.
3. **Implement Content Security Policies (CSP):** Use CSP headers to control where resources can be loaded from and mitigate against frame-based attacks.
4. **Keep Software and Libraries Updated:** Stay current with software updates and patches to address known security issues.

**Vulnerability Name:** X-Content-Type Options Header Missing

**CWE -918:** Server-Side Request Forgery (SSRF)

**OWASP Category:** A10:2021 – Server-Side Request Forgery (SSRF)

**Description:** The web server receives a URL or similar request from an upstream component and retrieves the contents of this URL, but it does not sufficiently ensure that the request is being sent to the expected destination.

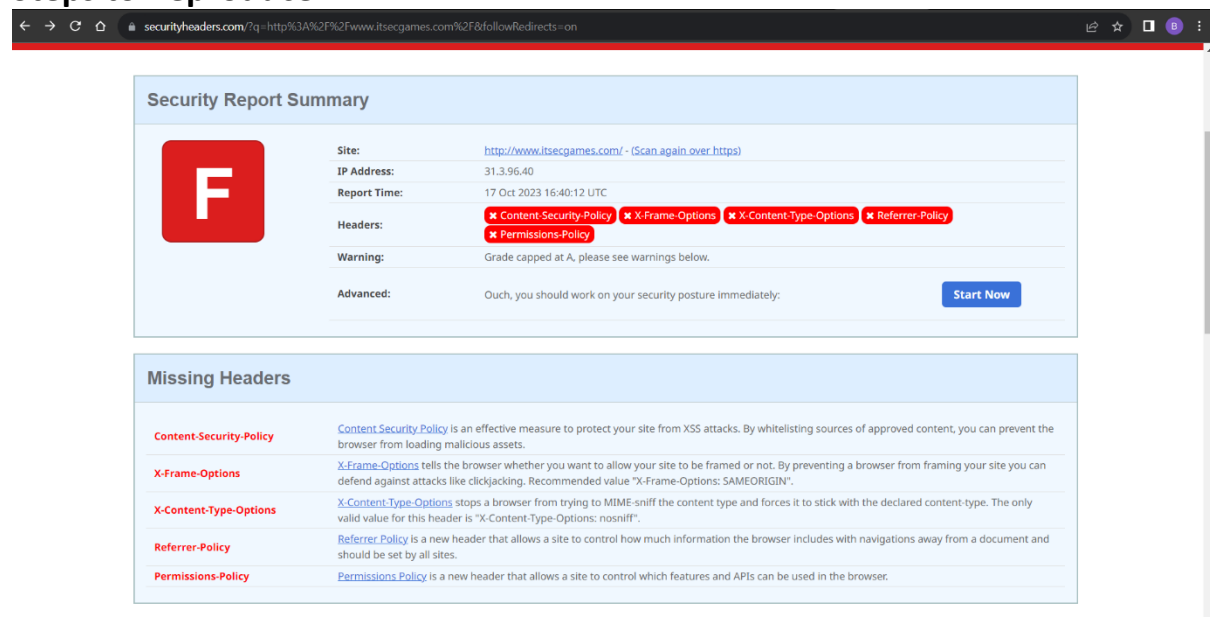
**Business Impact:** The header missing impact Data Exposure(Attackers might force the server to render sensitive data as HTML, leading to data exposure),Security Risks(Content type manipulation can increase security risks, potentially leading to breaches, data leaks, and unauthorized access), Brand and Reputation Damage(Successful attacks that exploit SSRF vulnerabilities can harm the organization's reputation and trustworthiness)

### Vulnerability Path :

<https://securityheaders.com/?q=http%3A%2F%2Fwww.itsecgames.com%2F&followRedirects=on>

### Vulnerability Parameter:

### Steps to Reproduce:



The screenshot shows a web browser window with the URL `securityheaders.com/?q=http%3A%2F%2Fwww.itsecgames.com%2F&followRedirects=on`. The page displays a "Security Report Summary" for the site `http://www.itsecgames.com/`. The report includes the following details:

- Site:** `http://www.itsecgames.com/` (Scan again over https)
- IP Address:** 31.3.96.40
- Report Time:** 17 Oct 2023 16:40:12 UTC
- Headers:** Content-Security-Policy, X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Permissions-Policy
- Warning:** Grade capped at A, please see warnings below.
- Advanced:** Ouch, you should work on your security posture immediately. [Start Now](#)

Below the summary, there is a section titled "Missing Headers" which lists the following headers and their descriptions:

Header	Description
Content-Security-Policy	Content-Security-Policy is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets.
X-Frame-Options	X-Frame-Options tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site you can defend against attacks like clickjacking. Recommended value "X-Frame-Options: SAMEORIGIN".
X-Content-Type-Options	X-Content-Type-Options stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type. The only valid value for this header is "X-Content-Type-Options: nosniff".
Referrer-Policy	Referrer-Policy is a new header that allows a site to control how much information the browser includes with navigations away from a document and should be set by all sites.
Permissions-Policy	Permissions-Policy is a new header that allows a site to control which features and APIs can be used in the browser.

```
(thanmayer08@THANMAYER)-[~]
$ nikto -h http://www.itsecgames.com/
- Nikto v2.5.0

+ Target IP: 31.3.96.40
+ Target Hostname: www.itsecgames.com
+ Target Port: 80
+ Start Time: 2023-10-17 09:03:40 (GMT5.5)

+ Server: Apache
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /com.tgz: Drupal 7 was identified via the x-generator header. See: https://www.drupal.org/project/remove_http_headers
+ /com.tgz: Drupal Link header found with value: <http://31.3.96.40/>; rel="canonical",<http://31.3.96.40/>; rel="shortlink". See: https://www.drupal.org/
+ /: Server may leak inodes via ETags, header found with file /, inode: e43, size: 5d7959bd3c800, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: OPTIONS, GET, HEAD, POST .
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ 8075 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time: 2023-10-17 09:29:14 (GMT5.5) (1534 seconds)

+ 1 host(s) tested
```

## Recommendation:

- Implement "X-Content-Type-Options" Header: Configure the "X-Content-Type-Options" header with the value "nosniff" in your web server or application to prevent content type sniffing.
- Secure SSRF: To directly mitigate SSRF vulnerabilities, ensure proper input validation, access controls, and allow-list-based URL validation to prevent attackers from making unauthorized requests to internal resources.
- Security Headers: Implement other security headers like "X-Frame-Options" and "Content-Security-Policy" to further enhance security and mitigate various types of web-based attacks.

While the "X-Content-Type-Options" header may not be a direct countermeasure to SSRF, it is a valuable security control that, when configured correctly, can help protect against content type-based attacks that may be exploited in SSRF scenarios.

**Vulnerability Name:** **Drupal Ajax API shows JSONP not disabled by default**



## **CWE-937,1035:** OWASP Top Ten 2013,2017 Category A9 - Using Components with Known Vulnerabilities

### **OWASP Category:** A06:2021-Vulnerable and Outdated Components

**Description:** Vulnerable and outdated components refer to using software libraries, modules, or dependencies within a web application that have known security issues or are not updated to the latest secure versions. This can include plugins, frameworks, or APIs that are no longer maintained or have publicly documented vulnerabilities.

**Business Impact:** The presence of vulnerable and outdated components in a web application poses significant security risks. Attackers can exploit these weaknesses to gain unauthorized access, inject malicious code, steal sensitive data, or disrupt the system's operation. The potential business impacts include compromised user data, damaged reputation, legal consequences, and financial losses due to breach recovery and remediation efforts.

**Vulnerability Path:** <https://www.shodan.io/host/31.3.96.40>

**Vulnerability Parameter:**

**Steps to Reproduce:**

**CVE-2020-  
13666**

**4.3** Cross-site scripting vulnerability in Drupal Core. Drupal AJAX API does not disable JSONP by default, allowing for an XSS attack. This issue affects: Drupal Core 7.x versions prior to 7.73; 8.8.x versions prior to 8.8.10; 8.9.x versions prior to 8.9.6; 9.0.x versions prior to 9.0.6.

### **Recommendation:**

1. **Regular Updates:** Keep all components, libraries, and dependencies up to date by applying security patches and upgrades.
2. **Vulnerability Scanning:** Conduct periodic vulnerability assessments to identify and address potential issues.
3. **Dependency Management:** Implement a strategy for managing third-party dependencies, including continuous monitoring of their security status.
4. **Security Testing:** Perform regular security testing, including penetration testing and code review, to identify and remediate vulnerabilities.
5. **Change Control:** Maintain a change management process to assess the impact of new components and ensure they don't introduce new vulnerabilities.



**Vulnerability Name:** [Drupal Api Vulnerability](#)

**CWE-20:** Improper Input Validation

**OWASP Category:** A03:2021 – Injection

**Description:** The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

**Business Impact:** The impact of a successful SQL injection attack can be devastating for businesses. It can result in unauthorized access to sensitive information, data theft, financial losses, damage to reputation, and legal consequences. Depending on the nature of the application and data involved, SQL injection can lead to compliance violations, customer trust erosion, and operational disruption.

**Vulnerability Path:** <https://www.shodan.io/host/31.3.96.40>

**Vulnerability Parameter:**

**Steps to Reproduce:**

**CVE-2022-  
25271**

**4.3** Drupal core's form API has a vulnerability where certain contributed or custom modules' forms may be vulnerable to improper input validation. This could allow an attacker to inject disallowed values or overwrite data. Affected forms are uncommon, but in certain cases an attacker could alter critical or sensitive data.

**Recommendation:**

1. **Input Validation:** Implement rigorous input validation on all user inputs and ensure that only safe, sanitized data is passed to the database.
2. **Parameterized Queries:** Use parameterized queries or prepared statements to separate SQL code from user input, preventing malicious injections.
3. **Escaping User Input:** If parameterized queries are not feasible, escape user input before incorporating it into SQL queries to neutralize special characters.
4. **Least Privilege Principle:** Limit the permissions and access rights of database accounts to the minimum required for specific tasks to minimize the potential damage of an attack.

5. **Web Application Firewalls (WAF):** Employ WAFs to filter and block malicious SQL injection attempts.
6. **Regular Security Testing:** Conduct regular security assessments, including penetration testing, to identify and address vulnerabilities.

**Vulnerability Name:** **Improper API Filtration**

**CWE-79:** Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

**OWASP Category:** A07-2017: Cross-site Scripting (XSS)

**Description:** The product does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

**Business Impact:** If exploited, this vulnerability allows attackers to inject malicious scripts into web pages, which can be viewed by other users. The potential business impacts include data theft, unauthorized access, compromised user accounts, damaged reputation, and financial losses. Attackers can use XSS to steal sensitive information, such as user credentials, personal data, or financial details, leading to regulatory compliance violations and legal liabilities. Additionally, defacement or manipulation of website content can erode customer trust and result in lost revenue.

**Vulnerability Path:** <https://www.shodan.io/host/31.3.96.40>

**Vulnerability Parameter:**

**Steps to Reproduce:**

**CVE-2020-13672**

**2.6** Cross-site Scripting (XSS) vulnerability in Drupal core's sanitization API fails to properly filter cross-site scripting under certain circumstances. This issue affects: Drupal Core 9.1.x versions prior to 9.1.7; 9.0.x versions prior to 9.0.12; 8.9.x versions prior to 8.9.14; 7.x versions prior to 7.80.

**Recommendation:**

1. **Update and Patch:** Ensure that Drupal and all related modules and components are up to date with the latest security patches to mitigate known vulnerabilities.
2. **Input Validation:** Implement strong input validation and output encoding to sanitize user inputs and prevent the injection of malicious scripts.
3. **Content Security Policy (CSP):** Implement CSP headers to restrict the sources from which content can be loaded, reducing the risk of XSS.
4. **Secure Coding Practices:** Train developers in secure coding practices, emphasizing the importance of validating and escaping user inputs.
5. **Security Scanning:** Conduct regular security scans and penetration tests to identify and address XSS vulnerabilities.

**Vulnerability Name:** Denial of Service

**CWE -400:** Uncontrolled Resource Consumption

**OWASP Category:**

**Description:** The Denial of Service (DoS) attack is focused on making a resource (site, application, server) unavailable for the purpose it was designed. There are many ways to make a service unavailable for legitimate users by manipulating network packets, programming, logical, or resources handling vulnerabilities, among others. If a service receives a very large number of requests, it may cease to be available to legitimate users. In the same way, a service may stop if a programming vulnerability is exploited, or the way the service handles resources it uses.

Sometimes the attacker can inject and execute arbitrary code while performing a DoS attack in order to access critical information or execute commands on the server. Denial-of-service attacks significantly degrade the service quality experienced by legitimate users. These attacks introduce large response delays, excessive losses, and service interruptions, resulting in direct impact on availability.

**Business Impact:** If successfully exploited, it involves overwhelming a system or application with excessive demands on its resources. This can lead to a range of disruptive consequences. The targeted system may experience

significant slowdowns, or even complete unresponsiveness, making it difficult or impossible for legitimate users to access. Critical resources like CPU, memory, and network bandwidth can become depleted, causing further degradation in performance. In severe cases, the exploited vulnerability may lead to system crashes or failures, potentially resulting in data loss. This disruption not only impacts business productivity and financial stability but also risks reputational damage, eroding customer trust and confidence. Additionally, organizations may face regulatory compliance issues if the uncontrolled resource consumption leads to violations of service level agreements or industry standards. Mitigating this vulnerability requires proactive resource management practices and a well-prepared incident response plan to minimize the potential impacts on business operations and customer satisfaction.

### **Vulnerability Path:**

### **Vulnerability Parameter:**

**Steps to Reproduce:** exploit open ports 80 to perform SQL injections, cross-site request forgeries and DDoS attacks.

```
(paavan@kali)-[~/Desktop]
$ sudo su
[sudo] password for paavan:
(root@kali)-[/home/paavan/Desktop]
# sudo nmap --top-ports 65536 31.3.96.40
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-18 14:18 IST
zsh: segmentation fault sudo nmap --top-ports 65536 31.3.96.40

(root@kali)-[/home/paavan/Desktop]
# sudo nmap -sV -p 80 31.3.96.40
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-18 14:18 IST
Nmap scan report for web.mmebvba.com (31.3.96.40)
Host is up (0.043s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.28 seconds

(root@kali)-[/home/paavan/Desktop]
# msfconsole
```

```
Metasploit tip: Open an interactive Ruby terminal with  
irb  
Metasploit Documentation: https://docs.metasploit.com/
```

```
msf6 > use auxiliary/dos/tcp/synflood  
msf6 auxiliary(dos/tcp/synflood) > set RHOST 31.3.96.40  
RHOST => 31.3.96.40  
msf6 auxiliary(dos/tcp/synflood) > set RPORT 80  
RPORT => 80  
msf6 auxiliary(dos/tcp/synflood) > RUN  
[-] Unknown command: RUN  
msf6 auxiliary(dos/tcp/synflood) > run  
[*] Running module against 31.3.96.40  
[*] SYN flooding 31.3.96.40:80 ...
```

**Recommendation:** Employ a robust Web Application Firewall (WAF) capable of filtering and mitigating suspicious traffic targeting port 80. Configure it to block excessive requests or patterns indicative of an attack. Utilize load balancing techniques to distribute traffic across multiple servers, ensuring no single server becomes overwhelmed. Additionally, implement rate limiting and throttling mechanisms to restrict the number of requests a single IP address can make within a defined time frame. Regularly monitor traffic patterns and set up alerts to detect unusual spikes in activity, enabling rapid response to potential attacks. Finally, keep software and systems up-to-date with the latest security patches to address known vulnerabilities that attackers may exploit.

**Vulnerability Name:** TLS Version 1.0 Protocol Detection

**CWE-327:** Use of a Broken or Risky Cryptographic Algorithm

**OWASP Category:** A02:2021-Cryptographic Failures

**Description:** The product uses a broken or risky cryptographic algorithm or protocol.

**Business Impact:** TLS 1.0 is an outdated and vulnerable encryption protocol, and its use can expose sensitive data to potential security threats. This not only jeopardizes the integrity of customer information and payment details but also tarnishes the company's reputation. The business may face legal

consequences, loss of customer trust, and decreased revenue as security-conscious users avoid the site due to its security vulnerabilities. Immediate action is necessary to upgrade to a more secure TLS version to safeguard data, ensure compliance, and maintain customer confidence.

### Vulnerability Path:

### Vulnerability Parameter:

### Steps to Reproduce:

1. Open a terminal or command prompt
2. Type the Command: `openssl s_client -connect 31.3.96.40:443 -tls1`
3. The connection is successfully means the server supports TLS 1.0

```
thanmaye08@THANMAYE: ~  
File Actions Edit View Help  
(thanmaye08@THANMAYE)-[~]  
$ openssl s_client -connect 31.3.96.40:443 -tls1  
CONNECTED(00000003)  
Can't use SSL_get_servername  
depth=0 CN = web.mmebvba.com  
verify error:num=18:self-signed certificate  
verify return:1  
depth=0 CN = web.mmebvba.com  
verify return:1  
-----  
Certificate chain  
0 s:CN = web.mmebvba.com  
i:CN = web.mmebvba.com  
a:PKKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256  
v:NotBefore: May 25 09:07:54 2015 GMT; NotAfter: May 22 09:07:54 2025 GMT  
-----  
Server certificate  
-----BEGIN CERTIFICATE-----  
MIICxDCCAaygAwIBAgIJALpeeeDC90PLMA0GCSqGSIb3DQEBCwUAMBoxGDAWBgNV  
BAMMD3dlYi5tbWVidmJhLmNvbTAeFw0xNTA1MjUwOTA3NTRaFw0yNTA1MjUwOTA3  
NTRaMBBoxGDAWBgNVBAMMD3dlYi5tbWVidmJhLmNvbTCCASIwDQYJKoZIhvcNAQEB  
BQADggEPADCCAQoCggEBAK/hFLEmVRyv2fbXel6oUukbiGdpX0Xps5sqis20xF9Y  
/YXJcxaPoI/fUyfSdJ6prrr2/YoEcUUFZaUIJWbsSz/qzklja0M+GMAVzL+wqAi+V  
HjgujqsgeEQgxgTiNoQu9Zld5NY7Ac9qQRKbLRLRubQV/9+Dud0Isx3z1gAsJda6  
bi3JuL0a27EMztRssKeQCLJhWLAdidjuXH6kc+3jBNPmvedRF+Mw5mk32nNtyZ78  
qjIFT7Bny/CNGN31PFgQWq+6pTgCzokui04zV0pYyaGX9hs6roLiQWLT9UqB0wx0  
Jr+GM175u2YLvXba+UZ2cWMOp5IZ3590Htkf0uuKSzUCAwEAAMNMAswCQYDVR0T  
BAIwADANBgkqhkiG9w0BAQsFAAOCAQEAghKAH91qFsuLDarYKIrjuUxFeI2bE1fLd  
VGiYBzmaQddzehzBu9Akn9g73DHwrs9+gW9cDfKCLVr3ayzS5KAMhvnZzga1B0+q  
NYfX2y61SMnS4w9p8PT3iqZdF/Rywu658BXjpJWojwoA8Hu1mhwT0DdSSeOfBnIq  
azNm62lQsxdTLU7wEanZ0/ExUFNktwKean7Jle3EvH94fkNg2T0er1WuEB/QF6sv  
dJqxsYSZ3toPEuqlOryWis0E8pfo3Hauzz6fBGijLcEnMoFp7oCZNz0Xh9Q93a0s  
5c9hNmHtQcu3vMrYUYArpj+a/euAEI2HaGJZf5g75mf4gAI/2B7iyg=  
-----END CERTIFICATE-----  
subject=CN = web.mmebvba.com  
issuer=CN = web.mmebvba.com  
-----  
No client certificate CA names sent  
Peer signing digest: MD5-SHA1  
Peer signature type: RSA  
Server Temp Key: ECDH, prime256v1, 256 bits  
-----  
SSL handshake has read 1404 bytes and written 274 bytes  
Verification error: self-signed certificate  
-----  
New, TLSv1.0, Cipher is ECDHE-RSA-AES256-SHA  
Server public key is 2048 bit  
Secure Renegotiation IS supported  
Compression: NONE  
Expansion: NONE
```



```

No ALPN negotiated
SSL-Session:
  Protocol      : TLSv1
  Cipher       : ECDHE-RSA-AES256-SHA
  Session-ID: 88C081DC5A60B2165E5ECF1174DFFA77732428624861E0BFAC934E3DE969B9E2
  Session-ID-ctx:
  Master-Key: 9A895BD34544E92181A631AB636C46EEC2788E9BFEE0DA5A6BDCA50E65C55250BA88EA
  PSK identity: None
  PSK identity hint: None
  SRP username: None
  TLS session ticket lifetime hint: 300 (seconds)
  TLS session ticket:
0000 - 11 e0 e2 5c 44 79 52 91-7b 32 b1 10 22 9f 80 b5    ... \DyR.{2.. " ...
0010 - 54 b6 b7 89 fe c7 7d 4a-18 bd 8c e3 e6 31 47 63    T.....}J.....1Gc
0020 - 38 1b 52 ec 16 ac 18 b6-bd e1 76 e9 4b 41 24 ad    8.R.....v.KA$.
0030 - e3 94 d2 60 65 35 8b e0-6e d2 76 db 1b e1 03 e7    ... `e5..n.v.....
0040 - e7 76 cc f9 70 6a 42 91-94 42 e2 e8 29 d5 fd 38    .v.. pJB..B..) ..8
0050 - 63 5f da 9f 09 af e0 d9-22 29 45 70 a7 a4 36 1f    c_.....")Ep..6.
0060 - 51 b1 cd d5 e7 b9 ef 9d-5d 20 99 ad 00 4d 21 df    Q.....] ...M!.
0070 - 38 a6 c9 1b 76 29 87 d9-14 79 7a 73 33 f8 4c 8f    8 ... v) ... yzs3.L.
0080 - 12 ae 66 c1 ad c3 6c ba-68 63 11 6c b2 18 e0 89    .. f ... l.hc.l....
0090 - 97 54 50 d7 ee f5 e1 37-f4 b0 be 55 2f d1 25 63    .TP....7 ... U/.%c
00a0 - c7 61 10 b3 ad 74 5a b7-07 fe e5 20 b1 da a1 a2    .a ... tZ.... ....
00b0 - be dd c8 45 a7 74 b7 eb-ee 9a 15 dc 1b bb 0b d9    ... E.t.....

Start Time: 1697691531
Timeout      : 7200 (sec)
Verify return code: 18 (self-signed certificate)
Extended master secret: no

4037A3514B7F0000:error:0A000126:SSL routines:ssl3_read_n:unexpected eof while reading:

```

## Recommendation:

1. **Upgrade to a Secure TLS Version:** Transition to a more secure TLS version, such as TLS 1.2 or TLS 1.3. This will provide stronger encryption and improved security against modern threats.
2. **Regularly Update SSL/TLS Configuration:** Keep your SSL/TLS configuration up to date to ensure it aligns with the latest security best practices and vulnerabilities.
3. **Implement a Strong Cryptographic Suite:** Use strong cipher suites and secure configurations to further enhance security. Disable weak ciphers and outdated protocols.
4. **Compliance Checks:** Ensure that your website complies with relevant data protection regulations and industry standards (e.g., GDPR, PCI DSS) to avoid legal issues.

**Vulnerability Name:** [web.config File Information Disclosure](#)



## **CWE-497:** Exposure of Sensitive System Information to an Unauthorized Control Sphere

**OWASP Category:** A3:2017-Sensitive Data Exposure

**Description:** An information disclosure vulnerability exists in the remote web server due to the disclosure of the web.config file. An unauthenticated, remote attacker can exploit this, via a simple GET request, to disclose potentially sensitive configuration information.

**Business Impact:** It exposes sensitive configuration details, including database connection strings and encryption keys, to potential attackers. This could lead to unauthorized access, data breaches, and system compromise. The impact includes financial losses, damage to reputation, and legal liabilities as the organization's security and customer trust are compromised.

**Vulnerability Path:**

**Vulnerability Parameter:**

**Steps to Reproduce:**

```
GET /web.config HTTP/1.1
Host: web.mmebvba.com
Accept-Charset: iso-8859-1,utf-8;q=0.9,*;q=0.1
Accept-Language: en
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
Pragma: no-cache
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
```

```
This produced the following truncated output (limited to 5 lines) :
----- snip -----
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
<system.webServer>
<!-- Don't show directory listings for URLs which map to a directory. -->
<directoryBrowse enabled="false" />
[...]
```

### **Recommendation:**

- Implement proper access controls to restrict access to the web.config file.
- Prevent directory traversal and ensure the web.config file cannot be directly accessed through the web server.
- Encrypt sensitive data within the configuration file.

- Regularly review and update the web.config file, removing any unnecessary sensitive information.
- Ensure that the web.config file is properly secured and not accessible to unauthorized users.