System That Uses AI For Real-Time Data Classification, Leak Detection, And Prevention To Enhance Data Security.

Main Website: vtop2.vitap.ac.in

Practice Website: testphp.vulnweb.com

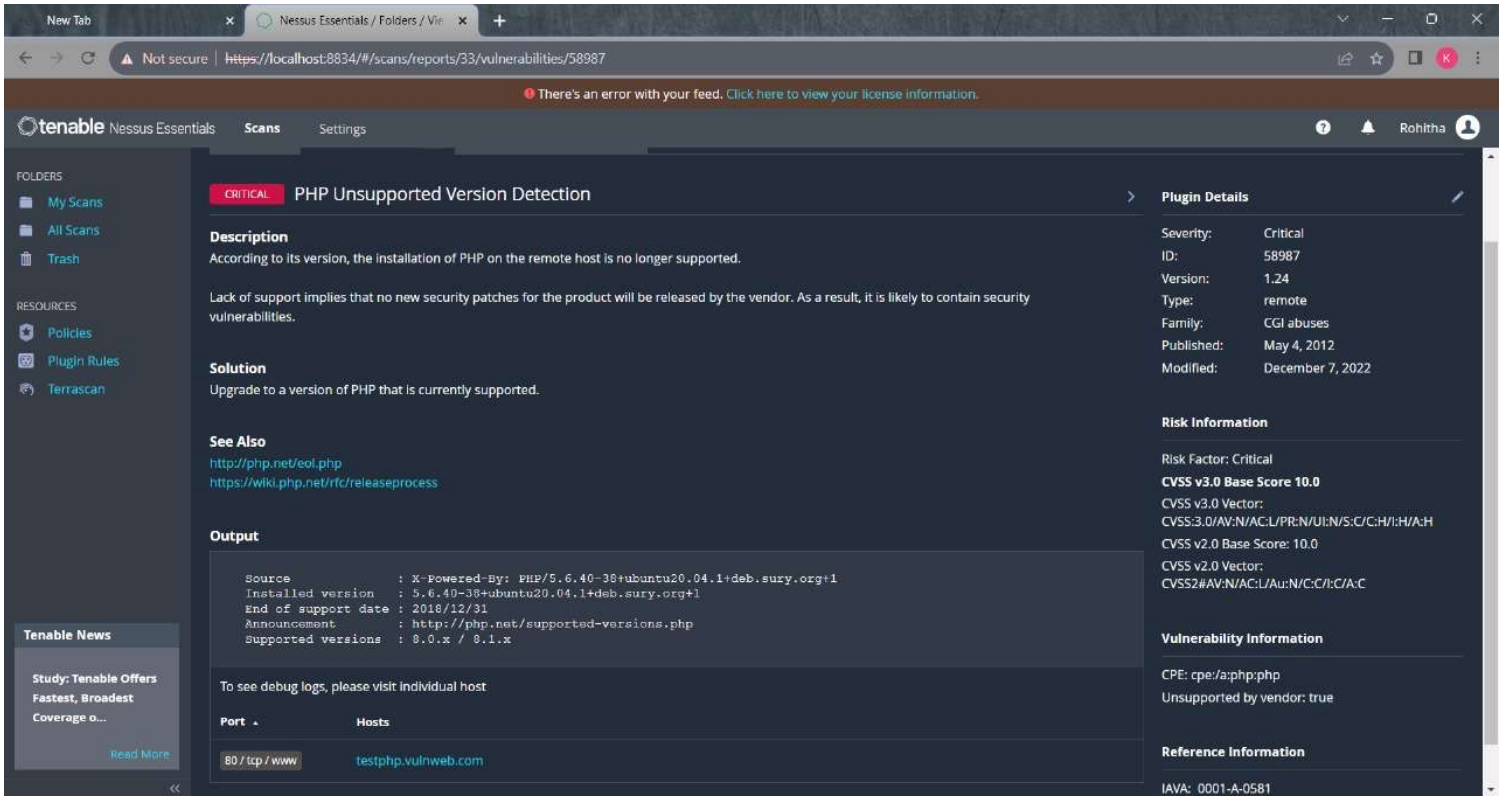
Vulnerabilities found in Practice Website:

1. **58987 - PHP Unsupported Version Detection**

   Synopsis
   The remote host contains an unsupported version of a web application scripting language.

   Description
   According to its version, the installation of PHP on the remote host is no longer supported.

   Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.



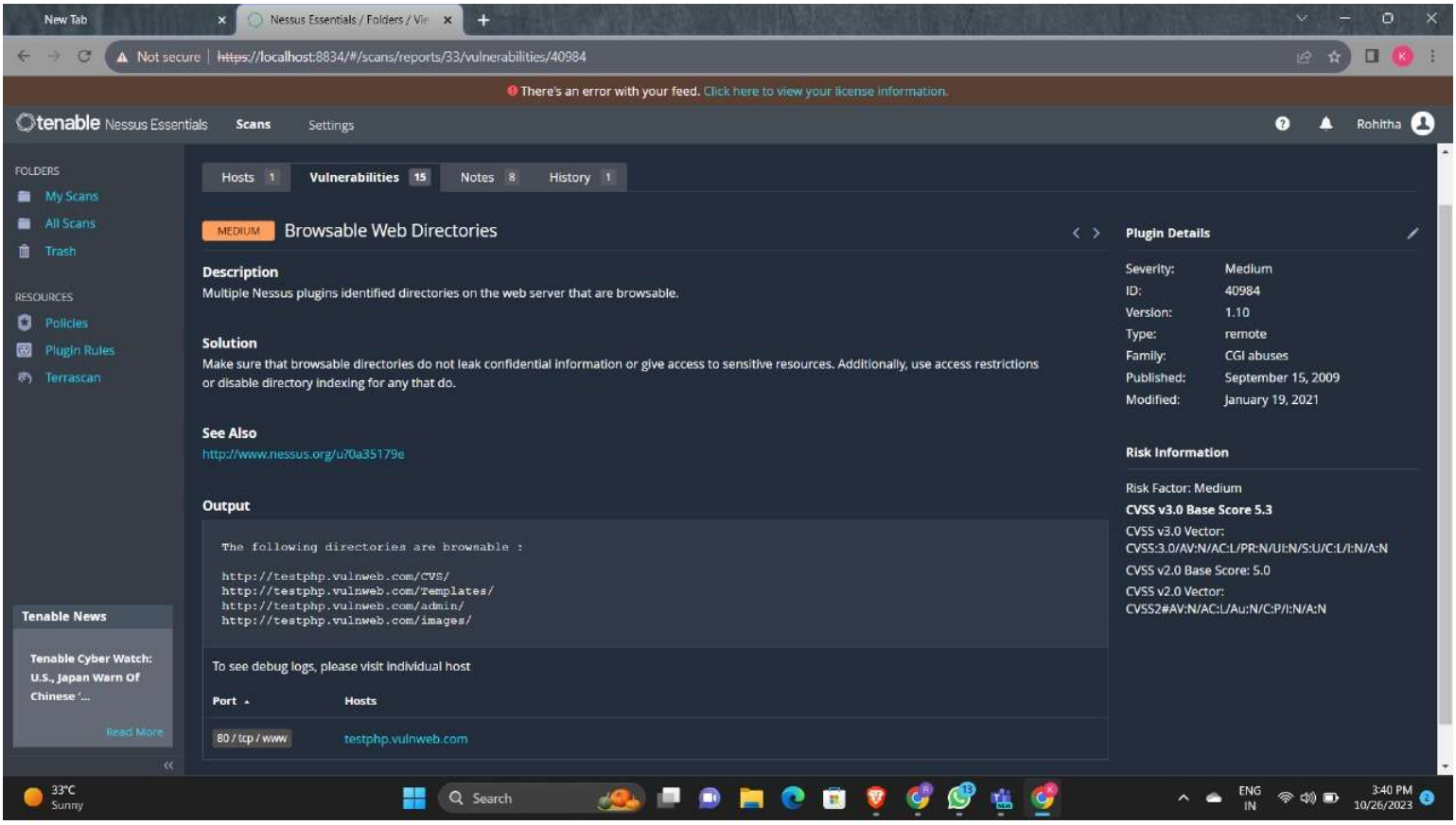
2. 40984 - Browsable Web Directories

Synopsis

Some directories on the remote web server are browsable.

Description

Multiple Nessus plugins identified directories on the web server that are browsable.



3. 85582 - Web Application Potentially Vulnerable to Clickjacking

Synopsis

The remote web server may fail to mitigate a class of web application vulnerabilities.

CWE: CWE-693

Description

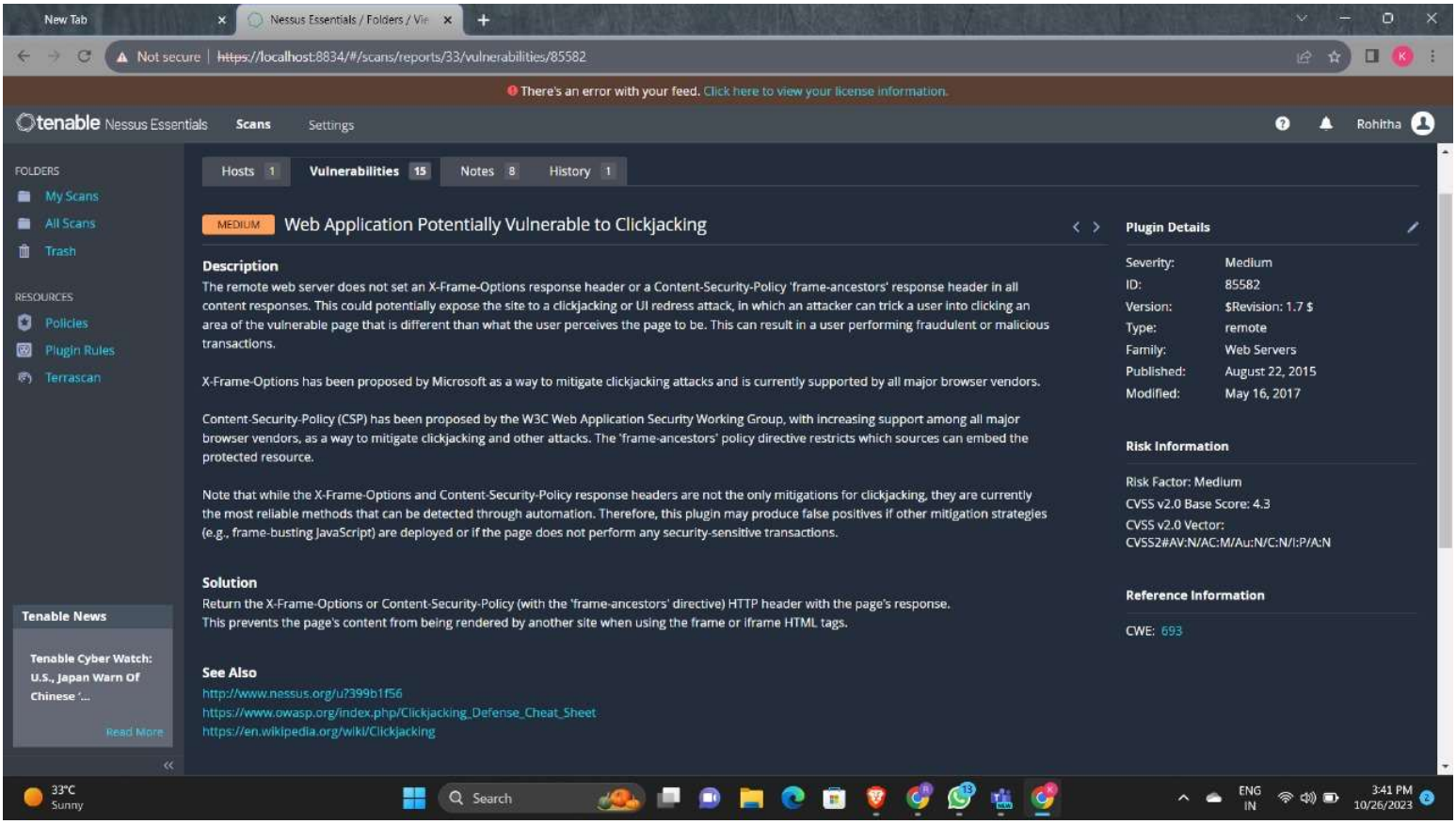
The remote web server does not set an X-Frame-Options response header or a Content-Security-Policy 'frame-ancestors' response header in all content responses. This could potentially expose the site to a clickjacking or UI redress attack, in which an attacker can trick a user into clicking an area of the vulnerable page that is different than what the user perceives the page to be. This can result in a user

performing fraudulent or malicious transactions.

X-Frame-Options has been proposed by Microsoft as a way to mitigate clickjacking attacks and is currently supported by all major browser vendors.

Content-Security-Policy (CSP) has been proposed by the W3C Web Application Security Working Group, with increasing support among all major browser vendors, as a way to mitigate clickjacking and other attacks. The 'frame-ancestors' policy directive restricts which sources can embed the protected resource.

Note that while the X-Frame-Options and Content-Security-Policy response headers are not the only mitigations for clickjacking, they are currently the most reliable methods that can be detected through automation. Therefore, this plugin may produce false positives if other mitigation strategies (e.g., frame-busting JavaScript) are deployed or if the page does not perform any security-sensitive transactions.



4. 26194 - Web Server Transmits Cleartext Credentials

Synopsis

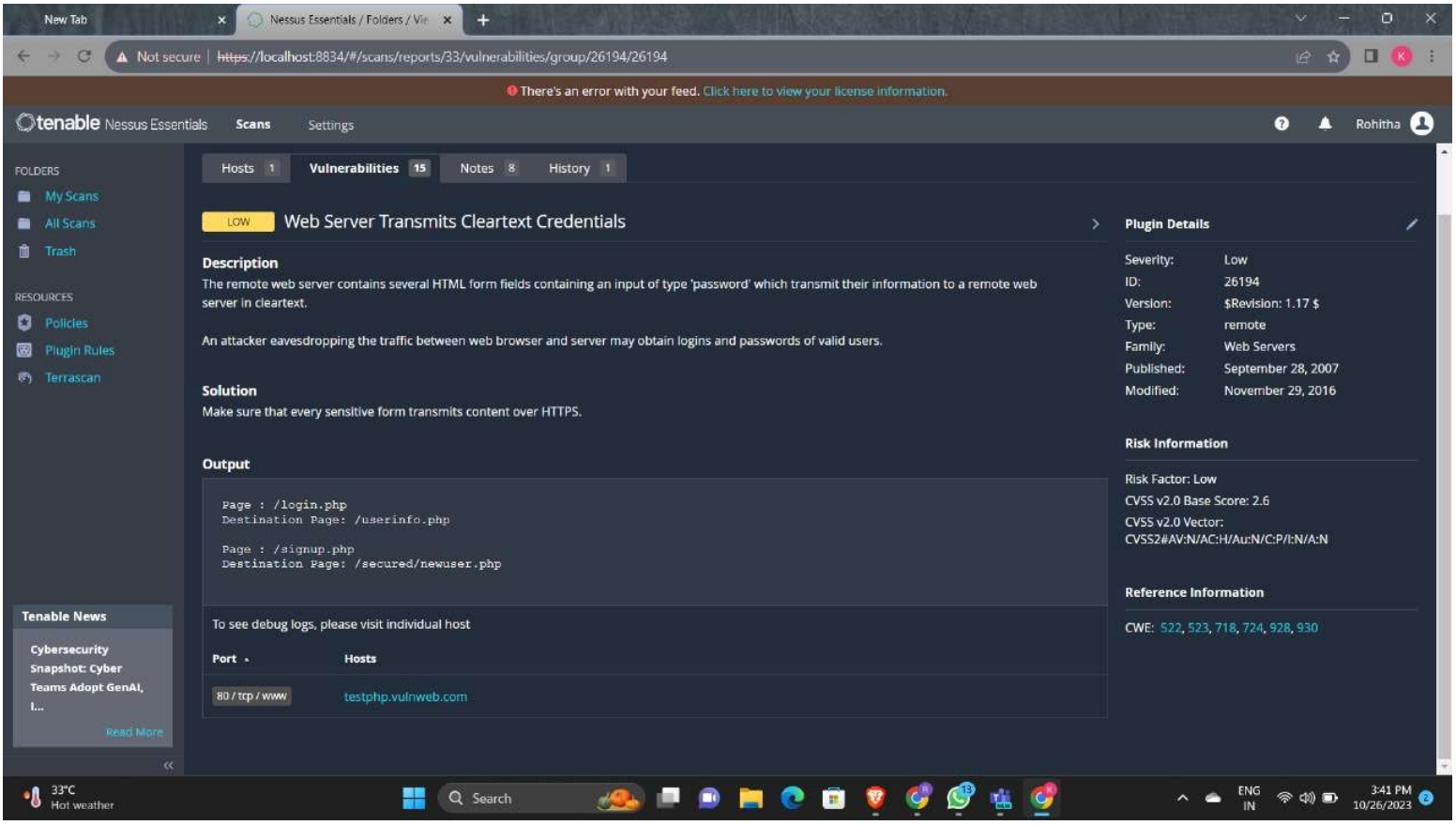
The remote web server might transmit credentials in cleartext.

CWE: CWE-522/ CWE-523/ CWE-718/ CWE-724/ CWE-928/ CWE-930

Description

The remote web server contains several HTML form fields containing an input of type 'password' which transmit their information to a remote web server in cleartext.

An attacker eavesdropping the traffic between web browser and server may obtain logins and passwords of valid users.



5. **42057 - Web Server Allows Password Auto-Completion**

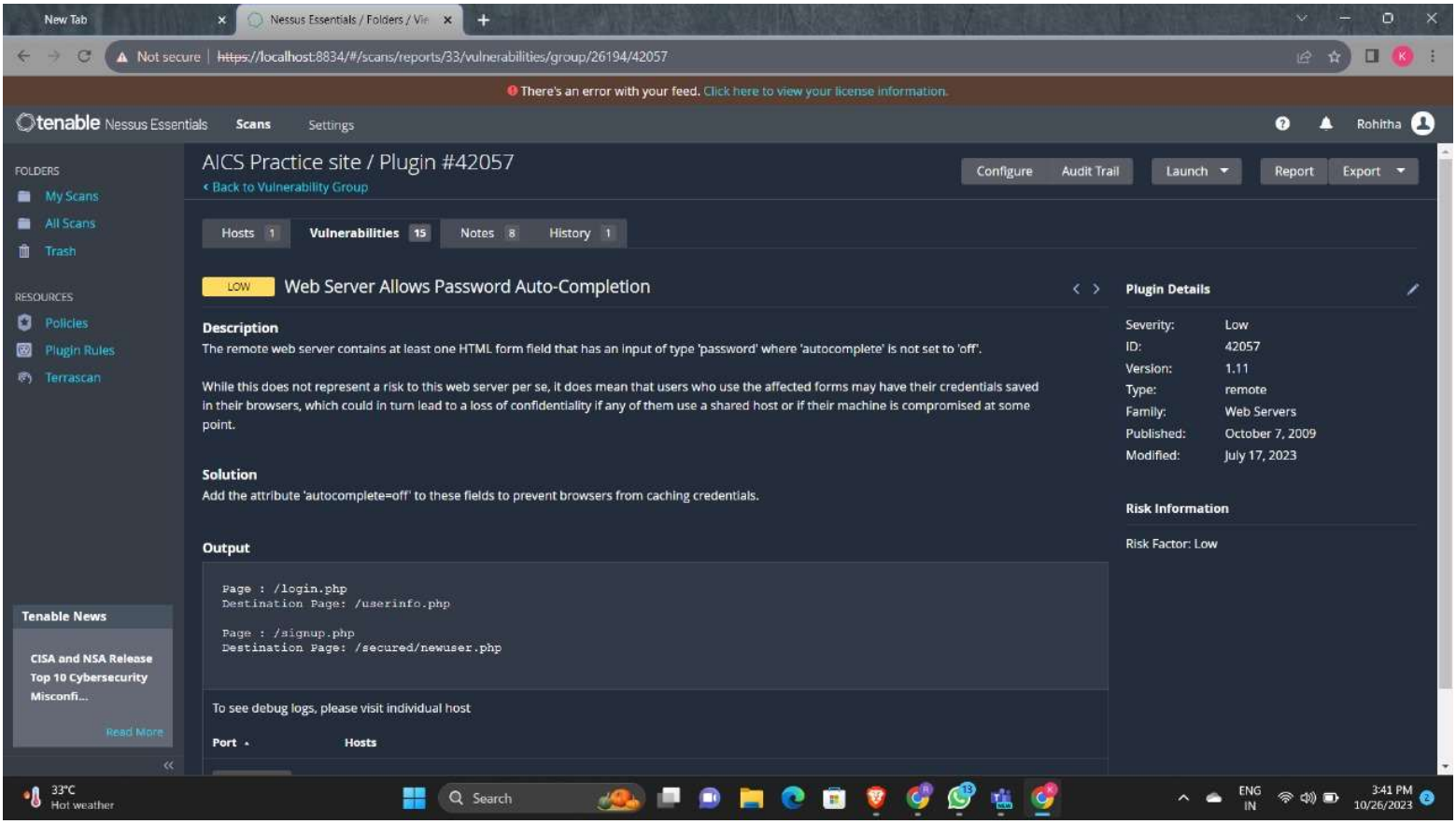
Synopsis

The 'autocomplete' attribute is not disabled on password fields.

Description

The remote web server contains at least one HTML form field that has an input of type 'password' where 'autocomplete' is not set to 'off'.

While this does not represent a risk to this web server per se, it does mean that users who use the affected forms may have their credentials saved in their browsers, which could in turn lead to a loss of confidentiality if any of them use a shared host or if their machine is compromised at some point.



6. CGI Generic HTML Injections (quick test)

Synopsis
The remote web server may be prone to HTML injections.

CWE: CWE-80/ CWE-86

Description
The remote web server hosts CGI scripts that fail to adequately sanitize request strings with malicious JavaScript. By leveraging this issue, an attacker may be able to cause arbitrary HTML to be executed in a user's browser within the security context of the affected site.

The remote web server may be vulnerable to IFRAME injections or cross-site scripting attacks:

- IFRAME injections allow 'virtual defacement' that might scare or anger gullible users. Such injections are sometimes implemented for 'phishing' attacks.

- XSS are extensively tested by four other scripts.

- Some applications (e.g., web forums) authorize a subset of HTML without any ill effect. In this case, ignore this warning.

MEDIUM CGI Generic HTML Injections (quick test)

Description

The remote web server hosts CGI scripts that fail to adequately sanitize request strings with malicious JavaScript. By leveraging this issue, an attacker may be able to cause arbitrary HTML to be executed in a user's browser within the security context of the affected site.

The remote web server may be vulnerable to IFRAME injections or cross-site scripting attacks :

- IFRAME injections allow 'virtual defacement' that might scare or anger gullible users. Such injections are sometimes implemented for 'phishing' attacks.

- XSS are extensively tested by four other scripts.

- Some applications (e.g. web forums) authorize a subset of HTML without any ill effect. In this case, ignore this warning.

Solution

Either restrict access to the vulnerable application or contact the vendor for an update.

See Also

http://www.nessus.org/u?602759bc

7. **CGI Generic SQL Injection**

Synopsis

A web application is potentially vulnerable to SQL injection.

CWE: CWE-20/ CWE-89/ CWE-77/ CWE-722/ CWE-751/ CWE-801/ CWE-928/ CWE-713/ CWE-727/ CWE-810/ CWE-929/ CWE-203/ CWE-209 /CWE-933/ CWE-717

Description

By providing specially crafted parameters to CGIs, Nessus was able to get an error from the underlying database. This error suggests that the CGI is affected by a SQL injection vulnerability.

An attacker may exploit this flaw to bypass authentication, read confidential data, modify the remote database, or even take control of the remote operating system.

HIGH CGI Generic SQL Injection

Description

By providing specially crafted parameters to CGIs, Nessus was able to get an error from the underlying database. This error suggests that the CGI is affected by a SQL injection vulnerability.

An attacker may exploit this flaw to bypass authentication, read confidential data, modify the remote database, or even take control of the remote operating system.

Solution

Modify the relevant CGIs so that they properly escape arguments.

See Also

https://en.wikipedia.org/wiki/SQL_injection
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://www.nessus.org/u?ed792cf5
http://projects.webappsec.org/w/page/13246963/SQL%20Injection
https://www.owasp.org/index.php/SQL_Injection

8. **Web Application SQL Backend Identification**

Synopsis

A web application's SQL backend can be identified.

Description

At least one web application hosted on the remote web server is built on a SQL backend that Nessus was able to identify by looking at error messages.

Leaking this kind of information may help an attacker fine-tune attacks against the application and its backend.

Vulnerabilities 27

MEDIUM Web Application SQL Backend Identification

Description

At least one web application hosted on the remote web server is built on a SQL backend that Nessus was able to identify by looking at error messages.

Leaking this kind of information may help an attacker fine-tune attacks against the application and its backend.

Solution

Filter out error messages.

See Also

http://projects.webappsec.org/w/page/13246925/Fingerprinting

Output

```
The web application appears to be based on MySQL

This information was leaked by these URLs :
http://testphp.vulnweb.com/
```

9. **CGI Generic Cookie Injection Scripting**

Synopsis

The remote web server is prone to cookie injection attacks.

CWE: CWE-722/ CWE-715/ CWE-472/ CWE-642

Description

The remote web server hosts at least one CGI script that fails to adequately sanitize request strings with malicious JavaScript.

By leveraging this issue, an attacker may be able to inject arbitrary cookies. Depending on the structure of the web application, it may be possible to launch a 'session fixation' attack using this mechanism.

MEDIUM CGI Generic Cookie Injection Scripting

Description

The remote web server hosts at least one CGI script that fails to adequately sanitize request strings with malicious JavaScript.

By leveraging this issue, an attacker may be able to inject arbitrary cookies. Depending on the structure of the web application, it may be possible to launch a 'session fixation' attack using this mechanism.

Please note that :

- Nessus did not check if the session fixation attack is feasible.

- This is not the only vector of session fixation.

Solution

Restrict access to the vulnerable application. Contact the vendor for a patch or upgrade.

See Also

https://en.wikipedia.org/wiki/Session_fixation
https://www.owasp.org/index.php/Session_Fixation
http://www.acros.si/papers/session_fixation.pdf
http://projects.webappsec.org/w/page/13246960/Session%20Fixation

10. CGI Generic SQL Injection (2nd pass)

Synopsis

A web application is potentially vulnerable to SQL injection.

CWE: CWE-20/ CWE-89/ CWE-77/ CWE-722/ CWE-801/ CWE-928/ CWE-713/ CWE-727/ CWE-810/ CWE-929

Description

By providing specially crafted parameters to CGIs, Nessus was able to get an error from the underlying database. This error suggests that the CGI is affected by a SQL injection vulnerability.

An attacker may exploit this flaw to bypass authentication, read confidential data, modify the remote database, or even take control of the remote operating system.

HIGH

CGI Generic SQL Injection (2nd pass)

Description

By providing specially crafted parameters to CGIs, Nessus was able to get an error from the underlying database. This error suggests that the CGI is affected by a SQL injection vulnerability.

An attacker may exploit this flaw to bypass authentication, read confidential data, modify the remote database, or even take control of the remote operating system.

Solution

Modify the relevant CGIs so that they properly escape arguments.

See Also

https://en.wikipedia.org/wiki/SQL_injection
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://www.nessus.org/u?e5c79f44
http://www.nessus.org/u?11ab1866