Directory Traversal Vulnerability

https://campus.barracuda.com/doc/42049342/

Description

The directory traversal/path traversal attack (also known as *dot dot slash* attack) is an HTTP exploit that allows an attacker to access restricted files, directories and commands that reside outside the web server's root directory. Directory traversal attacks are executed through web browsers. An attacker may manipulate a URL in such way that the website will reveal the confined files on the web server.

Typically, web servers provide two security mechanisms to restrict user access:

- Access Control Lists (ACLs)
- Web Document Root Directory

The access control list determines which users or groups are privileged to access, modify or execute files on the web server. Users are restricted from accessing the specific part of the file-system on the server, which is known as "root", "web document root", or "CGI root" directory. The attacker uses special-character "../"sequence to escape web document root, or alternate encodings of the "../" sequence to bypass security filters and access files or directories that reside outside the root directory. Some directory traversal attack variations include:

- URL encoded characters - "..%2e%2e%2f" for forward slash character (../), "..%2e%2e%5c" for backslash character (..\).
- Unicode encoding - "..%u2216" for "../" and "..%c0%af" for "..\".
- Double encoding - "..%252F" for "../" and "..%255C" for "..\".

These techniques employ special characters such as the dot (".") or NULL ("%00") character obfuscate directory traversal exploits.

A directory traversal vulnerability can exist either in web servers or web applications. Web applications that fail to validate input parameters (i.e. form parameters, cookie values, etc.) are vulnerable to directory traversal attacks.

Effects

If a web server or web application is vulnerable to directory traversal attack, the attacker can exploit the vulnerability to reach the root directory and access restricted files and directories. The attackers

can modify critical files such as programs or libraries, download password files, expose source code of the web application, or execute powerful commands on the web server, which can lead to complete compromise of the web server.

Methods

The directory traversal attack occurs when the user-supplied input is not properly filtered or sanitized. The following data must be sanitized properly before being processed:

- URL Parameters
- FORM Parameters (GET and POST parameters)
- Cookies
- HTTP Request Headers

Examples

If the server is vulnerable to directory traversal attack:

When a normal website visitor accesses http://www.vulnerable.com/index.html, he would be able to view the content in this page. The index.html resides in a web directory in the server where all web pages are stored. The web user can navigate and access the pages to which user access privileges has been defined by the administrator. In this example, the web user is allowed to access other files that are not in the “web root” directory of the server. Hence, the server is vulnerable to directory traversal attack. If the attacker intends to access the password file which is on the server, then the attacker would send the below request and access the sensitive information from the server.

http://www.vulnerable.com/../../../etc/passwd

If the application is vulnerable to directory traversal attack:

Consider a daily online news website (www.example.news.com) which has different sections and news articles in it. The user clicks on the first news article, the request goes to http://www.example.news.com/news=new_page1.html, where new_page1.html content is displayed to the user. In the same way, clicking on the second article, the URL browser shows http://www.example.news.com/news=new_page2.html.

If the user is an attacker and wants to view other files on the server, he would try to replace new_page1.html or new_page2.html with /etc/shadow i.e. http://www.vulnerable.com/news=/etc/shadow. This may not be successful as /etc/shadow might not

be residing in the same directory as of new_page1.html. The attacker may execute different trial and error methods to find the exact way to get access to /etc/shadow. The attacker sends the below request and retrieves the file in the server:

http://www.vulnerable.com/news=../../../../etc/shadow

With this, the attacker can steal the sensitive information about the user accounts in the server.

Prevention

To prevent directory traversal attack, it is necessary to perform proper input validation on all the entities mentioned in the **Methods** section above. This includes normalizing the input data to account for obfuscations and encodings.

For applications being actively developed, such filtering and validation should be part of the SDLC and developers or testing teams should be trained to identify and prevent such vulnerabilities.

For legacy and third party code, however, this might not be an easy thing to enforce. The Barracuda Web Application Firewall dynamically remediates this vector by identifying and blocking directory traversal patterns across all the HTTP entities where they can occur.

Tags

OWASP Top 10, PCI-DSS

See Also

CWE 22, OWASP, WASC