Brute Force Attack

https://campus.barracuda.com/doc/42049329/

Description

Brute force attack is a technique used to explore an unknown value by systematically trying every key combination to gain access to the targeted resource. In the context of web applications, such attacks appear as a volley of HTTP requests that successively cycle through a user input value till the “right” value is hit. This value could be a GET or POST parameter, usernames and passwords, URL paths or header values. Such attacks are carried out using automated tools and scripts that try every possible character combination to explore the value that is sought.

Attackers often make use of the fact that invalid inputs to web applications yield a different page than valid values. For example, an invalid username could yield one error message and an invalid password could yield another and a successful login yields a totally different page. An attacker can then write a script that cycles through username values till the error message is “invalid user”. When the error changes to “invalid password” the attacker can identify a valid username, and then proceed to cycle through passwords for that valid username, until the correct password is hit.

The other weakness that facilitates this attack is the lack of a policy to enforce a maximum attempt count to access a particular resource.

In addition to targeting login credentials, a brute force attack could also be used for guessing hidden pages or content, session ID values, one time passcodes, credit card numbers, and even reversing cryptographic hash functions.

Because brute force attacks from a single client could be easy to spot and block, attackers frequently use multiple attack sources that try to attack the web application in concert. Therefore , a common by-product of brute force attacks is resource exhaustion on the server that could degrade the quality of service to genuine client.

Indications of a Brute Force Attack

Since brute force attacks require trial and error of a large set of values, the most common indicator is an unusual volume of failed requests. When a parameter is being attacked (like username) then the requests are all to the same page. If the attacker is trying to find hidden pages, then each request would be different but the server response codes will be 404: Page Not Found.

Effects

A successful brute force attack can result in the following:

- It can leak confidential and private data (for example: user's profile data, bank details, financial status).
- It can leak hidden files or interfaces (for example: admin interface).
- It can disrupt the service if the service is attacked to the point of causing a denial of service (DoS).

If the attackers succeed in gaining access to administrative panels, they can modify/delete/add web application content, modify user privileges, and more.

Example: Brute Force Attack to Identify a URL in a Web Application

The attacker uses a word list of known pages to execute a brute force attack on a web application. In the example below, the attacker tries a brute force attack on a popular content management system. The attacker sends request to each known page and then analyzes the HTTP response code to determine if the requested page exists on the target server.

[root@localhost wfuzz-2.1-beta]# python wfuzz.py -c -z file,wordlist/general/common.txt --hc 404 http://X.X.X.X/FUZZ

****

* Wfuzz 2.1 - The Web Bruteforcer *

****

Target: http://X.X.X.X/FUZZ

Total requests: 950

==========

ID Response Lines Word Chars Request

==========

```
00213:  C=200     2 L       1 W     8 Ch   "default"

00457:  C=301     7 L      20 W           239 Ch          "lost%2Bfound"

00472:  C=301     7 L      20 W           235 Ch          "manual"

00584:  C=301     7 L      20 W           235 Ch          "portal"

00759:  C=200   828 L    2150 W       1275626 Ch        "script"

00783:  C=301     7 L      20 W           233 Ch          "test"


Total time: 19.71608

Processed Requests: 950

Filtered Requests: 944

Requests/sec.: 48.18401
```

How to Limit Attacks

Brute force attacks are difficult to stop completely, but with proper countermeasures and a carefully designed website, it is possible to limit these attacks. Use the following measures on your login pages to defend against brute-force attacks:

- Enforce long and secure passwords.
- Limit the number of failed login attempts and block users who attempt to log in using different passwords within a short period of time. Note that this could potentially end up blocking genuine users, if attackers use their usernames too many times in failed login attempts.
- Challenge suspicious requests with CAPTCHA or other challenges to prevent automated attacks.

The Barracuda Web Application Firewall allows you to restrict the maximum attempts to access resources in a given time window. The counting can be done per source IP or across all sources. When clients violate the access policy, they can be either presented with a CAPTCHA to prove they are humans and not scripts or locked out for a time period you specify.

Tags

OWASP Top 10, PCI-DSS

See Also

CWE 307, CWE 799, OWASP, WASC