Finding 3: Lack of Password Brute-force Defenses

Severity	MEDIUM
Likelihood	Medium
Impact	Medium
CVSS v3.1	4.8 AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N
Status	Open

Current Behavior

The Acme CRM application allows an unlimited number of authentication attempts against an account without locking the account or slowing the login process.

Impact

There are a number of common tools that will automatically submit lists of usernames and passwords to a website. If an attacker can discover or guess a valid user account, and if that user has chosen a common password, then it is likely that the credentials can be discovered. While there are a number of controls that can prevent these attacks, limiting the number or rate of login attempts that can be made is the most effective control that does not significantly impact the user experience.

Evidence

Meristem used an automated tool to rapidly submit 50 incorrect passwords for a known valid account, administrator@test.acme.com. The tool then submitted the correct password on the 51st attempt. The application accepted this login and created a new session. The output from the tool is shown below along with the successful final response.

```
Request      Payload      Status Response ms  Error  Timeout      Length
48           1111         302    29           false  false        254
49           austin       302    28           false  false        254
50           william      302    28           false  false        254
51           TEST         200    53           false  false        1280
```

Response to the final request:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=UTF-8
Content-Length: 1117
Date: Wed, 19 Oct 2022 01:44:44 GMT
Connection: close

<HTML>
  <HEAD>
    <TITLE>Wait...</TITLE>
```

```
<SCRIPT LANGUAGE="javascript" SRC="../javascript/cookies.js"></SCRIPT>
<SCRIPT TYPE="text/javascript">
  <!--
    // status = 1

    var dtInf = new Date(2030, 11, 30, 0, 0, 0, 0);

    setCookie ("profilenm","hipergate",dtInf);
    setCookie ("domainid","2050",dtInf);
    setCookie ("domainnm","TEST",dtInf);
    setCookie ("skin","xp",dtInf);
    setCookie ("face","crm",dtInf);
    setCookie ("userid","7f0000011821e62e6e1100004b81585f");
    [TRUNCATED]
```

Recommendation

Implement a control that limits the number of login attempts that can be made for a single account, the rate at which attempts can be made, or both. One approach is to increment a server-side counter each time a failed login attempt is made for a specific account. If a limit, typically 10 or less, is reached, all further login attempts should be rejected, even if the correct password is submitted. With this approach an unlock process must also be implemented that allows an administrator or customer service representative to reset the failed count. The count could also be reset automatically after an appropriate time has passed, typically 15 minutes or more.

Another approach is to track the number of failed login attempts and add a delay to the login response after a limit has been reached, typically 5 attempts or less. While this 1 to 20 second delay could be a small inconvenience to a legitimate user, it makes the time required for a brute force attack too long to be attractive to an attacker. The counter can be reset when the successful password is entered, when the password is reset, or when sufficient time has passed.

References

- https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html
- https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks