Twitter Account Lockouts Continue to Plague Users

Lori MacVittie, 2009-05-11

Brute force attacks by spammers seeking easy access causing frustration for users with no resolution in sight

At least once a day I see someone on Twitter broadcast that they have been "locked out of their Twitter account, temporarily." A search for "locked out" returns thousands of tweets with a good mixture of some folks who've (amusingly) been locked out of apartments/houses/buildings and many that have been temporarily locked out of Twitter. The more technically savvy tweeters like Ray Valdes often mention that it is most likely the result of spammers and miscreants attempting to brute force their way into their account, but usually it's just the beginning of rant against Twitter and how "stupid" it is to lock them out of their web account. Some of those rants are quite, shall we say, colorful and don't need to be reproduced here. You can use your imagination, I'm sure.

WHY ACCOUNT LOCKOUTS WILL CONTINUE TO HAPPEN

rayval: **Locked out** of my Twitter account temporarily due to "too many failed login attempts". Except I hadn't tried to login. Someone else probing.
31 minutes ago from *web* · Reply · View Tweet

Brute force attempts to gain access through users by spammers and other miscreants is a common occurrence in web applications. For Twitter, and really any Web 2.0 application providing API access through which third party applications can connect, the methods of determining what is a brute-force attack and what is simply a user who has (1) forgotten their password or (2) forgotten to change all passwords in all applications that access the application is exceedingly difficult.

S_Wash: Stupid twitter has me **locked out**! You can stifle me, ha!!
about 3 hours ago from *txt* · Reply · View Tweet

johnreeder: trying to change my PW on twitter site but am "locked" out. but tweetdeck is still working. weird.
about 1 hour ago from *TweetDeck* · Reply · View Tweet

Locking users out of their accounts when they are the victim of a brute force attack is a common security practice, designed to prevent compromise and, in many cases, it's the *only* option to prevent continued attempts through such persistent attacks. The problem is that the application can't take into account the subtle indicators that differentiate between a brute force attack and a user who's simply forgotten to update one password or another, or forgotten their password entirely.

RoxetteMabellon: part2 : These scammers try to access your account over and over. This result in a #twitter "lock out" !!! #twitter #warning #locked out
about 3 hours ago from *web* · Reply · View Tweet

kairostcheck: Yay! Back online after being hacked then **locked out** on Monday/Tuesday. Sorry to those who received that "I made $ on Google" update.
about 3 hours ago from *web* · Reply · View Tweet

That's because brute force attacks attempting to compromise an account are an ***application layer attack*** and there's no good way for Twitter – or any other application – to recognize them. What indicators there are that a brute force attack is occurring require the ability to evaluate individual requests in the ***context*** of all requests. For example, a suddenly high volume of requests for "login.html" or URIs/API calls associated with the authentication process coupled with increasing load on servers is a good indicator that *something* is going on and that "something" is probably not good.

fourthversion: There is nothing more infuriating than Twitter telling you to chill out when your account has been **locked out**.
about 3 hours ago from *web* · Reply · View Tweet

The application, Twitter in this case, when processing the "login" request, does not know that the same request has been attempted X times in the last second and is probably the victim of a brute force attempt. The application can't know that there are three other servers running at 90% of their CPU capacity all trying to process "login" requests. It just knows about this ***single*** request and it evaluates it in that very limited context. A flag in the user account somewhere keeps track of failed login attempts and when it that counter hits X, the account is locked out. Period.

Locking users out after X attempts frustrates the user but does nothing to prevent subsequent attempts. The user will eventually regain access, change their password, and eventually a spammer/miscreant will try again. Nor does preventing access to one account stop the attacker from simply moving to the next one and trying again. This is one of the - albeit few - instances in which a web application firewall (WAF) is capable of providing security that an application simply can't.

A BETTER SOLUTION

A web application firewall has access to what applications don't: the big picture. It has the proper ***context*** in which to recognize and prevent brute force password attacks. A WAF can see the pattern of connections and requests across the entire application and can use historical request patterns to recognize when it is likely a brute force password attack is occurring. Using various mitigation techniques including limiting the maximum number of failed login attempts on a per browser-session and IP address level along with recognizing an abnormally high rate of failed login attempts, a WAF can trigger preventive mechanisms that protect an application against these types of attacks.

Brute force attacks, which can generate up to a million requests per second, can also put considerable strain on the application and its supporting infrastructure. This can lead to a degradation of performance and availability for ***all*** users, not just those under attack. Using a WAF to mitigate attacks and regulate requests relieves the application and its infrastructure of that burden and thus preserves availability and performance for ***all*** users.

Employing an intelligent solution capable of interpreting failed login attempts in a broader context leads to the recognition and prevention of brute force password attacks. An application simply does not have the historical context nor a view of the big picture required to prevent these attacks; it can't, for example, recognize the latency between requests. The latency between login attempts of a real user versus that of a brute force script is very different. The only solution for the application is to lock users out of their accounts quickly or risk compromise. Even if we rethink thresholds for account lockouts and increase the allowed number of attempts the result will almost certainly be the same: the user is locked out. This does nothing to address the strain on the infrastructure, degrading performance of the application, and the frustration users experience when locked out of their accounts.

One thing Twitter can do **now** is to make users aware of ***why*** they were locked out and perhaps provide an additional message tacked onto the "too many failed login attempts" that explains the situation better. An explanation that Twitter is well aware that the user may not be the one responsible and that the account was locked to protect the user from compromise might go a long way toward relieving some of the angst users – especially the less technologically savvy ones – experience when they don't understand ***why*** something is happening.

- Out of Twitter
- Rethinking Thresholds for Account Lockouts
- Twitter's account lockout vs API
- Intelligent Layer 7 DoS and Brute Force Protection for Web Applications [PDF] [TEXT VERSION]
- Taking Down Twitter as easy as D.N.S.
- Layer 4 vs Layer 7 DoS Attack
- Rip and Replace Won't Solve Twitter's (Or Your) Security Problems
- Twittergate Reveals E-Mail is Bigger Security Risk than Twitter
- Web Application Security at the Edge is More Efficient Than In the Application

- Denied!

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com