**EMCS2020: Advanced Topics in Computer Security**

Assignment: Evaluation of Brown's Two-Factor Authentication System

Brian Russel Davis, [brian\_davis@brown.edu](mailto:brian_davis@brown.edu)

***What are the advantages and disadvantages of Duo Push in comparison with a standard OTP app? Are one or more of the three authentication methods vulnerable to man-in-the-middle attacks? ( I am combining these two to demonstrate a point )***

The notable advantage of Duo Push is its ability to narrow the scope of devices that it interacts with. In the Duo ecosystem, a “push” would only send a notification to a trusted device and a trusted device must be defined with the inception of the account by the user. A standard OTP device usually is not paired with a mobile device and relies exclusively on the algorithm in the hardware ( usually a keep fob ). Duo Push is not limited to push notifications, their complete system has many different ways for a user to verify themselves. While this is a clear advantage in terms of usability and availability, it does provide a larger attack surface for malicious actors and vulnerabilities that arise from configuration or implementation mistakes.

Quite notably there are 24 know Common Vulnerabilities and Exposures ( CVEs ) for standard OTP systems and only one for DUO.[[1]](#footnote-0) This doesn’t mean there aren’t more vulnerabilities for DUO. As a matter of fact, another simple search revealed what I would consider a substantial weakness in the DUO system in an article published on a pen-testing blog called “Abusing Duo 2FA”.[[2]](#footnote-1) The article describes how an attacker can easily change the host records for DUO desktop app, make the service unreachable and then leverage a ‘feature’ called “No Fail” in DUO to bypass 2FA. I think the feature is misnamed, it should be called “Turn Off DUO”, because that is basically the result.

Getting back to standard OTP’s if one were to read through the CVE it consists of mostly downgrade attacks or attacks on a poor configuration that expose the secret keys. These seem pretty bad considering how preventable they are. So naturally when comparing this to the DUO system, which can point to a very low CVE record, one would think that DUO is more secure. But as I mentioned above, more features and having availability on more devices provides a bigger surface area. For example, I found it trivially easy to extract the DUO app from my phone ( Android Pixel 2XL ) and decompile it. It took me less than 15 secs, and the decompiled APK not only gave me access to all the DUO Push source code but also all of the endpoints, the Android Manifest, the signatures and the graphics. It wouldn’t take much for me to switch the labeling of the Approve and Deny buttons, recompile the app, and using NFC silently install it on a target’s phone. When they get an elicit DUO Push ( trigger by me ) and they tap what they “think” is the decline button, they would give me access to the target site. On web site, DUO uses Javascript which is virtually impossible to obfuscate, so once again all the endpoints are exposed. An attacker on the network can see the communication between the user and DUO. This communication is encrypted but once again an attacker can inject javascript on the page and present an ilicit option to make the user verify their phone number for example. Once the attacker has the phone number, and clones it, they simply need to run a busy phone attack and use the cloned phone for verification.

In conclusion, having push notifications ( vs. not having them wiith standard OTP ) seems more modern and advanced, but unless those new options are completely secure they just provide more opportunities for attackers. SMS, Phone calls and any type of web applications are vulnerable to Person-In-The-Middle attacks. However Person-In-The-Middle attacks don’t always appear in a templated fashion. Attackers are not trying to fill in a Person-In-The-Middle template, they are looking for weakness. If the weakness is a traditional Person-In-The-Middle, then I am sure attackers will exploit these lines of communication, however much of the weakness I see is more in line with code injection as I demonstrated above.

**What is the impact of selecting "Remember me for 30 days" on the security of the system?**

“Remember me for 30 Days” is an extremely naive feature. It basically makes DUO susceptible to the exact same attack vector as Automatic Auto Fill. It’s not exactly clear how DUO does this remembering but it seems like it may use the information stored in a cookie? I am 90% certain this the case because when II clear my cache the 30 day remember feature is forgotten. The attack vectors are pretty obvious from here, steal the cookie and get access on another machine. Furthermore it seems the Brown implementation of Canvas is a missing a major feature : Log Out. There is no way to Log Out of the canvas dashboard as there is no Log Out button. DUO does also does not let the user see where they have logged in, so there is no way of knowing if the system is activelt being abused. Android SMS Messages for Web is an example of an app that lets the user see a list of current logins and invalidate. Allowing a user to set themselves up for automatic PWNing ( remember for 30 days, no log out, no list of active sessions ) seems liek a huge recipe for disaster.

**Would you change any aspects of the system (e.g., functionality, authentication methods, user interface, documentation) to increase its security and usability?**

In my opinion the DUO system has a beautiful interface and is relatively user friendly but does so at the sacrifice of solid state security. All of the fancy ways of using the system ( apps, push notifications, phone calls, sms messages with passcodes ) offer a large attack surface. In the African-American community we often crystalize this type of behavior as “high-posting”. Quoting one of my favorite artist Mos Def[[3]](#footnote-2) :

*“Now I like to have nice things just like you*

*But I'm from Brooklyn, certain shit you just don't do*

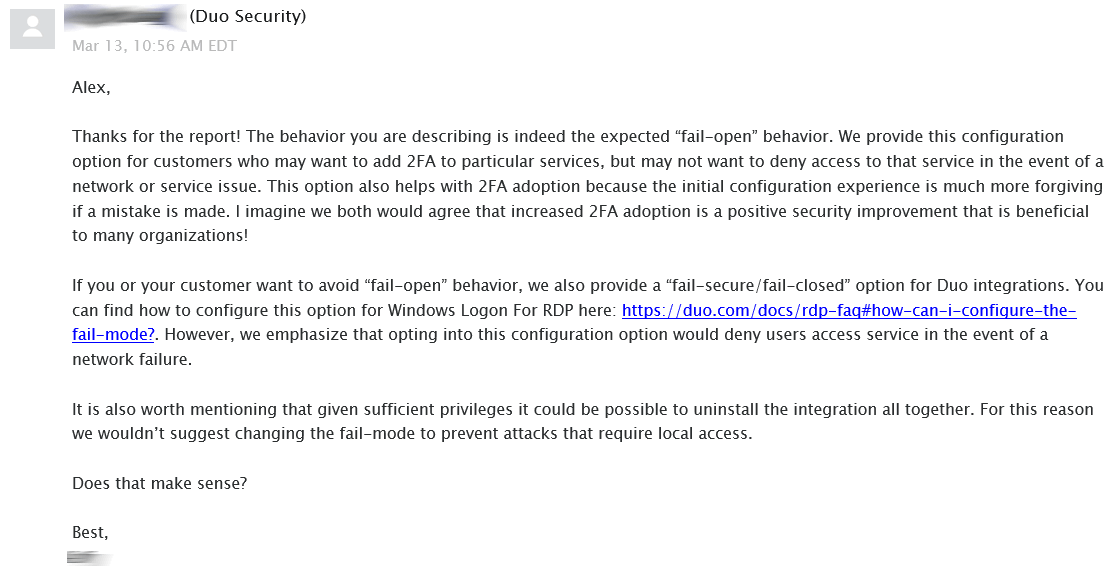
*Like, high postin' when you far from home*

*Or like, high postin' when you all alone*

*Now, this would seem to be clear common sense*

*But, cats be livin' off, sheer confidence”*

The lesson here is advertising your wealth is very naive when you are surronded by hungry killers. Much in the same way that DUO app advertises how easy and widely accessible it is. “*I like to have nice things just like you … But, cats be livin' off, sheer confidence*” ; It’s nice to have all the options, but at what risk? If it was up to me I would remove all the options until I could be certain that they could propely secured. DUO doesn’t seem to share this sentiment. Their response to Pen Test Partners illustrates this :



The writer is trying to get a Pen Tester to be happy with the fact that the company/client even uses 2FA? Seems like a pretty low bar, in my opinion. In that sprit here is a short list of changes I would make:

1. If Remember Me for 30 Days is a required feature, then it needs to tracked in more than just a cookie ( like the cookie and IP must match ) and the user should be prompted with a window of active session before they continue. This would stop the automatic autofiill related attacks because it requires
2. DUO should only be used on sites that have a clear LOG OFF function. With no log out function the site is removing a basic security feature from the hands of users.
3. Obfucate the source code. There is no reason make decompling the app so easy.
4. Serve the approve and decline buttons from the server using encrypted communication and use random positions to make sure user is paying attention.

1. Search Results. (n.d.). Retrieved October 27, 2019, from https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=otp. [↑](#footnote-ref-0)
2. Lomas, A. (2018, March 16). Abusing Duo 2FA. Retrieved October 27, 2019, from https://www.pentestpartners.com/security-blog/abusing-duo-2fa/. [↑](#footnote-ref-1)
3. Def, M. (1999, October 12). GOT. Retrieved October 27, 2019, from https://genius.com/17762388. [↑](#footnote-ref-2)