# Understanding and Using Microsoft 365 App Passwords

## Summary: Application (or App) Passwords Provide a Way for Applications to Authenticate in Microsoft 365 when MFA Policies are Enforced

In an environment of increasing security restrictions, I have noticed in my role as a Microsoft 365 administrator a misunderstanding among users regarding application (or app) passwords. The core idea of how app passwords came about was the need for Microsoft 365 application designers to provide an authentication mode for a non-interactive login in a Microsoft 365 environment with multi-factor authentication (MFA) policies enforced. There is a bit more to it than that, but an app password provides a way for an application connecting to Microsoft 365 to authenticate with a username and a single password without using a second factor. That begs the question, if the application can authenticate through just a username and password, and multi-factor authentication is enforced in the domain, then how does that satisfy the requirement for multi-factor authentication? This question is perhaps why so many users misunderstand app passwords or have perhaps never thought to create an app password in the first place. We will try to clear up some of the confusion in our post today.

With the recent increase in malware and hacking activity, most Microsoft 365 users are by now familiar with using a second factor authentication method to sign into their Office accounts and systems linked through Azure Active Directory integration. Users are now conversant with using applications on their smart phones such as the Microsoft Authenticator, or in receiving numerical codes sent via an SMS (text) message to authenticate within a Microsoft 365 environment. Using the Microsoft Authenticator, and SMS code, or perhaps a smart key USB device is great when provisions in the particular application connecting to Microsoft 365 provide an interactive login experience. However, what happens when an application needs to log in without the user interacting? In this case, how can the application access an MFA challenge when a human user isn't involved? This is where the app password comes in. A user can first create an app password for the particular instance when a non-interactive login is required, and then enter this unique password (along with the regular Microsoft 365 username).

## Scenario: Web Site Won't Send Transaction Emails

Consider the following scenario as perhaps the most common situation in which the need for an app password arises. Bob, a marketing department employee, is responsible for the company's web site and needs to send emails from within his content management system through the use of SMTP. Up until his company enforced MFA for every user in the company, he simply entered the username and password for his webmaster account in the SMTP settings in the web site. After his company enforced MFA, he noticed that the web site was no longer sending transactional emails from elements within the site such as contact forms and plugin update advisories. It is not a coincidence that this issue came about the same time that the company implemented MFA polices for ever user, including system accounts. The solution is this case would be for Bob to create an app password in Microsoft 365.

Before we move on, it is actually a bit more complicated than the scenario we outlined, because Bob would have had to validate his webmaster account with his desired MFA method when the company rolled out the new policies. For the sake of illustration let's consider that another employee already took care of this task. Second, sending mail using the Microsoft 365 SMTP service isn't what we consider a best practice. Instead, transactional emails should be sent using only transactional email systems such as Amazon SES, Mailjet, etc.

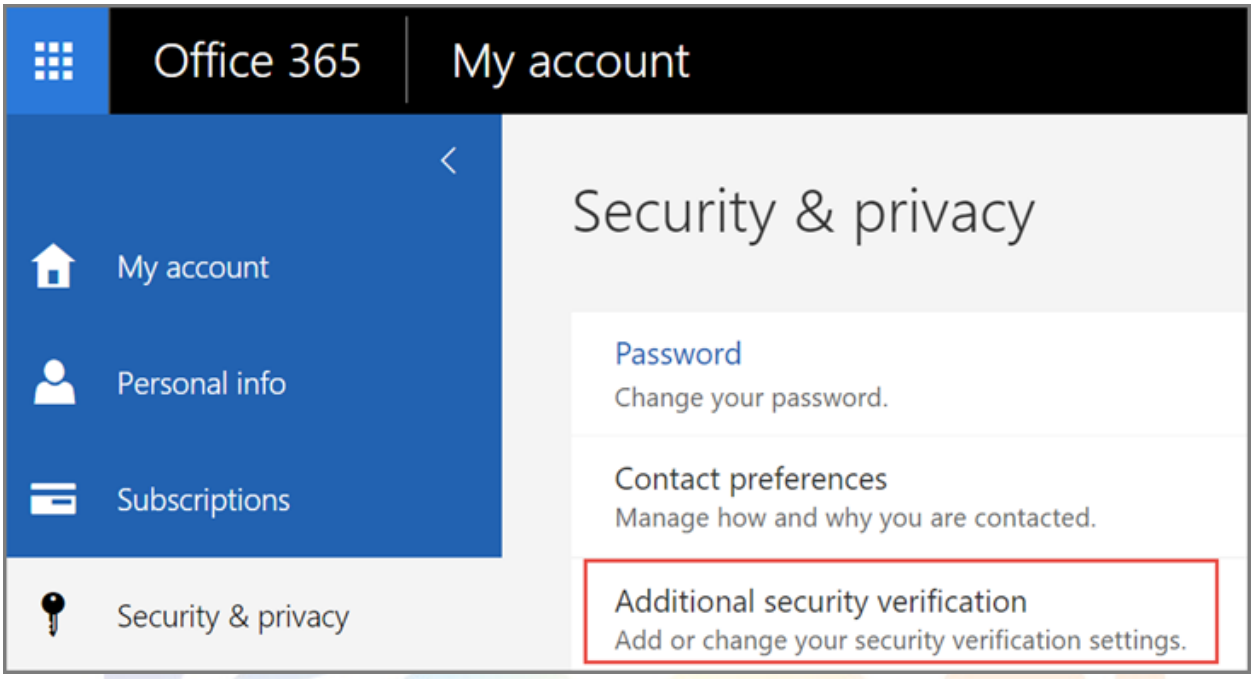
This is because of several reasons including the afore mentioned security issues when MFA is enforced, and Microsoft 365 enforces transactional limits on the sending of email. A case could be made for exceptions where the use of a Microsoft 365 account to send mail using SMTP would be acceptable, such as in very small companies with low email volumes.

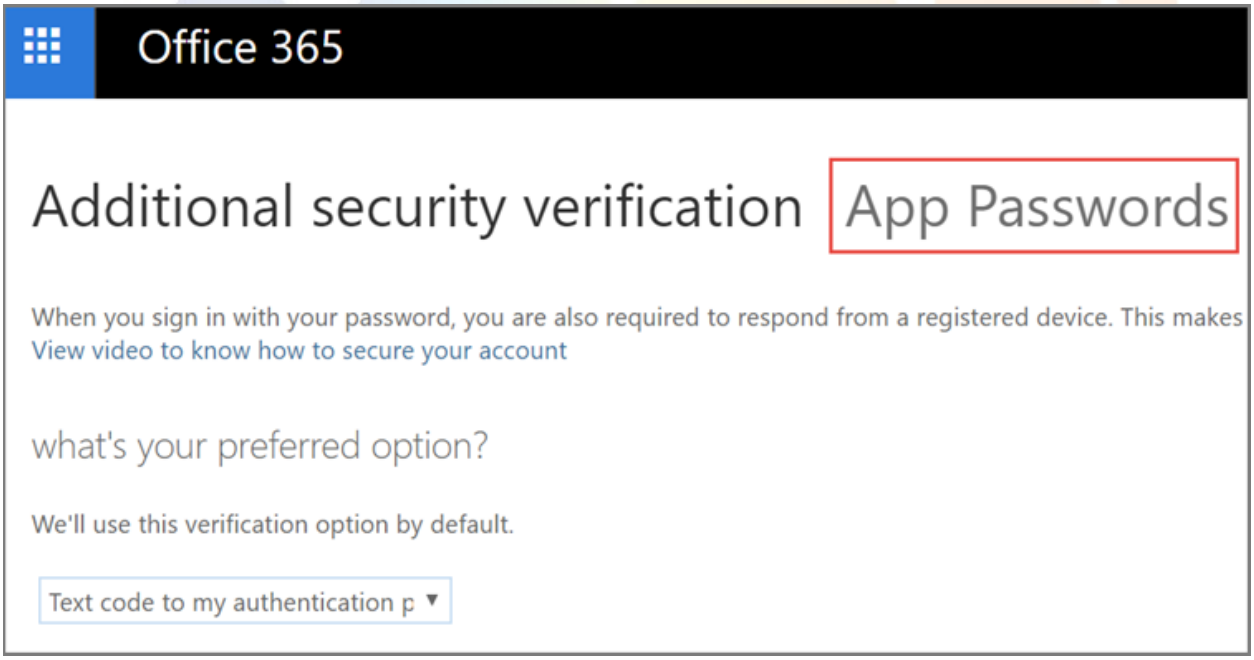
Back to Bob's scenario. Since the web site was not able to send email, he logged into the web site admin console and double checked the original credentials. However, the web site would still not send email. In this case all Bob would need to do would be to log into the Office.com account for his webmaster user and create an app password. Then, he could enter that app password into the web site SMTP settings to send mail. The app password would then in effect satisfy the need for a *second factor* of authentication, albeit in kind of in a **pseudo-MFA** way! Even though he is satisfying the application's requirement for authentication in an MFA environment, this is not a perfect solution as his password can still be breached!

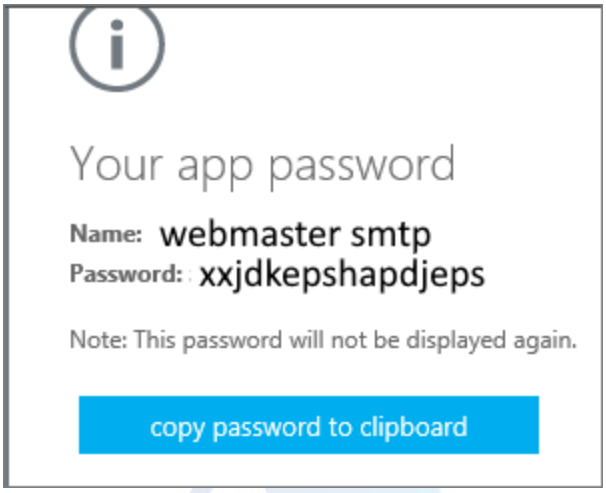
## How to Create an App Password in Microsoft 365:

The steps to create and app password in Microsoft 365 are very simple, just be certain that you remember to record the password because it will not be shown to you again. Should you lose the password you can just create a new app password.

1. Log into your Microsoft 365 **Office.com** account with the user for whom you need to create the app password.
2. Choose **Settings** > **Office 365**.
3. Choose **Security & Privacy** > **Additional security verification** (Figure 1).
4. At the top of the page, choose **App Passwords** (Figure 2).
5. Choose "**create**" to get an app password.
6. If prompted, type a name for your app password, and click **Next** (Figure 3).
7. Choose "**copy password to clipboard**." Save this new app password somewhere in your records or password manager tool. If your password manager tool provides the method to share with your colleague this would useful if the account for which you are creating the app password is a system account, such as webmaster@companyname.com.


Office 365
My account
My account
Personal info
Subscriptions
Security & privacy
Security & privacy
Password
Change your password.
Contact preferences
Manage how and why you are contacted.
Additional security verification
Add or change your security verification settings.


Office 365
Additional security verification
App Passwords
When you sign in with your password, you are also required to respond from a registered device. This makes
View video to know how to secure your account
what's your preferred option?
We'll use this verification option by default.
Text code to my authentication p




## Some Common Scenarios When App Passwords May Be Required:

SMTP connections on web sites are just one scenario where an app password may be required, however there are many more cases where they are necessary. Such cases include CRM systems which send email or connect to user's Microsoft 365 Inboxes through MAPI (please don't use IMAP connections to Microsoft 365, they are a security breach waiting to happen), transactional systems such as e-commerce systems, Internet of Things (IOT) systems including environmental controls and thermostats, security cameras and alarms, etc. Remember, once MFA is enforced on the Microsoft 365 account and that account needs to authenticate in a non-interactive manner, an app password will be required.

## Closing Thoughts:

App passwords provide a means for user accounts to authenticate in cases where MFA is enforced within applications where a non-interactive login is required. Such a case could include things like a smart thermostat which is programmed to send alerts through email using a SMTP. Another is a web site which sends submissions from contact forms through email. Although an app password satisfies the technical requirements for the two-factor authentication, it is still technically just another password that could be hacked. For this reason, as a best practice we recommend that all Microsoft 365 users utilize transactional systems for sending email. We recognize that the use of an app password may be necessary in certain situations, such as when an Office 365 mailbox is required to sync via MAPI to a CRM system. In such cases an app password provides a work around to continue the operation.