Enterprise Email: A Strategy for Securing the Phishing Deluge

As mentioned in the social engineering article [reference-1], phishing is one of the email social engineering delivery mechanisms that can use any number of social engineering manipulation techniques [reference-2]. Email is a deluge of phishing campaigns but with a decent protection strategy the majority of malicious content can be negated. SPF, DKIM, and DMARC are foundational to verifying sender legitimacy. Ensuring mailboxes are properly accessed by authorized entities whom are within an authorized geofence and whom are not using VPNs, anonymizers, or proxies is critical. Ensuring email cannot experience tampering in transit using encrypted protocols is basic while ensuring service providers cannot access corporate email at rest can be a bit trickier. Creating filtering rules that are specific to organizational needs and threats further reduces common phishing risks. Creating file type blocking and sand-boxing suspicious attachments and URLs is a surefire way to catch malware that has slipped by other defenses. With the last line of defense being user training and the ability to report suspicious and/or suspected phishing emails.

I remember years ago when I used telnet to send friends emails that commended them for being a stellar human and an asset to their nations and entered a return email address of santa@north.pole. Sadly, this spoofing capability still exists widely and is used for much more nefarious purposes in phishing, spear-phishing, and whaling campaigns. Spoofing (in the context of email) is when an attacker is able to send an email using another person's email address without authenticating to that person's account. This is so reliable because many organizations do not have technical countermeasures configured that will verify if the sender is an authorized sender. This is where Sender Policy Framework (SPF) comes into play. SPF is a technically configured security mechanism that was designed to ensure a sender header contains the IP address of an authorized sender [reference-3]. Email headers can be manipulated meaning attackers were able to circumvent the effectiveness of SPF so DKIM [reference-4] was born. Domain Key Identified Mail (DKIM) signs email messages with a cryptographic key with a corresponding key stored as a DNS record. The DKIM signature is used to verify the email did legitimately come from the alleged sender and was not modified. So, SPF provides info stating the email came from an authorized sender and DKIM signs the message to prove the message did not experience tampering. Knowing if an email is legitimate is good but knowing what to do with that information is better. This is where Domain-based Message Authentication, Reporting & Conformance (DMARC) comes into play. DMARC [reference-5] is another DNS entry but is created by the sending organization to tell all receiving organizations what they should do with illegitimate messages. An example, Notional Inc. will create a DNS record that will tell Fake Inc. what to do with emails that claim to be from Notional Inc. The choices Notional Inc. can configure are "do nothing" where by Fake Inc. will accept all messages without regard to failure, "quarantine" messages with failed checks, or "block" messages with failed checks. The combined use of SPF, DKIM, and DMARC are critical to ensuring only legitimate email is sent on behalf of your organization but can be a little tricky to implement when third-parties are permitted to send email on behalf of your organization.

Without configuring these technologies receiving organizations may, (and have been known to,) block email from organization based on the actions of spoofers that send high-volumes of spam or which send malware. SPF, DKIM, and DMARC are the main technical legitimacy checking tools but legitimate access verification needs to be ensured for humans as well.

If an attacker can access a corporate emailbox very few technical counter measures will be able to stop the spread of phishing, spam, and malware-laced email messages. The first way to verify the legitimacy of humans is by verifying authorization to emailboxes based on normal intended use. The best way to verify a human is permitted to access an emailbox is through the use of multi-factor authentication. It is the opinion of the author that certificate embedded hardware tokens (such as CAC or PIV) devices are the best choice followed by (in order of preference) mobile auth OTP apps, mobile mail client push notifications, then text messages. Biometrics can be a good solution as well if implemented properly. Even when two factor is used attackers have been known to use clever tactics to circumvent the protections so additional measures should be taken. Users whom log into email systems should conform to geofencing restrictions in conjunction with not being able to use a non-enterprise VPN, proxy, or anonymizer. If an attacker is geofenced but can use a VPN, Anonymizer, or Proxy the attacker could "hop" into the geofence before attempting authentication. Once the login portion of email has been reasonably secured the email network protocols need to be secured to ensure there is no eaves dropping or manipulation of data in transit. This can be achieved by using secure protocols such as S/MIME, SMTPS, POPS, and IMAPS. At this point, we know the user is authorized to access the mailbox and we know the email in transit is encrypted. Now we need to ensure the email is safe in the cloud-service provider that hosts the services. This can be achieved with the use of a zero knowledge architecture or zero access architecture or zero trust architecture [reference-6]. As very few email providers currently offer this because they were designed to data mine every user whom use their service it is unlikely most organizations have acces to this protection. There are a couple of providers that offer this by default such as ProtonMail [reference-7] but the majority are only now working on trying to achieve some form of implementation. Once this capability is more widely adopted it is likely to come at a high asking price. So, check to see if your provider offers a service that will encrypt all corporate email in a way that prevents message content read access by cloud service provider personnel. Once DMARC and human verification have been successfully implemented (or in a parallel project) it will be time to create some basic filtering rules.

Most email providers have a great intel and filtering capabilities that are offered by default so this section will be limited. The first to email blocking signatures are ones that have been widely seen across many industries and which have been highly effective in tricking users into providing financial information or the purchase of store gift cards. The most common being an email asking if someone is busy then asking the employee to purchase a store gift card because the attacker is stuck in a meeting and forgot to buy one earlier. The signatures are "*domain.com@publicemail.com" and "*domain@publicemail.com". When used by an attacker, the asterisk is usually replaced by a name or a username and the "domain.com" and "domain" are replaced by the victim organization's domain. The next set of blocking rules are to assist with email legitimacy when parent organizations have not

created DMARC entries in DNS. The rules that are created should block all email that have an SPF failure message in the header and a rule that blocks all messages that have a DKIM failure in the header. These rules should be turned on in "audit" mode as all organizations will have partners that do not have SPF and DKIM properly configured in their DNS servers. Once the partners have created the proper DNS entries (or by the drop-dead date) these signatures should be transitioned into enforcement mode and partners should be forced to create SPF and DKIM DNS entries. The next set of signatures is enterprise dependent and may not offer a great deal of benefit to some organizations. These rules constitute the blocking of emails that appear to come from specific departments, roles, or personnel within the organization. For instance, a rule might be created to block all emails that originate from outside the organization but which have a sender email that contains "HR@", "CEO@", "[CEO's name]@" and the like. This can help users from accidentally thinking an email cam from HR, the CEO's office, or the CEO (despite the fact it came from an external email provider). The last filtering rules that are recommended are for any email addresses that include Unicode or any messages contents that include hyperlinks to URLs that include Emoticons. There has been a number of very successful phishing campaigns that have leveraged Unicode/emoticons to trick users into clicking links. This is a very basic and minimal list of filtering rules that can be added to an existing program. There are many more possible rules to cover than is practical for this article so we shall now turn our attention to file attachment blocking and sand-boxing.

The default configuration for nearly all email service providers is to use a file type block list to prevent the email receipt of files that are not commonly sent legitimately through email. One example is .reg (registry) files. Although a registry file could be emailed to permit a user to activate Microsoft's software write blocker, it is extremely uncommon. Equally uncommon is the use of a block all attachments by default and permit file-types by exception policy despite this technique being widely pushed in all other security domains. A deny all and permit by exception is the safest way to permit file-types to be emailed. For any files that are received through email, a security check should be performed with any potentially risky emails being sent to a sandbox for testing [reference-8]. To ensure a sandbox has maximum detection capabilities it should be tested with a program such as ParanoidPhish to determine how easily malware can detect the sandbox environment [references-9 & 10]. Only when the sandbox is undetectable by malware will it be most able to detect the greatest amount of malware. One caveat to using a sandbox for malware though, some malicious documents have not started requiring user interaction to trigger the malicious activity. For instance, a spreadsheet might ask the user to enter a password before the malware will spring or a website may ask a user to complete a CAPTCHA. These interactions cannot be (or are very difficult to) automated by sandboxes so analyst interaction in the sandbox will occasionally be required.

Any email which is able to get past the legitimate sender checks, the filtering rules, and the sand-boxing will most certainly have a very high probability of being a legitimate email from a legitimate sender but could still be malicious. This is why all organizations need an easy way for users to report suspected phishing and malicious emails. Many email providers and third-party email security solutions simplify this reporting capability by offering a “clickable” menu button, use this menu button

to make it easy for users to report suspect emails and review reported emails. When this capability is first deployed there will be a large number of spam messages reported as potential phishing which can lead to poor security results on the back-end. Continued end-user education on the difference between phishing and spam will help improve security, decrease alert fatigue, and improve moral (people like being able to brag about what they know and/or how they have helped the organization). Once reports are received the emails should be sand-boxed, tested, and analyzed. Any delivered emails that are found to be malicious should be remotely wiped/rescinded from users' mailboxes to help protect them from accidental interaction or infection.

Email contains a deluge of phishing emails but with a decent protection strategy the majority of malicious content can be blocked from entering the network. SPF, DKIM, and DMARC are foundational to verifying sender legitimacy. Ensuring mailboxes are properly accessed by authorized humans whom are within an authorized geofence and whom are not using VPNs, anonymizers, or proxies is critical. Ensuring email cannot experience tampering in transit using encrypted protocols is key while ensuring service providers cannot access corporate email at rest can be a bit trickier. Creating filtering rules that are specific to organizational needs and threats further reduces common phishing risks. Creating file type blocking and sand-boxing suspicious attachments and URLs is a surefire way to catch malware that has slipped by other defenses. With the last line of defense being user training and the ability to report suspicious and/or suspected phishing emails.

Reference Links:

1. HTTPS://www.linkedin.com/pulse/social-engineering-9-delivery-mechanisms-andrew-kosakowski

2. HTTPS://www.linkedin.com/pulse/social-engineering-6-ways-manipulate-humans-andrew-kosakowski

3. HTTP://www.dmarcanalyzer.com/SPF/

4. HTTP://www.dmarcanalyzer.com/DKIM/

5. https://blog.mxtoolbox.com/2017/03/03/what-is-dmarc/comment-page-1/

6. HTTPS://csrc.nist.gov/publications/details/sp/800-207/draft

7. HTTP://protonmail.com/security-details

8. HTTP://www.itproportal.com/features/kick-suspicious-email-attachments-to-the-sandbox/

9. HTTP://www.trishtech.com/2016/09/paranoid-fish-tests-sandbox-protection-programs/

10. HTTP://resources.infosecinstitute.com/pafish-paranoid-fish/

Thank you for reading

-Andrew Kosakowski

www.Anthko.com