The Future of Authentication

State of Industry and Solutions

Swoop In Technologies LLC

April, 2024

James Kassemi, Brandon Trebitowski
Swoop In Technologies LLC
5501 Eagle Rock Ave NE Suite E3
Albuquerque, NM 87113

Media Contact:
Patrick Killoran
Partner at Swoop
patrick@swoopnow.com

Investment Contact:
Charlie Chamberlain
President, TigerIP Ventures
cc@tigeripventures.com

Despite considerable advancements in identity, authorization, and authentication systems over the past decades, login processes are still confusing and insecure. This is complicated further by the proliferation of third party tools and myriad protocols, which further confuse users and cause further security issues (often more severe than just having a single account hacked). Email - the dominant form of online communication - can solve these problems.

Swoop is an email-based authentication system. Users exchange email with Swoop to register and or access their online accounts. This paper examines two foundational components of Swoop's advanced email authentication verification systems, focusing on the use of the Sender Policy Framework (SPF), and Domain Keys Identified Mail (DKIM). We'll evaluate the state of email security and web authentication, discuss several potential security threats to both other methods and email-based authentication, and describe mitigations the Swoop email authentication verification systems apply to the problems.

Authentication

This passage aims to clarify industry-specific terms: authentication, authorization, and identity. Consider identity a description that uniquely defines an individual, such as their name, or in the case of most digital services, an email address or username. Authentication refers to the process of a user presenting that identity to the service, and authorization is the process that decides what access or privileges they have.

As an example, when a user accesses their online banking account, they authenticate themselves by providing their login credentials. The bank's system then validates the user's

identity and determines the actions the user is authorized to perform, such as checking their account balance, transferring funds, or paying bills.

These concepts are essential in the development of software that handles sensitive user data. A robust authentication and authorization mechanism is crucial to ensure that only authorized users can access the system's resources. Failure to implement a secure authentication and authorization system can lead to unauthorized access, data breaches, and other security risks.

Passwords

The earliest and most common method of authentication on the internet is the password. A user who provides an appropriate password is assumed by the system to be authenticated. While we mentioned above that authentication provides an identity to the service, a password authentication doesn't necessarily need to provide enough of it to uniquely identify the user. In most cases, however, a password is provided alongside some sort of identification property, such as a username.

A password is compromised when a user who is not permitted gains access to that password. The compromise may involve either a leak of the password from the user themself, from the users' system, or from the authentication system responsible for authenticating with that password. Short, repeated, or easy-to-guess passwords resulted in significant loss, prompting industry to push requirements such as regular password resets, more complex passwords, security questions, and image selections (sequences of images have even been used ("PassImages : An alternative method of user authentication")). In each case, additional complexity was added to the user's process, resulting in increased call center support costs to aid with account resets and password changes.

Password managers became more common, allowing users to generate and store unique passwords for sites. (References to common solutions 1password, lastpass). Users could generate unique username/password combinations that satisfied more complex password requirements, and store any additional authentication notes required. These systems are still popular, but become a single point of attack. Exploits and data losses by these systems continue to cause major issues for users. (Include examples of these losses here).

Public Key Infrastructure

Executing financial transactions online motivated the introduction of public key infrastructure to web communications technology. The early internet, powered by the HTTP protocol, exchanged plain-text communications between nodes. A request to a web server for a resource was made through an unencrypted channel, and it was possible to record or extract sensitive information from that channel as an observer. This was particularly problematic on shared wifi networks, where passwords were often exchanged and easily compromised.

With the introduction of public key / private key encryption with SSL and TLS, web servers could provide users a means of communicating over an encrypted channel. In SSL, a key pair is exchanged and validated with a user. This key pair is used to generate a shared encryption key that can then be used for encryption of traffic between the two nodes. All secure systems now either encrypt the communications channel between two users directly, or the password that a user authenticates with.

Multiple-Factor Authentication

Security researchers proposed the development of a new process to address issues related to password loss, which requires users to submit multiple forms of authentication to successfully authenticate to a system. Some of the most secure uses of this include a “factor” that the user “knows” - such as a password, and a factor that the user “has” - such as a device.

While the probability that a user loses both is still relatively high, the approach effectively mitigates the compromise of a large number of passwords from an authentication system. When paired with an effective rate limiting system, the cost for an attacker at scale becomes unwieldy.

Single Sign-On

With the proliferation of web services within organizations, where a single user may log in to many systems throughout the day (email, ticket tracking, human resources, etc), Single Sign-On (SSO) allows sharing a single authentication instead of continuous authentication prompting. Consumers use SSO solutions (often built on OAuth standards) provided by companies like Google, Facebook, and Microsoft to share their authentication with third-party services. By preventing a user from moving through a complicated authentication process many times, more complex and effective security mechanisms can be implemented during that first authentication workflow. With consistency, users can also rely less on password managers to store their single complicated password.

Email

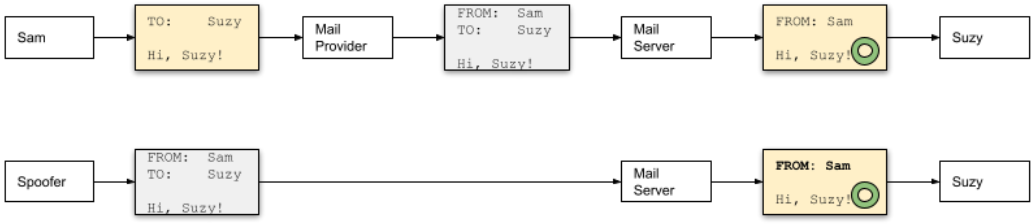
Given the ubiquity and importance of email, major organizations spend considerable time and resources securing these systems. They have become the cornerstone of private OAuth SSO services, and it's widely considered best practice to use email as an identifier for users on all online registration systems. Swoop's innovation's rely on a system that all users are already expected to have and know how to use. It effectively extends the boundaries of SSO to use access to email as a means of authentication. Users can authenticate with any service, without a password, through their existing email service. The benefits of SSO are applicable here, but users maintain control of their accounts, instead of being forced to register a new account with a provider. Because users are generally logged in to their email account, they do not need to remember or store another password. And because no password databases are maintained with the Swoop service, there's no risk of a mass compromise through any single organization.

Email Security

Historical Design

Email was originally used by researchers and academics who trusted each other and the systems they used to exchange messages. As it gained popularity, the need for authentication and identification standards became clear. Problems with spoofing - or sending a message as another person - continued after developing the first authentication systems, which acted solely at the account level of the email provider. The protocol itself lacked any sort of encryption or authenticity guarantees.

The following diagram represents an exchange over one of these old email servers. A message from a spoofer (impersonating Sam) could be crafted with a FROM header that indicated the message was from Sam. The email server would gladly process and deliver the message. IP addresses could still be recorded, and for small networks it was possible to ensure that messages came from relatively trusted sources.

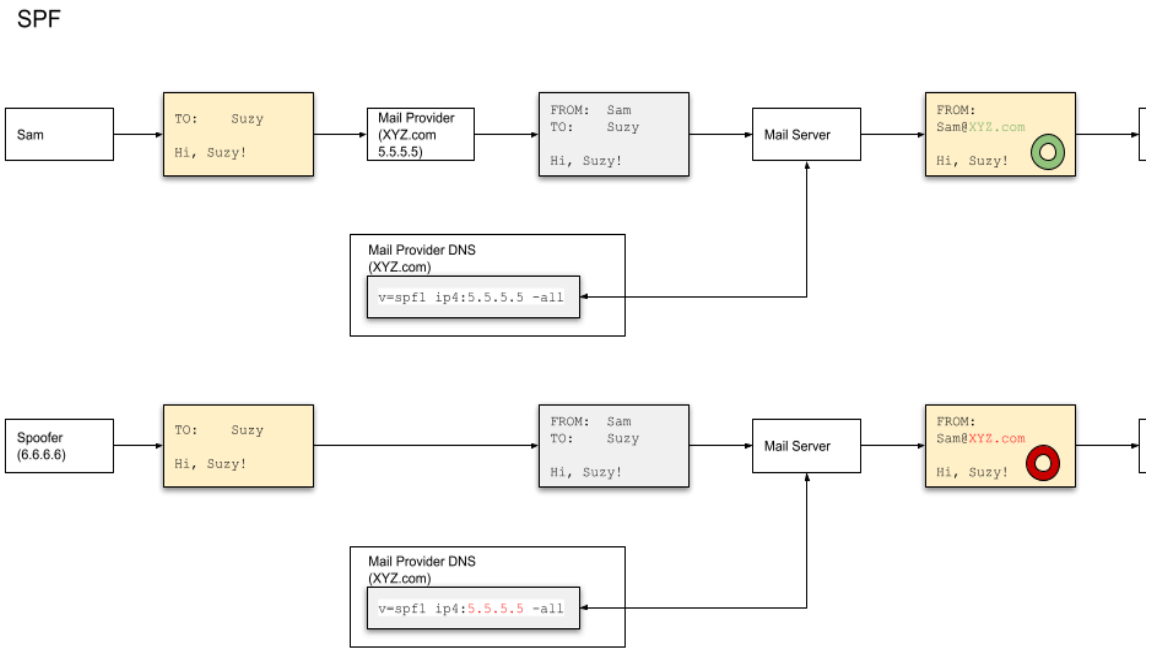


SPF

Building on top of the recorded IP records, [Sender Policy Framework (SPF)] enabled mail providers to define a range of IP addresses and domain names that the email provider managed mail for. To communicate this range, a DNS TXT record was added to the sending domain. With SPF, the recipient mail server examines the domain of the sender in an email

message, and then queries that domain's DNS for the SPF record. If the address and the domain were associated with that server, the message could be considered authentic.

The following diagram represents the use of SPF by a mail provider and receiving mail server to validate a message from Sam. Since the spoofer does not control the DNS for the sending domain (see the "DNS Compromise" review in the "Threats and Mitigations" section), they cannot send a message to the receiving mail server from a server they control. With vigilant operators and strong controls, the Mail Provider can confidently authenticate messages it's delivered.



The idea of verifying SMTP MAIL FROM addresses from DNS records was first mentioned December 14, 1997 in an email message by Jim Miller to now VP of Security at Amazon Web

Services. Most details regarding the format and specific record type used were mostly resolved through 2003 with the Designated Mailers Protocol specification, version 3.

Determining the rate of adoption since its first introduction is difficult, due to the distributed nature of the system and the continued lack of dedicated research focus. We know that the protocol has near universal adoption by the majority of US-based email service providers, but tracking usage statistics is still difficult. We can analyze domains that provide SPF headers, but many domains are not responsible for or configured to send email.

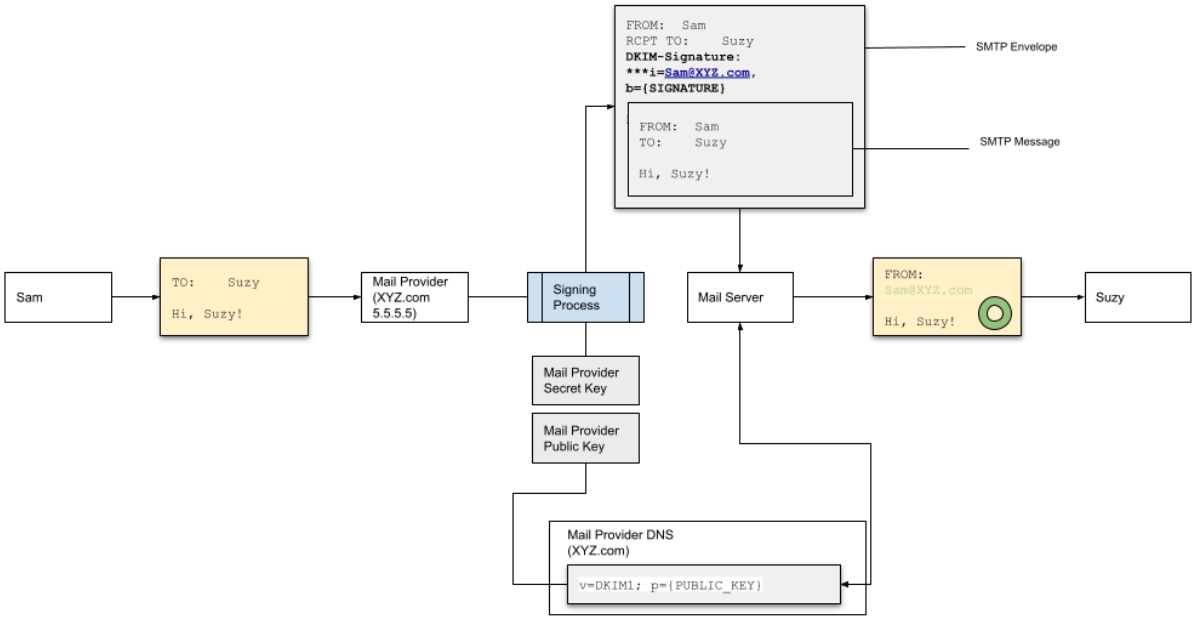
If an email service provider is not providing headers, Swoop's systems will detect and prompt the user for further action.

DKIM

Domain Keys Identified Mail (DKIM) is a signature-based email authentication protocol introduced by an industry consortium on July 11, 2005, which was based on Yahoo's DomainKey's email authentication technology and Cisco's Identified Internet Mail. It extended the concept of authenticating a message to the user of an associated email system.

DKIM builds on top of the idea of using DNS to manage authentication policy. A public key is added to a DNS TXT record, and outgoing messages from the sender's mail server are signed with the corresponding private key. The signature is provided to the receiver's mail server, which can validate the signature against the publicly available and independently retrieved public key. As demonstrated in the following diagram, since the entire message is signed, the receiver's mail server can consider that message authenticate.

DKIM



Measuring adoption of DKIM is just as difficult as SPF, for the same reasons. While DKIM configuration can be considered complex, it's generally more difficult to maintain a valid SPF record than a DKIM configuration. A DKIM public/private keypair can be used until they expire, and configuration does not require potentially complicated changes to server inventory.

We see nearly universal adoption of DKIM for 1-Click user's email providers, and send appropriate instructions to users when we determine their email is being delivered without it.

DMARC

Sender Policy Framework and Domain Keys Identified Mail are brought together under a third major standard, called Domain-based Message Authentication, Reporting, and

Conformance (DMARC). DMARC allows a receiver's email server to verify that any particular DKIM and SPF configuration of a sender's email system is correct.

Encryption

Email, based on SMTP, is generally a plain-text message exchange system, and the contents of messages may be exposed to entities other than the intended recipients. STARTTLS is an encrypted layer over email that provides encryption for all server communications. STARTTLS is a command sent to a mail client from a mail server after a connection is initialized. The client and server begin a key exchange, and ultimately negotiate a secret symmetric key which is used to encrypt the message in flight to the server.

Protocols such as GPG allow users to encrypt and sign messages prior to interacting with mail servers. A keypair can be generated and public keys for two parties exchanged through a non-email system (often in person). After which, messages signed with or encrypted by one user can be verified and or decrypted by the recipient. Some mail clients include GPG capabilities, but these are generally harder to configure and use, and haven't witnessed widespread adoption.

Threats and Mitigations

Email Spoofing

Email spoofing is eliminated by SPF, DKIM, and DMARC processing and validation, when configured correctly. Swoop requires valid and properly functioning SPF and DKIM

configurations for inbound authentication email, and prompts the user with instructions when problems are detected.

Phishing

Email-based phishing involves the delivery of an email to an individual purporting to be from a legitimate source. Spoofing can be one factor in a phishing attack, but hyperlink masking and homographs (using a word or letter that's very similar to another - bob@microsoft.com vs bob@microsolt.com) are also widely employed.

A phishing email claiming to be from a 1-Click customer's website may include an email to a malicious website, prompting the user to provide sensitive information. This attack is mitigated by the user training, reporting features, and anti-phishing software solutions common to industry. But because 1-Click users are not prompted for a password during a standard authentication flow with the legitimate website, they expect an email-based transaction, and should be less likely to interact with an attacker's website. 1-Click and other SSO systems improve the user's security by minimizing the exchange of passwords with internet systems.

Reply-All Authentication Spoofing

An attacker may attempt to send an email to 1-Click users that includes 1-Click's email authentication address in addition to the 1-Click user's address in a FROM or CC header. For example, assume Bob is attempting to gain access to Suzy's account on a 1-Click enabled application:

TO: suzy@XYZ.com

FROM: bob@XYZ.net

CC: sam@XYZ.com, OneClickSetup-324@swoopapp.co.uk

Hey Suzy!

Hit reply-all and let Sam and I know what you think about lobster for dinner tonight.

Bob

If Suzy were to respond to this message with her email client's "Reply All" function, a message would be sent to the unique OneClickSetup-X address at 1-Click's servers, and could be considered a successful authentication.

1-Click systems mitigate this technique by disallowing messages it determines are part of forwards or addressed originally to multiple recipients.

DNS Compromise

The Domain Name System (DNS) is a registry system originally created in November of 1983 (RFC882) that provides a mapping of human readable names to IP addresses. In addition to allowing users to determine how to access a domain, it can be used to store additional authoritative information about those domains and systems hosted on them. Configuring DNS is an important component to launching any new website or service.

There are several types of DNS servers that perform similar functions at several different levels and granularities. DNS servers are configured on a domain through the use of an "NS" DNS record. A domain lookup begins a recursive query of NS records that ultimately

results in an IP address that can be used to interact with the domain. SPF and DKIM configurations are stored in a TXT DNS record, which can be managed by the domain's authoritative DNS server administrator.

Cache poisoning is the introduction of false data into a DNS resolver's cache, which causes the resolver to return incorrect IP addresses for a domain name. A DNs resolver will save responses to queries for a certain amount of time so that it can respond more quickly. For example, a query to an ISP's DNS server for Google could store the response for a google.com nameserver for a period of time, allowing all the ISP's DNS users to more quickly resolve the domain and access google's services. If an attacker is able to trick a caching server into storing a record for google.com that points to the attacker's servers, then all traffic to google.com would be directed to the attacker's systems. An attacker may be able to overwrite or change DKIM or SPF DNS configuration at this layer. Executing a cache poisoning successfully is a difficult, considerable effort. In most cases, the attacker will need access to another compromised system in the network or the caching resolver itself.

If an attacker can control a mail provider's SPF and DKIM records through any means - either cache poisoning or mail provider DNS server compromise, they may be able to send a valid email to a receiving mail provider. DNSSEC, which verifies DNS records and their origins, addresses this issue, but is not currently well adopted.

In all cases, except cache poisoning - where the impact may be mitigated depending on the number of DNS server users - We feel that the scope of this vulnerability is identical to that of a full email provider compromise. 1-Click relies on internal detection systems and cache verification in an attempt to recognize DNS changes and signal operators to escalate readiness.

Email Account Compromise

If an email account is compromised (most commonly through weak passwords), all accounts associated with that email, particularly those on systems without MFA, should be considered compromised, as well. Major email providers spend significant resources on improving account security, and many require the use of MFA, which mitigates the password compromise risk.

1-Click authentication, like all SSO systems, allows users to focus their security efforts on important accounts like email, optimizing their ability to harden their accounts and ultimately improving their security posture in the broader ecosystem of tools and services they use.

Email Provider Compromise

The DMARC specification (RFC7489), states that “It is important to note that the authentication mechanisms employed by DMARC authenticate only a DNS domain and do not authenticate the local-part of any email address identifier found in a message, nor do they validate the legitimacy of message content.” Email systems are complex, internet-facing systems, often composed of hundreds or thousands of servers running modern software. If an email service provider is compromised, it is theoretically possible for an attacker to send an authenticated email from that server.

Since the email system an attacker controls is likely operating with SPF and DKIM configured properly, messages received from the attacker would be impossible to differentiate from legitimate email.

We must consider the depth of impact such compromise would have. Given the near universal reliance on email for password reset processes, an email provider compromise would have a cascading effect on almost all a user's accounts. The scope of systems that fall into this category include all major authentication systems (without MFA), including Swoop and typical password-based login.

Conclusion

Email is the most successful communications platform in the world. Continued, focused, and well-backed refinements in the security of email and the protocols on which it depends have resulted in an incredibly secure, performant, resilient, and reliable system with arguably the most well-known and capable user experience in existence.

While there's work to do, particularly around further adoption of email encryption systems and DNS security improvements, email is ready for broad adoption as an authentication mechanism. Swoop significantly improves both the security and user experience of authentication processes.