EMERGING TRENDS IN DIGITAL AUTHENTICATION

A PREPRINT

Prajwal S Nayak
Department of Computer Science
Manipal Institue of Technology
Manipal KA, 576104.
prajwal.s6@learner.manipal.edu

June 30, 2019

ABSTRACT

This manuscript attempts to shed the light on the authentication systems' evolution towards Multi-factor Authentication (MFA) from traditional text based password systems. The evolution of authentication systems is commensurate with that of security breaching techniques. While many strong authentication products, such as multi-factor authentication (MFA), single sign-on (SSO), biometrics and privileged access management (PAM), have existed for a long time, the constant deluge of data breaches and password database leaks has re-illustrated the weakness in many authentication paradigms. As a result, the industry is both re-thinking they way we approach authentication and making efforts to simplify previously complex or expensive authentication technologies for the every human being.

***K*eywords** Authentication · cryptography · MFA · Oauth

1 Introduction

The era of digital world has culminated through various devices like smartphones, computers, wearable technologies and IOT enabled devices. Owing to the celerity if data penetration, the need for secure and reliable frameworks for authentication is conspicuous. The first part of the paper curates information about authenticating users or people on digital platforms. The last part expounds the same for applications, web services and other softwares which may or may not interact with the end user directly. While all of these authentication methods work to safely confirm that all transactions are legitimate, no single technology will secure online financial transactions with 100% certainty. The key is to stay one step ahead of cybercriminals while preserving a low-friction customer experience. Passwords do neither of those. It's time to step up and embrace the latest in what strong multifactor authentication has to offer.

2 Background

The analog world has witnessed passwords for a long time. Back in the mid 90s computers were huge in size and not a viable tool for the most. Though mainframe like machine were found in research labs, universities and large enterprises. The time sharing systems emerged and passwords were the means to protect private files. When used properly, following very diligent security practices, passwords are a decent authentication factor. The problem is that most humans don't follow the arduous best practices, while many organizations that manage passwords don't follow good practices themselves. The result of this password mishandling was countless password database leaks over the past few decades which have proven that passwords alone are insufficient to protect our online identities.

2.1 Passwords and hashes

The passwords have to be stored in order to be compared against, when a auth request is made. The vulnerability of this file can expose the sensitive information, as witnessed in CTSS password leak of late 1960's. Instead of plain text,

the passwords were then saved using hash functions. The hash function had to be made more unique by adding an element of randomness called as salt; As the attackers found new workarounds for hashing algorithms. Authentication factors have evolved to leverage the opportunities for end-user authentication that are built into smartphones and tablets. While users often have multiple devices, they commonly use the same device to perform certain tasks. Forward-looking businesses recognize this trend and are streamlining the login process accordingly. Using a known device – in this case, a cell phone – users can log into a platform by simply scanning a code with their phone's camera, avoiding the need to input a password. In addition to providing an enhanced user experience, this has the potential to reduce successful phishing attacks. As passwords become less relevant to authentication, phishers will no longer significantly benefit from obtaining end-user credentials.

2.2 Asymmetric cryptography

Though cryptography and authentication belong to two different domains, they share similar technologies and have a symbiotic relationship. In 1977, RSA (Rivest–Shamir–Adleman) introduced at the time when the era of electronic email was expected to soon arise, implemented two important ideas:1. Public-key encryption. This idea omits the need for a "courier" to deliver keys to recipients over another secure channel before transmitting the originally-intended message. In RSA, encryption keys are public, while the decryption keys are not, so only the person with the correct decryption key can decipher an encrypted message. Everyone has their own encryption and decryption keys. The keys must be made in such a way that the decryption key may not be easily deduced from the public encryption key. 2. Digital signatures. The receiver may need to verify that a transmitted message actually originated from the sender (signature), and didn't just come from there (authentication). This is done using the sender's decryption key, and the signature can later be verified by anyone, using the corresponding public encryption key. Signatures therefore cannot be forged.

It emerged as one of the first public-key cryptosystems that is widely used for secure data transmission. RSA is a strong encryption algorithm that has stood a partial test of time. RSA implements a public-key cryptosystem that allows secure communications and "digital signatures", and its security rests in part on the difficulty of factoring large numbers

2.3 One Time Passwords

One of the biggest risks with a normal, persistent password system is that if an attacker can guess, steal, or intercept your password, they could replay it. Algorithmically new, non-predictable passwords were created and received by the users in a way that the central system can still validate them. sFrom the 1980s on, many different OTP standards were develop like S/Key and OTPW which eventually led to OAuth, a standard to standardize types of OTP. SMS-delivered one-time passwords (OTPs) eliminated the need to distribute physical tokens since they leverage the ubiquitous cell phone. Unfortunately, the communication systems used to transmit SMS messages are unencrypted and can be easily intercepted, making them no more secure than traditional passwords.

2.4 Public key infrastructure

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store & revoke digital certificates and manage public-key encryption. It's completely possible for a threat actor to create a new public/private key pair and publish the public key as though it belongs to someone else. To solve this problem, there was need some sort of trusted third party that creates these, thus validating their legitimacy. X.509 emerged as a standard defining the format of public key certificates. X.509 certificates are used in many Internet protocols, including TLS/SSL, which is the basis for HTTPS[1], the secure protocol for browsing the web. They are also used in offline applications, like electronic signatures. An X.509 certificate contains a public key and an identity (a hostname, or an organization, or an individual), and is either signed by a certificate authority or self-signed.

2.5 CAPTCHAs

"Completely Automated Public Turing test to tell Computers and Humans Apart" can limit password retries to human level of speed, greatly reducing the efficacy of brute-force attacks. Though it can authenticate users it can tell humans apart from computers. Modern deep learning models were successfully able to bypass the captchas, hence invalidating their future prospectus.

3 Multi-Factor authentication

The industry categorizes multiple factors of authentication into three buckets: something we know, have or are. Something we "know" would be things like our favorite color, a password, a PIN, or a one-time password. Something

we "have" includes things like special hardware, a digital certificate, a smart card or device containing a digital certificate. And something we "are" is our face, iris, fingerprint, or heartbeat.

3.1 Biometrics

Biometrics are the "something you are" in authentication. They got cheaper and more accessible through smart phone penetration. Strong, frictionless biometrics: Financial institutions are also taking advantage of biometric recognition technology in novel ways. Biometrics can be used to analyze an end user's unique physical characteristics to confirm that they are who they say they are. The process is highly secure but not overly invasive – and thus strikes a balance between security and ease of use. It leverages smartphone technology – like fingerprint scanners, voice recorders or the phone's camera – to enable the user to conveniently confirm their identity, either via fingerprint, voice or facial recognition technology. The best part is, customers are comfortable using these features, as it's as easy as using their phone the way they do every day (e.g., taking a selfie photo, talking on the phone or tapping the screen).

3.2 Two Factor authentication

Before the 2010s, MFA and 2FA solutions were pretty proprietary, expensive, and complex systems that many small and mid-sized businesses couldn't afford to manage.

Challenges Ahead The most recent problems with smartphone MFA have to do with SMS-based options. Text messages are sent in the clear, and attackers can intercept them whether through some man-in-the-middle attack, with phishing or smartphone malware. Over the past few years, we've seen multiple cases of attackers bypassing authentication systems despite their use of SMS-based 2FA. This means we need stronger smartphone 2FA solutions to survive.

3.3 Geolocation

Geolocation Identification: Geolocation also utilizes a user's mobile device to provide authentication services wherever and whenever they are needed. If a customer is shopping at a big box store and their bank needs to authenticate them, one option is for the institution to send a push authentication to authorize the transaction. Alternatively, geolocation allows the bank to access the customer's location via their mobile phone, verifying that the user is in the same physical location in which the transaction is being requested. In this case, there is no need for the customer to respond to a notification, creating a more transparent and frictionless authentication experience.

3.4 Digital signitures

There are different types of encryption techniques are being used to ensure the privacy of data transmitted over internet. Digital Signature is a mathematical scheme which ensures the privacy of conversation, integrity of data, authenticity of digital message/sender and non-repudiation of sender. Digital Signature is embedded in some hardware device or also exits as a file on a storage device. Digital Signature are signed by third party some certifying authority.

4 Authenticating client applications

4.1 Oauth

OAuth is an open-standard authorization protocol or framework that describes how unrelated servers and services can safely allow authenticated access to their assets without actually sharing the initial, related, single logon credential. In authentication parlance, this is known as secure, third-party, user-agent, delegated authorization. OAuth was released as an open standard in 2010 as RFC 5849, and quickly became widely adopted. Over the next two years, it underwent substantial revision, and version 2.0 of OAuth, was released in 2012 as RFC 6749. The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf. This specification replaces and obsoletes the OAuth 1.0 protocol described in RFC 5849. Oauth is widely criticized for various reasons, and it is not a authentication protocol.

4.2 OpenId

OpenID Connect is an open standard published in early 2014 that defines an interoperable way to use OAuth 2.0 to perform user authentication. RFC 6749 includes the definition of a Web API called "authorization endpoint". The API

requires response_type as a mandatory request parameter. OpenID Connect has defined flows to issue ID tokens by extending the specification of the response_type request parameter.

The OpenID Connect ID Token is a signed JSON Web Token (JWT) that is given to the client application along side the regular OAuth access token. The ID Token contains a set of claims about the authentication session, including an identifier for the user (sub), the identifier for the identity provider who issued the token (iss), and the identifier of the client for which this token was created (aud). Additionally, the ID Token contains information about the token's valid (and usually short) lifetime as well as any information about the authentication context to be conveyed to the client, such as how long ago the user was presented with a primary authentication mechanism. Since the format of the ID Token is known by the client, it is able to parse the content of the token directly and obtain this information without relying on an external service to do so. Furthermore, it is issued in addition to (and not in lieu of) an access token, allowing the access token to remain opaque to the client as it is defined in regular OAuth. Finally, the token itself is signed by the identity provider's private key, adding an additional layer of protection to the claims inside of it in addition to the TLS transport protection that was used to get the token in the first place, preventing a class of impersonation attacks. By applying a few simple checks to this ID token, a client can protect itself from a large number of common attacks. Since the ID Token is signed by the authorization server, it also provides a location to add detached signatures over the authorization code (c_hash) and access token (at_hash). These hashes can be validated by the client while still keeping the authorization code and access token content opaque to the client, preventing a whole class of injection attacks.

4.3 HTTP authentication

RFC 7235 defines the HTTP authentication framework which can be used by a server to challenge a client request and by a client to provide authentication information. The challenge and response flow works like this: The server responds to a client with a 401 (Unauthorized) response status and provides information on how to authorize with a WWW-Authenticate response header containing at least one challenge. A client that wants to authenticate itself with a server can then do so by including an Authorization request header field with the credentials. Usually a client will present a password prompt to the user and will then issue the request including the correct Authorization header. As the user ID and password are passed over the network as clear text (it is base64 encoded, but base64 is a reversible encoding), the basic authentication scheme is not secure. HTTPS / TLS should be used in conjunction with basic authentication. Without these additional security enhancements, basic authentication should not be used to protect sensitive or valuable information.

4.4 JWT authentication

JSON Web Token (JWT) and Transport Layer Security (TLS) are the two primary approaches for authentication of the things on the Internet. JSON Web Token (JWT) is used extensively today for authorization and authentication within the OAuth and the OpenId framework. Recently, the Google Cloud IoT has mandated the use of JWT for both HTTP and Message Queuing Telemetry Transport (MQTT) protocol based clients connecting to the cloud service securely over TLS. MQTT is the protocol of choice in IoT devices and is the primary focus of this paper as the application protocol. Another popular cloud platform Amazon Web Service (AWS) uses the TLS mutual authentication for client authentication. Any comparison provided here between the two approaches is primarily from a constrained device client perspective.

4.5 SAML

Security Assertion Markup Language (SAML) provides a secure, XMLbased solution for exchanging user security information between an identity provider (our organization) and a service provider (ASPs or SaaSs). The SAML standard defines rules and syntax for the data exchange, yet is flexible and can allow for custom data to be transmitted to the external service provider. There are three roles involved in a SAML transaction – an asserting party, a relying party, and a subject. The asserting party (identity provider) is the system in authority that provides the user information. The relying party (service provider) is the system that trusts the asserting party's information, and uses the data to provide an application to the user. The user and their identity that is involved in the transaction are known as the subject.

4.6 SSL/TLS protocols

Transport Layer Security (TLS), and its now-deprecated predecessor, Secure Sockets Layer (SSL),[1] are cryptographic protocols designed to provide communications security over a computer network.[2] Several versions of the protocols find widespread use in applications such as web browsing, email, instant messaging, and voice over IP (VoIP). Websites can use TLS to secure all communications between their servers and web browsers.

5 Conclusion

We will design better ways to identify one another online, but I suspect attackers will also leverage new technologies to find better ways to trick our identification processes. This is precisely why MFA was born. The late 2010s and the 2020s should be the era of MFA for everyone due to major decreases in complexity and cost. As technologies used to digitally authenticate people over the decades have advanced, so too have the techniques attackers find to trick or bypass digital authentication. The digital arms race is likely to never end.

References

[1] R. Dhagat and P. Joshi New approach of user authentication using digital signature In *New approach of user authentication using digital signature*, doi: 10.1109/CDAN.2016.7570947.

[2] M. Cooper. Internet X.509 Public Key Infrastructure: Certification Path Building. In *The Internet Society (2005), RFC-4158*, September 2005.

[3] D. Hardt, Ed. The OAuth 2.0 Authorization Framework In *Internet Engineering Task Force (IETF), RFC-6749*, Microsoft, October 2012.

[4] Kelly D. LEWIS, James E. LEWIS, Ph.D. Web Single Sign-On Authentication using SAML In *IJCSI International Journal of Computer Science Issues, Vol. 2, 2009* , ISSN (Online): 1694-0784.

[5] R. Kaur and A. Kaur WDigital Signature, In *2012 International Conference on Computing Sciences, Phagwara, 2012, pp. 295-301.* , doi: 10.1109/ICCS.2012.25