User Authentication System for Online Services Using Cloud Computing

Nasir Ali[1] Mohammad Ishrat[2]

[1]Lecturer (IT), Musanaa College of Technology, Muscat, Sultanate of Oman

[2] Lecturer (IT), Al Musanna College of Technology, Ministry of Manpower, Sultanate of Oman

Abstract ***This paper focusing on methods that are used to implement strong user authentication for online-consumer identities, this paper aims to distill a comprehensive view of strong user authentication by examining its concepts, implementation approaches, and challenges/additional concerns at the architectural level. It discusses effective solution approaches, overall architecture design, and emerging developments.***

Key words – DNS, Web-based user-authentication systems, CAPTCHA, Knowledge-based authentication

.----------♦----------

1. INTRODUCTION

How to protect your identity in online and social world of Internet is a key question today. Digital identity fraud is an increasing threat now a days, with an increase in phishing, DNS poisoning, malware, social engineering and an expansion of attack vectors like, unregulated financial systems with lottery contests. Now we are going to enter in a new form of moving data and services using and cloud-based application, the management and control of access to confidential and sensitive data will become more complex than verifying simple user credentials at the onset of user sessions for one application, and with higher interconnectivity and interdependencies among multiple applications, services, and organizations [1][3].

One of the more exploited methods today is the gaining of account access by stealing reusable credentials for Web sites that have not yet implemented "strong" user authentication. This is so, because most common forms of credentials today are knowledge-based (that is, user ID and password) and are requested only once during sign-on, which provides a higher level of convenience to users, but also requires less effort for attackers to exploit. Many attacks are manifested as "phishing" messages that masquerade as ones that are sent by legitimate organizations and contain URLs that point to fraudulent Web sites that have the same appearances as genuine ones. The ease with which online identities can be stolen and used effectively has prompted many organizations and governing bodies to raise alarms. Subsequently, various methods have been developed to improve the "strength" of an authentication system in withstanding identity-theft attacks. While the spectrum of methods spans a wide range of concern areas—such as enterprises, consumers, hardware, mobility, and so on—this paper focuses on methods that are used to implement strong user authentication for online-consumer identities. It discusses effective solution approaches with the use of cloud computing as emerging technology.

2. STRONG USER AUTHENTICATION

So, how do we improve Web-based user-authentication systems without compromising usability and ubiquity, when the Internet is accessed mostly through a browser that has limited access to the client environment and hardware devices? The most common solution approaches that are used today involve, in more generalized terms, various forms of enhanced shared-secret and/or multifactor authentication [2][4][5].

Enhanced shared-secret authentication refers to extensions of conventional knowledge-based (single-factor) authentication—for example, additional passwords, site keys, preregistered graphical icons to support mutual authentication, challenge-response, randomized code selections that are based on input patterns, CAPTCHA, and so on.

Multifactor authentication refers to a compound implementation of two or more classes of human-authentication factors:

Something known to only the user—Knowledge-based (for example, password, pass phrase, shared secrets, account details and transaction history, PIN, CAPTCHA, and so on).

Something held by only the user—Possession-based (for example, security token, smart card, shared soft tokens, mobile device, and so on).

Something inherent to only the user—Biological or behavior biometric traits (for example, facial recognition, fingerprint, voice recognition, keystroke dynamics, signature, and so on).[6][7]

For example, many enterprise extranet/VPN solutions today require both simple credentials (something known, such as ID and password) and hardware tokens (something held, such as secure ID with time-based one-time password generators, smart cards that use embedded PKI solutions, and so on) in order to gain access. The combination of the two "known" and "held" factors makes up the multifactor authentication method, and significantly improves the authentication strength, as it curtails the threat of stolen digital identities.

3. NEW APPROACHES TO SOLVE THE PROBLEM

Now, not all of the available strong-authentication options that are available today lend themselves well to the Web. Conventional multifactor authentication methods (that involve the deployment of custom hardware tokens, such as RSA Secure ID, smart cards, and so on) are effective for closed communities—such as employees and partners—but they are too costly, inconvenient, and logistically difficult—for example, distribution, administration, management, support, and so on—for the open consumer communities on the Web. In this case, authentication solutions have to work primarily within the confines of the browser's security sandbox. Here, we discuss a few solution approaches that are relatively cost-effective to implement for online consumers, based on today's standards:

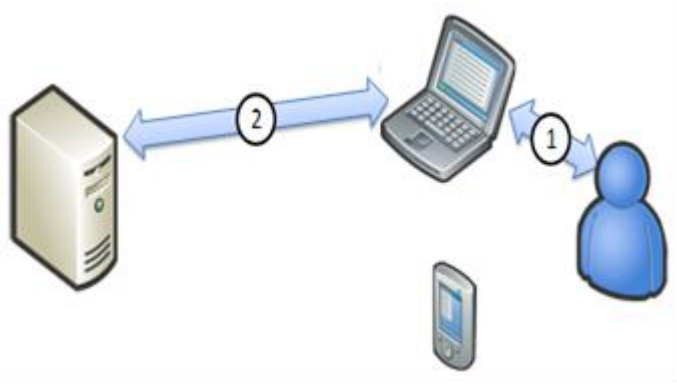

Figure 1.1 Knowledge-based authentication

Knowledge-based authentication (KBA) typically is implemented as extensions to existing simple-password authentication. Knowledge-based credentials include chosen information, personal and historical information, on-hand information, deductive and derived responses, patterns, images, and so on. However, it generally boils down to additional set(s) of challenge-response that allows users to prove that the claimed identities belong to them. Some well-known examples include Bank of America's "SiteKey," HSBC's virtual keyboard, and so on. KBA is used also as an identity-verification method for self-service password-reset processes; but, when implemented effectively, they can be used as methods to complement primary authentication.[8][10] This approach moderately improves authentication strength, as it is still single-factor (in-band within the browser) and prone to phishing attacks, but it might be sufficient for some Web sites.

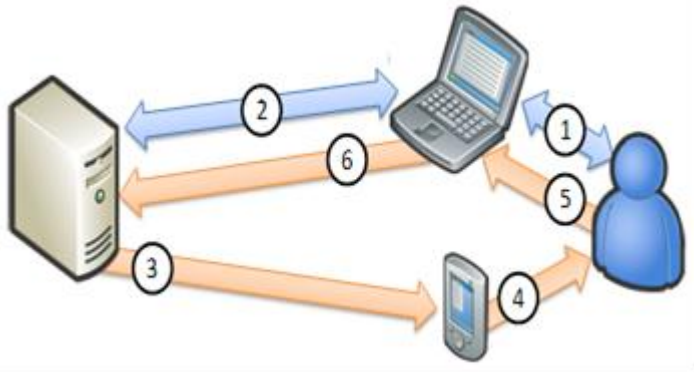

Figure 1.2 Server-generated OTP

Server-generated one-time passwords (OTPs) commonly are implemented as randomized password strings that are generated in real time after verifying simple-password credentials. Some more advanced implementations combine KBA elements to facilitate derived OTPs (such as

server-generated grid cards for shared pattern recognition, digitally signed OTPs that are based on server-generated data, and so on). The generated OTPs then are delivered to users via a different channel (out-of-band) from the session in the browser, such as e-mail, SMS (Short Message Service) text messaging to mobile devices, direct phone calls that use computer-generated speech, and so on. Users then can use the OTP to sign-in to the application, by entering it into a designated field on the page.[9][11][12]

Many organizations in the public sector have started to deploy this type of solution to implement strong user authentication. This approach significantly improves authentication strength as it employs two-factor authentication and out-of-band delivery of OTPs. However, it still is not completely secure, as it is prone to the "man-in-the-middle" real-time phishing attacks that try to capture and use the OTP in real time. Plus OTP delivery latencies potentially could affect overall user experience.

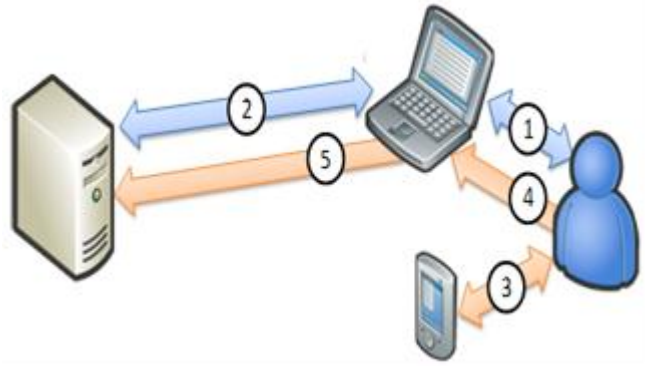

Figure 1.3 Client-generated OTP

Client-generated one-time passwords are similar to conventional OTP hardware tokens (such as RSA Secure ID, VeriSign VIP OTP, and so on). However, with the level of near-ubiquitous proliferation of mobile devices today, cellular phones have become a viable alternative as the soft-token (or "something held") authentication factor. In this case, individualized cryptographic software components can be installed on mobile devices to generate time-based or event-based OTPs. Users then can use the OTP to sign-in to the application after authenticating simple Web-based credentials (examples include RSA Secure ID software, Java ME applications, and so on). This approach has the benefits of OTPs, not having to deal with out-of-band delivery latencies and inconsistent service coverage.[13][14]

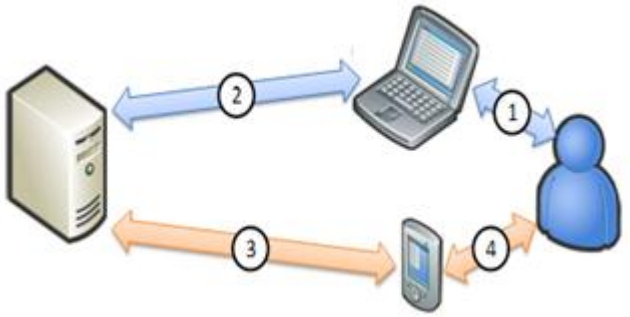

Figure 1.4 Out-of-band authentication

Out-of-band (OOB) authentication leverages the second factor for authentication, instead of delivery of individualized information. Current implementations include speech recognition that is facilitated via out-bound or in-bound voice calls (to/from land or cellular lines) or KBA via SMS request/reply. This type of solution offers higher authentication strength, as both the browser and a second factor are used to verify credentials, which works to impede common phishing attacks.

In general, these high-level approaches rank in increasing relative authentication strength. However, higher authentication strength does not necessarily represent the best-of-breed solution, as security often requires trade-offs in user convenience. Studies indicate that different user populations respond differently to strong-authentication methods. The choice of an approach (or combinations of approaches) should consider user segments, as well as the types of activities and interactions that they perform with the application. For example, online consumers who view account balances (considered a relatively low-risk type of activity) should require comparatively less effort to access than consumers who conduct account transfers or physicians who view patient's medical records.

Finally, the implementation of strong user authentication often is a balancing act between security and usability. Many implementations have not achieved their intended effectiveness by either delivering significantly deteriorated user experiences or compromising security postures by simply satisfying requirements on paper. Furthermore, there is no one-size-fits-all solution approach. For example, not all Web sites require multifactor authentication, while enhanced knowledge-based authentication might be "secure enough"; and some carefully designed single-factor methods actually might be "stronger" than some multifactor methods. A well-balanced design tends to be more cost-effective than force-fitting various best-of-breed approaches into one solution.

4. USE OF CLOUD COMPUTING

From a user-authentication perspective, moving data into the cloud and integrating cloud-based services should be implemented with the same level of overall effective authentication strength as the enterprise perspective of authentication architecture. However, organizations have significantly less control over the authentication strengths of the interdependent cloud-based services of their counterparts/partners. For example, whether via identity federation or delegation, the overall security posture of the resulting interconnected architecture can be compromised if the integrated services themselves have comparatively lower-strength authentication systems in place.

The same is true for SaaS (software-as-a-service) providers, as extra attention must be focused on ensuring appropriate levels of authentications strengths for different user communities in a multitenancy model, without compromising overall and individual security and usability.

Thus, the focus on authentication systems becomes one of the primary evaluation factors for organizations that are looking to adopt cloud-based services. Organizations must ensure that service providers provide the flexibility to deliver varying levels of strong authentication to meet required security policies, or extend existing security implementations by leveraging identity federation (via SAML or WS-Federation) or authentication delegation to support single sign-on (SSO) or reduced sign-on (RSO). However, in these cases, organizations must incur the costs to deploy secure and accessible identity-federation and/or authentication-delegation services.[15]

From a capabilities perspective, many of the authentication architecture components are being deployed as cloud-based services—for example, identity-proofing services that are deployed by credit bureaus, consumer-identity frameworks and providers, vulnerability-management networks, PKI and certificate-management services, secondary-factor channel providers (voice telephony, SMS messaging, speech recognition, patterns recognition, and so forth), fraud detection, strong-authentication service providers, and so on. These services provide much-needed capabilities to compose a strong-authentication system; however, the same integration-security concerns remain such that any one weak link in the connected-systems architecture will compromise the overall security posture.

5. IDENTITY META SYSTEMS

The consumer-identity frameworks that are available now as cloud-based platforms and their growing adoption means that organizations eventually will need to integrate these identity meta systems to improve user convenience—for example, Open ID identity providers, Google Account, Windows Live ID, Yahoo! ID, and so on—although, in order to integrate these online communities, the authentication strengths that are implemented for these services must be evaluated against the security policies and requirements for the organizations that are looking to leverage them.

Similarly, online identity providers increasingly will need to add flexibility to configure varying levels of authentication strengths for different user segments, in addition to integrating various authentication form factors and standards (Higgins, PKCS, OpenID, Windows Cardspace, and so on) if they intend to provision services to data-sensitive organizations.

SMART-CARD PROLIFERATION

With the availability of more sophisticated smart-card solutions and ecosystem support, more physical credentials are adopting smart-card (standard plastic cards embedded with microprocessors and/or integrated circuits) deployments. For example, many countries and states (for example, Austria, Belgium, Estonia, Hong Kong, and Spain) already have rolled out government-sponsored electronic ID programs to national citizens. Subsequently, smart cards are becoming another form of authentication factor, where smart-card readers are available and are integrated into authentication systems.

Furthermore, many vendors are consolidating multiple authenticators into the ISO 7816 smart-card form factor—for example, integrated LCDs to display OTPs, and biometric (fingerprint) readers. We might find smart-card deployments materialize in more cases, such as from financial institutions that already are issuing physical credentials (that is, credit cards, debit cards, and so on). Cryptographic smart cards that use biometric readers provide very high identity assurance, as they tightly bind the private keys to the users' biometrics (multifactor authentication).

MOBILE IDENTITY

From a physical-hardware perspective, SIM (Subscriber Identity Module) cards have improved significantly in terms of storage capacity and capability to perform cryptographic processing. Computing power and memory capacity also have improved exponentially in mobile devices. Subsequently, the SIM card and mobile phone have become the smart card and smart-card reader that constitute the most ubiquitous "something held" (or in-possession) authentication factor.

This makes it possible to store symmetric keys on SIM cards and, along with simple cryptographic software

modules, to turn the mobile device into a seeded OTP generator. The generated OTPs can be used as credentials for out-of-band, multifactor authentication.

Furthermore, with wireless data plans, mobile devices can communicate directly with authentication systems by using wireless PKI. In this case, SIM cards provide secure storage of users' private PKI keys. The private keys then can be used to facilitate strong authentication—implemented with corresponding certificates to facilitate digital signatures—and, in some cases, to facilitate client-authenticated SSL. Some of the projected applications of this approach include mobile banking, contactless/proximity mobile payments, identity and credential verification, and so on. At some point, mobile devices might become the most ubiquitous form of mobile digital identities for consumers.

6. CONCLUSION

To conclude we can say that a user-authentication system for consumer communities on the Web is growing beyond the traditional database-driven and/or directory-driven component of a Web application, for organizations that have higher data-confidentiality requirements. Implementation approaches for strong authentication span a full spectrum that ranges from highly integrated and interconnected/dependent to simple extensions of existing stand-alone architectures.

The escalating trend of moving data and services into the cloud also necessitates methodical planning to ensure secure access to authorized users over the Internet. While existing simple-password–based authentication might continue to work for many consumer-oriented Web sites, its inherent vulnerabilities have been identified as security risks for institutions that have higher data-privacy requirements. To mitigate the risk of online identity fraud, organizations look to strong user authentication as the solution for improving their Web-based authentication systems.

However, implementing strong user authentication often is not a straightforward task, as projects have myriad options from which to choose, a multitude of trade-offs to consider, and a cluster of intricacies to manage. This article has intended to distill a comprehensive view of strong user authentication by examining its concepts, implementation approaches, and challenges and additional concerns at the architectural level.

REFERENCES

1. Authentication in an Internet Banking Environment, Federal Financial Institutions Examination Council. http://www.ffiec.gov/pdf/authentication_guidance.pdf

2. Electronic Authentication Guideline v1.0.1, National Institute of Technology Special Publication 800-63 (NIST SP 800-63).

3. http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf

4. Policy for a Common Identification Standard for Federal Employees and Contractors, Homeland Security Presidential Directive-12 (HSPD-12).

5. http://www.whitehouse.gov/news/releases/2004/08/20040827-8.html

6. Personal Identity Verification of Federal Employees and Contractors, Federal Information Processing Standards Publication 201-1 (FIBS PUB 201-1).

7. http://csrc.nist.gov/publications/fips/fips201-1/FIPS-201-1-chng1.pdf

8. Electronic Signatures in Global and National Commerce Act, United States Congress E-SIGN Act.

9. http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=106_cong_bills&docid=f:s761enr.txt.pdf

10. A Guide to Understanding Identification and Authentication in Trusted Systems, National Computer Security Center (NCSC-TG-017).

11. http://www.csirt.org/color_%20books/NCSC-TG-017.pdf

12. Computer Security Guidelines for Implementing the Privacy Act of 1974, FIPS PUB 41.

13. http://www.itl.nist.gov/fipspubs/fips41.pdf

14. Security Requirements for Cryptographic Modules, FIPS PUB 140-2.

15. http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf