Password-Based Authentication: A System Perspective

Art Conklin[1], Glenn Dietrich[2], Diane Walz[3]
The University of Texas at San Antonio
Information System Department
6900 North Loop 1604 West
San Antonio, TX 78249
1. aconklin@utsa.edu 2. gdietrich@utsa.edu 3. dwalz@utsa.edu

Abstract

User authentication in computer systems has been a cornerstone of computer security for decades. The concept of a user id and password is a cost effective and efficient method of maintaining a shared secret between a user and a computer system. One of the key elements in the password solution for security is a reliance on human cognitive ability to remember the shared secret. In early computing days with only a few computer systems and a small select group of users, this model proved effective.

With the advent of the Internet, e-commerce, and the proliferation of PCs in offices and schools, the user base has grown both in number and in demographic base. Individual users no longer have single passwords for single systems, but are presented with the challenge of remembering numerous passwords for numerous systems, from email, to web accounts, to banking and financial services. This paper presents a conceptual model depicting how users and systems work together in this function and examines the consequences of the expanding user base and the use of password memory aids.

A system model of the risks associated with password-based authentication is presented from a user centric point of view including the construct of user password memory aids. When confronted with too much data to remember, users will develop memory aids to assist them in the task of remembering important pieces of information. These user password memory aids form a bridge between otherwise unconnected systems and have an effect on system level security across multiple systems interconnected by the user. A preliminary analysis of the implications of this user centric interconnection of security models is presented.

1. Introduction

The concept of a user id and password is a cost effective and efficient method of maintaining a shared secret between a user and a computer system. Identifying a user is essential for the application of security in the form of permissions to various objects, processes and access to resources. User authentication in computer systems based on passwords has been a cornerstone of computer security for decades. The authentication process is embedded in many systems, in many different variations. In each case, one common aspect is the focus on mapping authentication data to specific authorized users for a specific application. And this central focus, the mapping, is designed from the perspective of the specific system or application, encompassing its set of valid users.

The implementation of user authentication using a password, from an application point of view was a valid assumption when there were only a few applications compared to numbers of users. Today, with the rise of the Internet and a push for ubiquitous computing, this low application count per user assumption does not hold true. Users have multiple accounts on multiple systems. Users must to remember multiple IDs and multiple passwords for the wide range of computer based services they use. This has placed a strain on user memory and users have developed memory aides, such as password lists, to assist them in the task of keeping accounts and passwords straight.

The purpose of this paper is to present a conceptual model of password-based security across multiple systems connected by user activity. We emphasize the effect of user generated schemes to assist in the user's management of IDs and passwords, and the effect of these memory aides on system security. Examination of system security from a user perspective illuminates

0-7695-2056-1/04 $17.00 (C) 2004 IEEE

interconnections and pathways between systems which are not visible from a specific application perspective.

User password memory aids affect overall system security at the individual application level in two ways. Users' security is decreased, with the memory aid itself becoming a source of risk. And application security also suffers because of the inter-system relationships created by the memory aids. The application-centric focus of software developers and system engineers fails to the user-centric and system-wide implications of password-based risk. The acceptance of these issues requires a change in mindset on the part of system developers, embracing each new application as part of a larger, greater system (as opposed to an application centric view). To achieve desired levels of system-wide security will require an understanding of the cognitive limitations of users and the behaviors which result from these limitations. The conceptual model presented here illustrates some opportunities for system wide improvement of password-based risk.

2. Conceptual Development

2.1 Authentication

In order for a computer system to perform specific acts on behalf of a specific individual, an identification and authorization step is needed. This process of identification and authorization has been extensively studied and formally described in principle [1, 2]. These formal descriptions include the user as the principal and document all relevant constructs and issues from a single system perspective.

The concept of distributed computer systems does not change the need for identification and authentication, although the logic for these tasks was developed from a single-system connection point of view [2, 3].

Authentication is a simple function where one party presents a set of credentials to a system. If the credentials match a given set on the system, the system returns a value that represents authorization; otherwise it does not. The purpose of authentication is to verify that the specific information presented represents a request to be authentic from a specified entity [1, 2]. This is important, for verifying the identity of an entity is the basis for all future rights and privileges granted to the entity [4]. Whether the presenting entity is a computer program or a user makes no difference to the authentication process.

In the basic authentication process, the entity desiring authentication presents credentials, usually an account ID and some additional information, to prove that the request is coming from a legitimate owner of the ID. This is a relatively straightforward process that has been in use for decades. This can be represented as presenting something you know and others would not know. An example is a user ID and password combination, one of the simplest forms of user authentication [4-7].

A more complicated example is the smartcard system [8, 9], where a user typically has an ID, a password, and also a time-generated passkey from the smart card which changes every 60 seconds. This represents the case of something you have, as in the smartcard, or possession of a physical key. The authenticating server has the same time changing numerical sequence as the specific smart cards assigned to that ID and if the ID, password and card generated number are all correct, authentication is granted. This scheme verifies not just the knowledge of an ID and password, but also possession of the specific smart card assigned to the ID. Frequently smartcards are combined with passwords for an account to increase security. This is an example of two-factor authentication and is more secure because it requires more items for authentication.

A third form of authentication involves the concept of demonstrating "what you are" or biometrics. Biometrics can take the form of several measurements, from fingerprints, to retinal scans to pupil images. The idea is again the same, the presentation of unique information proving identity. The advantage of biometrics is that, for most cases you don't leave home without them, and they can not be forgotten. Disadvantages are many, including not being able to change them if needed, or use them for all functions as they are not secret and are not possessed by non-human entities needing authentication [10].

Many modern systems have adopted a simple id/password method of achieving the goals associated with the identification and authentication function, and numerous technical methods exist to achieve this end [4, 5, 8, 11-21]. The wide variety of implementation schemes for identification and authentication are a result of individual design decisions appropriate to specific circumstances at the time of design of a specific system or class of systems.

The need for differing levels of protection is based on an assessment of the risk associated with a particular system. Designing security levels commensurate with risk levels has a long history and is formally described in a number of sources [4, 22].

The end result is an authorization system that establishes specific security levels based on the needs of the application. This is an application centric point of view which models the user in isolation from the user environment which may consist of numerous other applications, each with their own authentication method.

The specific implementation of an authentication method from an application point of view is a valid assumption whenever the number of applications per user is small. Today, with the rise of the Internet and a push for ubiquitous computing, the assumption of a low application count per user is seldom valid. Users interact with multiple systems and can act as a bridge between them connecting their security mechanisms through password memory aids. The proliferation of numerous single method solutions has forced users to remember numerous IDs and passwords, a task that is becoming increasingly difficult.

2.2 Human Cognitive Ability

Important research on human cognitive ability has generated a lot of practical knowledge on the issue of what an individual can remember [23]. Phone numbers are split into chunks to assist the memory, and personal phone books supplement human memory systems. Domain names, such as www.microsoft.com, are used because people cannot remember IP addresses. Browsers have Favorites functions to remember web site addresses for users. It has been argued that one of the main purposes of a personal digital assistant (PDA) is to act as a high tech memory aid. All of these examples are immediate reminders of human memory limitations, and the systems users have created to handle the limits of human memory.

The effect of human cognitive ability in the authentication process is a central element, though often overlooked by developers. Remembrance of passwords is one of the cornerstones of the current password-based authentication system [24-31]. Widespread usage of password protected systems accessed by the Internet has caused an explosion in the number of accounts per user and is revealing issues associated with users' difficulty in remembering passwords [32-35].

One impact of the spread of computing is the users' requirement to manage multiple computer accounts and passwords. Software and system designers are working from the perspective of the system, where the system has many users. The number of other password-based accounts "owned" by a system user is usually not considered. Thus, users are developing their own systems for dealing with the memory issue surrounding multiple account names and passwords. And these systems may or may not, conform to the security needs and requirements of the various application systems. In fact, such systems are physically exogenous to the applications accessed by the users.

2.3 System Design

The overall design of a complex system is the venue of the system architect/engineer. Systems engineering is the process of designing and developing multiple interconnected components in such a way that they function efficiently together to perform specific tasks which meet specific needs in an organization. A basic tenet of system engineering is the concept that the whole can do more than the sum of the parts.

Security is frequently an emergent property of a system [36], not specifically pegged to a single component, but one where several components interact to produce the desired result. To design security into a system requires a system level of thinking, for the design must take into account the interaction between components of the system, and the resulting emergent properties or lack thereof, when attempting to achieve specific levels of security [37].

Before the rise of highly interconnected, distributed, network based computing, most computer systems ran in isolation, and the number systems for each user were small. Today's highly distributed computer systems exist in a different environment and users now access multiple different computer systems and have separate logons for both disparate and interconnected systems. Attempts to resolve the issues surrounding multiple logons with concepts such as single sign-on have been attempted, but are frequently expensive and do not scale to the entire span of distributed programs in most environments [33, 38].

The locus of traditional software engineering is system-specific with consideration for the customary ergonomic aspects of the users' environments, with little or no regard for the multiplicity of other systems and accounts managed by those users. One principal concept of both systems and software engineering is scalability, i.e., can a specific solution to a problem continue to perform effectively when it is increased in proportion to its increased use or spread. In the case of the user authentication process, the concept of user IDs and passwords is technically scalable. Limitations of human cognitive function and memory, however, create exogenous barriers to scalability. Users are faced with an ever-increasing task of managing account names and passwords, and are building their own methods to address this problem. Yellow sticky notes, lists in wallets, re-use of passwords across systems, key fobs that store passwords, and personal "password books" are all methods for addressing memory limitations with respect to passwords.

The expanding pool of Internet users (now including late adopters) often has even less tolerance for remembering account names and passwords. This places an even greater need and emphasis on the

memory aids for passwords. Memory aids act as user-centric extensions of security systems, assisting a user in the management of their user ids and passwords. Because of their user centric position, they operate at a user level, and between disconnected systems. If a user uses a specific aid of a common password, then this aid connects their Amazon account and their banking account through this common element. Discovery of the Amazon account information can lead to compromise of the banking account. And increased use of memory aids increases the connections between systems by users, weakening overall system security.

Common memory aids include password lists in wallets, lists in personal digital assistants, common personal items such as child name and birth date, pet names, etc. Each user adapts what works for themselves to assist in the remembering of multiple passwords. One of the simplest memory aids is a user selected common password between accounts. When a user gets to pick a password, picking one that the user currently uses elsewhere represents a simple method to reduce the number of passwords a user must remember. But in this simple act, the user can tremendously lower security of other systems by providing easy access for an unauthorized user. Assume system A is a high security system, such as a bank, ecommerce or investment site. Here a system would carefully guard the user password and would not do such security lessening functions as emailing the password to a user. But if the user has the same password on a less secure site, say a marketing site, or library site, then less security may be employed by the site, making the password vulnerable to discovery. Discovery of a password for Site B, the low security site, gives an unauthorized user a good choice for the high security site, a choice rewarded by the user's desire to remember fewer passwords.

2.4 Password-Based Risk

Security risk from un-authorized entry involves more than the risk to a single user via their system account. While an individual user may not bear risk from unauthorized access (as when the user stores no personal or sensitive information on the account), the system itself can be at risk. A first step in breaking into a system and causing damage is to obtain user level access. It is not the un-authorized user level access that is a primary concern of a system administrator as much as the next step – privilege escalation. This is where an un-authorized user gets the ability to do real damage, and it starts as simple user access. So, while an individual user may not recognize the system level risk in compromising their account, the system administrator should. Security begins with blocking the initial access for un-authorized users, as the problem increases in complexity after initial access is obtained.

The basic premise behind password-based security is that an authorized user can keep and remember a secret. And that secret, in turn, is used to authenticate the identity of the authorized user for access to a particular system. From the system's perspective, a password should be easily remembered, yet hard for an intruder to guess [26, 30]. Although other system level solutions exist, much of the effort to secure password-based systems is focused on thwarting unauthorized access through better password selection [6, 19, 39-41].

Many known weaknesses exist in password-based systems, and various fixes have been applied over time [5, 11-13, 15, 20, 42-46]. The types of attacks can be divided into three categories: technical (brute force), discovery, and social engineering. To counter these types of attacks, designers have responded with three types of safeguards; password rules, system rules, and training and awareness. In the middle of all of these elements is the construct representing the user generated password memory aid. These seven constructs are the basic elements in the models of password-based risk presented here.

2.4.1 Attacks. In the brute force (technical) attack, two methods can be used. The first is just attempting passwords against the system, but this is easily stopped with account lockouts. The second is an offline attack against the password hash file. This is a processor intensive search through the entire password keyspace, calculating and comparing hash values of potential passwords to the values in the stolen hash file. This exploit can also be performed off line and on a high speed PC, attempting millions of keys per second. Various defenses exist, including increasing keyspace through the use of salts, and physically protecting the password hash file.

Passwords may also be compromised by discovery. Forms of password discovery may vary and include interception of a script file, an exploit on another system, a Trojan program capturing keystrokes, or the discovery of default passwords associated with other systems or programs. Whether performed by an un-authorized user by looking for the yellow sticky note, or a network sniffer recording network traffic, the end result is the same, a plaintext password is 'discovered' and then used in an un-authorized manner. This is a targeted, directed system level exploit aimed at specific access, whether through another account to increase privilege or across systems for common users and accounts. The primary defense against discovery is proper system design rules that do not allow discovery

of passwords through scripts or default system accounts. A common system level rule of emailing passwords at the users' request is actually a destructive rule. Although this provides an automated method to assist a user, emails are plain text in nature, and this gives an un-authorized user with a properly deployed network sniffer an ability to request passwords on demand.

Social Engineering represents an attempt by an intruder to elicit password and account information from a user. This attack is exogenous to the computer system in question, coming via phone, fax, email, or casual contact. This is a common method of obtaining user level access, and the attack is often disguised in a very official sounding, persuasive manner. This method of attack takes advantage of a person's willingness to help. All requests are typically round about, indirect, and subtle, and often the victim is not aware they are divulging information. The primary defense against this type of exploit is training and awareness directed at the user with respect to this specific vulnerability.

2.4.2 Safeguards. Password rules are either optional or enforced specifications about the length of the password and the diversity of the characters that comprise it. The length and diversity contribute to the size of the domain set containing all possible passwords (commonly referred to as keyspace), that increases the difficulty of brute force detection. Prevention of easily guessed passwords reduces discovery. However, the same rules that increase password resistance to brute force attack directly reduce the ability of a user to remember a password and increase the need for password memory aids.

System rules relate to the procedural aspects of gaining access and are enabled in a system. For example, the automatic user lockout after three failed attempts is a system enforced rule. More sophisticated mechanisms include expiring passwords and the forcing of password changes, or prescribing the amount of change at password change time. The reporting of failed access attempts is another system rule designed to improve security. System rules can also have an opposite effect though, as they can lead to discovery patterns. There are systems that will email an unencrypted password back to a user if requested, presenting an opportunity for discovery.

System rules that make passwords harder to remember can increase the need for user based password memory aids. System rules for recovering forgotten passwords, such as emailing forgotten passwords lessen the need for memory aids, but increase risk of discovery. In general, rules that enforce higher quality passwords do so at the expense of user memory ability. This was an acceptable tradeoff when the count of passwords per user was low, but this tradeoff today forces users to use memory aids.

One of the weakest links in a security system is an untrained user. Formal and informal activities of training and awareness can alleviate a wide variety of actions that weaken a system, such as choosing poor passwords, writing them down, sharing them with others, and inadvertently giving information to strangers that have no need to know. Training can address issues associated with discovery and social engineering attacks. The primary issue with training is its temporary nature, users forget or become complacent over time and re-training is time consuming and costly. The additional issue that the effect of training diminishes over time only exacerbates the training difficulty.

2.4.3 System Level Issues. The password-based risk to an information system must be considered at both the system and the user level. The system level password risk is the potential for harm to the system that results from the design of the password authentication procedures, specifically through the safeguards (password rules, system rules, and training and awareness). This risk can be considered to be the probability of unauthorized access, times the amount of damage that an intruder can inflict, where the probability of unauthorized access is a function of the quality of the instantiated safeguards.

Password risk at the individual user level has the potential for harm to the system from an individual user's password selection. This risk is also conceptualized as the probability of unauthorized access times the amount of damage that an intruder can inflict. The probability, however, is determined by the password selected and the manner in which the password is protected, or not.

Adherence to password rules does produce passwords that are more difficult to break. The problem is that the passwords are also more difficult for users to remember. Adherence to system rules produces passwords that are more difficult to discover. Again, the problem is that this also makes passwords more difficult to remember. For this reason, user memory aids will be developed by the user, distinct from the overall system design, and the existence and use of these memory aids will serve to increase the risk from discovery and social engineering attacks.

3. Risk Models

Figure 1 presents a password-based risk model for a single system, in isolation. The model shows how the system safeguards mitigate the system's vulnerabilities to brute force, discovery, and social engineering attacks. Training and awareness mitigates both attacks of discovery and social engineering. Appropriate system rules and password rules decrease the likelihood of brute force attacks. System rules also allay discovery attacks with procedures and requirements for password construction, changes and transmission.

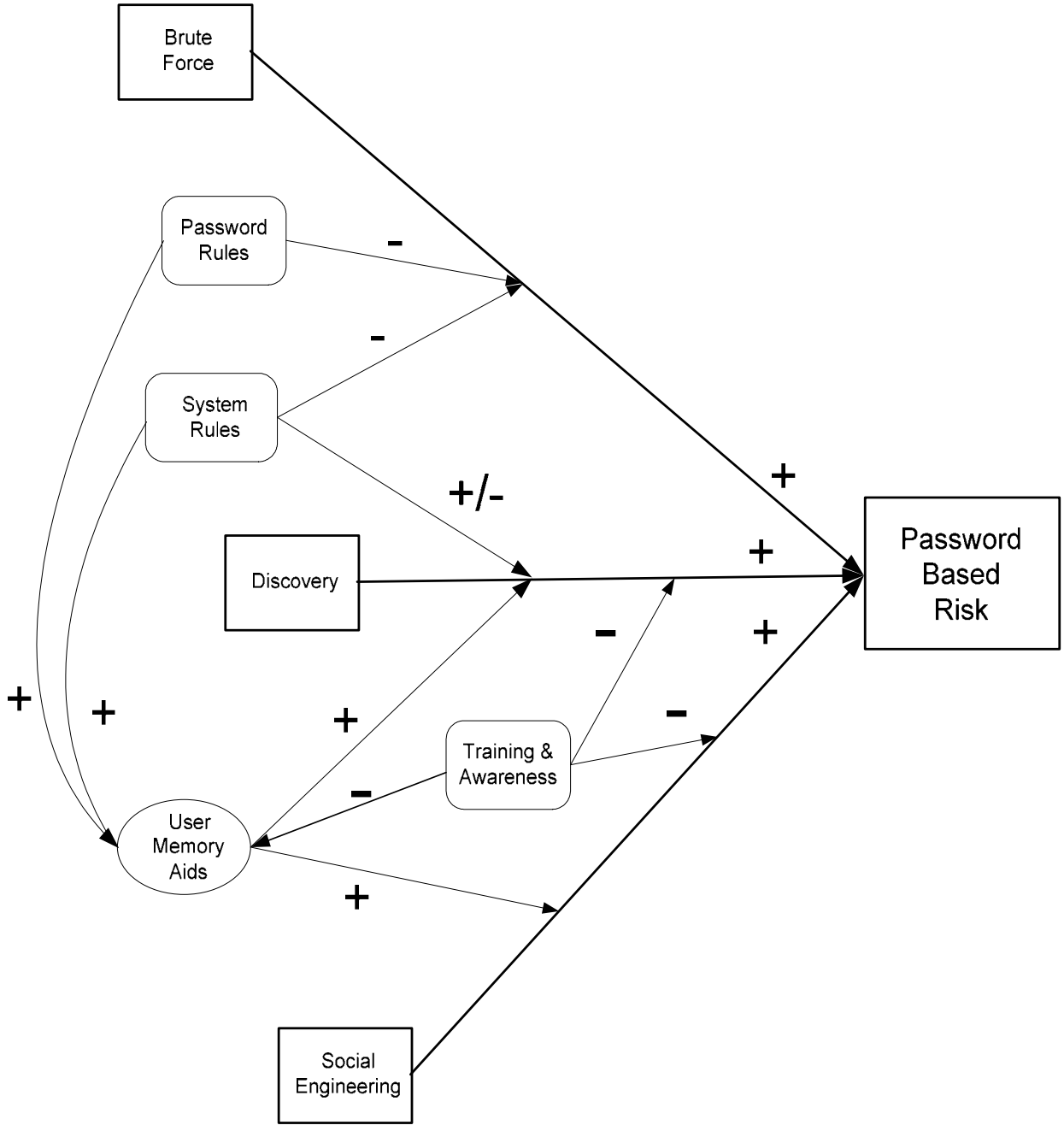

Figure 1 Password-Based Risk model for a Single System in Isolation

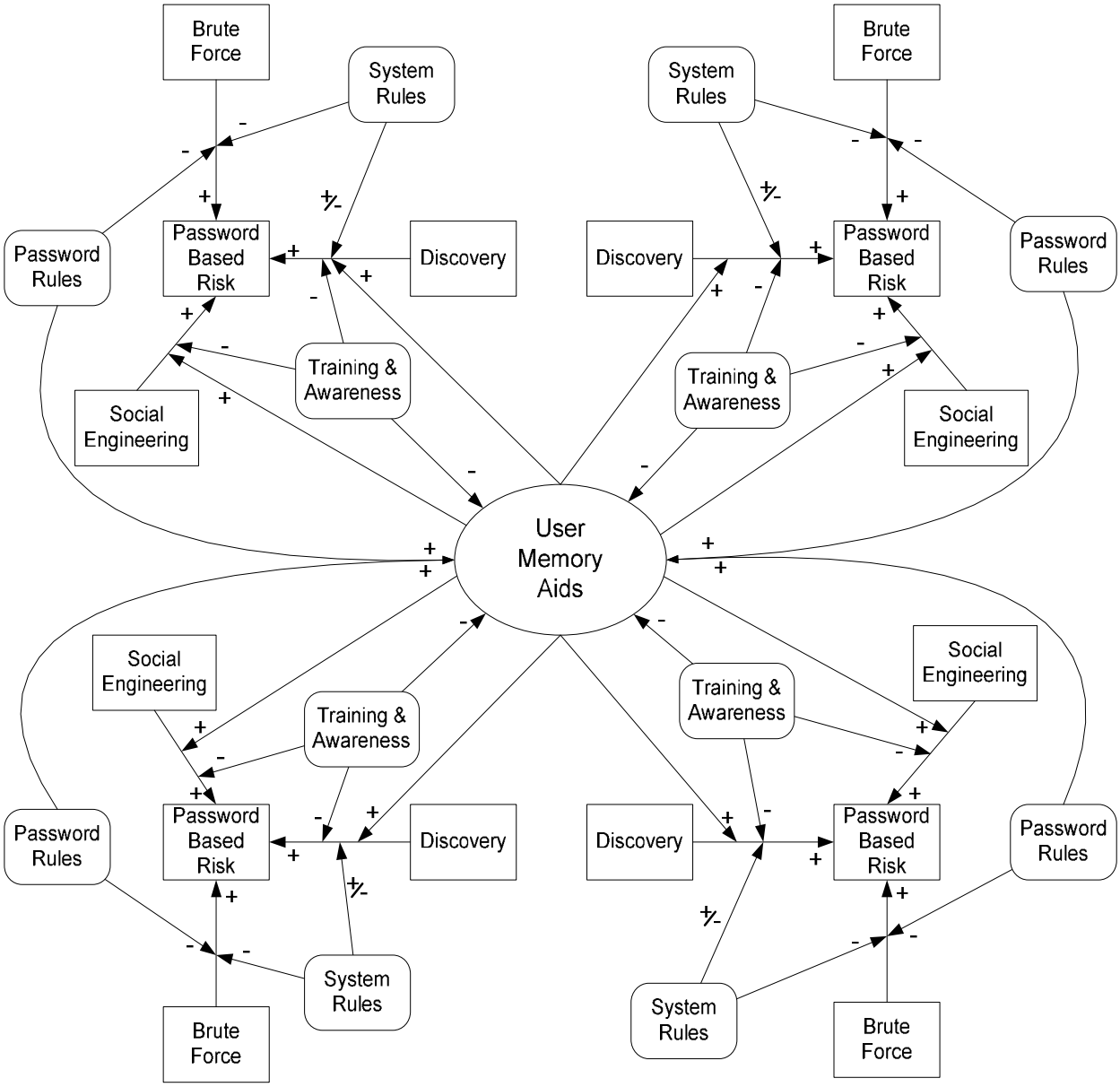

Figure 2 Multiple System Security Design with User Memory Aid

As users access more and more systems, the interconnections for a single user begin to propagate quickly. The number of systems for which a user is authorized can range from two to literally hundreds. The user's need for a memory aid is exacerbated as the number of accounts and passwords climbs. Even training and awareness cannot change the basic cognitive limitations which make it impossible for users to simultaneously accommodate the security recommendations of multiple systems.

In Figure 2, it can be seen that the user memory aid can be a point of connectivity between the four systems, in addition to any other hardware-based connections that may exist between the systems. If you imagine several different users, each with access to a set of systems, then the set of systems to which any user has access are connected to one another via the user memory aid itself. Thus, a successful discovery attack on System A may jeopardize the security of System B. The route into a system, before privilege escalation, begins with a simple user account. This also is an avenue for distributed denial of service attacks and a whole set of other vulnerabilities. In an environment where every user account must be

considered sacred in a secure system, a dangerous situation has evolved. Users are changing the connectivity and configuration of multiple systems.

4. Conclusions

The representation of multiple systems connected through the functionality of a single user's memory aid is a viewpoint not typically considered by designers and architects [2]. They view systems with respect to an established set of requirements that are typically narrower in scope and do not address the needs associated with ubiquitous computing and total system wide scalability. Developing this new view of a complete system, composed of independent systems and user developed constructs such as password memory aids allows designers and developers to assess the impact of their decisions in a large scope of computing.

Different memory aids can result in different system cross connections, as does differing system level security implementations. A common password type memory aid can bridge two or more systems. If one of these 'connected' systems has a lower level of security implementation, then this can be carried over to a higher level system. A password revealed for a library login can give an unauthorized user access to an investment site for a user sharing a password across these accounts. The actual degree of system connection depends upon the memory aid and the level of security implementation on each system.

5. Implications

A user's choice of password, the specific implementation of the authentication subsystem, can have a system level effect on security. The fact that system inputs can effect system operation is a commonly understood tenet of systems engineering. But when applied to today's distributed interconnected systems, the scope of "a system" can become quite large and out of the domain of control of any specific entity. This leads to situations where a security breach on one system due to poor password authentication can directly lead to a system outage at a partner company via an interconnected transaction system that allows cross system infections. User-based password memory aids can act as just such a bridge.

Something as simple as a stolen PDA with passwords in it, or a password sniffed from a network connection while it is being emailed in plaintext as a response to a 'send me my password request' or a scrap of paper left with account id and password information opens the door not to just a single system, but potentially to many systems. Many of these implementations of user memory aids are essential for the user, yet compromise results in a spreading loss across multiple systems without the user or designer truly being aware of the impact until afterwards.

Although the developer community is aware of the linkages between systems and the first order effects that these linkages can cause, the rate of growth of interconnected systems and the human limitations in managing passwords across numerous distributed accounts results in systems being interconnected in ways never imagined. The Slammer worm attack in 2002 infected machines at an unprecedented rate, doubling its number every 8.5 seconds, and finding 90% of its targets worldwide in 10 minutes [47, 48].

For ubiquitous computing to achieve its goal of transparent computing across all aspects of life, the systems level issues need to be resolved by design. System designers must view their systems and their users within an interconnected web of authority and access. Architectural review of security related implications of designs across systems are needed prior to implementation. Future research is planned which will attempt to eliminate the "sunk costs" of password-based authentication and identify and evaluate alternative methods in this framework.

With the advent of net-centric computing and distributed systems such as web services, there is a move towards completely decentralized computing. System models are becoming more complex as the interconnections grow in number. One common element across this network is the concept of a principal party, whether human user or computer account, under which permissions and privileges are extended based on authentication and authorization [2]. Users form the central theme of these distributed systems, and the same Jane Smith that orders a book from Amazon, may pay her bills on-line, access city services on-line and buy theater tickets on-line, each separate system having separate authentication methods for the same end user, Jane Smith. Ignoring the role of the user in these systems has led to slowed acceptance of many on-line functions in the user community [38]. Addressing system level impediments to user access will have a positive effect on the adoption of new applications of computer assisted services, from e-commerce to e-government to entertainment and more.

6. References

[1] M. Burrows, Abadi, M., Needham, R., "A Logic of Authentication," *Proceedings of the Royal Society of London*, pp. 233-271, 1989.

[2] B. Lampson, Abadi, M., Burrows, M., Wobber, E., "Authentication in Distributed Systems: Theory and Practice," *ACM Transactions Computer Systems*, vol. 10, pp. 265-310, 1992.
[3] C. Cachin, "Modeling complexity in secure distributed computing," presented at International Workshop on Future Directions in Distributed Computing (FuDiCo), Bertinoro, Italy, 2002.
[4] J. Anderson, Vaughn, Rayford, "Guide to Understanding Identification and Authentication in Trusted Systems (Light Blue Book)," National Computer Security Center NCSC-TG-017, September 1991 1991.
[5] U. Manber, "A Simple Scheme to Make Passwords Based on One-Way Functions Much Harder to Crack,," *Computers & Security*, vol. 15, pp. 171-176, 1996.
[6] B. Menkus, "Understanding the Use of Passwords," *Computers & Security*, vol. 7, pp. 132-136, 1988.
[7] B. L. Riddle, Miron, M. S., Semo, J. A., "Passwords in Use in a University Timesharing Environment," *Computers & Security*, vol. 8, pp. 569-579, 1989.
[8] W. Yang, Shieh, S., "Password Authentication Schemes with Smart Cards," *Computers & Security*, vol. 18, pp. 727-733, 1999.
[9] M. Abadi, Burrows, M., Kaufman, C., Lampson, B., "Authentication and delegation with smart-cards," *Science of Computer Programming*, vol. 21, pp. 91-113, 1993.
[10] J. Vaclav Matyas and Z. Riha, "Toward Reliable User Authentication Through Biometrics," *IEEE Security & Privacy*, vol. I, pp. 45-49, 2003.
[11] K. Dehnad, "A Simple Way of Improving the Login Security," *Computers & Security*, vol. 8, pp. 607-611, 1989.
[12] M. Bishop, and Klein, D.V., "Improving System Security via Proactive Password Checking," *Computers and Security*, vol. 14, 1992.
[13] N. Ahituv, Lapid, Y., Neumann, S., "Verifying the Authentication of an Information System User," *Computers & Security*, vol. 6, pp. 152-157, 1987.
[14] J. Evans, Arthur, Kantrowitz, William, Weiss, Edwin, "A User Authentication Scheme Not Requiring Secrecy in the Computer," *Communications of the ACM*, vol. 17, pp. 437-445, 1974.
[15] D. L. Jobusch, Oldehoeft, A. E., "A Survey of Password Mechanisms:Weakness and Potential Improvements. Part 1," *Computers & Security*, vol. 8, pp. 587-604, 1989.
[16] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, vol. 24, pp. 770-772, 1981.
[17] C. Lin, Hwang, T., "A password authentication scheme with secure password updating," *Computers & Security*, vol. 22, pp. 68-72, 2003.
[18] G. Mcgraw, Viega, John, "Making your software behave: Cryptographic essentials Using hashing algorithms for data integrity and authentication," vol. 2003: IBM developerWorks, 2000.
[19] R. Morris, Thompson, K., "Password Security: A Case history," *Communications of the ACM*, vol. 22, pp. 594-597, 1979.
[20] M. Peyravian, Zunic, N., "Methods for Protecting Password Transmission," *Computers & Security*, vol. 19, pp. 466-469, 2000.
[21] M. Zviran, Haga, William, "A Comparison of Password Techniques for Multilevel Authentication Mechanisms," *The Computer Journal*, vol. 36, pp. 227-237, 1993.
[22] D. B. Baker, "Assessing Controlled Access Protection (Violet Book)," National Computer Security Center NCSC-TG-028, May 1992 1992.
[23] G. A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *The Psychological Review*, vol. 63, pp. 81-97, 1956.
[24] A. Adams, Sasse, M. A., "Users Are Not The Enemy," *Communications of the ACM*, vol. 42, pp. 41-46, 1997.
[25] R. Pond, Podd, J., Bunnell, J., Henderson, R., "Word Association Computer Passwords: The Effect of Formulation Techniques on Recall and Guessing Rates," *Computers & Security*, vol. 19, pp. 645-656, 2000.
[26] S. N. Porter, "A Password Extension for Improved Human Factors," *Computers & Security*, vol. 1, pp. 54-56, 1982.
[27] R. Shimonski, "Create effective passwords: strategies for computer systems," vol. 2003: IBM developerWorks, 2002.
[28] S. L. Smith, "Authenticating Users by Word Association," *Computers & Security*, vol. 6, pp. 464-467, 1987.
[29] D. Weirich and M. A. Sasse, "Pretty Good Persuasion: A first step towards effective password security in the real world," presented at New security paradigms, Cloudcroft, New Mexico, 2002.
[30] J. Yan, Blackwell, A., Anderson, R., Grant, A., "The Memorability and Security of Passwords Some Empirical Results," Cambridge University Computer Laboratory.
[31] M. Zviran, Haga, William, "Password Security: An Empirical Study," *Journal of Management Information Systems*, vol. 15, pp. 161-185, 1999.

[32] Microsoft Canada, "Information Overload:Canadians Have Too Many Passwords," vol. 2003: Microsoft Canada, 2000.
[33] Microsoft, "Microsoft .Net Passport Q & A," vol. 2003: Microsoft, 2003.
[34] T. Jones, "Too many secrets? Password proliferation leads to user fatigue," in *Columbia News Service - Columbia University Graduate School of Journalism*. New York, 2002.
[35] S. Swanson, "Way too many passwords, not enough protection," in *Chicago Tribune*, online edition ed. Chicago, 2003, pp. 1.
[36] I. Sommerville, *Software Engineering*, 6th ed. Essex, England: Pearson Educational Limited, 2001.
[37] J. J. Whitmore, "A method for designing secure solutions," *IBM Systems Journal*, vol. 40, pp. 747-768, 2001.
[38] Liberty Alliance Project, "Business Benefits of Federated Identity," vol. 2003: Liberty Alliance Project, 2003.
[39] E. Spafford, "Opus: Preventing weak password choices," *Computers & Security*, vol. 11, pp. 273-278, 1992.
[40] E. Spafford, "Observing Reusable Password Choices," presented at Proceedings of the 3rd Security Symposium, Usenix, 1992.
[41] D. V. Klein, "Foiling the Cracker; A Survey of, and Improvements to Unix Password Security," presented at Proceedings of the USENIX Security Workshop, 1990.
[42] A. Adams, Sasse, M. A., Lunt, P., "Making Passwords Secure and Usable," presented at People & Computers XII (Proceedings of HCI '97), 1997.
[43] R. Fagin, Moni, N., Winkler, P., "Comparing Information Without Leaking it," *Communications of the ACM*, vol. 39, pp. 77-85, 1996.
[44] R. Hauser, Janson, Philippe, Tsudik, Gene, Van Herreweghen, Els, Molva, Refik, "Robust and Secure Password and Key Exchange Method," *Journal of Computer Security*, vol. 4, pp. 97-111, 1996.
[45] D. P. Jablon, "Strong password-only authenticated key exchange," *ACM SIGCOMM Computer Communication Review*, vol. 26, pp. 5-26, 1996.
[46] D. L. Jobusch, Oldehoeft, A. E., "A Survey of Password Mechanisms:Weakness and Potential Improvements. Part 2," *Computers & Security*, vol. 8, pp. 675-689, 1989.
[47] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "The Spread of the Sapphire/Slammer Worm," vol. 2003, 2003.
[48] CERT/CC, "CERT® Advisory CA-2003-04 MS-SQL Server Worm," vol. 2003, 25 January 2003 ed: CERT/CC, 2003.