Global Information Assurance Certification Paper

Randy Herbert
GSEC 1.3
User Account Security

Abstract

User accounts are one of the most fundamental components of any information system. It is fundamental, but very critical. It is realized just HOW critical the instant that there is a failure in account security.

Very recently E-Bay had a user whose account id was stolen, the password for the account was changed and the hacker set up an auction. If the user had not been diligent about checking her account regularly, it would have been discovered too late and the hacker would have walked away with the money for the auction, leaving the real user and E-Bay to clean up the mess. [1] How this was possible will be discussed later in greater detail.

This paper will discuss various topics on securing user accounts. First, and throughout the paper, we will cover the need for a formal documented user account policy. This will lead into a more detailed discussion of some of the pieces of the account policy and account procedure. We will take a look at portions of good and bad account policies. Finally, we will revisit the E-Bay example above and see a real world example of why a secure account policy is needed.

The conclusions drawn will show that and organization is much better served from a security standpoint by having a documented policy that makes their users aware of the rules for their accounts and the potential harm that can be realized if they do not adhere to these rules.

User Account Policy

Initially, it seemed like a safe assumption that account policies were well defined for all companies or institutions that have users requesting accounts on their systems. Research revealed that nearly every site did indeed have some sort of policy; but there was a great deal of variance in the quality of the policies.

Let's start with a definition of a user account policy. According to Michele Crabb-Guel, a policy can be defined as "[The policy] outlines the requirements for requesting and maintaining an account on the systems" [2]. This is a pretty simple definition, but is sufficient. Michele included the following items in her list of components in a user account policy:

- Have user read and sign account agreement prior to account creation
- State who has the authority to approve account creation
- State if users can share accounts or have multiple accounts on one host
- State when accounts are disabled or enables
- State when accounts will be disabled due to inactivity

- State the password policy

We will discuss a couple of these items as well as these topics:

- Account approval process
- User ID construction
- Social Engineering education
- Security implications

There is more to an effective account policy than what will be discussed below. But, for this paper the focus is more on the account management policy rather than the account usage rules.

Account Agreement

A user account agreement is overlooked for many systems. Although it may not always be possible to get a signed copy of this agreement back from the user, an electronic version should be made available. This paper will not go into details on what should be part of the account agreement; that will be left up to the lawyers. The agreement should state all legal liabilities and implications of using an account on the system.

This is also where you will see most of the account usage policies. For instance, you would see information about acceptable usage on the system. A university computer lab would not want a user running a web business from their terminals.

Now, let's move onto getting the account requested and approved.

Approved Account Requestor

There should be a defined method of determining who can request an account for a user. For instance, in an educational setting, perhaps only the professor can request an account for a student. In a business setting, maybe this will be a supervisor. In a large environment, it could be that there are specific account administrators who are responsible for actually entering the account request into the system. It should be clearly stated in the policy who is responsible/authorized to request an account.

The account requestor should also be responsible for modifying accounts and deleting accounts when no longer needed. Ideally, this would be a person who deals with the users and is familiar with their needs. If a user is requesting a higher level of authorization on the system, this person should know whether or not the user has the expertise and need for the additional privileges. You don't want to give SQL privileges to a user that does not have any background with databases. The account management system should allow users to change their own personal profiles. Account administrators will only change security relevant areas of a users account.

If an account administrator is responsible for submitting an account creation or modification request, should they also be authorized to approve the request? Probably not. Let's talk a little bit about who should be responsible for approving accounts and the process by which the approval is carried out.

Approval Process

An effective account management process will have a defined approval process for new accounts and modifications to existing accounts. The process definition should include who is able to approve account requests and what portions of the account need to go through the approval process.

Generally, the account approver will be someone in an administrative position. This person should know the network and understand the different permissions that are available. The account approver should be someone that is security conscious. It is not likely that a help desk person is going to be right for this position. The help desk is an area where social engineering can promote vulnerabilities. This will be discussed later in the paper.

An administrator who is responsible for approving account creation requests should not have the authority to submit a new account request. This is simple separation of roles. If this person is also going to approve account modification requests, perhaps not every part of the request needs to go through the approval process. For instance, if a user needs to change his/her location or phone information, the approver probably does not want to be bothered with this. However, if the user is requesting additional privileges on the system, this will need to be approved by the proper authority.

User Ids

When considering user IDs, a couple of factors need to be considered. First, the format for the user ID should be mandated by the system and not by the user requesting that account. This allows a uniform secure policy to be implemented. It also makes it simple to ensure that all accounts are created uniquely.

Next, especially for a large corporation, some sort of a department designator should be used to differentiate between accounts at a quick glance. If the system is a multi-level security system, perhaps a character to denote the level that the user should be operating on should be used.

Finally, implement something that will make sense to the user, such as initials. The combination described above will allow enough flexibility but allow the user IDs to be intuitive to learn and give some insight about the account.

For instance, let's suppose you are responsible for accounts for an accounting office. You could have the following policy for User IDs:

- User IDs will be seven characters in length.
- First three characters will denote the employee's department
    - San Jose, support staff – SJS
    - San Jose, accounting staff – SJA
    - New York, support staff – NYS
    - New York, accounting staff - NYA
- The third character will be a randomly selected character
    - This could also be used to designate a security level
        - A-M = Level 1
        - N-Z = Level 2
        - 0-9 = Level 3
- The last three characters will be the user's initials. A user without a middle initial will have an "x" as the middle character.

The security indicator would make a security analyst's job a little easier when doing a scan of security logs. The above convention is also something that makes sense to the user. However, this convention will help to deter someone from quickly generating a list of user IDs after obtaining a list of the names of users on the system. To contrast this, suppose that same person had the list of names and realized that the company decided that all user ids will be generated using the formula of first name initial, followed by the first six characters of the last name. You have a list of all user ids and can start working immediately generating passwords. There are many institutions that use this exact convention or some derivation of it. [10]

The User ID method is not something that would be part of a published policy. If the IDs are being protected, they are given away as soon as you publish the methodology.

Emurl is an application that provides web based email hosting. A vulnerability was discovered in it's 2.0 version where the user's ids were generated using a simple ASCII based method and were easily stolen and generated. SecuriTeam posted the vulnerability with the following information: [13]

> After logging into a Emurl web based email account you are transferred to an URL such as:
>
> http://www.example.com/scripts/emurl/RECMAN.dll?
> TYPE=RECIEVEMAIL&USER=113100104114116111123
>
> This identifier is based solely on your account name. Therefore, if you create an account with the same name on another site, you'll end up with the very same identifier.
>
> Furthermore, this identifier can easily be determined since it is "encoded" using the ASCII value of each character of the account's name and incremented by its position.
>
> In this example, the user ID 'Pbenoit' results in the identifier:
> 113100104114116111123.
>
> p = 112 + 1 = 113

b = 98 + 2 = 100
e = 101 + 3 = 104
n = 110 + 4 = 114
o = 111 + 5 = 116
i = 105 + 6 = 111
t = 116 + 7 = 123

The following perl code could be used to quickly generate a user id based on Emurl's policy: [13]

```
print "Enter your ID: ";
$_=lc(<STDIN>); chomp;
print "Your identifier is: ";
@letters=split(//, $_);
for ($i = 0; $i < length($_); $i++) {
$mychar = ord($letters[$i])+$i+1;
if ($mychar < 100)
{ $mychar = (0).$mychar;}
print $mychar
}
```

This particular vulnerability has been corrected; however, using Emurl services, I was able to create an account at webmail.capcity.com with only a two-digit user id and a four digit common English word for a password. [14] Seems like there is still some work to do with this application and how it is used. Speaking of weak passwords, let's talk a bit about password policies.

Password Policy

Passwords are a key element in a secure account management process. They are the frontline defense for user accounts. A compromised password opens the door for extensive damage to a system. How important are password? So much that no or weak passwords made the SANS top 20 list of Internet vulnerabilities, that's pretty serious. [12]

We could define a **password policy** as the rules that users must abide by for account passwords on the information system. A defined password policy is an absolute requirement for a secure account management process. A password policy is made up of two parts: rules for selecting a password and rules for protecting your password.

Password selection is the first part of the password policy. Most, but not all, policies will dictate minimum length, types of characters required, restrictions on expiration and re-use, and password age restrictions. What gets left out many times is the strong wording necessary to enforce these policies. Policies too often use the word "should" rather than "must. How can a company enforce the "must part of the policy? Simple, don't allow passwords on the system that do not meet the minimum criteria. The following is a list of the criteria that should be checked:

- Password at least 8 characters in length
- Password must be changed every 4-6 months

- Password cannot be re-used for 5 iterations
- Password must contain upper and lower case letters
- Password must contain a number, and special character
- Password must not contain words that can be found in a dictionary
- Password must not contain personal information (names of family, friends, pets, etc., or birth date, anniversary, etc.)

This list may not be complete and can be expanded as deemed appropriate. To implement this policy, password checking can be done via custom software or with the use of the many COTS products that are available today. A few are listed at the end of this section.

The second part of the password policy is the rules that are enforced about password protection. Although it may seem like common sense would prevail, it is necessary to document the rules for account passwords. This is all about education. Some users are not aware of the dangers of writing down their password, choosing a simple password, or sharing their password. If the password policy does not state the rules, you can't expect the users to follow.

It must be stated what is expected of the user if their password is shared, stolen, or forgotten. To protect against passwords being guessed, a lockout feature should be implemented. If three failed login attempts occur, the account should be locked.

People are afraid of complex, difficult to remember passwords. Some examples of strong passwords and methods to select them will go a long way in getting people to choose strong passwords.

An example of an ill-defined, weak password policy comes from the Bloomberg School of Public Health [8]:

> Your password is the first level of protection from a would-be intruder. Once into the system, a knowledgeable user of UNIX could do considerable damage. Thus, as a member of the user community it is your responsibility to choose a good password.
>
> A good password is one that is at least 6 characters long (8 is better), is a mixture of both lower and upper case characters, and is NOT in a dictionary, a proper name, or information that can be deduced about you (like your phone number, birth date, children's names, etc.) It should not be easily guessable. For example, "SPH123" is not an ideal password for an account on JHSPH.

The only parts to this policy are simply suggestions and only cover length, case, dictionary, names, and personal information. Even if the users only followed these suggestions, this site is still leaving itself very open to password crack attacks. Password crack programs are readily available and fairly simple to use. A couple of popular ones are John the Ripper and WebCracker.

How can a system fight against these tools? There are many programs available that will check the strength of the passwords on your system. One such program, Password Defender, will check the passwords based on the strength that you set. Further, this application will at slow times of CPU usage, try to crack existing passwords on the system. [15] [16] Here are a couple of other tools that can be used for password checking:

- Npassword, http://www.utexas.edu/cc/unix/software/npasswd/
- Password Policy Enforcer 2.4, http://freedownloadscenter.com/Utilities/Access_Control_Utilities/Password_Policy_Enforcer.html

A very good password policy is provided at the SANS website. [6] This policy can be implemented in an organization with little editing. One very good area of the policy is that it covers social engineering that affects passwords. This will be discussed in further detail in the next section.

Social Engineering

The Security Focus compiled a definition of Social Engineering and boiled it down to the following: [3]

> The one thing that everyone seems to agree upon is that social engineering is generally a hacker's clever manipulation of the natural human tendency to trust. The hacker's goal is to obtain information that will allow him/her to gain unauthorized access to a valued system and the information that resides on that system.

People tend to trust others, especially when they think that they are being helpful. Social Engineering should be discussed as part of any account policy. This is simply educating the user. Systems have been compromised due to this in the past and will continue to be compromised in the future. If a new person is being granted access to an information system via a user account, they must be informed about how to handle the account. Below is a table provided by Security Focus that describes several "types" of social engineering: [4]

Area of Risk	Hacker Tactic	Combat Strategy
Phone (Help Desk)	Impersonation and persuasion	Train employees/help desk to never give out passwords or other confidential info by phone
Building entrance	Unauthorized physical access	Tight badge security, employee training, and security officers present
Office	Shoulder surfing	Don't type in passwords with anyone else present (or if you must, do it quickly!)
Phone (Help Desk)	Impersonation on help desk calls	All employees should be assigned a PIN specific to help desk support
Office	Wandering through halls looking for open offices	Require all guests to be escorted
Mail room	Insertion of forged memos	Lock & monitor mail room

Machine room/Phone closet	Attempting to gain access, remove equipment, and/or attach a protocol analyzer to grab confidential data	Keep phone closets, server rooms, etc. locked at all times and keep updated inventory on equipment
Dumpsters	Dumpster diving	Keep all trash in secured, monitored areas, shred important data, erase magnetic media
Intranet-Internet	Creation & insertion of mock software on intranet or internet to snarf passwords	Continual awareness of system and network changes, training on password use
Office	Stealing sensitive documents	Mark documents as confidential & require those documents to be locked
General-Psychological	Impersonation & persuasion	Keep employees on their toes through continued awareness and training programs

How serious is the social engineering problem? Simply take a look at one of the most famous attackers ever. J. Phillip Craiger stated: [17]

> Kevin Mitnik, the most infamous hacker in history, said that 98% of his computer break-ins were facilitate through "social engineering."

Every account policy must cover social engineering. User must be aware of the dangers. Let's now take a look a couple of example policies to see in practice what is being done.

Example policies

In doing research for user account policies, I came across some very well defined policies. There were many that covered most of the items discussed in this paper while there were some that only touched on a couple of items. Others simply did not use strong enough policies or wording to convince anyone that they are serious about securing their user accounts.

One good example is the University of Florida. [5] It is quite lengthy and will not be copied here. It has a well-defined acceptable use policy and covers many items that are related to user education. The one thing missing is a strong password policy. They would be well served to take the policy provided by SANS and implement it as part of their policy.

Here is an example of a policy that could certainly use stronger wording as well as more definition. This is the entire individual user account policy from Social Science Research Computing, the social science research and instruction facility at the University of Chicago:[7]

1. No user should use more than one (1) PC at any time.
2. No software can be installed on any PC without the prior consent of SRC staff.
3. Printing job will stay in the print queue for 4 hours after which it will be removed automatically.

4. Users are only allowed to print one (1) copy of a document. Please use a copy machine for multiple copies. Users found in violation of this policy may have their printing privileges suspended or revoked.
5. User files can be saved on the user disk(U:). Files saved elsewhere will be routinely removed. There is a quota of 30MB on the user disk. Any user exceeding this quota will be warned and given one week to meet the quota. After that week, SRC staff reserves the right to remove files from the user disk without prior notification.
6. You must log out at the end of your session. Failure to do so compromises the security of your account. You cannot leave a PC unattended for more than 15 minutes, after which the machine will log off automatically. SRC assumes no responsibility for unsaved files. SRC staff reserves the right to log you off and allow someone else to use the machine if there are no other open PCs.
7. It is the responsibility of each user to maintain secure passwords. Users should establish appropriate passwords in the first instance, change them occasionally, and not share them with others.

Let's take a look at this policy and compare it to the items that were mentioned above.

- No account agreement form
- No apparent approval process
- No information about account disabling
- Password policy (if that's what you want to call it) is non-existent
- No user education about the dangers of not protecting the account id or password

It is pretty obvious that this is not an effective user account policy. There are many more examples such as this one. This includes business, education, and retail sites. Simply having a policy is not enough to maintain security. It must be strong and enforceable. Now, for a visit back to E-Bay.

Problems with no or little policy

Let's revisit our example introduced way back in the beginning. E-Bay had a problem where a user account was stolen and a hacker was able to take control. The reason this was possible was due to a kink in the E-Bay policy. E-Bay does not enforce strong passwords and does not lock out accounts due to multiple unsuccessful login attempts. This leaves the door wide open for a hacker to crack passwords. Since there is not a strong policy on username and password generation, someone could easily create an account with a very weak username and password. In playing around with E-Bay it was discovered that the password could be set to something very simple and violate almost every part of a normal password policy.

This is a growing concern for companies such as E-Bay. Lee Curtis, a high-tech investigator noted "If they lose the confidence of their customer base, they're out of business." [1] E-Bay is well aware of this problem and is considering steps to prevent it in the future.

E-Bay is not alone in having lax password policies. Amazon.com will allow a user to create an account with a 5 character everyday word as the password. Considering these sites are storing personal information such as address, phone number, AND credit card numbers, there are certainly plenty of holes to be exploited.

Conclusion

This paper has discussed some of the aspects of a user account policy. While most companies will not have an implementation of all of these items, it should strive to educate its users about what is acceptable use of their systems.

Protecting against hackers is an ever-changing science. However, using some very basic procedures such as protecting the accounts and privileges given, implementing a strong password policy, and making users security aware will always give some layers of defense against attacks.

References:

1. E-Bay account ID theft recently
http://news.com.com/2100-1017-868278.html

2. Crabb-Guel, Michele - User Account Policy
http://www.sans.org/newlook/resources/policies/bssi3/sld018.htm

3. Social Engineering Fundamentals, Part I: Hacker Tactics
http://www.securityfocus.com/cgi-bin/infocus.pl?id=1527

4. Social Engineering Fundamentals, Part II Combat Strategies
http://online.securityfocus.com/infocus/1533

5. University of Florida CLASnet Account Policy:
http://www.clas.ufl.edu/clasnet/policy/account.html

6. SANS Password Policy
http://www.sans.org/newlook/resources/policies/Password_Policy.pdf

7. Social Science Research Computing
http://www.spc.uchicago.edu/src.cgi?clusterp

8. Bloomberg School of Public Health
http://www.jhsph.edu/is/pol_pass.html

9. *Stanford University School of Medicine,* **User Account Policy Agreement**
http://medit.stanford.edu/web/hosting/policy.html

10. University of Ottowa, First Class Groupware

http://www.edteched.uottawa.ca/FirstClass/default_e.htm

12. SANS Resources, The twenty most critical Internet security vulnerabilities
http://www.sans.org/top20.htm

13. Beyond-Security, Emurl's User ID generation mechanism cracked
http://www.securiteam.com/exploits/5DP0I201FS.html

14. Emurl Start, Webmail Capcity
http://webmail.capcity.com/emurl/

15. Beyond-Security, Password Defender - protect yourself against NT password cracking tools
http://www.securiteam.com/securityreviews/3S5Q2S0QAU.html

16. Password Defender 2.0
http://www.brd.ie/ntsecurity/

17. J. Philip Craiger, Traveling in Cyberspace: Computer Security
http://craiger.ist.unomaha.edu/pc/readings/Craiger_TIP_April_01.pdf