2006-331: PASSWORD AUDITING TOOLS

Mario Garcia, Texas A&M University-Corpus Christi

Password Auditing Tools

Abstract

A goal of computer system security is to prevent an attack, and authentication mechanisms can prevent a compromise on parts of a system. Most if not all forms of access are granted based on a single authentication scheme, and passwords are currently the most widely used authentication mechanism. Weak passwords have been cited by experts from industry, government, and academia as one of the most critical security threats to computer networks. However, various applications are available today which allow system administrators to assess the strength of their passwords in order to take the necessary precautions. The purpose of this report is to conduct a study of how well some of the more popular password auditing applications perform for Windows and UNIX operating systems.

Introduction

The three basic components of computer security are confidentiality, integrity, and availability. To ensure the integrity of a system, prevention and detection mechanisms are used to handle improper or unauthorized change. Prevention mechanisms specifically seek to maintain integrity by blocking any unauthorized attempts to access or change the data in a system [1]. Authentication, also known as origin integrity, is the binding of an identity to a subject. Thus, an authentication mechanism is used to prevent a compromise on some parts of a system. Currently most forms of access are granted based on a single authentication scheme, and passwords are the most widely used authentication mechanism [1].

Password auditing is a method of ensuring that user passwords are strong, thus strengthening the authentication mechanism used by the organization. Only the system administrator implements password auditing. This method tests the strength of user passwords by executing similar attack techniques to what a hacker might use to compromise the system. Password auditing is an important method to use for securing a system. The auditing process can help organizations to protect against password attacks that could compromise their systems. There are many attacks that could potentially be used by hackers. These attacks include dictionary attacks, hybrid attacks, pre-computed attacks, brute force attacks, mask attacks, and distributed network attacks.

A dictionary attack uses a pre-defined dictionary file to compare against the passwords. Before the passwords in the file can be used for comparison, they must first go through the same hashing method used by the system on which the passwords are stored. Another type of attack is a hybrid attack. This attack checks for dictionary words that have significant letters replaced by special characters (i.e. $unshine). A pre-computed attack is similar to a dictionary attack but slightly more efficient because it uses a dictionary list that has already been hashed so it can do a straight comparison against each password. A brute-force attack tries every combination of letters, numbers, and special characters specified for the software. A mask attack uses a pre-defined mask that is known to be part of the password. Then, it uses a dictionary attack to compare against the remaining characters in the password. This method is more efficient than a dictionary attack, but more knowledge is needed to stage the attack. This method is more

effective for password recovery than password auditing. A distributed network attack uses a group of two or more computers to stage an attack on a system, thus cracking the passwords more quickly.

To implement an auditing mechanism, the administrator must first decide what the standards are for a good password. Then, the administrator should choose a software application whose features best fit the organization's standards. When performing the audit, the administrator should choose which features to use and which external documents to use for comparison so that the standards are met. The administrator should then take measures to ensure that users with weak passwords change their passwords based on guidelines chosen. The purpose of this paper is to compare and contrast auditing software, and to give some conclusions and recommendations on good authentication procedures. In this experiment, different auditing software tools were tested. After comparing the results, it was found that Password audits tools and proactive password checkers are excellent methods for ensuring the security of a system.

Applications for Password Auditing and Recovery

Applications designed for auditing passwords can be used for password recovery, and conversely, password recovery applications may effectively be used for auditing passwords. An obviously important consideration for choosing an application is the type of operating system that is being used. The operating system will determine how passwords are stored in regards to both location and manner. Most operating systems store passwords and other user information in either a file or database and can use several encryption algorithms. Consequently, the password auditing software must be tailored to handle each operating system or else it will need to be able to be highly configurable.

Windows

Current versions of the Windows operating system use a directory database known as SAM, the Security Accounts Management Database. The hashed passwords are stored in SAM, and the SAM is further encrypted with a locally stored system key. The SysKey utility can additionally be used to secure the SAM database by allowing the SAM database encryption key to be moved off the host. Consequently, if an attacker were to obtain a copy of the SAM database, they could not extract the passwords without the key [7].

LC5

LC5 is a robust application with extensive features and a comprehensive graphical user interface. LC5 is high end professional software with versions available at various price ranges from $650 to $1,750 depending on the type of license purchased [1]. Among its many features, LC5 allows automated and schedulable password scanning, remote system scans from multiple domains, multiple assessment methods and rapid processing with pre-computed password tables and use of multiple dictionaries and international characters. Support is provided for both Windows and UNIX as well. The user interface provides real-time information on assessments and will report scores on recovered passwords from a baseline of password security and label them by strength [1].

SAMInside

SAMInside was developed by a Russian company called InsidePro primarily for password recovery. InsidePro offers a shareware version for demo purposes and a full version available for $40. SAMInside purports to be the first program in the world which can "break" SysKey protection [6]. SAMInside can conduct brute force, mask, dictionary and distributed attacks, and it provides a user friendly interface which makes it easy to choose which accounts to crack. However, because it was designed for password recovery, it does lack the rich output provided by LC5 as no report is given other than that of a password having been cracked. Therefore, it meets only minimal requirements to act as a password auditing application.

John the Ripper

John the Ripper is an open source application developed by Openwall Project and now in version 1.6 since 1998 can be found at their home page. It was originally developed for UNIX operating systems, but a version was ported for Windows as well that will run under DOS can crack AFS passwords and WinNT LM hashes [9]. Since John the Ripper will run under Windows as a DOS application, it does not have a graphical user interface and must be called via the command line. However, it does offer a wealth of command line options. John the Ripper will be described in detail in the section which follows.

UNIX

Traditional UNIX systems store the encrypted passwords and other user account information in the plaintext file */etc/passwd*. However, the */etc/passwd* file needs to be globally readable as it is used by many tools such as 'ls' to map user IDs to user names, and this poses a security risk as any user can then copy the file and run any number of password cracking programs against it [5]. This problem is alleviated by use of a shadow password system which leaves the */etc/passwd* file intact with the exception of the password field which is replaced by a special token such as "x" depending on the version of UNIX used. The encrypted passwords as well as other account information are then stored in a plain text file, named */etc/shadow* in many cases, which is only readable by the root account. [5]

Crack

Crack is an open source password cracker developed in 1996 by Alec Muffet. It is fast although not as easy to use as John the Ripper. Crack is not designed to break user passwords; it is designed to break password files. Muffet says this is a subtle but important distinction which gives credence to its use as a password auditing application [8]. Crack conducts the audit in a series of sweeps. First, it will try and use user account information from the password file which it claims is often highly successful compared to other techniques. Second, it will conduct a dictionary attack, and then further sweeps will be based on hybrid attacks depending on user configuration [8].

John the Ripper

As mentioned before, John the Ripper is an open source password cracker originally created for the UNIX operating system whose primary purpose is to detect weak UNIX passwords. It is designed to be powerful and fast, and it has been tested with Linux x86/Alpha/SPARC, FreeBSD x86, OpenBSD x86, Solaris 2.x SPARC and x86, Digital UNIX, AIX, HP-UX, and IRIX. John the Ripper supports standard and double-length DES-based, BSDI's extended DES-based, FreeBSD's (and not only) MD5-based, and OpenBSD's Blowfish-based. It packs its own highly optimized modules for different ciphertext formats and architectures [9]. Unfortunately, like Crack, it does not offer a graphical user interface and therefore must be run using the command line. It does provide reporting however at any point it is running and stores cracked password information in a separate file. It does not provide time to crack for those passwords it does discover, so the program must be closely monitored to estimate the crack time for each user.

Passwords

Although the experimentation was conducted on both UNIX and Windows operating systems, the user account information was the control of the experiment. A total of 10 accounts were created for use on each system differing in both length and strength (Table 1). The passwords for the 10 accounts were chosen from a group of four criteria that were best representative of varying degrees of password strength. The number of accounts assigned to each group was based on its anti-cracking strength. Because passwords with a lower anti-cracking strength are easier to crack, the lower the anti-cracking strength of the criteria for a group, the more accounts were used for that group. Within each of these groups of passwords, passwords were chosen with unique character lengths of at least five characters.

Table 1 Usernames and Passwords for Test Accounts

Password Basis	Username	Password
common dictionary word	john	paper
	steve	turtle
	randy	plastic
	desi	security
common dictionary word with some letter replaced with a special character	robert	h@ppy
	sally	fl#wer
	jason	sun$hine
common string of characters with appended number	grace	pemdas123
	kelly	tamu2004
random string of all possible characters	george	\sy/uR%OEl

Dictionary File

The wordlist chosen for the dictionary attacks by Crack and John the Ripper was taken from the home page of A.R.G.O.N. [2]. It is the most comprehensive wordlist found taking over 2GB of disk space and containing mote than 200 million entries. It was also used to conduct the

Windows password audit using John the Ripper. The default wordlists that came with LC5 and SAMInside were used to conduct their dictionary attacks.

Windows

The Windows applications were easy to install. Setup files were provided for both LC5 and SAMInside which automated the entire process quickly and easily. Because John the Ripper is a DOS executable it did not require any setup once the compressed file was decompressed.

LC5

To set up LC5, the option to retrieve the passwords from the local machines was selected. The other options to retrieve from a remote machine or the network did not fall into the scope of this paper. The attack options were customized by choosing all four of the available attacks. For the dictionary attack, the default dictionary list was used. For the pre-computed hash attack, the hash table was generated from the default dictionary list. For the brute-force attack, a character set of "alphabet + numbers + all symbols", and the English language was selected.

SAMInside

Using SAMInside, each attack must be staged individually. For this experiment, a brute-force attack and a mask dictionary attack were used. Other attacks available are a mask attack and a distributed attack. For the brute-force attack options the alphabet [A-Z], [a-z], special symbols, digits, and space were used. The minimum password length was 1 character, and the maximum was 12 characters. For the dictionary attack, the default dictionary list was used.

UNIX

Since they are capable of being run on many different versions of UNIX systems, John the Ripper and Crack required much configuring to install than the Windows applications. Both were run on a Dell Optiplex GX270 running Redhat Fedora 2. John the Ripper was found to be comparatively much easier to install than Crack. Once installed, their default settings were sufficient for use with both John the Ripper and Crack in regards to running the audits. The only default option not used on both applications was the specification of the wordlist that was downloaded from the internet. Therefore, running the applications "out of the box" was easier with the UNIX applications than the Windows applications once installed.

Windows results

LC5 reported the speed at which passwords were processed for a brute force attack which averaged around 4 million keys per second. LC5 reports the type of attack that it is used to crack the password giving useful information to determine if the wordlist being used was effective or not. LC5 was able to find passwords faster than the other 2 Windows applications. Since SAMInside is designed for password recovery, it did not give any indication of how long it took to crack each password, but it did report its cracking speed which averaged around five million passwords per second. It was not able to find all of the passwords before exhausting its crack

methods, but it did find a majority of them. John the Ripper similarly did not provided any information other than its cracking speed which averaged about 800,000 password combinations per second and the indication of each password that was cracked. However, it did take up very little CPU resources. The test ran for 33 days. Despite running this long, it could not crack the some passwords.

UNIX results

Crack provided the least information of all programs conducted in this experiment. It would log each password once cracked, but otherwise, the user had no indication of the speed of the audit or what the program was doing at any time. It was only able to crack one of the passwords instantaneously, and after running for 2 days did not find any others. Because Crack gave no indication of its crack speed and it was unable to crack any other dictionary word based passwords, it was terminated after 2 days. John the Ripper for UNIX ran significantly slower than when it was run in Windows at only an average of 3,400 password combinations per second. This may be explained perhaps to the greater complexity in the MD5 hash used by UNIX than the old and deprecated LANMan hash used by Windows for backwards compatibility reasons. Furthermore, UNIX passwords must be hashed with the addition of its 12-bit salt before being encrypted adding to its complexity. Table 2 presets the results.

Conclusions

When choosing auditing software, organizations should first consider what operating system they are using. The two UNIX-based applications tested were John the Ripper and Crack. The Windows-based applications tested were LC5, SAMInside, and John the Ripper. The next step to choosing auditing software is to consider the purpose for its use. For instance, John the Ripper is more geared toward password auditing while Crack is more useful for password recovery. Similarly, for the Windows-based applications, LC5 is more useful for auditing while SAMInside is more useful for recovery. The final consideration when choosing an auditing application is the budget and size of the organization. LC5 provided the best results and most comprehensive summary for the experiment. However, small companies may not be able to afford the expense of LC5. SAMInside may be much more economical for some companies. Since the UNIX-based applications are freeware, the economic benefits are irrelevant.

Pedagogical Issues

To make the auditing process more efficient and effective, it is important to consider some guidelines in choosing a strong password. For instance, passwords should be 6-8 or more characters in length. There are 17 billion possible passwords of length 6 characters, while the number of possible passwords of length 8 characters is 33 trillion [4]. So, it is actually beneficial to add the extra two characters to the password. Strong passwords should also contain random character strings including numbers as well as special symbols. Passwords should be aged so that they change frequently, and the old passwords should not be reused for some length of time. Passwords should not be based on dictionary words or login information as this makes them much easier to guess [3.] Creating strong passwords is a huge step in implementing good authentication techniques.

Table 2 Password Audit Results

	Windows				**UNIX**	
	LC5		SAMInside	John the Ripper	Crack	John the Ripper
Password	**CRACK TIME**	**METHOD**	**CRACK TIME**	**CRACK TIME**	**CRACK TIME**	**CRACK TIME**
paper	0d 0h 0m 0s	Dictionary	instant	within a few minutes	Not Found	within a few minutes
turtle	0d 0h 0m 0s	Dictionary	instant	within a few minutes	Not Found	within a few minutes
plastic	0d 0h 0m 0s	Dictionary	instant	within a few minutes	< 1 min	within a few minutes
security	0d 0h 0m 0s	Dictionary	instant	within a few minutes	Not Found	within a few minutes
h@ppy	0d 0h 1m 23s	Brute Force	1-3 days	@ 2 days	Not Found	cracked
fl#wer	0d -1h -34m -56s	Brute Force	1-3 days	@ 2 days	Not Found	cracked
sun$hine	0d 12h 58m 7s	Brute Force	Not Found	@ 2 days	Not Found	cracked
pemdas123	1d 3h 50m 30s	Brute Force	Not Found	cracked	Not Found	cracked
tamu2004	1d 2h 58m 39s	Brute Force	Not Found	cracked	Not Found	cracked
\sy/uR%OEl	Not Found		Not Found	Not Found After > 33 Days	Not Found	Not Found After > 50 Days
	> 4,000,000 k/s		@ 5,215,000 p/s	@ 802520 c/s	Unknown	@ 3400 c/s

As it was demonstrated, password auditing is a very effective method of ensuring that an organization has strong authentication procedures. There are many password auditing applications available to companies. Among all the applications that were tested, LC5 proved to be the most efficient and comprehensive. It was shown how important strong passwords are for security.

Bibliography

[1] @stake LC5. Available from http://www.atstake.com/products/lc/ (visited September 25, 2004).

[2] A.R.G.O.N. Index of /archives/wordlists. Available from http://www.theargon.com/archives/wordlists/ (visited October 4, 2004).

[3] Bishop, M, *Computer Security Art and Science*, Addison-Wesley, 2003.

[4] Davis, E. A. Securing UNIX passwords. Available from http://www.nas.nasa.gov/Groups/Security/epasswd/article.html (visited October 28, 2004).

[5] Frampton, S. Linux Password & Shadow File Formats. Available from http://www.tldp.org/LDP/lame/LAME/linux-admin-made-easy/shadow-file-formats.html (visited October 4, 2004).

[6] InsidePro – Passwords recovery and encryption. Available from http://www.insidepro.com/eng/saminside.shtml (visited September 25, 2004).

[7] Microsoft Knowledge Base Article – 310105. How to use the SysKey utility to secure the Windows Security Accounts Manager database. Available from http://support.microsoft.com/default.aspx?kbid=310105 (visited October 5, 2004).

[8] Muffett, Alec. Index of alecm. Available from http://www.crypticide.com/users/alecm/ (visited September 25, 2004).

[9] John the Ripper password cracker. Available from http://www.openwall.com/john/ (visited September 25, 2004).