Security Threats to Microsoft Windows XP

Prof (Dr) P K Suri1[1] , Manoj Wadhwa[2] and Sachin Kumar[3]

[1] Professor, Dept of Computer Science and Applications Kurukshetra University, Kurukshetra, Haryana ,India
[2] Assistant Professor and Head, Dept of Computer Science and Engineering, Shri Krishan Institute of Engineering and Technology, Kurukshetra , Haryana, India
[3] Infosec Consultant, AKS IT Services Noida India

Summary
Privacy is the birth-right of a computer user, and it should not be a privilege to only a chosen few. This paper presents security loopholes in Windows XP User Management System and dissection of the SAM file used for storing the passwords using hexa editor followed by an experiment of invading in to some account of Windows XP without knowing the passwords and retrieving back the users profile and desktop features and provide some guidelines for securing the user Security features.

Key words:
SAM, Windows XP

Notation:

SAM	System Account Manager
LM	Lan Manager
NT	New Technology

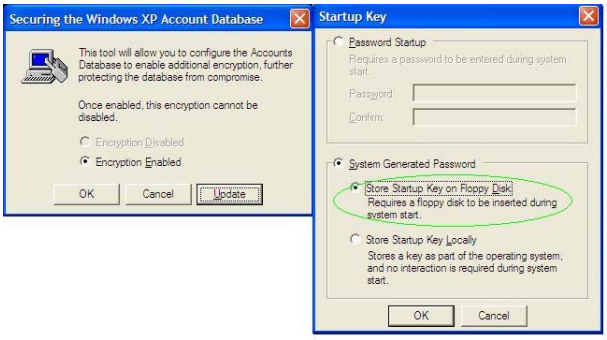
1. Introduction

Everyone in this world today needs some privacy and security. Being the reason, it has become important feature of every operating system, take Linux or Novell or any Microsoft's operating system. Everybody is trying to embed latest security. So it is done in Microsoft windows XP [4]. It gives you a very good feature to create user accounts and assign password to each user account. For security reasons the password are secured using hashing which is stored in SAM file, so that it may not be available to any unknown user so that the normal user can't access the passwords of other users.
Path of SAM File C:\Windows\system32\config
The SAM file appears to be fairly secure. But if physical access to the machine is achieved, it is not so secure. Microsoft has admitted this. If the SAM file is locked, it is not possible to delete/copy/move/rename it within windows via explorer and access to SAM file is also restricted if it is not in the administrator group [5].
The structure of a SAM [13] file is then described, with a byte-by-byte analysis using Hexa Editor, so that its various features and the storage and security methods employed are made apparent shown in Appendix A.

Manuscript received August 5, 2007.
Manuscript revised August 20, 2007.

2. Threats to Microsoft Windows XP

In this paper efforts have been made to show through the experiments, Windows XP has security loopholes and can easily invade into any user accounts of windows XP even if syskey[13] is installed and windows XP can be booted without permission.



2.1 Intrusion into the user accounts of Microsoft windows XP

In this case only formal tool that is required to do this is just a bootable diskette [2]. The user account details are stored in a SAM file in the system32 folder of the windows operating system. It is not known to the programmers that the windows operating systems have left a copy of SAM file in another folder of the windows folder, which doesn't have any user accounts and passwords, except the administrator account with a blank password, which can be used to hack into the system.
The process includes replacing the active SAM file with the backup copy.
Some security features included into the SAM file of any system doesn't allow to simply copy the SAM file from any other system and copying it into the system32, as this can result in a corrupted, unbootable system shown in Appendix B.

2.2 Retrieving Back the User profile and Desktop
In this case Windows XP is to be booted without permission, but subsequently the original user profiles can be restored, which would make the detection of intrusion next to impossible. It is another knack of Windows XP that the whole accounts [3] which are deleted, can be completely restored, with their desktop and folder settings intact (unless the files have been messed around with), and the password set to blank

2.3 Intrusion into the user accounts of Microsoft Windows XP even if the syskey is installed.
In this case you can invade a system even if syskey has been used. Microsoft claims syskey to be a very strong security feature but it also falls flat by replacing some of the backup files. As you already know the syskey uses a checksum of four files present in C:\windows\system32\config
so backing up all these files will break the syskey and the computer will become accessible.

3. Measures to Secure SAM File

SAM file has many security threats so there are some guidelines proposed to secure SAM file.

3.1 Use of hashes
Instead of storing your user account password [9] in clear-text, Windows generates and stores user account passwords by using two different password representations, generally known as hashes [11]. When we set or change the password for a user account to a password that contains fewer than 15 characters, Windows generates both a LM hash and a Windows NT hash of the password. These hashes are stored in the local SAM File. The LM hash is relatively weak compared to the NT hash, and it is therefore prone to fast brute force attack. Therefore, we can make Windows stores the stronger NT [13] hash of our password.

3.2 Use of Hardware Locks
Put up the Hardware Locks on cabinet. We know that SAM file is not possible to delete/copy/move/rename within windows via explorer. So we can use hardware locks on cabinet to restrict unauthorized access from using the hardware.

3.3 Use of Bios Passwords
Using Bios Passwords can also be used for securing the system to a good extent. Using supervisor password, we can restrict unauthenticated entry to change the boot sequence. Booting from hard disk is the best booting priority considering the protection of SAM file.

4. Results and Discussion

During the study and experimentation with Windows XP, it has been found that NT hashes passwords are good option for security but it is breakable. Bios passwords may be better option at some extent because it is also breakable by some hacking tools. So hardware locks are best option from unauthorized access to Windows XP

5. Conclusion

It is found that windows XP is not secure from security threats. Hardware locks, and bios setup are some of the ways to add on the security layer to minimize the physical exploitation of the operating system. Bios protection can also be broken by shorting the 1&3 pins of jumper of motherboard or by removing the batteries and reinserting them. So using bios setup with the feature of hardware lock is the best way to secure one's personal computer from any security threat.

6. References

[1] Ankit Fadia,," Unoffical Guide to Ethical Hacking ", Course Technology PTR 2002.
[2] Bill, David Pollino, Himanshu Dwivedi ,Tony Bradley," Hacker's Challenge 3" , Mcgraw Hill Osborne Media 2006.
[3] Charlie Russel,Sharon Crawford, "Microsoft Windows XP Professional Resource Kit",3/E Microsoft Press USA.
[4] David A Krap, "Fixing Windows XP Annoyances", O'Reilly Media Inc. 2006
[5] David Pogue, L.J Zacker "Windows XP Pro: The Missing Manual", O'Reilly Media Inc. 2004.
[6] Eric Cole, Ronald L. Krutz, James Conley "Network Security Bible", John Wiley & Sons 2005.
[7] Justin Clarke, Nitesh Dhanjani ,"Network Security Tools" O'Reilly Media Inc. 2005.
[8] Lawrie Brown , William Stalling, "Computer Security: Principles and Practice" PH-07 NewYork
[9] Rick Lehtinen, G.T Gangemi, Deborah, "Computer Security Basics", O'Reilly media Inc. 2006.
[10] Stuart Mcclure, Joel Scambray, George Kurtz, "Hacking Exposed", McGraw Hill Osborne Media 2005.
[11] William Stallings, "Network Security Essentials: Application and Standards", Upper Saddle River, NJ: Prentice Hall, 2000.
[12] http://www.beginningtoseethelight.org/ntsecurity/
[13] www.microsoft.com Article ID: 299656
[14] www.microsoft.com Article ID: 310105

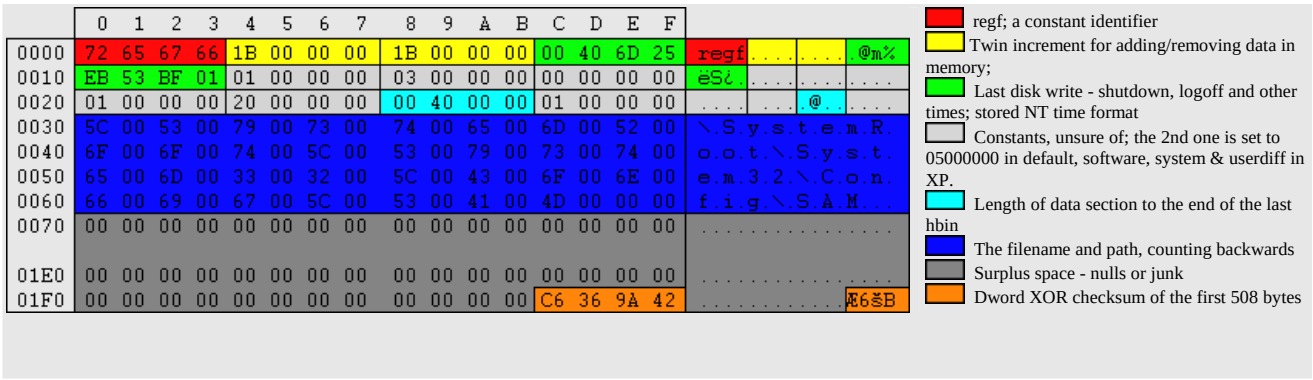
7. Author's Profile

Dr. P.K. Suri received his Ph.D. degree from Faculty of Engineering,Kurukshetra University, Kurukshetra, India and master's degree from Indian Institute of Technology, Roorkee (formerly known as Roorkee University), India. He is working as Professor in the Department of Computer Science and Applications,Kurukshetra University,Kurukshetra - 136119 (Haryana), India since Oct. 1993. He has earlier worked as Reader, Computer Sc. & Applications, at Bhopal University, Bhopal from 1985-90. He has supervised five Ph.D.'s in Computer Science and thirteen students are working under his supervision. He has more than 100 publications in International / National Journals and Conferences. He is recipient of 'THE GEORGE OOMAN MEMORIAL PRIZE' for the year 1991-92 and a RESEARCH AWARD –"The Certificate of Merit – 2000" for the paper entitled ESMD – An Expert System for Medical Diagnosis from INSTITUTION OF ENGINEERS, INDIA. His teaching and research activities include Simulation and Modeling, Software Risk Management, Software Reliability, Software testing & Software Engineering processes , Temporal Databases, Ad hoc Networks, Grid Computing , and Biomechanics.

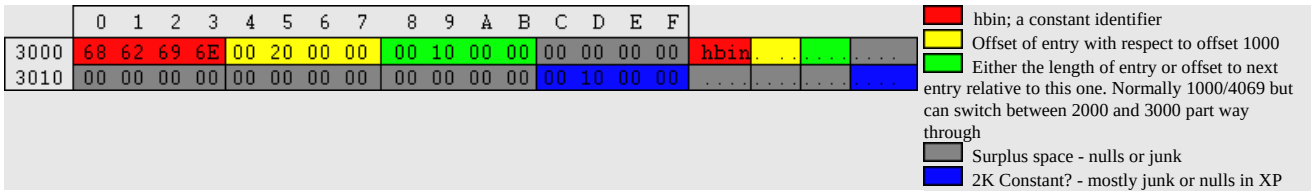
Manoj Wadhwa received M.Tech in ComputerScience and Engineering from Kurukshetra University Kurukshetra India and pursuing Ph.D from Kurukshetra University, Kurukshetra- Haryana (India). Presently, he is working as Assistant Professor and Head of Computer Science and Engineering Department in Shri Krishan Institute of Engineering and Technology, Kurukshetra. He possesses more than ten years experience of Teaching, Research, and Industry. His areas of interest include Software Engineering, Simulation and Modeling and Operating Systems.

Sachin Kumar received B.Tech in Electronics and Communication Engineering from MD University Rohtak- Haryana (India) and Presently working as Senior InfoSec consultant working with AKS Information Technology Services providing customized Security solutions in the Information Security domain. He has executed projects in Information Risk Management, External Penetration Testing, Web Application Security and Network Security. He was also involved in designing and conducting various information security workshops for various sectors like Corporate, Government, Educational institutions.

Appendix A



The hbin entry:



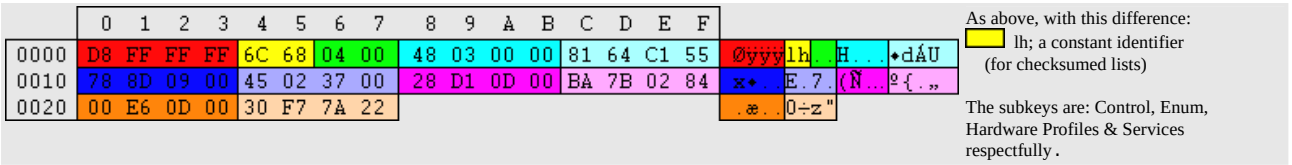
The registry appears to be made up of 7 different types of entries:
All offsets are relative to 1000, xx denotes no constant identifier

01. nk = (sub)keys (links to the following 4 types)

02. lf/lh = Subkey list
03. xx = Value list (links to type no. 6)
04. sk = Permissions
05. xx = Class information (regedt32 input on key creation)

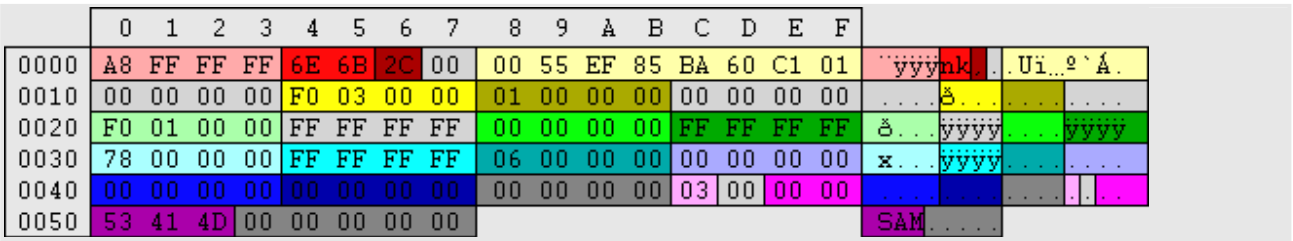
06. vk = Value (links to type no. 7 though data can be within the value)
07. xx = Data

The nk entry:



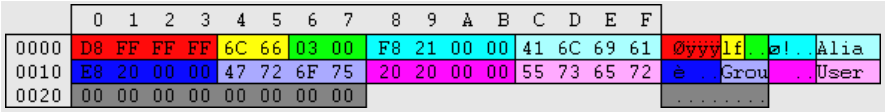
Calculating the checksum: Control = 43,6F,6E,74,72,6F,6C -> 43,4F,4E,54,52,4F,4C (CONTROL)
Use calc.exe, view = Scientific, length = Dword
43 + 4F = 92 + (43 x 24) = 9FE
9FE + 4E = A4C + (9FE x 24) = 17204
17204 + 54 = 17258 + (17204 x 24) = 357AE8
357AE8 + 52 = 357B3A + (357AE8 x 24) = 7BAC3DA
7BAC3DA + 4F = 7BAC429 + (7BAC3DA x 24) = 1DFE4ED1
1DFE4ED1 + 4C = 1DFE4F1D + (1DFE4ED1 x 24) = 55C16481 -> 55,C1,64,81 -> 81,64,C1,55

Appendix B



Entry length, 100000000 - reversed bytes
nk; a constant identifier
Keytype: 2C = Rootkey. 20 = Subkey
NT date format, set at key creation and modified the key is renamed or if values within are added or changed. Does not change for any subkey changes
Number of subkeys,if none; filled with 00,00,00,00
Subkey list(lf) offset, if none; filled with FF,FF,FF,FF
Number of values,if none; filled with 00,00,00,00
Values list offset, if none; filled with FF,FF,FF,FF
Audit and permissions(sk) offset
Class entry offset, if none; filled with FF,FF,FF,FF
The maximum subkey's name length x 2, if none; filled with 00,00,00,00.
The maximum subkey's class size, if none; filled with 00,00,00,00
The maximum value's name length, if none; filled with 00,00,00,00
The maximum value's data size, if none; filled with 00,00,00,00
Surplus space - nulls or junk
Keyname length - NB: If the length is set to 1 more than the "real" length and a null suffixed to the end of the name, the key will be handled similarly to these subkeys.
Class length
Keyname; stored in ASCII format.

Figure1. Byte wise description of SAM file



Entry length, 100000000 - reversed bytes
lk; a constant identifier
No. of subkeys
Offsets to subkeys, followed by the first
four characters of the subkey name
Surplus space - nulls or junk

Figure2. SubkeyList