WINDOWS ENUMERATION: USER2SID AND SID2USER

INTRODUCTION TO THE TOOLS

User2sid and Sid2user are two small utilities for Windows NT, created by Evgenii Rudny, that allow the administrator to query the SAM to find out a SID value for a given account name and *vice versa*. User2sid.exe can retrieve a SID from the SAM (Security Accounts Manager) from the local or a remote machine and Sid2user.exe can then be used to retrieve the names of all the user accounts and more. These utilities do not exploit a bug but call the functions; LookupAccountName and LookupAccountSid respectively. These tools can be called against a remote machine without providing logon credentials except those needed for a null session connection. These tools rely on the ability to create a null session in order to work.

NULL SESSION BACKGROUND

Null sessions allow an anonymous attacker to extract a great deal of information about a system--most importantly, user account names. They are dangerous because they allow attackers to enumerate juicy user data remotely across the LAN or internet. Windows NT, 2000 and even Server 2003 domain controllers are susceptible to enumeration using null sessions. There is a lot more information available on null sessions and SMB enumeration. The key point to take away on null sessions and enumeration is that you can obtain account names to use with dictionary attacks and other information like last logon, privileges, and when and if the user's password expires. It even gives you the logon hours so we aren't knocking on the door when the user should be asleep and not able to log in.

Ideally people block UDP 137 & 138, TCP 139, and TCP 445 at the firewall and that will not allow null session from outside your network but you are still vulnerable to internal attackers or if the attacker finds a way through the firewall (source port spoofing or application exploits). But you will find many machines and networks that do not block 139 to the internet.

WINDOWS SECURITY IDENTIFIER (SID) BACKGROUND

SID is short for security identifier, a security feature of the Windows NT, 2000, XP, 2003 operating systems. The SID is a unique name (alphanumeric character string) that is used to identify an object, such as a user or a group of users in a network of NT/2000/XP/2003 systems.

Windows grants or denies access and privileges to resources based on ACLs, which use SIDs to uniquely identify users and their group memberships. When a user requests access to a resource, the user's SID is checked by the ACL to determine if that user is allowed to perform that action or if that user is part of a group that is allowed to perform that action.

All SIDs are unique within a given system and are issued by what is known as an "Authority" such as a domain. There are five authorities:

SECURITY_NULL_SID_AUTHORITY

There is a universal Well-Known SID S-1-0-0 that represents a group with no members and is generally used when the SID of an object is not known. A universal well-known SID is a SID that is common to all machines. That is, the value of the Null_SID is the same on my machine as it is on yours.

SECURITY_WORLD_SID_AUTHORITY

This authority is responsible for the Everyone group. The well-known SID of this group is S-1-1-0

SECURITY_LOCAL_SID_AUTHORITY

Responsible for Local issues. Users with the right to Log on locally will have membership of the group SID S-1-2-0.

SECURITY_CREATOR_SID_AUTHORITY

There are two group well-known SIDs issued by this authority namely Creator Owner ID (S-1-3-0) and Creator Group (S-1-3-1)

SECURITY_NT_AUTHORITY

This is, as far as this document is concerned anyway, the most important SID issuing Authority. This will dish out the SIDs for all user accounts, default global (domain) groups, default local groups, as well as both non-default local and global groups. It must be noted that all of these share the same sub-authority except the default local groups and the special internal objects

Internal
DIALUP S-1-5-1
Network S-1-5-2
BATCH S-1-5-3

Interactive S-1-5-4
SERVICE S-1-5-6
ANONYMOUS LOGON S-1-5-7
SERVER LOGON S-1-5-9
Authenticated Users S-1-5-11
SYSTEM S-1-5-18
BUILTIN S-1-5-32

Local groups

Administrators S-1-5-32-544
Users S-1-5-32-545
Guests S-1-5-32-546
Account Operators S-1-5-32-548
Server Operators S-1-5-32-549
Print Operators S-1-5-32-550
Backup Operators S-1-5-32-551
Replicator S-1-5-32-552

All the SIDs below are relative to the domain. All other SIDs (i.e. those listed above) are universal and are the same on every machine.

Default Global groups (SidTypeGroup)

Domain Admins S-1-5-21-<number>-<number>-<number>-512
Domain Users S-1-5-21-<number>-<number>-<number>-513
Domain Guest S-1-5-21-<number>-<number>-<number>-514

Non-Default Global Groups (SidTypeAlias)

Example S-1-5-21-<number>-<number>-<number>-n=> 1000

Non-Default Local Groups (SidTypeAlias)

Example S-1-5-21-<number>-<number>-<number>-n=> 1000

Default Accounts (SidTypeUser)

Administrator S-1-5-21-<number>-<number>-<number>-500
Guest S-1-5-21-<number>-<number>-<number>-501

Non-Default User Accounts (SidTypeUser)

jsmith S-1-5-21-<number>-<number>-<number>-n=> 1000

Any group or user that is not created by default will have a RID of 1000 or greater. A RID is a Registered ID. This is the last portion of the SID. Once a RID has been issued it will never be used again even if the user and user account are deleted.

USING THE TOOLS

The readme text for user2sid and sid2user says the following:

--------snip---------

User2sid is a command line interface to a WIN32 function LookupAccountName.
Usage: `user2sid [\\computer_name] account_name`
Where `computer_name` is optional. By default, the search starts at a local Windows NT computer.
Sid2user is a command line interface to a WIN32 function LookupSidName.
Usage: `sid2user [\\computer_name] authority subauthority1 ...`
Where `computer_name` is optional. By default, the search starts at a local Windows NT computer. For example,

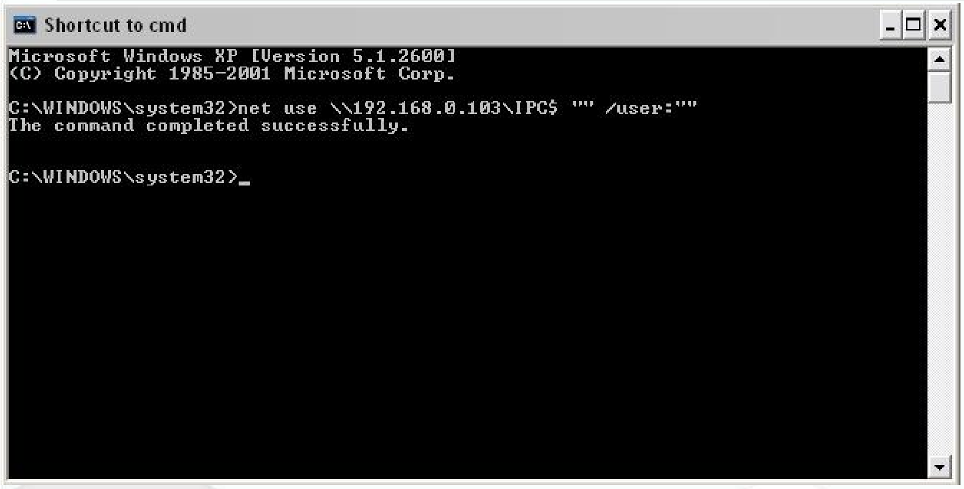
```
sid2user 5 32 544
```

--------snip---------

Basically you need to know a computer name/IP and an account name to get the SID. So I decided to run it against a Windows 2003 Domain Controller that I share a network with. Note that 2003 domain controllers allow null sessions where 2003 member servers do not.

The first thing you have to do is set up a null session. A null session connects to the IPC$ share on the remote machine. You can do this by issuing:

```
net use \\remote.compter.com\ipc$ "" /user:""
```

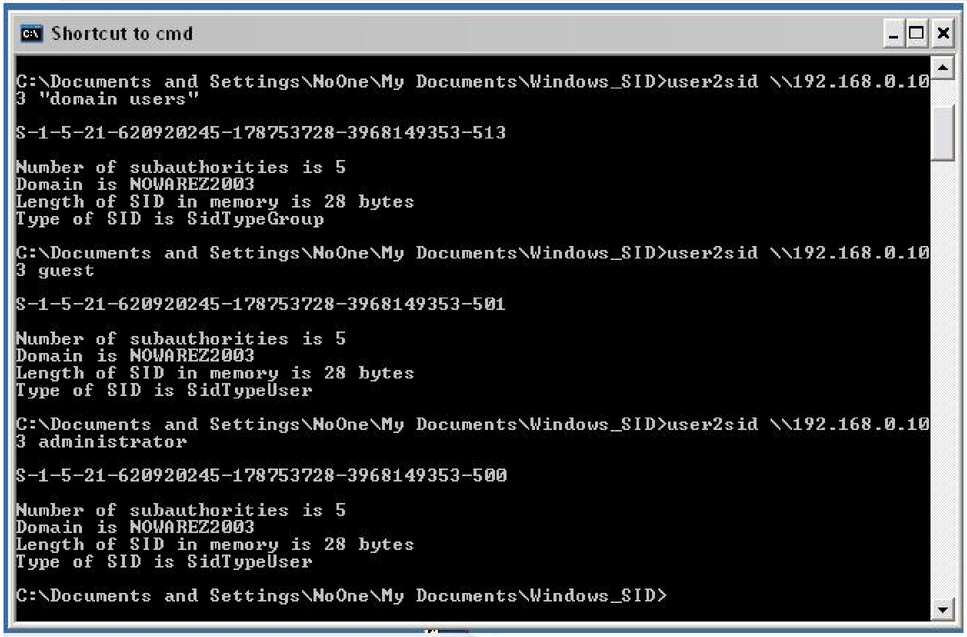


```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>net use \\192.168.0.103\IPC$ "" /user:""
The command completed successfully.


C:\WINDOWS\system32>_
```

Figure 1. Setting up the null session

To be able to suck down the user list from the remote machine we need to know the value of a SID issued by the SECURITY_NT_AUTHORITY. We will use user2sid.exe for getting such a value. Because the SIDs of the Global groups are issued by the SECURITY_NT_AUTHORITY, renaming the default accounts (Administrator and Guest) to something else will not help (as far as this attack is concerned anyway.) Groups like "Domain Admins" cannot be renamed so the attack will be launched providing "Domain Admins" as the lpAccountName placeholder. Note that this group only exists on domain controllers. For workstations or member servers the guest or administrator account will be tried and will fail if they have been renamed. Note #1: If this happens use UserInfo or Userdump. Note #2: Usually people don’t rename the guest account, they only disable it. Even if it is disabled UserInfo/UserDump will work. Note #3: If they have renamed both of the accounts use nbtdump or nbtstat to try to enumerate some usernames.

The first account we will try to enumerate is “domain users”



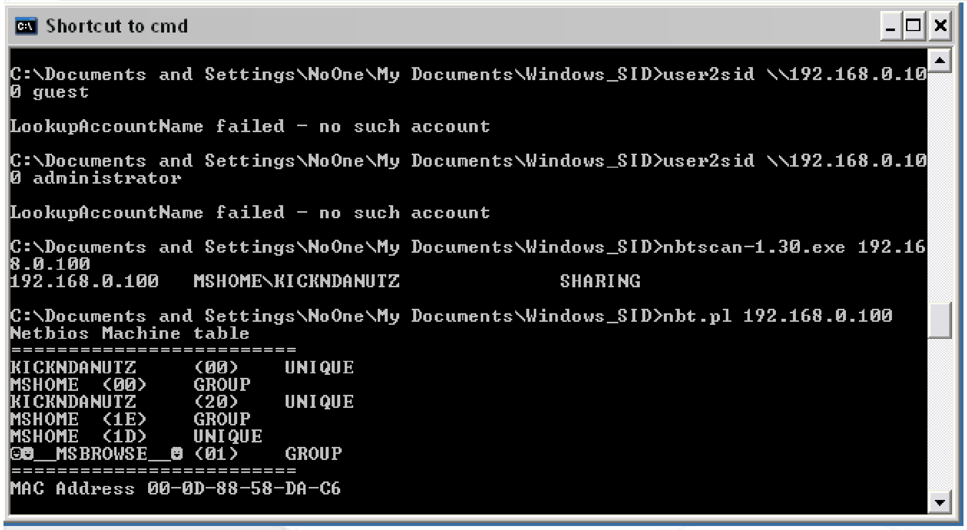
```
C:\Documents and Settings\NoOne\My Documents\Windows_SID>user2sid \\192.168.0.10
3 "domain users"

S-1-5-21-620920245-178753728-3968149353-513

Number of subauthorities is 5
Domain is NOWAREZ2003
Length of SID in memory is 28 bytes
Type of SID is SidTypeGroup

C:\Documents and Settings\NoOne\My Documents\Windows_SID>user2sid \\192.168.0.10
3 guest

S-1-5-21-620920245-178753728-3968149353-501

Number of subauthorities is 5
Domain is NOWAREZ2003
Length of SID in memory is 28 bytes
Type of SID is SidTypeUser

C:\Documents and Settings\NoOne\My Documents\Windows_SID>user2sid \\192.168.0.10
3 administrator

S-1-5-21-620920245-178753728-3968149353-500

Number of subauthorities is 5
Domain is NOWAREZ2003
Length of SID in memory is 28 bytes
Type of SID is SidTypeUser

C:\Documents and Settings\NoOne\My Documents\Windows_SID>
```

Figure 2. User2sid on “domain users”

Now the SID always starts with an **S,** and its components are separated with hyphens. The next value is the revision number. The second number is the identifier authority value (supposedly it is always 5 for windows server 2003 but it was also 5 for my windows XP box). Then, four subauthority values (in this case 21-620920245-178753728-3968149353). Lastly, a Relative Identifier (RID) which is 501 or guest in this case.

S-1-5-21-620920245-178753728-3968149353-501

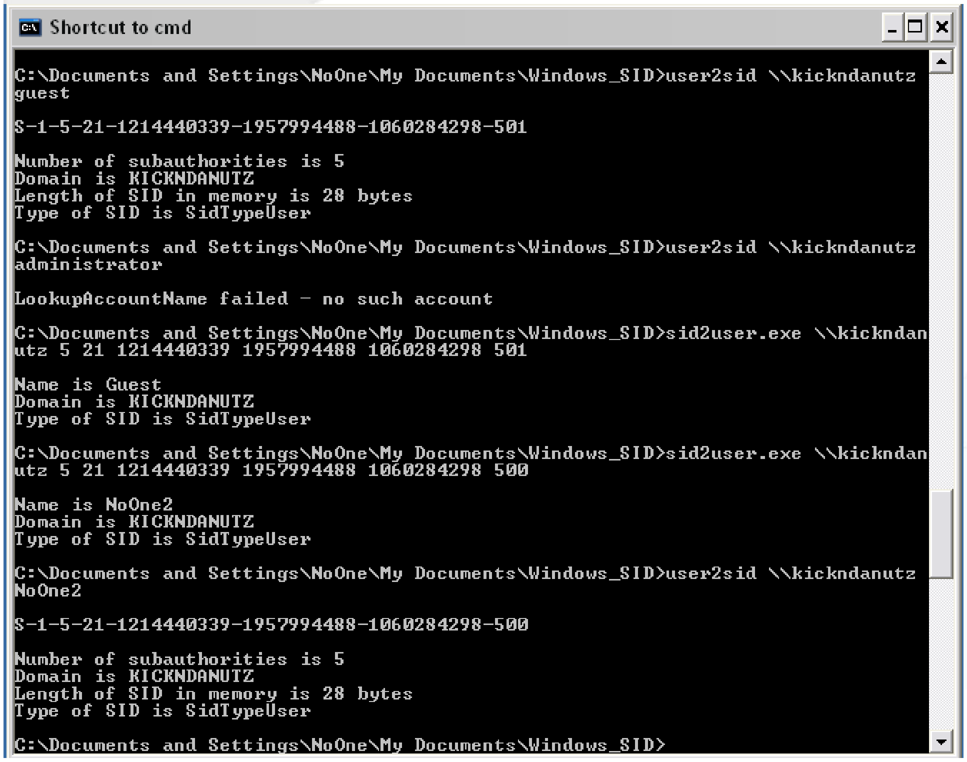
The SID will have different values depending on the OS and each subauthority values will be unique within a domain. What wont change is the RID. An SID with a RID of 500 is always the true administrator account, RID 501 is the guest account. User accounts start with 1001 (example 1003 is the third user account made in that domain). Renaming an account’s friendly name does nothing to its SID, so the account can always be identified, in fact it is important for it to **always** be able to be identified for tokens and authentication and what not within the domain. What that means to us is that even if you rename the administrator account which is part of most baselining and lockdown procedures, an attacker should be able to see who is an administrator by searching for an SID with a RID of 500. Let’s check it out on an XP box that has had the administrator account renamed.



```
C:\Documents and Settings\NoOne\My Documents\Windows_SID>user2sid \\192.168.0.10
0 guest

LookupAccountName failed - no such account

C:\Documents and Settings\NoOne\My Documents\Windows_SID>user2sid \\192.168.0.10
0 administrator

LookupAccountName failed - no such account

C:\Documents and Settings\NoOne\My Documents\Windows_SID>nbtscan-1.30.exe 192.16
8.0.100
192.168.0.100   MSHOME\KICKNDANUTZ              SHARING

C:\Documents and Settings\NoOne\My Documents\Windows_SID>nbt.pl 192.168.0.100
Netbios Machine table
=========================
KICKNDANUTZ     <00>    UNIQUE
MSHOME  <00>    GROUP
KICKNDANUTZ     <20>    UNIQUE
MSHOME  <1E>    GROUP
MSHOME  <1D>    UNIQUE
☺☻__MSBROWSE__☻ <01>    GROUP
=========================
MAC Address 00-0D-88-58-DA-C6
```

Figure 3. Locating the computer name

As you can see, user2sid is not finding the administrator or guest account. I then run nbtscan and nbt.pl to find the computer name of the remote machine. If we have already connected via null session we can talk to the computer with its computer name.



```
C:\Documents and Settings\NoOne\My Documents\Windows_SID>user2sid \\kickndanutz
guest

S-1-5-21-1214440339-1957994488-1060284298-501

Number of subauthorities is 5
Domain is KICKNDANUTZ
Length of SID in memory is 28 bytes
Type of SID is SidTypeUser

C:\Documents and Settings\NoOne\My Documents\Windows_SID>user2sid \\kickndanutz
administrator

LookupAccountName failed - no such account

C:\Documents and Settings\NoOne\My Documents\Windows_SID>sid2user.exe \\kickndan
utz 5 21 1214440339 1957994488 1060284298 501

Name is Guest
Domain is KICKNDANUTZ
Type of SID is SidTypeUser

C:\Documents and Settings\NoOne\My Documents\Windows_SID>sid2user.exe \\kickndan
utz 5 21 1214440339 1957994488 1060284298 500

Name is NoOne2
Domain is KICKNDANUTZ
Type of SID is SidTypeUser

C:\Documents and Settings\NoOne\My Documents\Windows_SID>user2sid \\kickndanutz
NoOne2

S-1-5-21-1214440339-1957994488-1060284298-500

Number of subauthorities is 5
Domain is KICKNDANUTZ
Length of SID in memory is 28 bytes
Type of SID is SidTypeUser

C:\Documents and Settings\NoOne\My Documents\Windows_SID>
```

Figure 4. Using Sid2user to locate the renamed administrator account.

So to find the true administrator account we tag 500 to the end of the SIDs we have already enumerated: S-1-5-21-1214440339-1957994488-1060284298-500

I also noticed that XP SP2 has disabled these tools. You can still connect via null session, but user2sid and sid2user aren't working. But they were working fine against the 2003 domain controller.

To undo the null session you can do **net use \\ServerIP /delete**

Ok, if you are like me I was left sitting there going wow that was neat but what's really the point of all that. Well, I'll tell you. First, User2SID/SID2User will help you locate that renamed administrator account. UserInfo/Userdump will fail if the account name is not there. The ability to find that renamed administrator account is probably the most important use of this tool. If you can pull off the SID of just one domain user using user2sid \\ServerIP "domain users" where domain users is any user account, you can put in your RID of 500 and find that renamed account. From there you can use tools like UserInfo/UserDump to enumerate the other users on the computer or domain. These tools will give you tons of user information like: User accounts, groups, policies, services, and other information are all enumerate-able via the "Null" user. You can also (usually) obtain these specific information fields:

Account Username
Account Lockout.
Account Disabled.
User cannot change password.
Password never expires.
Smartcard required for interactive logon (Win2k).
Account is trusted for delegation (Win2k).
Account is sensitive and cannot be delegated (Win2k).
All Dates, as well as Logon Hours, are at the controller, in GMT.
Any comments left by the admin.

These details can easily be enough to launch social engineering attacks or give the attacker key bits of information to more effectively coordinate his attacks.

Updated Note: If the server allows anonymous connections usually through file & printer sharing you can connect and enumerate this information using linux samba tools like smbclient and rpcclient.

I plan on covering this in another paper.