By abusing session specific Base Named Objects I create conditions where a folder resolves to one path when accessed in my security context and another for everybody else.

Since the data sharing service establishes whether a requested file copy operation is allowed by impersonating the caller, I can make the destination benign while impersonation is in effect, then when the actual copy operation is executed, without impersonation, the destination becomes a path where writing is not permitted by the caller.

Since the copy operation is executed in system context the file copy succeeds, even though the operation would not be permitted if executed by the caller.

I have named the user dependent destination folder a "trapfolder", such a folder is created by using the "fall through" property of certain NT object folders.

I create an NT object directory named GLOBALROOT in the user specific \??\ path ( like \Sessions\0\DosDevices\00000000-00065c77\GLOBALROOT ).

Accessing the path \??\GLOBALROOT will resolve to that directory for my user, but for other users it will "fall through" the user specific directory and resolve to shadow directory \GLOBAL??\GLOBALROOT .

To exploit this property I need to make a path that is controllable and valid for my user and others when the fall through mechanism is used.

By creating an additional folder named "RPC Control" in \??\GLOBALROOT\ , the path \??\GLOBALROOT\RPC Control resolves to the new directory when resolved by my user and \GLOBAL??\GLOBALROOT\RPC Control when resolved by everybody else.

In the two paths I create a symbolic link targetting the folder I want the trapfolder to resolve to when accessed by both possible identities.

Finally I make a junction folder targetting \??\GLOBALROOT\RPC Control\symlink that is the "trapfolder"

By using the trapfolder as the destination folder when requesting the data sharing service to copy a file, conditions are created where the folder used while verifying permissions is different then the actual folder used when copying the file.

The symbolic link the junction folder resolves to,in the session context of my user,( \Sessions\0\DosDevices\00000000-00065c77\GLOBALROOT\RPC Control\symlink ) is targeted at a random writeable(by me) folder.

The symbolic link the junction folder resolves to by everybody else( \Rpc Control\symlink ) is pointed at C:\windows\system32 .

Now when I request the data sharing service to copy a file into the junction folder the verification of permissions succeeds and when the file copy operation is executed the file will get copied into C:\windows\system32.

I have attached a proof of concept binary that will use this vulnerability to spawn a command prompt as SYSTEM, when executed in none privileged context.