

# Utilizing Domain Credentials to Enhance Nessus Scans

## Security Assurance Group White Paper

Ty Gast, Senior Security Engineer (tgast@securityassurancegroup.com)

### Overview

Nessus is often used to perform network-based assessments of Windows® domain computer systems. However, giving Nessus a little insider information can result in more thorough and accurate scans, and can allow “local” registry security checks to be conducted using a “remote” network scan. Specifically, Nessus can use a pre-configured domain username and password to access system registry settings that would not be accessible without the required credentials.

By default, remote registry access is typically only accessible to domain administrators. It is possible to create an account on a domain, give it domain administrator privileges, and configure Nessus to use that account when performing scans. However, this presents problems for both domain security and the scan results. It is undesirable to have more administrator accounts than absolutely necessary, so creating additional admin accounts just for the purpose of running scans may be unacceptable. During times when no scans are being conducted, individuals may attempt to use the domain admin account for purposes other than what was intended. While it is possible to disable accounts when not in use and enable them only when running scans, this can become burdensome from a management perspective.

Another shortcoming to using an account with full domain admin credentials is that scan results may not accurately reflect the true security posture of the target environment. For example, it may be difficult to detect improperly configured file shares on various domain servers because the Nessus scan using full domain admin privileges will have much more access than what a normal account would have. Domain admin accounts will typically be able to

access most (if not all) of the shares in a domain, and this will be reflected in the Nessus results.

One approach to solving these problems is to create a domain account that has minimal domain privileges while still having enough access for Nessus to be able to perform remote registry checks. This paper explains the steps required to create an account that has the minimal amount of privileges while still providing Nessus with the access it needs to perform its scanning. It also covers the steps to configure Nessus to use the account information.

### Account Creation and Configuration

All of the following steps need to be performed using a domain administrator account. First, a new global group called “Nessus Test Accounts” should be created on the domain. Nessus will be able to use any account members of this group for remote registry access. Next, a new domain account should be created called “nessustest”. This account should not be a part of any other domain group other than the newly created “Nessus Test Accounts” group.

For Windows NT 4.0 systems and later, remote access to the registry is regulated by the permission on a single key within the registry. In other words, the remote access permissions mirror those that are present on the key. The key to be modified is:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg

All Windows systems in the domain must have a change made to the permission of this local registry key to allow the newly created Nessus accounts to have remote access. For Windows NT 4.0 systems, this key may not exist and will need to be created before making the necessary permission changes. This is done as follows:



1. Use the registry editor and go to the key  
"HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control"
2. On the "Edit" menu click "Add Key" and enter: "SecurePipeServers" for the key name and "REG\_SZ" for the class
3. Go into the new key "SecurePipeServers"
4. On the "Edit" menu click "Add Key" and enter: "winreg" for the key name and "REG\_SZ" for the class
5. Go into the new key "winreg"
6. On the "Edit" menu click "Add Value" and enter: "Description" for the value name and "REG\_SZ" for the data type
7. Modify the new "Description" string value to be "Registry Server"

If the key already exists these steps will not be necessary. Once the key is in place, the next step is to modify the permissions of the key. These modifications can be accomplished in many different ways.

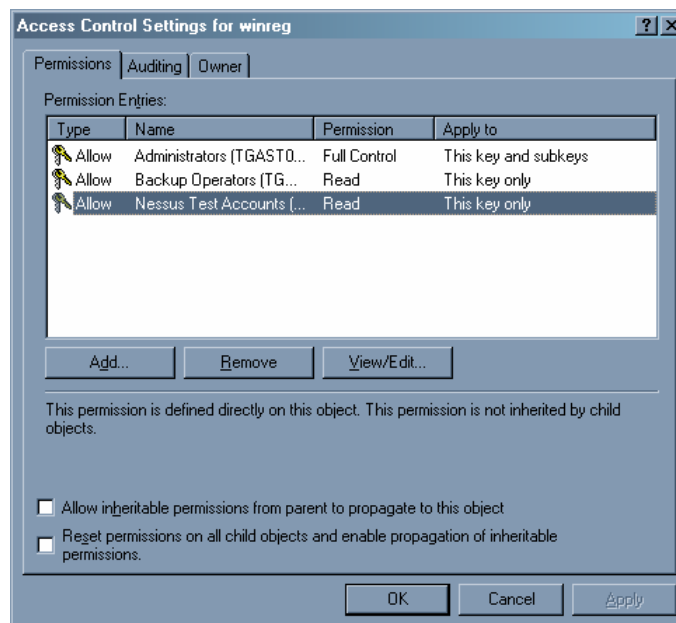
#### Manual Modifications

Typically the most time consuming method would be to use the "regedt32" application to

manually change the permissions on the key for every Windows machine in the domain.

1. Using "regedt32" connect to the computer system (if not running locally), navigate to the key, and select the "Permissions" option from the "Security" menu.
2. In the dialogue box click "Add...", select the domain group "Nessus Test Accounts" from the list, and click "Add" followed by "OK".
3. Click on "Advanced...", highlight the newly created entry, and click on "View/Edit..."
4. In the "Apply onto:" drop down box select "This key only", click "OK", then click "OK" again

The resulting new entry will appear not to have any permissions whatsoever. The checkboxes beside "Read" and "Full Control" will be unchecked. However, clicking on "Advanced..." will show that the group now has "Read" permission for "This key only," which will equate to the "Nessus Test Accounts" group having remote read permission for accessing the registry. The actual dialogue box will be similar to the following screenshot:

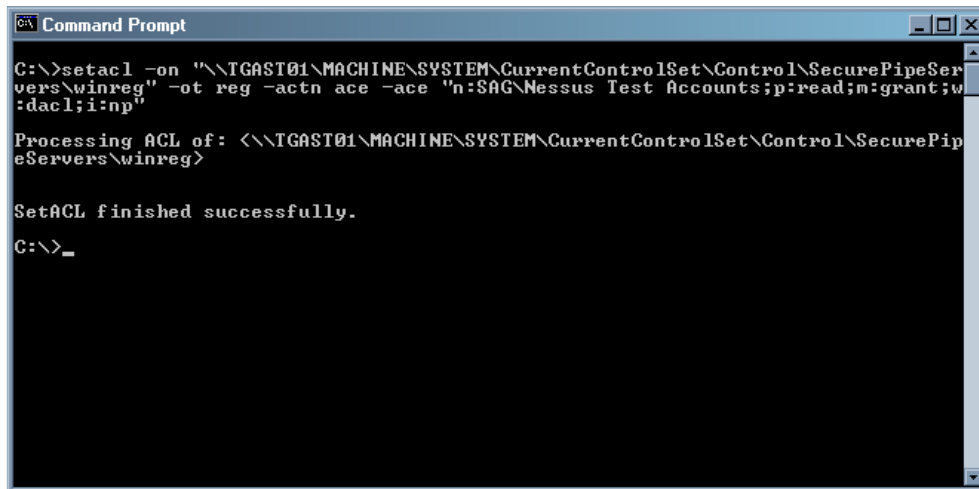


### Command Line Modifications

Another more efficient means to accomplish the same task would be to use a command line tool to script the changes remotely. One such tool is SetACL, available from <http://setacl.sourceforge.net/>. This provides a command line interface to changing many permissions settings, not just for the registry but for files and directories as well. SetACL can be used to change the registry permission settings in the following manner:

```
setacl -on "MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg" -ot reg -actn ace -ace "n:DOMAIN\GROUPNAME;p:read;m:grant;w:dacl;i:np"
```

When performing this remotely, the *machine name* can precede the key designation, as shown in the following screen shot:



```
Command Prompt
C:\>setacl -on "\\TGAST01\MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg" -ot reg -actn ace -ace "n:SAG\Nessus Test Accounts;p:read;m:grant;w:dacl;i:np"
Processing ACL of: <\\TGAST01\MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg>
SetACL finished successfully.
C:\>_
```

### Modifications Using User Logon Scripts

Another possible method would be to use a Windows domain logon script to automatically make the necessary changes when someone logs into the domain. It would be necessary to write a script that can handle all of the potential issues, such as different operating systems, missing keys and mapping drives to have access to the necessary tools. That effort is outside the scope of this document and is left to the reader to accomplish.

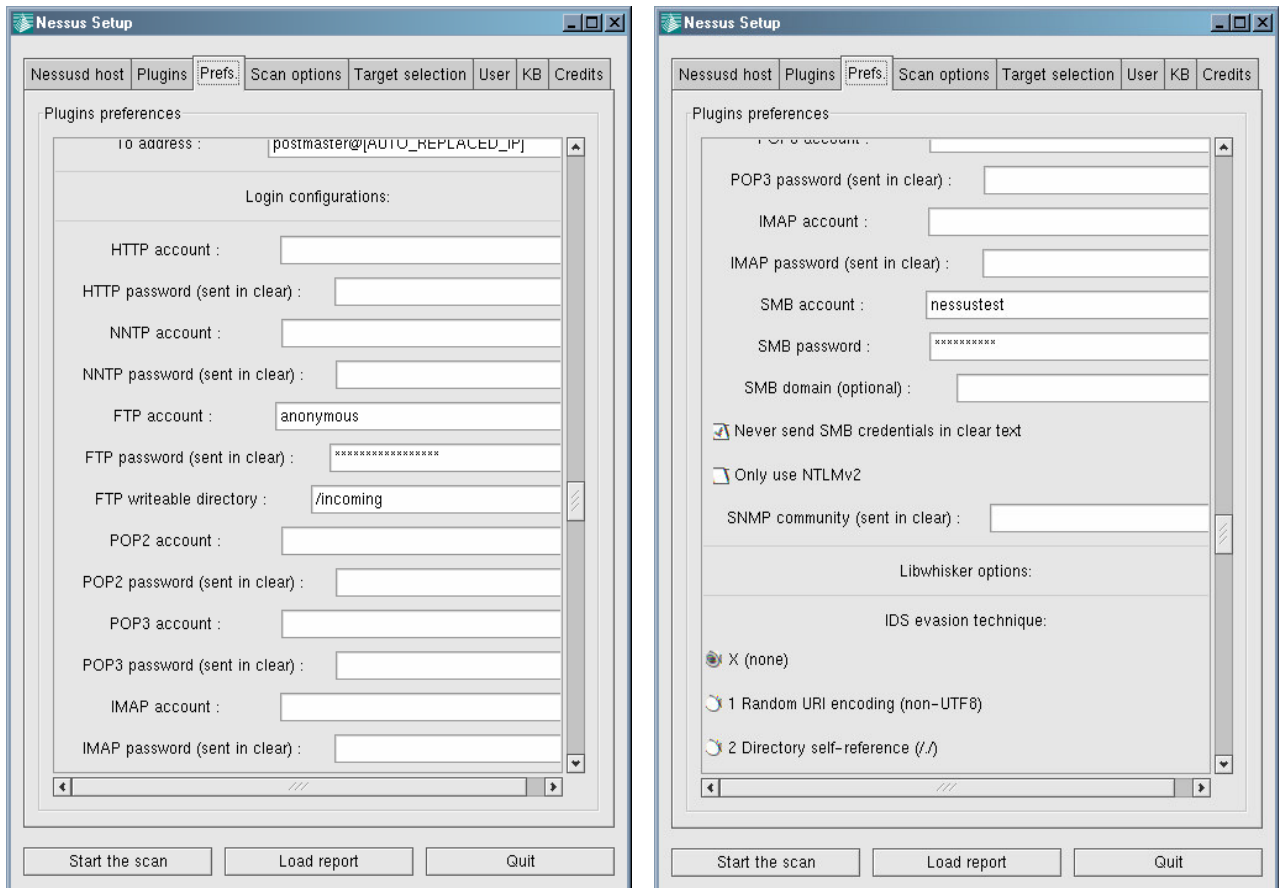
Once all of the systems on the domain have had the necessary registry permissions put in place, a Nessus scan can use the created accounts to gain access during a network scan. Configuring Nessus to use the account credentials is relatively simple, either by using the graphical interface or modifying the ".nessusrc" file directly.

## **Nessus Scan Configuration**

Setting up the Nessus client to use domain credentials via the graphical interface is very straightforward and can be applied to any existing scan configuration. Once all the other settings for the scan are entered, navigate to the "Prefs" tab. As shown in the Nessus Setup dialogue box to the left below, scrolling down will show the "Login configurations" section where username/password pairs are entered for various services. As shown in the dialogue box to the right below, scrolling down will show where the SMB username and password are listed. This is the location where the domain credentials for the "nessustest" account will be entered.



Once the scan starts, all security checks that require remote access to the registry will be able to use those credentials.



### Modifying the Nessus Configuration File

In addition to using the GUI, the Nessus configuration file “.nessusrc” can also be manually modified to use domain credentials. Using a text editor, navigate down to the preferences section, which is framed by the lines:

```
begin(PLUGINS_PREFS)
.
{preference settings here}.
.
end(PLUGIN_PREFS)
```

This section of the configuration file holds all the individual settings for the various plugin preferences, including any username/password settings. Within this preferences section find the entries for the SMB account information, which includes the following two lines:

```
Login configurations[entry]:SMB account : =
Login configurations[password]:SMB password : =
```

If these lines do not exist they can be added manually. Appending the respective username and password information at the end of these lines will make the information available to the plugins. The following example lines show the setup using the “nessustest” account (with a password of “changem3”) created above.



Login configurations[entry]:SMB account : = nessustest  
Login configurations[password]:SMB password : = changem3

Note that the password information is stored in plain text, which may pose a security issue in some environments. However, given the limitations that have been placed on the “nessustest” account above, the exposure is much more limited than if an actual domain administrator account and password were used in the configuration file for the testing.

## Comparing Scan Results

Three different scans were run using the following configurations:

- Scan 1 - No domain credentials used
- Scan 2 - Domain credentials with registry read rights
- Scan 3 - Domain administrator credentials

The differences in scan accuracy between the three scans are easily seen in the results output.

### Comparing Scan 1 with Scan 2

Scan 2 was able to perform many more localized checks using the domain credentials provided. These additional checks included registry-related findings, some examples of which included:

- Determining which Service Pack was installed on the target
- Permission settings and values for critical registry keys
- Vulnerabilities related to Internet Explorer
- Services running on the target
- A list of shares available

Obviously the additional access allowed Nessus to gain a better picture of the security implemented on the target system.

### Comparing Scan 2 with Scan 3

All of the potential vulnerabilities and findings discovered by Scan 2 were also discovered during Scan 3. However, Scan 3 had some results that were misleading as to the true security condition of the target system. Specifically, Scan 3 results indicated that all file shares on the target were accessible. This is misleading in that using a domain administrator account gave Nessus the level access that would allow any share to be accessed. The credentials used in Scan 2 would only identify those shares that were truly open to remote access from the network.

## Conclusion

The comparisons show a marked improvement in scanning results when using the special created account with remote registry access, even over those scans run under the domain administrator privileges. Conducting internal Nessus scans across corporate networks may not provide the truest assessment of security unless domain credentials are used. Creating an account with the necessary access, while not providing the complete and total access that domain admin privileges provide, increases the accuracy of testing while minimizing the impact to local domain security.

One last consideration for choosing to create the special account instead of using domain administrator credentials has to deal with the potential for some plugins to damage target systems. Specifically, it could prove disastrous if a Nessus plugin that has not been thoroughly tested were given full domain admin privileges. The potential for data to be corrupted or deleted exists, and putting complete trust in plugins to appropriately handle full domain admin privileges may be asking for trouble. In the interest of obtaining the most complete security picture for an environment, using the specially created account bridges the gap between too much access and enough to get a clearer view of the security situation.

