

## Comparing Windows Server 2008 UAC to Red Hat Enterprise Linux 5 sudo Command

---

As operating systems become increasingly complex and administrators become more comfortable obtaining utilities from a variety of sources (including the public Internet), there is an increasing risk of inadvertently damaging important system files or state through user error or by executing malicious code (i.e., Trojans and other malicious software, or *malware*).

Modern operating systems such as Windows Server 2008 and Red Hat Enterprise Linux 5 (RHEL5) help protect systems by requiring explicit user action before the OS performs restricted tasks. In Windows Server® 2008 (and Windows Vista®), the User Account Control subsystem provides this kind of system protection.

### User Account Control in Windows Server 2008

The Windows Vista User Access Control (UAC) is also a tool for user permission control. The permissions in UAC are divided into Standard User, Over-the-Shoulder (OTS) Credentials, and Admin Approval Mode. Along with setting the elevated privileges, UAC employs the highly secure desktop, which, in effect, isolates the process of looking for elevated permissions and helps ensure that until a user is authenticated that user cannot interact with the rest of the desktop.

#### Admin Approval Mode

The base idea of UAC is to allow all users to run in Standard User mode. However, administrators can use what is called the Admin Approval Mode feature. The Admin Approval Mode prompts members of the Admin group for their credentials when they perform a task that requires elevated privileges. By controlling these permissions and functions available to end users, UAC has also limited the security threats end users and administrators must face.

Admin Approval Mode permits expanded functionality, including the following:

- Adding printers
- Changing power-management settings
- Allowing administrator-approved ActiveX controls
- Installing critical updates

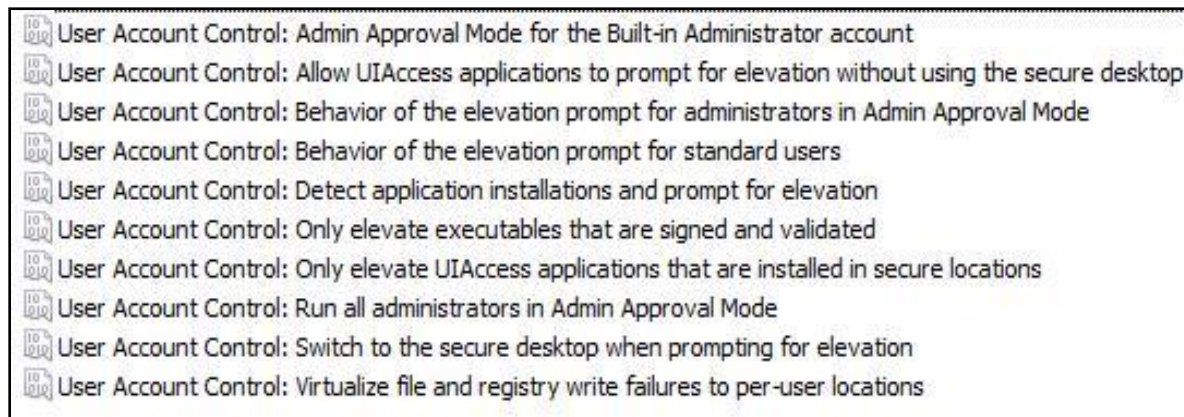
#### Over-the-Shoulder (OTS) Credentials

Any task that calls for installation of applications or drives, and the addition, removal, or modification of users' accounts or access to the Windows firewall, needs to use elevated privileges such as Over-the-Shoulder (OTS) Credentials. In OTS mode, users can provide the local admin permissions if they know them, or an administrator can input users' credentials to continue the task.

## Group Policy

Using the UAC Group Policy Management Console, an administrator can create a Group Policy Object (GPO) for users or groups of users. Group Policy gives you the ability to configure ten different UAC objects whose function can range from managing the behavior of the elevation prompts, to setting rights for Admin Approval Mode, and even choosing whether to use Secure Desktop when prompting for elevation. The console also makes it simple to modify the Group Policy options for UAC to best suit the environment (see Figure 1).

Figure 1.



Many Linux distributions (including RHEL5) provide a similar type of protection with the **sudo** (“superuser do”) command.

## sudo Command in RHEL5

**sudo** is a command-line function that allows administrative access and provides administrative commands that are defined in the sudoer configuration file. The sudo file also contains credentials for users who are allowed to run commands with administrative permissions.

A user who has been granted administrative privileges is first prompted for authentication credentials before he can execute a command. The user is then granted elevated access for a defined time, usually five minutes.

## Executing sudo Commands

To begin using **sudo** commands, a user must be granted permissions in the `/etc/sudoers` configuration file. To edit this file, use the **visudo** command as the root. Figure 2 is an example of how to edit the sudoers configuration file using the **visudo** command.

*Figure 2.*

```
# sudoers file.
#
# This file MUST be edited with the 'visudo' command as root.
#

# User privilege specification
root    ALL=(ALL) ALL
ronald  ALL=(ALL)  ALL

# alicia can manage the printers
# as well as add and remove users
alicia  ALL = PRINTING, /usr/bin/adduser, /usr/bin/rmuser

# emma can run anything on all machines except the ones
# in the "SERVERS" Host_Alias
emma    ALL, !SERVERS = ALL
```

A typical **sudo** command looks like the command stated in Figure 3. Here we see an example of Emma, the administrator, using the **sudo** command to reboot a user's machine within five minutes. Emma then adds a comment to let the user, Alicia, know what is going to happen. The five-minute time interval allows Alicia to close anything she's working on.

*Figure 3.*

```
[emma@raretech]$ sudo shutdown -r +5 " Alicia we need to reboot your system"
Password:
```

This method works, but it tends to be an involved process.

The root administrator needs to create a file and set the permissions per user or group; the admin also needs to suggest which hosts, users, services, and commands will be affected by the use of the **sudo** command. This method also works, but using the **sudo** command is with UAC is better and less time consuming.

## Conclusion

For Linux administrators who are looking for control over their environment while cutting back on the administrative overhead of their current network, Windows Server 2008 provides interface tools that can streamline the user permission and security process.

As we have seen, both Windows and Linux platforms address the need to help secure user permissions; however, the ease of deploying and securing user access in Windows Server 2008 makes it a highly

compelling product. UAC allows users access to their desktop while helping ensure better system security. An added bonus is that UAC can alleviate administrative overhead for the IT department.