Administration of large Windows NT networks with DoIt4Me

Alessandro Augusto, Célio Cardoso Guimarães, Paulo Lício de Geus

IC-UNICAMP – Campinas, SP, Brasil
{alessandro.augusto, celio, paulo}@ic.unicamp.br

Abstract

Remote administration of a large Windows NT network is a complex task. The tools provided by standard NT installations are, at best, inadequate. The explosive growth in network sizes over the last several years has resulted in large and complex sites[1] but no significant new tools were created. One major problem not fully solved is remote NT Registry auditing and configuring.

This paper describes the design and implementation of DoIt4Me, a simple and flexible tool that enables from a single console automation of most Windows NT administrative tasks, especially remote auditing and remote configuring of the NT Registry in a large network.

1. Introduction

With the increased proliferation of system networks, computer security has become an increasingly large problem for system administrators of large sites (with several hundreds or more systems). Most people would agree that keeping a watchful eye on a handful of workstations is a simple task, but not on several hundred workstations.

Unlike many other types of system administration tasks, which can be done at a later time, delaying the installation of a security patch, could leave a site more vulnerable to an intruder attack.

A remote automated procedure should not require that system administrators visit each workstation. This is a problem in many environments where the workstations are located in different rooms, buildings, towns and so on. Fixing each machine through physically visiting it requires a lot of manpower and be error-prone; operator errors can lead to machines being configured erroneously, improperly, or not at all.

2. Challenges Faced

Among the operating systems with wide prominence and use in several environments, Windows NT gets the attention with its growing use and its user-friendly interface [1]. The automation of system administration and security tasks has been discussed a lot, especially when applied to UNIX-like operating systems. However, solutions derived for the Unix environment are generally not applicable to the Windows NT one.

In a comparison of Windows NT with UNIX systems, NT lacks adequate remote network administration tools [8].

In organizations that have a considerably large Windows NT network, administrators always have a hard time when they need to apply some security configurations on each machine in the network. These hardships imply on high monetary costs to maintain a group of system administrators in service and normally take many hours of work.

[1] In this paper, we used the word "site" as a synonym for "network".

Furthermore, a prerequisite to gain efficiency is the knowledge of how to audit the system. To identify which vulnerabilities exist, it is important to regularly audit security by centrally scanning the whole network and identifying which workstations are vulnerable. Then, each of those systems must be correctly reconfigured to adequately secure the network.

In the last several years, there has been a large number of books and papers published on NT security, and on how to improve security of a site; nevertheless, Windows NT still lacks efficient remote administration of large sites, especially in the realms of remote Registry auditing and configuring.

3. Design Goals

One of the keys to administering large networks is to write tools to handle as many common tasks as possible. This may make it possible to automate common tasks, to spend less time on them, or even to hand them off to other people.

Accordingly, it was also necessary to find some way to cover the Windows NT defficiency of tools for remote automation of administrative tasks, and to scale whatever solution one finds to large numbers of machines. This had to be done with a large amount of configuration flexibility (so it could be tailored to the needs of different machines and administration methods) in an as automatic as possible way.

Faced with these Windows NT weaknesses, our solution should have some desirable properties:

?? Simple use and maintenance
?? Centralized
?? Well scalable
?? Configurable in order to meet specific user needs
?? Able to provide verification and notification of compliance with security policies
?? Capable of enforcing compliance with security policies and standards
?? Reduced overall cost of administration
?? Inexpensive
?? Minimal human interaction to install packages on each networked machine
?? Capable of alerting administration when a machine is having problems

When trying to figure all these desirable properties in a single solution, we decided to implement a new remote system administration tool, called DoIt4Me. Its goal was to automate administrative tasks across a Windows NT network, especially in regards to providing Windows NT remote Registry auditing and configuring in an easy fashion.

Automating tasks with scripts is an old technique from the UNIX community. Many different techniques and scripting languages where studied before we chose Perl to implement our solution.

4. Perl

Perl has been used on UNIX platforms for administration purposes for many years. ActiveState [2] provides a fairly complete distribution of Perl for Win32. It also has several modules for the NT environment that provide a convenient wrapper around the Win32 API, providing access and modification of NT security-relevant data [4].

Perl is able to do many unusual tasks. For example, the administrator can use Perl to have the machine send an e-mail back to him when it is running out of disk space, or to make it purge old database entries.

Perl can be useful in many Windows NT administrative tasks, as will be shown. It was desirable that all tasks presented previously should be grouped together in a single tool, or maybe in a toolkit, i.e., a collection of tools and scripts.

5. Previous Work

Harlan Carvey presents in [4] a framework of a few administrative scripts that had some similar goals to our project. For example, one of his scripts, called `regkeys.pl`, is devised to collect Registry values from a remote NT system. However, these scripts have some weaknesses: they are not scalable to a large NT network.

As a practical example of remote auditing and compliance, suppose the system administrator wants to collect the value of the `DontDisplayLastUserName` Registry key of all workstations. This can be done with the script presented by Harlan, but the administrator will have to write down the results of each workstation, because the script only checks one machine at a time.

The framework presented in [4] requires human intervention for each audited machine. There is no remote task automation for multiple machines. It is then clear that this approach is unable to handle a large number of machines. Also, once the system administrator knows which workstations are not in compliance with security policies, there is no ability to configure the machines with new values, i.e. to act upon.

However, these scripts also have their strength, since they show how to do the remote collection for a single machine, and as such can be used as a building block to achieve our goals.

6. Our Solution

Centralized security administration of NT systems can be performed in three phases:

1. Data collection
2. Filtering/Analysis
3. Modification

They are kept separate in order to maintain simplicity, scalability and functionality. It also makes it easier to build a working set of tools, by allowing testing and verification of one phase before moving on to the next. Additional functionality can be added to one phase without requiring any changes to the other phases [4].

1. In the data collection phase, the administrator specifies which configuration settings he or she wants to collect. It is only necessary to specify the subset of machines that will be scanned and the configuration settings that will be collected.
2. In the second phase, the administrator filters and analyses the results of the first step. This can be easily done using Perl's regular expression pattern matching abilities [4]. A proposed functionality of DoIt4Me will show the machines that are not in compliance with the desired configuration settings.

   These filters may check:

   ?? Known security issues, such as specific Registry values and ACLs (on files, directories, and Registry keys).

   ?? Compliance with corporate security policies, such as Service Pack versions and Hotfixes, and NT services status.
3. The most important stage is the last one. In this phase, the system administrator can apply his configuration to any subset of machines. A few examples of administrative tasks that can be automated are listed below:

?? To start or stop remote NT services
?? To add, delete or change Registry values
?? To enable or disable security auditing
?? To change Registry values
?? To directly access the Microsoft API
?? To reboot machines with a predefined grace period

We managed to build a single tool that meets all of the above mentioned requirements, called DoIt4Me.

7. DoIt4Me

7.1 Overview

DoIt4Me is an automated and remote administrative tool for Microsoft Windows NT operating systems. It can manage a large NT network from a single console. Infrequent trips to distant machines will only be necessary in case of hardware failures.

It is specifically aimed at administrating and securing Windows NT 4.0 machines, although some of the functionality could also be used on Windows 2000.

In order to achieve a better and easier solution than previous works, it was a requirement to be able to specify a subset of machines. All of DoIt4Me's options, showed on section 6, can be performed by any subset of machines.

The first feature of DoIt4Me, is the ability to scan the entire network and to report the results for auditing. The next goal after auditing was the ability to configure remote computers. DoIt4Me makes this easy. Another feature of DoIt4me is the ability to print the results online or print it to text files, which can be read by text editors and analyzed with more attention by the administrators.

Moreover, on NT networks, it is important to determine not only whether individual machines are up or down, but also whether services (daemons) they offer are available. DoIt4Me is able not only to check the status of remote NT services, but also to start or to stop any subset of services on any subset of workstations.

DoIt4Me does not depend on `Regedit.exe` or any other Registry editing tool.

By installing DoIt4Me on the PDC[2], the administrator can remotely control any subset of workstations served by the PDC. It is also necessary that the PDC be able to execute Perl scripts.

7.2 Interface

There is no single interface for configuring and administering an NT network. For example, the audit policy for a standalone NT system is set via the User Manager, while log specific settings and all monitoring activities are recorded in the Event Log. Furthermore, each object (file, directory, share, Registry key) has its own interface for enabling access control lists (ACLs). Rolling out a common audit standard across an NT enterprise and monitoring the Event Logs can be a daunting task [4].

A related issue is whether or not administration tools should be based on a “graphic user interface” (GUI). This kind of interface can be easier to use if the system administrator's goal is to build or configure a single machine. In general GUI tools are harder to automate and extend. DoIt4Me

[2] Primary Domain Controller

interface has a simple unified syntax and is used through the NT command line interpreter. A brief overview of DoIt4Me interface and its options is located in the appendix A.

7.3 Reporting

One problem became very apparent during the implementation. The output produced should be in a format fit for human consumption.

The reports enable the system administrator to identify quickly and easily, any problems related to the machines, ranging from a client being down to reporting a subset of machines that are not complying with security policies and standards. Appendix B and appendix C present examples of DoIt4Me reports.

7.4 Configuration Files

Global security policy changes are made on centrally located configuration files. This model works well for complying with changing security policies.

DoIt4Me has a few configuration files. Each file has its own function. For example, the file `pclist.cfg`, contains the subset of machines that DoIt4Me will scan, configure or reboot.

Another file, `srvnewstatus.cfg`, contains the subset of NT services followed by its new status, i.e., 1 to start the service or 0 to stop the service.

7.5 Limitations

Some management tasks cannot be performed remotely because of Windows NT limitations, such as remotely accessing some parts of the Registry. None Windows system export the whole Registry. Only two of the six Registry keys can be accessed remotely: the `HKEY_LOCAL_MACHINE` and the `HKEY_USERS`. Nevertheless, the main Registry key necessary to implement security is `HKEY_LOCAL_MACHINE,` which fortunately is remotely available.

If the administrator wants to modify any Registry value not present in these two keys, he or she may start the schedule service on the target machines via DoIt4Me and use the NT administration technique "Schedule Technique" presented in [1].

On the other hand, DoIt4Me can be wholly customized. System administrators can construct new customized functions.

8. Conclusion and further work

Even in a small NT network, the process of auditing a Registry value can be cumbersome. Early versions of DoIt4Me focused only on the identification of security vulnerabilities, not their correction. The current version of DoIt4Me addresses security weaknesses and eases standardization and adherence to NT network security policies.

Further versions of DoIt4Me will bring more automated tasks, such as applying ACLs to disk folders and files, and integrating DoIt4Me with ODBC and SQL, so the reports can be archived in a database [11].

Our experience has shown that with the right mix of administration techniques and DoIt4Me, it is possible to remotely manage a large NT network in an scalable way.

DoIt4Me is aimed at:

?? Security minded system administrators who are willing to put time and effort into securing their Windows systems.

?? Security consultants who find themselves having to secure Windows NT computers regularly, and who want to automate these tasks as much as possible without losing the flexibility of easy customization.

9. Availability

For further information on the availability of the current version, please send an electronic mail to `alessandro.augusto@ic.unicamp.br`.

10. References

[1] AUGUSTO, Alessandro and GUIMARAES, Célio and GEUS, Paulo Lício. ``Administration Techniques for Implementing Security on Large Windows NT Networks". Proceedings of SSI'2000: Symposium on Informatics Security, BRAZIL, 2000.

[2] ActiveState WebSite
http://www.activestate.com

[3] CARTER, Gerald. ``Patch32: A System for Automated Client OS Updates". Proceedings of USENIX LISA-NT: The Large Installation System Administration of Windows NT Conference, USA, 1998.

[4] CARVEY, Harlan, ``System Security Administration for NT``. Proceedings of USENIX LISA-NT: The 2nd Large Installation System Administration of Windows NT Conference, USA, 1999.

[5] Cox, Phil, ``Auditing: The Ugly duckling of Computers``, ;Login: The Magazine of USENIX & SAGE, 1998.

[6] DALY, Gregg and BUHRMASTER, Gary and CAMPBELL, Matthew and CHAN, Andrea and COWLES, Robert and DENYS, Ernest and HANCOX, Patrick and JOHNSON, Bill and LEUNG, David and LWIN, Jeff. ``NT Security in an Open Academy Environment". Proceedings of USENIX LISA-NT: The 2nd Large Installation System Administration of Windows NT Conference, USA, 1999.

[7] FULMER, Robert and LEVINE, Alex. ``AutoInstall for NT: Complete NT Installation Over the Network". Proceedings of USENIX LISA-NT: The Large Installation System Administration of Windows NT Conference, USA, 1998.

[8] GOMBERG, Michail and EVARD, Rémy and STACEY, Craig. ``A Comparison of Large-Scale Software Installation Methods on NT and UNIX". Proceedings of USENIX LISA-NT: The Large Installation System Administration of Windows NT Conference, USA, 1998.

[9] GOMBERG, Michail and STACEY, Craig and Sayre, Janet. ``Scalable, Remote Administration of Windows NT". Proceedings of USENIX LISA-NT: The 2nd Large Installation System Administration of Windows NT Conference, USA, 1999.

[10] KRANENBURG, Paul. ``Monitoring Utilization in an NT Workstation Lab". Proceedings of USENIX LISA-NT: The Large Installation System Administration of Windows NT Conference, USA, 1998.

[11] ROTH, Dave, ``A Networked Machine Management System". Proceedings of USENIX LISA-NT: The 2nd Large Installation System Administration of Windows NT Conference, USA, 1999.

Appendix A: DoIt4Me Interface

What follows is a brief overview of the DoIt4Me interface.

```
C:\> DoIt4Me.pl
----------------------------------------------------
DoIt4Me - Automate NT Administrative Tasks Remotely

Usage : DoIt4Me.pl <option>
Option: <1> Auditing
        <2> Configure the Registry
        <3> Check the status of ALL NT services
        <4> Check the status of a subset NT services
        <5> Change NT services status (Start/Stop)
        <6> Reboot a subset of workstations
        <7> Help
 -----------------------------------------------------
```

Appendix B: DoIt4Me Auditing Report

This example shows the report generated when the system administrator performs the above mentioned option 1 (auditing), for a subset of 2 machines: `mustang` and `porsche`. The system administrator wants to collect the values of the following Registry keys: `CSDVersion`, `DefaultUserName` and `DontDisplayLastUserName`.

The report to this option should look like this:

```
C:\> DoIt4Me.pl  1
-----------------------------------------------------
               Auditing Report
-----------------------------------------------------

COMPUTER           KEY                       VALUE
--------      --------------           --------------
mustang       CSDVersion                Service Pack 6
porsche       CSDVersion                Service Pack 5

mustang       DefaultUserName           Administrator
porsche       DefaultUserName           Administrator

mustang       DontDisplayLastUserName   0
porsche       DontDisplayLastUserName   0
```

Appendix C: DoIt4Me Service Status Report

This example shows the report generated when the system administrator performs DoIt4Me option 4, for a subset of 3 machines: `mustang`, `porsche` and `ferrari`. Also, the system administrator wants to check only the status of "alerter" service and "schedule" service.

Note that each report tries to print the result in an easily understandable way to the system administrator.

```
C:\> DoIt4Me.pl  4
-----------------------------------------------------
               Services Status
-----------------------------------------------------


alerter
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
COMPUTER                STATUS
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
mustang                [Stopped]
porsche                [Stopped]
ferrari                [Started]


schedule
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
COMPUTER                STATUS
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
mustang                [Started]
porsche                [Started]
ferrari                [Stopped]
```