Automatização de Administração e Segurança em Redes Windows NT

Alessandro Augusto

Dissertação de Mestrado

Instituto de Computação
Universidade Estadual de Campinas

Automatização de Administração e Segurança em Redes Windows NT

Alessandro Augusto

Setembro de 2001

Banca Examinadora:

- Prof. Dr. Paulo Lício de Geus (Orientador)
  Instituto de Computação, UNICAMP
- Prof. Dr. Carlos A. Maziero
  Pontifícia Universidade Católica do Paraná, PUC-PR
- Prof. Dr. Ricardo Dahab
  Instituto de Computação, UNICAMP
- Prof. Dr. Edmundo Madeira (Suplente)
  Instituto de Computação, UNICAMP

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Augusto, Alessandro
Au45a Automatização de administração e segurança em redes Windows NT/ Alessandro Augusto. – Campinas, SP: [s.n.], 2001.

Orientadores: Paulo Lício de Geus; Célio Cardoso Guimarães.
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Redes de computação - Medidas de segurança. 2. Computadores - Medidas de segurança. 3. Windows NT (Sistema operacional de computador). I. Geus, Paulo Lício de. II. Guimarães, Célio Cardoso. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título

Automatização de Administração e Segurança em Redes Windows NT

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Alessandro Augusto e aprovada pela Banca Examinadora.

Campinas, Setembro de 2001

Prof. Dr. Paulo Lício de Geus (Orientador)

Prof. Dr. Célio Cardoso Guimarães (Co-orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 13 de setembro 2001, pela Banca Examinadora composta pelos Professores Doutores:

Prof. Dr. Carlos Alberto Maziero
PUC - PR

Prof. Dr. Ricardo Dahab
IC - UNICAMP

Prof. Dr. Paulo Lício de Geus
IC - UNICAMP

Resumo

A administração, a manutenção da segurança e o gerenciamento de grandes redes de computadores baseados em Windows NT são tarefas desafiadoras e trabalhosas. Algumas tarefas podem se tornar extremamente laboriosas para os administradores dessas redes, como por exemplo: instalação remota de programas, auditoria e modificação de uma configuração de segurança remota ou melhorar o desempenho de cada máquina.

Este trabalho tem como objetivo desenvolver técnicas para automatizar as tarefas de administração de redes Windows NT, tornando-as menos complexas. Além disso, este trabalho apresenta e descreve DoIt4Me ("do it for me"), uma ferramenta de gerenciamento remoto capaz de melhorar a segurança, a administração e o desempenho de cada máquina dessas redes.

Abstract

The administration, the maintenance of the security and the management of large Windows NT networks are challenging tasks. Some tasks can be extremely laborious, such as: software remote install action, auditing and updating the security configurations or to improve the performance of each network machine.

The goal of this work is to develop techniques to automate network administrative tasks, turning them less complex. Besides that, this work presents DoIt4Me ("do it for me"), a network management tool to improve the security, the administration and the performance of each network machine.

Agradeço todo o meu sucesso e minha saúde
aos responsáveis por tudo isso ser verdade,
meu Senhor e meu Deus.

Agradecimentos

Aos melhores pais do mundo, Sr. Orlando e Dna. Cida, por toda força e apoio que me deram durante todas etapas da minha vida, nos momentos tristes e difíceis, como nos momentos alegres e divertidos pelos quais estivemos juntos.

À Dna. Ana, a vó mais alegre de todas, por todas as rezas e pelos pães caseiros que fez em minha homenagem.

Ao meu irmão Rafael, que um dia vai chegar até aqui também e seu sucesso irá muito mais adiante. Confie em você, reze e fique de olho na sorte.

Aos meus brothers André e Daniel, também conhecidos como AndGué e Sem Sono, pelos dias e noites os quais passamos dando risadas em nosso apartamento e fazendo planos pro futuro, pelos trotes a Dna. Elvira, pelas caronas inesquecíveis até a rodoviária, pela ajuda de vocês na tese e pelo amor que tenho a vocês. Conte comigo no que precisarem.

Ao grande brother Dioni, pela amizade incrível de sempre e pelas conversas noturnas no orelhão. Por sinal, será que tem como você me enviar uma cópia do nosso compilador da UEL?

À minha família americana, os Carrolls, que me acolheram durante meu intercâmbio e durante minhas visitas aos Estados Unidos, por todo o amor, paciência e ajuda. Sem eles essa conquista também não seria possível.

Aos professores e amigos Paulo Lício e o Célio, por terem acreditado no meu potencial, pela ajuda e paciência que tiveram durante todo esse tempo.

À Bosch, pelo patrocínio financeiro durante todo o projeto.

Aos malucos do LAS, o pessoal mais fera da área de segurança de computadores no Brasil.

À galera do Fut, pelas cervejas e divertido futebol das tardes de quarta feira. Ainda vamos nos encontrar muito durante o resto de nossas vidas.

E a você que está lendo isso aqui também!

Conteúdo

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

A evolução tecnológica e a conseqüente diminuição dos custos dos computadores tornou cada vez mais atraente a possibilidade da interconexão de computadores em redes. Em termos genéricos isto é conhecido como compartilhamento de recursos, cujo objetivo é colocar programas, equipamentos e dados ao alcance de todos os usuários autorizados da rede, independente da localização física do recurso e do usuário [59].

Nos últimos anos houve um crescimento muito grande de novas redes de computadores. Organizações têm desenvolvido redes cada vez maiores e mais complexas com relação à sua interconexão. Com esse crescimento acelerado, o sucesso de qualquer organização depende muito da administração e da segurança de seus recursos computacionais. Da mesma forma que tem aumentado a confiança nas informações provindas de sistemas computacionais, também se nota um aumento na exploração ilegal de sistemas e manipulação indevida de informações.

Com um significado maior que simples proteção contra usuários mal intencionados, a segurança deve ser considerada uma parte crítica da estrutura necessária para garantir a disponibilidade dos recursos computacionais, bem como proteger informações estratégicas e de caráter sigiloso.

Entretanto, muitas pessoas ainda não conseguem enxergar essa importância e imaginam que as soluções de segurança são caras e não trazem nenhum retorno financeiro. Isso faz com que gerentes, diretores e responsáveis das organizações prefiram aplicar seus recursos apenas em novas soluções que podem trazer vantagens visíveis aos olhos de todos.

De fato esse é o maior problema da área de segurança, ou seja, a solução de segurança é difícil de ser mensurada, sendo muitas vezes erroneamente desvalorizada. O fato é que poucos

percebem a existência de segurança; sua falta é sentida apenas quando um incidente acontece e resulta em prejuízos.

Deve-se superar a idéia de segurança ser um produto que funciona como um antídoto a incidentes ou ataques que comprometam informações ou recursos da organização. O conceito de segurança vai muito além disso. Segurança não é um produto, é um processo [49]. A segurança da rede não está garantida apenas com a compra e instalação de um novo utilitário ou hardware, mas sim através de uma administração eficiente de segurança de rede.

Dessa forma, na área de segurança também é válida o dito popular: "prevenir é melhor que remediar". Porém, o que se nota é uma visão reativa, com decisões de segurança sendo tomadas apenas após um incidente, o que resulta numa série de conseqüências negativas, principalmente no que se refere a perda de credibilidade [44]. É necessário eliminar o fato de segurança ser considerada opcional e geraknente estar em segundo plano. A segurança deve ser vista como elemento essencial para a prevenção de prejuízos. É importante identificar os valores das informações da organização e então calcular e avaliar os impactos causados após um incidente. Essa identificação permite entender os custos gerados se a organização sofrer um incidente [8].

Em condições ideais, todas as organizações deveriam ter uma equipe de administradores de segurança separada da equipe de administradores de redes e proporcionais ao tamanho de sua rede. Com isso a equipe de administradores de rede teria tempo suficiente para executar as tarefas exclusivas dessa área, enquanto os administradores de segurança teriam tempo e informações disponíveis para implantar e manter ativo um conjunto de regras e normas com relação à segurança dos recursos computacionais, garantindo assim alguns requisitos de segurança.

Na realidade, isso dificilmente acontece. Em muitas organizações, a equipe de administração de rede é responsável pelas decisões para implementação de segurança das informações, fazendo com que esse departamento assuma, conceitualmente, as funções de administração de segurança. Em algumas organizações o caso é ainda pior: além de não ter uma equipe de administradores de segurança e de rede proporcional a seu ambiente, os administradores têm ainda que responder por tarefas não ligadas diretamente à sua função e nem mesmo com segurança, por exemplo fornecer suporte técnico a usuários da rede, o qual deveria ser função do departamento de *help-desk*.

1.1 Objetivos da dissertação

O alvo dessa dissertação é tratar alguns problemas encontrados na área de administração e na área de segurança de redes em ambientes baseados no sistema operacional Windows NT.

O ambiente NT é considerado carente na área de administração remota de redes [18] [24] [25] [47] . Tarefas como instalar ou atualizar programas remotamente podem se tornar muito complexas dependendo do número de computadores presentes na rede.

A administração da segurança em um computador com Windows NT é muito trabalhosa. Não existe um programa único capaz de modificar todas as configurações de segurança do sistema. Para isso, o sistema padrão traz consigo vários utilitários, cada um capaz de configurar itens específicos de segurança.

Os sistemas Windows NT são considerados carentes com relação a ferramentas para administração de segurança em computadores remotos [24] [47]. Em [25] os autores afirmam ser o NT um sistema impossível de gerenciar remotamente, cuja administração não é escalável e que necessita interação manual para quase toda tarefa de segurança. É difícil encontrar programas que possibilitam realizar remotamente a mesma tarefa em todos os computadores da rede de uma só vez.

De qualquer forma, esse sistema operacional vem sendo amplamente adotado nos últimos anos, desde pequenas redes até em grandes instituições e corporações com milhares de computadores.

Agrupando as áreas de administração e segurança de redes e as deficiências encontradas nessas áreas em redes Windows NT, pode-se citar algumas razões para esta dissertação:

- suprir a carência do sistema operacional Windows NT em fornecer ferramentas para automatizar a administração remota da rede e da sua segurança,
- diminuir a dificuldade e tempo gasto em aplicar uma política ou recomendação de segurança,
- diminuir a chance de erro manual durante as configurações de segurança.

A partir dessas razões, os 2 principais objetivos deste trabalho são:

1. **Automatizar tarefas de administração de redes**

   Busca-se aqui tentar minimizar a interação humana e automatizar a execução de tarefas comuns na área da administração de redes Windows NT, independente do tamanho da rede. A título de exemplo, suponha que o administrador deseje instalar um conjunto de programas localmente em cada computador da rede. Cada instalação exige a presença física e interação do administrador. A complexidade e o tempo gasto para completar a tarefa aumenta bastante por ter que visitar e refazer a mesma modificação em todas as máquinas da rede. Desta forma, quanto maior a quantidade de computadores dessa rede, mais complexa e demorada será sua administração. O primeiro objetivo desta dissertação é automatizar uma instalação e/ou atualização de programas em todas as máquinas da rede de maneira escalável. Como instalar ou atualizar o(s) programa(s) uma vez em um computador e em seguida automatizar a tarefa fazendo com que o restante dos computadores da rede se auto-atualizem, ou seja, cada computador cliente da rede deve ser capaz de instalar ou atualizar programas sem interação do administrador.

2. **Implantar remotamente de maneira escalável segurança em cada computador da rede NT**

   Além das dificuldades em automatizar as tarefas de administração, o mesmo problema ocorre com a área de segurança. Suponha que o administrador de segurança deseje aplicar uma política de segurança, que é composta por uma lista de configurações que devem ser alteradas em cada computador da rede. Existe uma carência no ambiente Windows NT de uma ferramenta que permita especificar a lista de configurações a ser aplicadas e em quais computadores. Os objetivos são: como administrar a segurança local de cada máquina de maneira escalável, eficiente e rápida? Como fazer as modificações desejadas em todas as máquinas da rede com um esforço pequeno e no menor tempo possível? Como especificar as configurações a serem implementadas em um computador e propagar automaticamente essas configurações a todos os computadores da rede?

Para os dois objetivos serem alcançados, principalmente o segundo, o autor desenvolveu uma ferramenta chamada DoIt4Me ("do it for me"), que automatiza administração de segurança remota em grandes redes Windows NT.

1.2 Trabalhos Correlatos

Existem vários trabalhos sobre segurança no ambiente NT, porém poucos relacionados com automatização escalável de tarefas. A maior parte dos trabalhos apresenta apenas uma lista de configurações para aumentar a segurança dos computadores [11] [35] [38] [40] [43] [45].

Entretanto, um trabalho correlato chama a atenção pela questão da segurança remota. Harlan Carvey desenvolveu um sistema composto por alguns *scripts*, com objetivo de realizar remotamente algumas tarefas [10]. Um dos *scripts* apresentados por Harlan tem a finalidade de auditar remotamente as configurações de uma máquina NT. Embora esse *script* sirva para a área de segurança, tem uma deficiência séria, pois a auditoria é feita em apenas um computador. Nenhum dos *scripts* automatiza a tarefa para vários computadores ao mesmo tempo.

Com essa falta de escalabilidade o sistema de *scripts* apresentado por Harlan necessita de interação humana para cada computador, tornando-se muito complexo seu uso em uma rede com uma grande quantidade de computadores. Outra deficiência séria: suponha que o administrador utilize esse sistema de *scripts;* após consultar as configurações de cada máquina, não existe nenhum *script* capaz de configurar e aplicar remotamente segurança nas máquinas. Além disso o autor não trata o assunto de administração de redes, como por exemplo o problema da automatização da instalação e manutenção de aplicativos.

Pelo fato do Windows NT não fornecer ferramentas para instalar e atualizar remotamente programas, grande parte de trabalhos publicados sobre esse assunto utilizam o *Systems Management Server* (SMS), que é um produto da Microsoft para gerenciamento centralizado de rede [21] [33]. Com o SMS é possível realizar tarefas de administração de forma automática, como por exemplo, realizar inventário, instalação e distribuição de programas pela rede [52].

Com relação a segurança, o SMS não facilita a implantação de políticas de segurança. Não existe no SMS uma interface onde seja possível definir quais parâmetros de segurança o administrador quer aplicar nas máquinas da sua rede. Além disso, uma limitação desse produto é o preço. O SMS requer dois tipos de licenças, uma para o servidor da rede e uma licença individual para cada computador da rede. Portanto, o preço desse produto varia dependendo do número de computadores da rede: quanto mais computadores mais caro o preço final do pacote [51].

Outra possível solução é a ferramenta VNC [60]. VNC é um programa gratuito que tem como objetivo realizar console remoto. Instalando um cliente VNC em uma máquina Windows, é pos-

sível exportar a área de trabalho para um computador remoto. Essa ferramenta permite exportar ambiente Windows para máquinas Unix, ou vice-versa. Com o VNC o administrador é capaz de instalar realizar tarefas remotamente, porém a deficiencia dessa ferramenta é a questão de automatização e da escalabilidade. É necessário conectar em cada computador remoto para realizar tarefas. Não existe nenhuma forma de realizar tarefas em lote, automatizando o mesmo conjunto de tarefas em um conjunto de máquinas. Similar ao VNC, existem algumas outras ferramentas de console remoto como por exemplo PCanywhere [58].

1.3 Organização do trabalho

Esta dissertação está dividida em três partes. Inicialmente é mostrada a motivação e os objetivos pretendidos. Em seguida são apresentadas as áreas de administração e segurança de redes e as problemáticas dessas áreas no ambiente Windows NT. Por último são apresentadas as técnicas propostas para automatizar as tarefas de administração, a ferramenta desenvolvida pelo autor para automatizar a administração de segurança de uma rede NT e as conclusões deste trabalho.

O capítulo 1 apresenta a motivação e a necessidade de aplicar segurança nos ambientes de rede, seguido dos objetivos gerais da dissertação e trabalhos correlatos. O capítulo 2 apresenta as áreas de administração e segurança de redes, descrevendo suas tarefas, requisitos, uma introdução sobre política de segurança e encerrando com o ciclo de vida de um projeto de segurança. O capítulo 3 apresenta uma breve história do sistema operacional Windows NT e a relação entre a área de segurança e os sistemas operacionais. Em seguida descreve um dos tópicos mais importantes sobre segurança do NT, o *Registry*, e finaliza com a necessidade de automatização das tarefas em ambas as áreas, administração e segurança de redes. No capítulo 4 são apresentadas algumas das principais recomendações de segurança a ser aplicadas em computadores NT e são detalhadas 3 técnicas de administração, sendo 2 delas criadas pelo autor e que têm como objetivo automatizar tarefas de administração de redes, facilitando a instalação de programas remotamente sem a necessidade da presença física do administrador em cada máquina da rede. O núcleo desse capítulo é formado pelo artigo *"Administration Techniques for Implementing Security on Large Windows NT Networks"*, apresentado no 2° Simpósio de Segurança em Informática (SSI' 2000), realizado em São José dos Campos, São Paulo, em outubro de 2000 [7]. O capítulo 5 descreve o problema de auditoria e implantação remota de políticas de segurança em máquinas NT e propõe como solução a ferramenta de código aberto desenvolvida pelo autor, DoIt4Me [3]

[5] [6]. O conteúdo do capítulo 5 é formado pelo artigo *"Administration of Large Windows NT Networks with DoIt4Me"*, apresentado na *10th International Conference on System Administration, Networking and Security* (SANS' 2001), Baltimore, MD, USA, em maio de 2001 [4]. O capítulo 6 apresenta conclusões da dissertação, contribuições e extensões para este trabalho. Há 2 apêndices: o apêndice A apresenta detalhes sobre a implementação e código da ferramenta DoIt4Me. O apêndice B traz uma descrição e o manual da ferramenta DoIt4Me. A descrição, *"DoIt4Me: a Tool for Automating Administrative Tasks on Windows NT Networks"* e o manual, foram apresentados no 1° salão de ferramentas do 19° Simpósio Brasileiro de Redes de Computadores (SBRC' 2001), que aconteceu em Florianópolis, Santa Catarina, em maio de 2001 [5].

Os artigos apresentados nos capítulos 4 e 5 não sofreram modificações de seu conteúdo final publicado nos anais das respectivas conferências. Entretanto, por questão de estética, foram reformatados para o formato do resto da dissertação e suas seções de referências foram movidas para a seção de bibliografia desta dissertação.

Capítulo 2

Administração e Segurança de Redes

Este capítulo descreve algumas subtarefas envolvidas na área de administração de redes. Em seguida mostra os elementos conceituais da área de segurança, apresentando os requisitos e política de segurança. O capítulo encerra definindo o ciclo de vida proposto para os projetos de segurança.

2.1 A área de administração

Ao mesmo tempo que as redes de computadores criaram inúmeras oportunidades para usuários e empresas se comunicarem e compartilharem informações, essa tecnologia digital criou um grande desafio com relação às áreas de administração e segurança de redes.

O termo administração de redes refere-se ao trabalho de gerenciamento e manutenção de tarefas diárias em um ambiente com dois ou mais computadores conectados entre si.

Pode-se subdividir a área de administração em algumas tarefas ligadas a:

- Gerenciamento de Usuários

  É de responsabilidade do administrador do sistema adicionar, gerenciar e remover contas de acesso de usuários. Procedimentos devem ser seguidos para adição de novas contas de usuários, entre eles a política de segurança, que define regras com relação ao tamanho mínimo e máximo das senhas, normas e obrigações dos usuários, etc.

- Instalação e manutenção de *hardware*

  Quando um novo *hardware* é adicionado, modificado ou removido de uma máquina, o sistema deve ser configurado para reconhecer esse *hardware*. Isso pode variar entre uma

simples tarefa de adicionar uma impressora até uma tarefa mais complexa como bloquear o acesso ao disco flexível através do *hardware*.

- *Backup*

  Executar *backups* é uma das tarefas mais importantes do administrador da rede. O processo de *backup* é uma tarefa que consome tempo e normalmente é cansativo e desagradável. Os *backups* podem ser automatizados e programados para execução automática.

- Instalação e manutenção de programas

  Quando um novo programa é adquirido, o administrador deve instalar e testar a compatibilidade desse programa com os demais programas instalados na rede. Os programas podem ser instalados no servidor da rede ou localmente em cada máquina cliente. Adiante veremos como é complexo o trabalho de instalar ou atualizar programas remotamente nas redes baseadas em Windows NT.

- Monitoramento do sistema

  Existem diversas tarefas que devem ser monitoradas pelo administrador, como por exemplo: monitoramento dos servidores, disponibilidade dos recursos, monitoramento dos logs, entre outras.

2.2 A área de Segurança

Para minimizar os prejuízos e danos causados em caso de alguma indisponibilidade do sistema, organizações definem um conjunto de regras que protegem e gerenciam o sistema, o qual se chama política de segurança. Uma política de segurança é um conjunto de leis, regras e práticas que regulam como uma organização gerencia, protege e distribui sua informação e recursos [53]. Considera-se um sistema seguro quanto à política de segurança o sistema que cumpre as leis contidas no documento.

No contexto de segurança, há diferentes requisitos que um sistema deve possuir para que possa ser considerado ou comparado a um outro sistema seguro. Há vários requisitos diferentes que ambos, usuários e administradores de segurança, precisam estar cientes, como por exemplo [22]:

- Confidencialidade ou sigilo

Proteger a informação de ser lida ou copiada por qualquer usuário ou processo não autorizado.

- Integridade de dados

  Proteger a informação de ser apagada ou alterada de qualquer forma que seja, sem que se tenha a autorização do proprietário da informação.

- Disponibilidade

  Proteger os recursos computacionais de forma que eles não sejam indisponibilizados sem autorização.

- Consistência

  Garantia de que o sistema se comporta da maneira esperada perante os usuários autorizados. Considerar, por exemplo, os possíveis danos causados por um comando do tipo `ls` ou `dir`, que ao invés de realizar sua operação normal e esperada (listar arquivos), apagasse os arquivos.

- Auditoria

  Da mesma forma que se preocupam com acessos não autorizados, às vezes os próprios usuários autorizados cometem erros, ou executam ações maliciosas. Nesses casos, deve-se determinar o que foi feito, o que foi afetado, quem fez e quando o fez. A forma de se verificar esses resultados é através de auditoria, ou seja, verificação através de leitura de registros gravados pelo sistema. No decorrer deste trabalho usa-se também o termo auditoria como a função de verificação de configurações individuais de cada computador em uma rede.

Mesmo sabendo que todos esses aspectos e requisitos de segurança são importantes, diferentes organizações têm uma visão diferente para cada item, definindo prioridades diferentes. Em um banco, o requisito integridade teria prioridade máxima, garantindo por exemplo, que saldos bancários e valores de poupança não sejam modificados. Já em uma entidade autenticadora de cartão de crédito, o item disponibilidade seria o mais importante, pois alguns segundos ou minutos que o sistema fique fora do ar ou indisponibilizado, acarreta grandes prejuízos.

Um ponto crucial é definir a política de segurança através de um documento que deve ser claro e simples, do conhecimento de todos os usuários do sistema e que enfoque todos os pontos

referentes à segurança. Uma boa política de segurança é aquela que tenta abordar todas as situações com relação a segurança envolvendo a rede ou recursos da mesma [22]. Alguns exemplos de itens que são definidos em uma política de segurança:

- o que é e o que não é permitido aos usuários em termos de segurança,
- normas e obrigações dos usuários,
- política de senhas,
- descumprimento das leis e punições,
- política de *backup*,
- plano de contingência.

2.3 Ciclo de vida do projeto

Quando se aborda o termo segurança, deve-se imaginar um cenário mais amplo que simples equipamentos e produtos que se propõem a proteger uma determinada rede de computadores.

Segurança não é uma tecnologia, é um processo [49]. Não é possível comprar um dispositivo que torne a rede totalmente segura, assim como não é possível criar um software capaz de tornar um computador 100% seguro. A falácia dessas promessas se baseia na implicação da segurança ser um estado que se pode alcançar. Isso não é possível. A segurança é a direção em que se deve viajar, mas nunca chegando de fato ao destino. O que é possível fazer é administrar um nível aceitável de risco, tentando fazer com que o mesmo seja o máis próximo possível de zero, ou seja, de pequeno risco.

O risco é uma medida numérica ou relativa, que qualifica ou quantifica a probabilidade de ocorrência de um incidente. Embora possa ser considerado que um risco deva ser uma medida quantitativa, consideramos que a obtenção deste número nem sempre seja viável ou factível. A utilização de conceitos tais como "pequeno", "médio" ou "alto" por vezes são mais adequados que a busca de uma medida numérica.

Segundo [49] o ciclo de vida para um projeto de segurança ou implantação de uma política de segurança possui 3 fases:

- auditoria e análise de riscos

  A fase de análise de riscos é sempre indicada como a primeira etapa de qualquer solução de segurança. Quais são as ameaças e vulnerabilidades do sistema? O objetivo desta fase

é consultar e analisar as configurações de segurança atuais da rede e detectar quais computadores não estão de acordo com a política de segurança aplicada na rede. Suponha que o administrador conheça um conjunto de vulnerabilidades. Nesta fase de auditoria o administrador vai realizar uma busca em todas as máquinas de sua rede para identificar quais máquinas devem ser configuradas para melhorar o nível de segurança.

- Modificação ou implantação da política de segurança

  Após realizar uma auditoria e descobrir quais máquinas não estão de acordo com sua política de segurança o administrador pode implantar novas configurações corrigindo as vulnerabilidades das máquinas detectadas na etapa de auditoria e elevando o nível de segurança. É fundamental entender que não adianta apenas implantar uma política de segurança, é necessário manter a política em uso e que a mesma seja do conhecimento de todos usuários do sistema. Para isso, periodicamente o administrador deve fazer auditoria na rede e corrigir as configurações dos computadores que não estão de acordo com a política de segurança seguida pela organização.

- Avaliação

  O intuito dessa fase é avaliar as etapas de auditoria e de configuração, verificando ocorrências de possíveis erros durante o processo como por exemplo, detectar quais máquinas estavam desligadas ou sem conexão durante as etapas anteriores.

Essas fases são independentes e explicitamente separadas com o intuito de manter simplicidade. Nota-se que nem sempre uma fase depende diretamente de outra, i.e., para modificar as permissões e implantar uma política de segurança, não é necessário iniciar pelas fases anteriores. Pode-se implantar a política de segurança sem auditar as configurações existentes, porém o sucesso do processo como um todo fica mais restrito. Entretanto não faz sentido iniciar o processo pela fase de avaliação. Na Figura 2.1 está ilustrado o ciclo de vida de um processo de segurança.

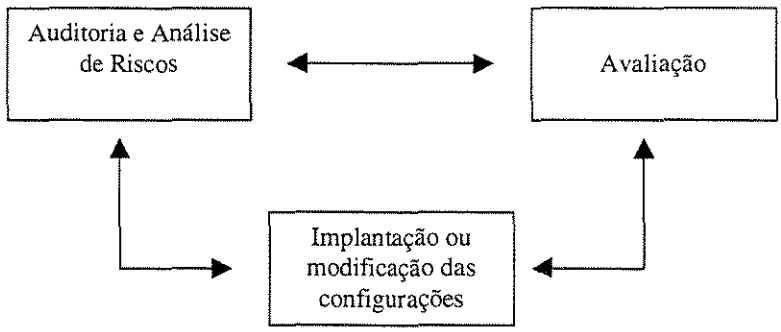

Figura 2.1: Ciclo de vida de um processo de segurança

2.4 Conclusão

Este capítulo foi dividido em duas partes. A primeira parte apresentou a área de administração de redes, citando algumas de suas subtarefas. Já a segunda parte apresentou a área de segurança de redes, descrevendo os requisitos de segurança, citando alguns itens necessários em uma política de segurança e detalhando o ciclo de vida de um processo de segurança.

Capítulo 3

Administração e Segurança do Windows NT

Inicialmente este capítulo apresenta uma breve história do sistema operacional Windows NT e a relação entre sistemas operacionais e segurança. Em seguida, descreve com detalhes uma das estruturas internas mais importantes do NT, o *Registry*. O capítulo encerra apresentando o objetivo principal desse trabalho: a necessidade de automatizar tarefas de maneira escalável nos ambientes NT.

3.1 Uma breve história do NT

Segundo a Microsoft, o sistema operacional Windows NT foi projetado para ser um sistema operacional robusto, portável, flexível e de fácil manutenção [15] [54] [55].

O Windows NT é o resultado de uma cisão entre a Microsoft e a IBM durante o desenvolvimento do sistema operacional OS/2. Enquanto os engenheiros da Microsoft projetavam o sistema operacional da Nova Tecnologia (NT, de *New Technology*), a IBM em parceria da Microsoft continuaram trabalhando para melhorar o OS/2.

Em 1990 a Microsoft lançou o Windows 3.0, sucesso absoluto de vendas, ofuscando o nascente mercado do OS/2. Um ano depois, após consolidar o mercado dos PCs, a Microsoft rompeu com a IBM, passando assim para o desenvolvimento de versões avançadas para a família Windows. Foi especificado que a versão do NT seria um sistema de 32 bits, portável, multiprocessável, preemptivo, multitarefa, porém compatível com aplicações de 16 bits [26]. Quanto à estabilidade, o sistema é considerável estável pois executa cada aplicação em um espaço sepa-

rado de memória, o que não acontecia com versões anteriores dos sistemas Windows. Com essa característica, o mau funcionamento de uma aplicação não afeta as outras.

Projetado para integrar uma rede, o Windows NT divide-se em dois produtos: no modo cliente, conhecido por Windows NT *Workstation*, o sistema padrão traz programas clientes de: ftp, correio eletrônico, telnet, etc. E no modo servidor, conhecido como Windows NT *Server*, o qual é um sistema operacional poderoso e escalável, desenvolvido para organizações que necessitem implementar uma rede com funcionalidades de servidor de usuários, servidor de arquivos, servidor de impressão, etc. O NT *Server* traz aplicativos como por exemplo: servidor web (IIS, de *Internet Information Server*), serviços de rede (DNS, DHCP, RAS).

Um dos atrativos dos sistemas Windows é a interface gráfica (GUI, de *Graphical User Interface*). Até a versão Windows 3.11 e Windows NT 3.51, a interface dessas versões nunca sofria alteração. Porém, com o lançamento do Windows 95, a Microsoft, após alterar e inovar a interface gráfica padrão, supôs que depois que uma pessoa usasse essa nova versão de interface por um período de tempo, ela consideraria antiquada a antiga interface gráfica do Windows NT 3.51. Para resolver isso a Microsoft liberou o mesmo modelo de interface GUI do Windows 95 na nova versão do Windows, chamado de Windows NT 4.

3.2 Relação segurança vs. sistema operacional

Vários motivos podem levar à adoção de um determinado sistema operacional: facilidade de utilização, interfaces, possibilidade de alterar códigos fontes, segurança, entre outros.

A segurança depende muito do sistema operacional, mas não somente dele. Pode-se dizer que a segurança depende mais do administrador que do sistema operacional. Qualquer sistema operacional bem administrado, com correções atualizadas do sistema e uma boa política de segurança em uso, pode ser considerado mais seguro que outro sistema mal administrado.

Em [30] é feita uma comparação de fatores como funcionalidade, confiabilidade, administração e desempenho de sistemas Windows NT e UNIX. O autor afirma que: assim como o UNIX pode comunicar-se com outros tipos de computadores, o NT também o pode; da mesma forma como o NT pode aplicar segurança a dados sensíveis e usuários, o UNIX também pode, e conclui que um sistema operacional é mais seguro que outro se ele for melhor configurado e administrado.

Seja qual for a escolha, sistemas operacionais e os próprios usuários devem ser capazes de proteger arquivos, memória e configurações contra modificações ou mesmo leitura por usuários não autorizados. A segurança de um sistema operacional inclui mecanismos de controle de acesso ao sistema e aos recursos. Entretanto, também deve-se incluir formas de proteger o sistema para que usuários não executem ações privilegiadas, isto é, ações a que estes usuários não têm direito, permissão ou autorização, e que apenas os administradores poderiam executar, por exemplo, reiniciar o computador ou incluir novas contas de usuários.

3.3 Registry

O Windows NT oferece um mecanismo de segurança unificado que pode ser usado para proteger os recursos do usuário de acessos não autorizados [26]. O modelo de segurança do Windows NT é baseado na existência de uma entidade unificada de segurança, chamada domínio. Um domínio é uma organização lógica de servidores de rede e *workstations* que compartilham informações comuns de segurança e contas de domínio. Dentro dos domínios, os administradores criam uma única conta de acesso para cada usuário, a qual permite "logar" no domínio, e não em servidores individuais dentro do domínio, pois a validação das contas é feita no controlador [42].

Existem os controladores primários de domínio (PDC, de *Primary Domain Controller*) e os controladores secundários de domínio (BDC, de *Backup Domain Controller*). Os BDCs são passivos, i.e., permanecem na rede e obtêm replicações de quaisquer modificações feitas no PDC. Caso o PDC se torne incomunicável, os BDCs, podem responder como se fossem um PDC, permitindo operação continuada do domínio. Em um ambiente de domínio as *workstations* são gerenciadas pelo PDC.

Quando bem configurado, o Windows NT se revela capaz de prover um bom nível de segurança [45]. Contudo, certas tarefas necessárias para se atingir um grau desejável de segurança revelam-se complexas e cansativas, devido à quantidade de pequenas ações.

Como em qualquer outro sistema operacional, não se pode pensar em segurança se existem nele vulnerabilidades conhecidas e que ainda não foram corrigidas. Para cada vulnerabilidade descoberta no Windows NT, a Microsoft disponibiliza um *hotfix*, isto é, uma correção específica para este problema. Periodicamente, a Microsoft agrupa todos os *hotfixes* lançados em um pacote maior conhecido como *service pack* [41]. É essencial que as máquinas estejam utilizando a versão

mais atualizada do *service pack*, mas só isso não garante um nível de segurança aceitável. Além do último *service pack* instalado, deve-se verificar por *hotfixes* disponibilizados após a versão do *service pack* instalado e garantir que uma boa política de segurança esteja ativa na rede. A implantação desses *hotfixes* ou *service packs* faz com que as configurações do sistema sejam corrigidas para um nível aceitável de segurança até aparecerem outros *hotfixes* e *service packs* [12].

A maior parte das configurações corrigidas pelos *service packs* estão contidas no *Registry*, que é um banco de dados central ao host, no qual todas as informações de configurações de *hardware* e programas do sistema estão armazenadas [19]. Cada máquina com Windows NT possui um *Registry* local.

Nas versões anteriores dos sistemas Windows, estas informações eram armazenadas em vários arquivos de configuração com extensão .ini e .sys, por exemplo o arquivo Win.ini que controlava as funções do sistema operacional enquanto que o arquivo System.ini controlava as aplicações e as configurações da área de trabalho [42]. Existiam vários outros arquivos com as mesmas extensões, os quais eram responsáveis por aplicações específicas. Um problema com relação aos arquivos de configuração é que qualquer usuário poderia facilmente editar esses arquivos com um editor de texto convencional, tornando-os inutizáveis e causando assim a indisponibilidade das aplicações controladas por esses arquivos. Além disso, havia o problema de segurança, a facilidade em ter acesso ao arquivo e as informações nele contidas. Frente a estas limitações, as versões mais recentes do sistema operacional e das aplicações têm buscado centralizar todas suas configurações no *Registry*.

Toda entrada do *Registry* controla uma função do usuário ou do computador. Um exemplo de função do usuário, é a configuração da sua área de trabalho. Função do computador é algo como a instalação de um programa ou *hardware*, alterações que realizadas naquele computador serão comuns a todos os usuários do sistema. Essa base de dados de configuração do sistema local oferece ao usuário uma visão hierárquica, composta de chaves e valores, similar a encontrada num sistema de arquivos, com diretórios e arquivos.

É no *Registry* que se encontram todas as informações sobre contas de usuários, grupos a que o usuário pertence, direitos e permissões, além de conter informações sobre todos os equipamentos de *hardware*, aplicações e protocolos de rede instalados, etc.

Qualquer alteração no *Registry* afeta diretamente a configuração da máquina. Se o *Registry* for danificado, o acesso ao *hardware* e aos programas podem ser drasticamente limitados, exis-

tindo a possibilidade do sistema não conseguir mais iniciar. Mesmo em caso de um problema menor, uma aplicação pode não funcionar e não responder como supostamente deveria.

A estrutura do *Registry* é constituída por 5 subárvores (*subtrees*) [42]. Essas subárvores são diferenciadas pelas informações contidas em cada uma delas. Em uma das subárvores estão armazenadas as informações do computador, as quais incluem dados sobre *hardware* e programas instalados. Em outra subárvore, estão gravadas as informações e perfis de usuários como por exemplo: as configurações da área de trabalho (*desktop*), as preferências individuais para determinados programas, as configurações da impressora pessoal, etc. A relação abaixo identifica e define cada subárvore [19] [31] [41]:

- HKEY_LOCAL_MACHINE

  Subárvore onde estão contidas algumas configurações sobre *hardware*, programas, sistema operacional, memória, etc. As informações sobre segurança, direitos e compartilhamentos também estão armazenada nessa subárvore.

- HKEY_CLASS_ROOT

  Contém as informações necessárias para carregar as aplicações. Dentre essas informações estão contidas as associações entre aplicações e tipos de arquivos (extensão do arquivo), os nomes dos drivers, os ícones usados pelas aplicações e documentos, etc.

- HKEY_CURRENT_USER

  É onde está armazenado o perfil de usuário para o usuário que atualmente tenha efetuado o "logon", incluindo variáveis de ambiente, configurações da área de trabalho, preferências de aplicativos, conexões de redes.

- HKEY_USERS

  Essa subárvore contém todos os perfis de usuários atualmente carregados. Apesar dessa subárvore conter configurações de todos os usuários separadamente, cada configuração só é acessível pelo usuário a ela associado. Os usuários que estão tendo acesso remoto a um servidor não tem perfis sob essa chave no servidor: para esses usuários os perfis são carregados no *Registry* de seus próprios computadores.

- HKEY_CURRENT_CONFIG

  Nesta subárvore se armazenam as informações sobre configurações de *hardware* utilizadas pelo computador local na inicialização.

O *Registry* pode ser acessado (consultado e configurado) remotamente [19]. Porém apenas 2 das 5 subárvores são acessíveis através da rede: HKEY_LOCAL_MACHINE e HKEY_USERS. Na área de segurança, a maioria das alterações necessárias para viabilizar um nível aceitável de segurança devem ser feitas na chave HKEY_LOCAL_MACHINE. Caso o administrador precise alterar valores de chaves que não são exportadas, o capítulo 4 apresenta 3 técnicas capazes de realizar essa tarefa remotamente.

3.4 Automatização de tarefas

Por muito tempo o Windows NT desfrutou de sua interface gráfica intuitiva para administrar um único sistema. Porém, com o aumento do número de servidores e da sua dispersão geográfica, algumas das deficiências arquitetônicas do NT para administração do sistema ficaram mais aparentes.

O argumento de que o Windows NT é fácil de administrar devido à sua interface gráfica é questionável. Infundado também é o argumento de que uma interface gráfica torna mais simples o trabalho de administrar se comparado com uma interface de linha de comando (CLI, de *Command Line Interface*), onde o administrador tem que digitar os comandos manualmente [24].

A dificuldade não é a interface gráfica em si, mas sim os programas de código fechado que rodam sob ela. Geralmente é difícil expandir ou utilizar uma aplicação gráfica em uma automatização de tarefas no NT. Suponha, por exemplo, um programa com interface gráfica de código fechado que gerencie as permissões de disco de um computador NT. Se esse programa, em sua interface gráfica, não tiver a opção de configurar computadores remotos da rede e se caso o administrador tentar automatizar essa tarefa através de *scripts*, ele não conseguirá realizar essa função se não tiver acesso à documentação de implementação ou ao código fonte do programa, pois dificilmente ele saberá qual função chamar ou qual linha de comando executar.

A grande barreira encontrada na administração do Windows NT aparece quando existe uma grande quantidade de computadores na rede. A dificuldade e o tempo gasto na administração é diretamente proporcional a essa quantidade. Quanto maior o número de estações na rede, mais complexa e mais demorada será a administração. Administrar o sistema Windows NT é uma tarefa complexa devido à carência de ferramentas de automatização de tarefas e de administração remota [18] [24] [25] [47].

A melhor solução para esse caso seria automatizar o processo de maneira escalável, isto é, executar a tarefa automatizada em todos as máquinas da rede independente do número de computadores.

Todo processo de automatização tem como objetivo excluir ou minimizar a participação do componente humano durante a realização das tarefas. No processo de automatização de tarefas em uma rede, não deveria ser necessária a presença física do administrador em cada máquina da rede. Isto é um problema em ambientes Windows NT que possuem computadores clientes (*workstations*) às vezes localizadas em prédios e localidades distantes entre si. A necessidade da presença física do administrador em cada computador da rede, além de requerer muita mão-de-obra, gera:

- um custo elevado,
- um tempo muito maior para finalizar as tarefas,
- possibilidade de erro manual durante o processo.

Como dito anteriormente na Seção 1.2, página 5, a ferramenta VNC permite o administrator gerenciar a rede remotamente sem ter que estar presente em cada máquina. Porém para atualizar ou realizar qualquer tarefa, o administrator precisa fazer a mesma tarefa em cada computador da rede. Um erro manual durante o processo de implantação de uma política de segurança em uma grande rede pode ocasionar em um resultado catastrófico e pode por abaixo todo o tempo, dinheiro e trabalho dos administradores. No objetivo desse trabalho, o VNC não serve como solução pois não automatiza as tarefas.

A segurança de uma rede pode ser medida pelo seu computador mais vulnerável e exposto [9]. Portanto, de nada importa que 99% dos computadores de uma rede estejam bem configurados com relação à segurança se um deles for vulnerável ou possuir um nível de segurança baixo. Isso significa que caso o administrador cometa um erro manual em uma das máquinas durante a configuração de segurança, toda a rede pode estar vulnerável por causa de um simples erro em uma das máquinas.

No modelo tradicional, a segurança em redes limitava-se aos extremos da rede, garantindo segurança máxima nos servidores e confiando em tudo e em todos no interior da rede. É como imaginar um castelo antigo com um lago ao redor e com uma ponte levadiça. Dentro do castelo não há fechaduras e todos que estão dentro confiam uns nos outros.

Já no modelo atual existe um novo paradigma. A segurança não deve ser mantida somente nos servidores, mas sim em todos os computadores [44]. Cada computador da rede deve ser o máis seguro possível sem impacto negativo quanto à sua utilização. Neste caso, imagine uma grande cidade moderna onde ninguém confia em ninguém e todas as portas têm no mínimo uma fechadura. Uma coisa é proteger uma máquina, outra coisa bem diferente é proteger uma rede com centenas ou milhares de computadores.

3.5 Conclusão

Este capítulo apresentou uma breve história do sistema operacional Windows NT. Em seguida relacionou a área de segurança com o sistema operacional. Logo após detalhou o *Registry*, componente principal de todas máquinas Windows com relação a segurança. E por último apresentou o tópico principal dessa dissertação, a necessidade e os benefícios trazidos através da automatização de tarefas nas áreas de administração e de segurança de redes.

Capítulo 4

Técnicas Propostas para Automatizar Tarefas de Administração de Redes Windows NT

Este capítulo apresenta algumas das principais recomendações de segurança para o sistema operacional Windows NT e descreve com detalhes 3 técnicas para automatizar o problema da área de administração de redes, principalmente a tarefa de instalar ou atualizar programas remotamente.

Prólogo

Este capítulo é composto pelo artigo *"Administration Techniques for Implementing Security on Large Windows NT Networks"*, publicado nos anais do 2° Simpósio de Segurança em Informática (SSI' 2000), realizado em São José dos Campos, São Paulo, em outubro de 2000 [7].

O início do artigo contextualiza o problema descrito nos capítulos anteriores com relação à segurança e administração remota. Sem perda de continuidade, o leitor poderá pular as seções 4.1 a 4.4.

A seção 4.5 apresenta as principais recomendações de segurança para possibilitar o Windows NT um nível de segurança semelhante a C2 do departamento de defesa dos EUA [11] [35] [38] [40] [43] [45]. Um sistema com nível de segurança C2 (proteção por acesso controlado), entre outras coisas, deve ser capaz de: permitir ou negar uso e/ou acesso a recursos do sistema para certos usuários ou grupos de usuários; garantir que quando um bloco de memória é liberado,

seu conteúdo é explicitamente sobrescrito antes de ser designado a outro processo; proteger-se de modificações a arquivos e componentes do sistema. Em novembro de 1999, o Windows NT com *service pack 6a* e atualização C2 foi incluído na lista dos sistemas operacionais de nível C2 [45].

A seção 4.6 detalha técnicas de administração para tornar o processo de instalação e/ou atualização remota de programas em Windows NT simples e principalmente escalável. A habilidade de controlar máquinas remotamente, porém uma de cada vez, não é uma solução conveniente para administrar grandes redes [58] [60]. É desejável possuir ferramentas que de forma simples permitam gerenciar todas as máquinas da rede ao mesmo tempo, independente do número de máquinas [10] [25].

A título de exemplo prático suponha que o administrador precise instalar um programa localmente em cada máquina da rede, por exemplo `abc.exe`.

As técnicas descritas nessa seção utilizam o conceito de clonagem e pacote. O objetivo da etapa de clonagem é criar um arquivo, chamado pacote, composto por todas as modificações e programas que serão instalados ou atualizados nas máquinas da rede. Para criar os pacotes, será utilizada a ferramenta `sysdiff.exe`, a qual vem junto com o sistema operacional Windows NT. No exemplo prático, inicialmente o administrador precisa escolher alguma máquina da rede, em seguida instalar o programa `abc.exe` nessa máquina e com a ferramenta `sysdiff.exe` o administrador verifica quais configurações do *Registry* e quais arquivos do disco rígido dessa máquina foram alterados ou adicionados. Tudo o que sofreu alguma alteração estará presente no pacote, o qual será instalado no resto das máquinas.

Após criar e disponibilizar o pacote no servidor, o administrador deve escolher qual técnica vai utilizar para aplicar esse pacote em cada máquina da rede. A primeira técnica, na Seção 4.6.2, página 41, denominada "técnica da conta especial" é utilizada com sucesso no Instituto de Computação da Universidade Estadual de Campinas há algum tempo [28]. Ela requer que o administrador crie uma nova conta de acesso no PDC com direitos administrativos e por questões de segurança essa conta deve ser configurada para executar apenas o *script* de instalação do pacote e em seguida desconectar do sistema. Desta forma, basta o administrador ir fisicamente até cada máquina da rede e efetuar "logon" na conta criada. Com isso, a única tarefa possível de ser realizada com essa conta é executar o *script*, o qual servirá para conectar a máquina local com um diretório compartilhado do servidor e instalar o pacote.

A Seção 4.6.3 descreve a segunda técnica, "técnica dos serviços NT", a qual requer que o administrador crie um novo serviço em cada máquina da rede para que, quando a máquina for ligada, ela se auto-atualize instalando o pacote sem que o usuário tome conhecimento. Nesse caso toda vez que a máquina for ligada, o serviço é executado. Esse novo serviço irá conectar a máquina local com o servidor e verficar se existe algum pacote disponibilizado pelo administrador e que ainda nao foi instalado na máquina local. Caso positivo, essa tarefa é executada automaticamente nesse momento. Caso contrário, a máquina segue sua inicialização normal.

A terceira técnica, apresentada na Seção 4.6.4, página 43, e denominada "técnica do serviço *schedule*" é a que requer menos presença física do administrador em cada computador da rede. Ela possibilita o administrador agendar remotamente um *script* de instalação de pacote para ser executado em horários preestabelecidos, evitando assim sobrecarregar o tráfego da rede. Para utilizar essa técnica o administrador precisa iniciar o serviço *schedule* em cada máquina da sua rede. Isso pode ser feito remotamente. Após iniciar esse serviço, o administrador pode agendar tarefas em cada máquina cliente. No exemplo prático anterior, após iniciar o serviço em cada máquina cliente, o administrador poderia agendar um *script* para conectar as máquinas clientes com o servidor e aplicar o pacote localmente.

O artigo apresentador a seguir detalha desde a criação do pacote até um exemplo prático da utilização dessas técnicas.

Administration Techniques for Implementing Security on Large Windows NT Networks

Alessandro Augusto,
Célio Cardoso Guimarães, Paulo Lício de Geus

IC - UNICAMP
University of Campinas - Campinas, SP, Brazil
alaugusto@yahoo.com.br, {celio, paulo}@ic.unicamp.br

Abstract

The process to secure a Windows NT computer can be easy if the administrator knows which configurations and security settings he needs to do. But, even when the administrators knows the changes that needs to be done on a single NT computer, the process to apply the same configuration in an environment with hundreds of NT-based computers can be really frustrating. Most solutions to this problem require some expensive tool such as Systems Management Server (SMS). But there are many companies and institutions that cannot purchase this kind of tool or the SMS's licenses. In this case, the solutions presented until now don't solve the problem of administering NT and applying security to a NT network. This paper describes some highly security recommendations and propose three solutions to solve the difficulty to apply security or to upgrade NT-based computer networks without any extra tool.

4.1 Introduction

During the last years it is unquestionable the advantages that institutions had with the increase in the use of computers, with the interconnection of these computers in networks and with the sharing of resources. Even so, it is also unquestionable that the institutions need to be prepared before migrating to this new "digital territory".

Thus, these have been discussion a lot on system administration and security, especially in UNIX operating systems, but little aspect deals with Windows NT operating system. 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.

In spite of easiness to use the system, comparing Windows NT with other UNIX systems, NT can be considered "lacking" in the subject of network administration, especially when the topic is applying security on a NT network.

In institutions that have a considerable group of interconnected computers through a network based on Windows NT, it always existed difficulties when the administrators need to do apply some security configurations on each network computer. These difficulties generate high monetary costs to maintain a group of system administrators in service.

Windows NT's environments has a reputation to be a system requiring hands-on administration, that is, it needs a manual by-hand work [24]. It is necessary the administrator's physical presence in each one of the machines every time it needs to do some modification or configuration. With that, it can be concluded that the associated costs would increase as the amount of the network computer gets bigger. Remote software installation and configuration is another problem in this kind of environment.

A large portion of configuring security on NT is modifying some Registry values. As the administrator begins to look at configuring values keys on the Registry and to read the papers about NT security, he starts to think that it is almost impossible to administer NT-based computer networks without some expensive administrative tool such as SMS (Systems Management Server) and without a large number of system administrators available [33]. For a practical example, the usual software installation methods on NT requires the administrator to sit in front of an individual machine, answer some questions interactively, wait some minutes for the software to load and maybe reboot the machine. This approach doesn't scale to hundreds of NT machines. With this example, the administrator can't imagine the problem that he will face when he starts to configure security. He knows what he needs to modify on the Registry, but how does he do all this modifications without sitting in front of each computer?

The goal of this work is to define a good level of security for NT computers and also, to create techniques and propose solutions, where the system administrator is able to configure the security and have it automatically distribute to each machine of a given type. But there are some restrictions about these techniques, one of the restrictions is that the administrators cannot copy the whole Registry and paste it on another computer.

Fortunately, there are ways to by-pass most of these problems. As a result, the solutions presented here solve this problem of applying security on NT network and also, techniques to facility any software installation, software upgrading and also any other kind of communication between the workstations and the server without expensive tools such as SMS. For another example, if the administrator needs to know the exactly time that each network computer was turned on, he can create a batch script that logs the time and use this script with one of the proposed solutions to send these logs to the administrator.

The techniques presented in this paper can be considered good solution for institutions that don't want spend money with this kind of problem. With these techniques, system administrators can deploy up to 100 PC's per hour depending on which technique he chooses and also, it depends of the package size that is being installed. These techniques will increase a lot the deploy ratio and turn much more easily the administration process.

This paper is structured. After a brief introduction on Seção 4.1, it describes the Windows NT operating system on Seção 4.2. After that, it describes the Registry and the problem on how to administering the NT network, respectively on Seção 4.3 and Seção 4.4. Seção 4.5 presents some security recommendations. Seção 4.6 proposes and explains the solutions to administer the NT networks. Seção 4.7 shows a practical example with its solution of how can the administrator applies some security configurations to a NT environment. The paper finishes it with the conclusion and the references used by it.

4.2 Windows NT

Since its initial release in 1993, the operating system Windows NT appeared as an outstanding operating system with multiple purposes. Projected to integrate a client-server network, Windows NT is divided in two products: Windows NT Workstation and Windows NT Server [42].

Combining an application server with a file system and a print system, it was created to be easy to use and to manage. Besides that, it is much more reliable and stable than the previous versions of the systems Windows 9.x and Windows 3.x.

The client is known as Windows NT Workstation. The default system already brings applications to execute in the network such as ftp clients, electronic mail, telnet, etc. In the same way, NT Server default system brings some different applications, for example a web server, IIS.

In Windows NT, all configurations are stored centrally in only one database denominated Registry, which is one of the most important topics about this system, especially when deals about security [31].

4.3 Registry

Registry is a central and organized database that contains all the information about *hardware* and software configuration.

In previous versions of Windows, configuration files with extension .ini and .sys executed the functions exercised by the current Registry. The problem of these configuration files was the restriction with relationship to its maximum size to be of 64 Kb. Beyond that problem, any user could easily edit some configuration file and could cause damage on it [31].

Inside of the Registry is stored all information about user's account, user's groups, besides information about all hardware and software installed in the computer.

To modify the Registry values, it is necessary to have writing permission, because each of its items has access permission. Every change in the Registry affects the configuration of the machine directly.

Developed with a hierarchical structure, the Registry can be compared with a country, which it is divided in states, which is divided in cities, in neighborhoods and so on.

4.4 The Problem

One of the hardest tasks that system administrators have with NT environment is to configure the security of its network and to install or upgrade software. Some people don't agree with this, they say it's much easier to install an application under Windows NT than under UNIX. On NT, the administrator just need to put the CD or the floppy in the drive, maybe click setup (if autorun is not configured automatically), answer some Installshield questions and wait for it goes to work [24]. These people would expect that UNIX software management would be much harder, since there is no installshield there.

The argument that NT is easier to administer due to its graphic interface (GUI-based) it is questionable. Besides being questionable due other operating systems also possess graphic interface, it is also doubtful the argument that the graphic interface is simpler than command line

interface (CLI-based). Generally, GUI-based tools are easier to install but harder to automate and extend [24].

However, the fact that software installation under NT is easier than UNIX can be considered true for an isolated NT machine. Managing a large environment is entirely different. Command-line tools are easier for any system administrator to use when managing a large environment.

The difficulty found in Windows NT administration occurs when the environment, which is the amount of network computers, is larger than 1. The difficulty and the time spend in the administration is directly proportional to the amount of computers. The larger is the amount of computers in the environment, more complex and delayed will be its administration.

Another item that hinders the administration of Windows NT networks, is the fact of having a heterogeneous network, that is, when there are computers with different hardware profiles.

In spite of the same security configuration in two different machines add or modify the same keys and fields in the Registry, it is impossible to copy the Registry of the first computer and paste into another NT computer where it was not configured yet. The impossibility is because of the remaining keys, which contains hardware and software configurations specific to each computer.

The two main problems here are: (1) what changes and security settings should the administrator do to make a NT computer secure? (2) How to apply these settings to the whole NT network computers without having to sit in front of each machine? How to administer a Windows NT network and its security in some easy way, also with a cheap solution and fast results, considering the amount of computers present in the network is larger than 1? How to do the needed modifications in all the network machines with a small effort and in the least possible time? How to automate the tasks of software installation or upgrading to the remaining network machines? The solutions for these questions are describe in the next sections.

4.5 Security Settings Recommended

Windows NT provides a rich set of security features, however, the default configuration is highly relaxed. This is because the operating system is sold as a shrink-wrapped product with an assumption that an average customer may not want to worry about a highly restrained but secure system on their desktop. This assumption has changed over the years as Windows NT gains popularity largely because of its security features [43].

This section describes the most important security recommendation settings. It will follow some recommendations of a Windows NT C2 configuration [40] [45].

A particular installation's requirements can differ significantly from another. Therefore, it is necessary to evaluate the environment and requirements before implementing a security configuration.

Windows NT allows the administrator to establish a full range of security levels, from no security at all to the C2 level of security. These levels are arbitrary, and the administrator will probably want to create his own level by blending characteristics of the levels presented in this section.

One reason to not have maximum security level at all times is that the limits the administrator sets on access to computer resources make it a little harder for people to work with the protected resources. And if the security is too tight, users will try to circumvent security in order to get work done [43].

The first step in establishing security is to make an accurate assessment of the needs. Then choose the elements of security that the administrator wants, and implement them.

The following subsections describe some recommendations to apply security configuration on Windows NT.

4.5.1 Operating System and Service Pack Installation

The first step to start armoring the NT system is the operating system (OS) installation. Install it on a NTFS file system. With NTFS, the administrator can assign a variety of protections to files and directories, specifying which groups or individual accounts can access these resources in which ways. During the OS installation, select only the services that will run and the protocols that will be need. The fewer services that are running, the fewer exploits or security issues the system will have [43].

Following the installation, install the latest service pack (current service pack 6a - october 2000). Staying current with the latest exploits is critical for a secure system.

Once the administrator finishes the OS and the service pack installation, he can start to configure the system. All unnecessary devices and services must be disable. The services that should be enable depend of the needs.

4.5.2 OS/2 and Posix Subsystems

OS/2 and Posix are subsystems designed to run with other system but not specifically with Windows NT and that may not be able to take full advantage of all Windows NT features (such as memory management).

Most of the administrators don't need these subsystems, so it can be disabled. To remove OS/2 and POSIX subsystems, the administrator needs to delete the `\winnt\system32\os2` directory and make the following Registry configurations [40]:

Hive	**HKEY_LOCAL_MACHINE\SOFTWARE**
Key	\Microsoft\OS/2 Subsystem for NT
Action	Delete all sub keys

Hive	**HKEY_LOCAL_MACHINE\SYSTEM**
Key	\CurrentControlSet\Control\Session Manager\Environment
Value Name	Os2LibPath
Action	Delete

Hive	**HKEY_LOCAL_MACHINE\SYSTEM**
Key	\CurrentControlSet\Control\Session Manager\SubSystems
Value Name	Optional
Action	Delete values

Hive	**HKEY_LOCAL_MACHINE\SYSTEM**
Key	\CurrentControlSet\Control\Session Manager\SubSystems
Action	Delete entries for Posix and OS/2

Table 4.1: removing Posix and OS/2 subsystems

4.5.3 ShutDown Button

Normally, any user can shut down a computer running NT without logging on by choosing Shutdown in the Logon dialog box. This is appropriate where users can access the computer's operational switches; otherwise, they might tend to turn off the computer's power or reset it without properly shutting down. However, the administrator can remove this feature requiring users to

log on before shutting down the computer [40]. The configuration to remove the shutdown button from logon dialog box is on Table 4.2.

Hive	**HKEY_LOCAL_MACHINE\SOFTWARE**
Key	\Microsoft\Windows NT\CurrentVersion\Winlogon
Value Name	ShutdownWithoutLogon
Action	Set the value 0

Table 4.2: removing shutdown button from dialog box.

4.5.4 Files and Directories

Among the files and directories to be protected are those that make up the operating system software itself. The standard set of permissions on file system and directories provide a reasonable degree of security without interfering with the computer's usability. For a high-level security installations, however, the administrator might want to additionally set directory permissions to all subdirectories and existing files. To protect the files and directories, the administrator needs to use the ACL editor in Windows NT Explorer to change access on the system drive (by default "C:\") to grant full control to Administrators and SYSTEM, and grant read permission to Everyone. In [40] and [42] it gives the following recommendations:

Directory	**Permissions**
C:\	Administrators: Full Control SYSTEM: Full Control Everyone: Read
\WINNT	Administrators: Full Control SYSTEM: Full Control Everyone: Read CREATOR OWNER: Full Control
\WINNT\REPAIR	Permit only Administrators: Full Control
\TEMP	CREATOR OWNER: Full Control

Table 4.3: protecting files and directories.

Directory	Permissions
\WINNT\Profiles\<user>	User: Full Control
\WINNT\Profiles\administrator	Remove Everyone

Table 4.3: protecting files and directories.

4.5.5 Protecting the Registry

In addition to the considerations for standard security, the administrator of a high-security installation might want to set protections on certain keys in the registry. By default, protections are set on the various components of the registry that allow work to be done while providing standard-level security.

For high-level security, the administrator can assign rights to specific registry keys, but this should be done with caution, because programs that the users require to do their jobs often need to access certain keys on the users' behalf.

Normally, the keys in the registry are changed indirectly, through the administrative tools such as the control panel. The registry can also be altered directly with any registry editor.

Open regedt32.exe and grant full control to Administrators and SYSTEM and read access to Everyone for the followings Registry subkeys [40]:

- HKEY_LOCAL_MACHINE\Software: locks the system in terms of who can install software.
- HKEY_LOCAL_MACHINE\Hardware
- HKEY_LOCAL_MACHINE\System
- HKEY_USERS\.Default

4.5.6 Restricting Remote Access to the Registry

The default permissions do not restrict which users can have remote access to the registry. Only administrators should have remote access to the registry.

To restrict network access to it, select the hive `HKEY_LOCAL_MACHINE\System`, the key `CurrentControlSet\Control\SecurePipeServers` and the value name `winreg` and set the Administrators permission to full control, and make sure no other users or groups are listed [11]. Table 4.4 shows this configuration.

Hive	HKEY_LOCAL_MACHINE\SYSTEM
Key	\CurrentControlSet\Control\SecurePipeServer
Value Name	Winreg
Action	Administrators: Full Control

Table 4.4: restricting remote access to the registry.

4.5.7 Trojan Horses

Restrict untrusted users' ability to plant Trojan horse programs on the system. Trojan horses can take advantage of the Run utility if its is unguarded [36]. There are some trojan horses that are written to execute during an Uninstall operation.

To restrict the ability of users to plant trojan horses programs, set the values of Table 4.5.

Hive	HKEY_LOCAL_MACHINE\SOFTWARE
Key	\Microsoft\Windows\CurrentVersion
Value Name	Run, RunOnce, Uninstall, AEDebug
Action	Everyone and all untrusted users: Read

Table 4.5: protecting from trojan horses.

4.5.8 Share

To allow only the administrator to control which users can access a computer from its network interface and what information is shared over the network interface set read permission for Everyone and all untrusted users on the value of Table 4.6 [40].

Hive	HKEY_LOCAL_MACHINE\SYSTEM
Key	\CurrentControlSet\Services\LanmanServer
Value Name	Share
Action	Everyone and all untrusted users: Read

Table 4.6: shares.

4.5.9 Cache Logon

The default configuration of Windows NT caches the last logon credentials for a user who logged on interactively to a system [11]. Even though the credential cache is well protected, administrators may want to disable the cache. This results in a somewhat longer logon time, but prevents malicious users from tapping logon information from short-term memory. Table 4.7 shows how to disable caching.

Hive	HKEY_LOCAL_MACHINE\SOFTWARE
Key	\Microsoft\Windows NT\CurrentVersion\Winlogon
Value Name	CachedLogonCount
Action	Set the value 0

Table 4.7: disabling cache logon.

4.5.10 Hiding the Last User Name

By default, Windows NT places the user name of the last user to log on the computer in the user name text box of the logon dialog box. This makes it more convenient for the most frequent user to log on. To help keep user names secret, the administrator can prevent Windows NT from displaying the user name from the last log on [34]. To prevent it, the administrator needs to set the values of Table 4.8.

Hive	HKEY_LOCAL_MACHINE\SOFTWARE
Key	\Microsoft\Windows NT\CurrentVersion\Winlogon
Value Name	DontDisplayLastUserName
Action	Set the value 1

Table 4.8: hiding the last user name.

4.5.11 Fpnwclnt.dll

There is a security issue that may occur due to the way Windows NT handles the file `\winnt\system32\fpnwclnt.dll`. This file is a dynamic link library that let files and prints services for a netware and directory service manager for netware perform password synchroni-

zation with Novell netware servers. If there is no Novell netware servers on the NT network, this file should be removed [37].

4.5.12 User Rights and More Security Configurations

Above, the paper described a few recommendations to set a Windows NT as a C2 security level. There is a lot more security configurations that can be changed. There are several user rights that the administrators should be aware of and possibly audit. This permissions can be simple changed using the Microsoft Manager Console.

MMC integrates all the set of administration components. Together, these services provide a model of system administration and coherent delegation that reduces the time of administration. MMC hosts the programs, called snap-ins, that administrators use to manage their servers.

MMC allows administrators to configure account police, local polices, event log, restricted groups, system services, registry and file system.

The security template snap-in is a stand-alone Microsoft Management Console (MMC) snap-in that allows the creation of a text-based template file that contains security settings for all security areas.

4.6 Proposed Administratives Solutions

The first of the main problem present before is accomplished, which was the changes and security settings that the administrator should do to have a NT computer more secure then the default installation. Now the problem is how can the administrator applies these settings to the whole NT network computers without having to sit in front of each machine and configuring one by one?

To solve this problem, this paper proposes 3 techniques that will help the administrator. In order to decide on which technique would be easier to use, the system administrator needs to evaluate and consider as many options as possible.

The procedure for any of these techniques has basically two steps:

1. create a package that will be installed or applied
2. choose the technique to apply the package created

For a better understanding, the term "change" is standardized in this paper as being the task or the modification that the administrator wants to do in his network computers. This change can

be for example a simple modification in the standard wallpaper, or a new software installation, or to restrict the permission to read the Registry or any other alteration.

Also the term "model machine" defines the computer where the change was accomplished for the first time. This model machine can be any of the present computers in the network and it will be consider a reference during the whole process. The package will be created in this model machine.

"Target system" or "target machine" are the terms used to define the network computers where the changes accomplished in the model machine will be applied, that means, the computers where the package created will be installed.

Next section will detail the process to create the package.

4.6.1 Packaging

The heart of successful automated installations is the first step, which is called packaging. As the name says, packaging is the process to create a file, which contains all the changes that the administrator wants to apply in the network [21].

The packaging process is similar to the clone process.

With the impossibility of copying the whole model machine Registry to the target machines, the goal of the packaging is to create a system exactly equal, that is, to clone the model machine, including in the clone system all the configurations and security policy of the model machine, all them installed software and configured, besides considering each system as being a different machine in the network.

The initial idea of the clone process is to discover what changed in the NT system of the model machine after any change and to create a package just contends the modifications that happened after the administrator execute its change, it will be detailed better in the Seção 4.6.1.3. After creating the package, it needs to be applied in the target machines with some of the techniques that will be explained below.

When someone change any property in NT or when someone install a new software or hardware, new files and new keys are added and modified in the Registry of that system. The clone process seeks to discover which were those modifications. After discovering the modifications that were done in the model machine, it creates the package that will be applied in the target machines.

To discover the alterations happened in the model machine, it is necessary some application that does a "sweeping" of the file system and the Registry, to discover what changed in the system after the administrator's modification, and as result, create the package. In this paper the application used to sweep the system and to generate the package is called sysdiff.exe, which comes on the Windows NT cdrom.

4.6.1.1 Sysdiff.exe

Sysdiff.exe is a practical example of a tool that helps to clone NT systems. With this tool it is possible to discover all the new files that were added or modified in the file system and all the keys and values that were inserted or modified in the Registry.

The application Sysdiff.exe doesn't install the operating system, it just discovers the modifications happened in the model machine, it generates a package contends those changes and it creates an installation for that package.

To run correctly, there are some requires that need to be done for sysdiff.exe:

- It is necessary to define a "model machine", which is a reference computer, where the changes will be accomplished firstly. This model machine should run the same operating system of the target machines, where the package will be applied.
- It is also necessary to define a distribution point. A shared folder (share), where should be stored the application Sysdiff.exe and the package created. It is necessary that the target machines have access to this share.

The clone process should be fast enough, otherwise it can be unviable. Depending on the small number of network computers, the clone process can delay more than the accomplishment of the same task in all the target machines.

4.6.1.2 Options of the application Sysdiff.exe

Sysdiff.exe possesses the following parameters:

Option	Command Line
/Snap	Sysdiff /snap snapshot_file

Table 4.9: Options of sysdiff.exe.

With this option, the application takes a snapshot of the current system configuration. The parameter snapshot_file is already the name of the file in which the photo of the current configurations will be recorded.

Option	Command Line
/Diff	Sysdiff /diff snapshot_file package
This option generates the distribution package. That package is a file contends the differences found among the first snapshot token of the system, with the configuration of the system registered immediately after the changes are accomplished. The new parameter included in this option is the name of the package that will be create. This is the same package that will be applied in the target machines.	

Option	Command Line
/Apply	Sysdiff /apply package
This option is used to apply the package generated by the option /diff in the target system, that means, in the machine that is executing the application Sysdiff.exe.	

Option	Command Line
/Dump	Sysdiff /dump package dump_file
This option allows to create a report contends the modifications accomplished by a certain package. The parameter dump_file is the name of the file that will contains the changes applied by the package.	

Table 4.9: Options of sysdiff.exe.

4.6.1.3 Creating the package - step by step

After defining the necessary requires for Sysdiff.exe, described in the Seção 4.6.1.1, the steps to create the package are:

1. in the model machine, install the application Sysdiff.exe.
2. execute the application Sysdiff.exe with the option /snap to take a snapshot of the model machine before any change. This photo will contain all the configuration of the current NT system. This step should be done before the administrator configure any security setting.

3. Soon after, the administrator should accomplish the change wanted in the machine, e.g. accomplishes the security recommendations presented on Seção 4.5.
4. After accomplishing the configurations, execute the application Sysdiff.exe again with the option `/diff`. The objective of this step is to find the differences happened in the model machine and to create an installation package for the target machines. To create the package, Sysdiff.exe receives as entrance the snapshot took before the changes (step 2) and it supplies as result the generated package, which contains all the changes that were done on that machine by the administrator.
5. The last step is to leave the application Sysdiff.exe and the generated package in the share distribution folder (share).

With that, the process of creating the package is concluded. This stage is necessary to use with any of the techniques that this paper presents. Seção 4.6.2 presents the first one, which uses a login account. The other one, uses NT Services and will be present in Seção 4.6.3. And the last technique, presented in Seção 4.6.4 uses the schedule service.

4.6.2 Solution 1: Using a special account

This technique is considered the easiest for networks where the physical location of each computer is close to the others.

The goal of this technique is simple: it discovers the entries that were added in the Registry of the model machine, creates a package and apply it on the target machines using a batch script.

To discover the modifications done in the model machine, it uses the process described in the Seção 4.6.1.3 to create the package. This technique consists of:

1. Create an installation package for the changes that the administrator wants to do and leave it on a share drive (Seção 4.6.1.3).
2. Create a batch script (.bat), that connects the current machine to the above share and apply that package to this computer, the computer that is executing the batch (described on Table 4.10).
3. Create a new user account login with administrator's rights and configure this account to execute the script from the last step (step 2) when the system administrator logon in.

4. On each target machine, the administrator should logon in using the user account create on step 3.

The goal of the batch script (from step 2) is to apply the package on every network machine. Suppose that the administrator already created the package, leave it on a share drive and created the new user account login. After this, the administrator needs to go to each computer, and logon in this new account. Then, the account will execute the script, which will do:

(1) connect the current computer to the share drive

(2) execute the sysdiff.exe application with the option to apply the package (`sysdiff /apply`)

(3) logout the account

With these steps, any modification on the first machine will create a new package. So the process to install or customize anything on the network can be really easy. The system administrator just needs to go to each network computer and logon in this new account.

4.6.3 Solution 2: NT services

This technique uses the concept of NT services. It came up for the system administrator that cannot be present on each network computer every time he needs to install a new package or do any other modification.

It is a good solution for companies and institutions that have a large environment, where the network computers stay far away from each other.

The solution is similar to the first one, but in this technique, the system administrator will need to go to each computer only one time, to configure the service, and not all the time that he wants to apply a new package.

Installing a new service, NT is capable to self-upgrade when the machine is booted, without any user interaction. The only requirement in this technique is to turn on the machine, so the system starts the service automatically and upgrades itself [43].

Among the benefits offered with the installation of the new service, there are:

- Possibility to automate tasks without needing administrator's interaction
- When applications are executed as being services, the applications are not concluded in the moment that an user makes the logout of the machine. The service will execute even if the user logout the system.

- If the application that is executed as service is a client-server application, this application can respond commands all the time, even when there is no user logged in the machine.

The second solution proposed by this paper consists of the following steps:

1. Create an installation package for these changes and leave it on a share drive (Seção 4.6.1.3).
2. Create a new batch file (.bat) different from the first technique. The batch file here should connect the current machine to the share drive where the package is, and compare if it needs to apply that package or not. Maybe that package has already been applied to the current computer.
3. Create a new NT service on each network computer and configure the service to be the batch file from step 2 and to startup automatically when the computer turns on [39].

In this technique, the administrator needs to create a batch file which will compare the packages from the share with the packages already applied on the current computer that is running the service.

With these steps, the system administrator needs to configure each computer only the first time, when he creates the new service. This service will run automatically when the computer turns on. So, every morning, when any user starts his work, he turns the computer on, and doesn't even know that the system is automatically self-upgrading.

4.6.4 Solution 3: Schedule Service

The third technique described in this paper is very similar to the second solution. It was deployed for the system administrator that can't be present on each computer during the upgrading.

This solution requires that the system administrator goes to each network computer one time and configure the NT´s schedule service to startup automatically. With this configuration, this service will always be startup when the computer turns on.

The schedule service allows the system administrator to schedule, remotely or locally, any task at some specific time. Also known as the AT command, the schedule service is used to schedule tasks to run automatically at a present time.

The third proposed solution consists of:

1. Configure each network computer to start the schedule service automatically. This step is done just one time. The next times that the administrator applies different packages, this step can be ignored.
2. Create an installation package for the changes and leave the package on a share drive (Seção 4.6.1.3).
3. Create a batch file similar to the first technique (special account) which will connect the current computer to the share drive and apply the package on the current computer (Table 4.10 shows the script).
4. Schedule the batch file created above to run at a specific time. There are options to schedule it to run several times, every day, every week, so it can be handled by the system administrator choices.

In [24], the authors say they were unsure with the schedule service because of security aspects. With the right configuration, schedule service is secure because only the administrator can schedule tasks to execute.

4.7 Practical Example

For a practical example, suppose a NT-based network, with one server and 10 workstations. Also, suppose that the workstations are located at the same room and the server is in a different place, where the users don't have physical access.

The administrator wants to configure the security of each workstation and also wants to install a new software. The steps to automate this tasks are:

1. The administrator selects one of the workstations and installs the sysdiff.exe application.
2. Before he makes any change, he executes sysdiff.exe with the option /snap to take the first snapshot of the machine.
3. After take the snapshot, the administrator starts to configure the security and the installation of the software. The paper won't get in details about what security setting the administrators is changing, but he can follow the recommendations described on section 5 of this paper and apply it here. The administrator installs the software in this step too.

4. When its done, the administrator executes the sysdiff.exe with the option /diff to create the package.
5. Now he needs to create a share drive (folder) and put the package created and the sysdiff.exe there.
6. The administrator needs to choose one of the techniques. In this case, lets suppose that the administrator choused the first technique described in this paper, the special account technique (Seção 4.6.2). The reason is because this solution is easier for network where the computers are close to each other's.
7. The next step is to create the batch script that connects the current computer to the share drive, and execute the `sysdiff.exe` with the option `/apply` (Table 4.10).
8. After that, the administrator needs to create a new user account, with administrator's rights that will execute the script create on the last step.
9. To finish the process and to apply the package, the administrator needs to go to each workstation and logon in the special account that he created on step 8.

An example of a batch script is presented on Table 4.10.

4.8 Conclusions

Automating NT tasks without administrative tools such as SMS can be by-passed using some techniques. The difficulty found to deploy this solutions was that all the published papers before this one, explain how it could be done assuming that SMS was installed.

After learning about NT application issues, Registry, and trying out various options, the paper suggest some security settings and proposes 3 solutions to apply this settings on every kind of NT-based environment, either when the amount of computers is small and the computers are close to each other, or in large environments and large networks where there is one system administrator that can't be in front of each network computer.

```
@REM Script to automate and install security packages
@REM Packages are created with the application sysdiff.exe

@echo Applying the Package
@REM connect the current computer to the share
@net use g: \\share\drive

@REM Change the work directory
@g:

@REM Apply the package
@sysdiff /m /apply package

@REM Return to drive C:
@c:

@REM Disconnect from the share drive
@net use g: /d
@echo off
```

Table 4.10: Batch Script.

Capítulo 5

DoIt4Me

O capítulo 4 sugere aplicar uma lista de configurações de segurança a todas as máquinas da rede. A dificuldade e barreira encontrada nas redes NT é como fazê-lo remotamente, de maneira escalável, eficiente e rápida. Este capítulo tem como objetivo resolver o problema encontrado por um administrador de segurança no momento de realizar remotamente uma auditoria ou implantar uma lista de recomendações de segurança em cada máquina da rede NT.

Prólogo

Suponha uma grande rede NT gerenciada por apenas um administrador, o qual deseja implantar uma política de segurança de forma rápida, simples e utilizando as etapas definidas no ciclo de vida do projeto de segurança (seção 2.3).

Com o desenvolvimento da ferramenta DoIt4Me, o autor conseguiu tornar as tarefas de auditoria e configuração remota do *Registry* e de serviços um processo simples e escalável . De uma forma centralizada é possível facilitar muito a administração de segurança remota. O capítulo também apresenta detalhes da interface, arquivos de configuração, e impressão de auditorias realizadas pelo DoIt4Me [3] [5] [6].

Saindo um pouco da questão de segurança e entrando na área de gerenciamento de rede, suponha que o administrador deseje melhorar o desempenho de cada máquina da rede e que para isso é necessário modificar alguns valores do *Registry* referentes a memória. Isso também pode ser realizado com a ferramenta DoIt4Me, da mesma forma e simplicidade como os exemplos sobre segurança que serão apresentados.

O núcleo do capítulo é constituído pelo artigo *"Administration of Large Windows NT Networks with DoIt4Me"*, apresentado na *10th International Conference on System Administration, Networking and Security* (SANS' 2001), em maio de 2001, Baltimore, MD, USA [4]. O artigo contextualiza o problema de como realizar uma auditoria remota e uma configuração remota do *Registry* e os objetivos desejáveis para facilitar esse problema. Em seguida apresenta a linguagem utilizada para desenvolver a ferramenta e o trabalho correlato (apresentado na Seção 1.2, página 5). A parte principal, apresentada a partir da seção 5.6, descreve características e vantagens trazidas com a utilização da ferramenta DoIt4Me.

A interface e os relatórios gerados pelo DoIt4Me, apresentados nas seções 5.11, 5.12 e 5.13 como apêndice deste artigo, sofreram pequenas alterações depois da sua publicação. Uma versão atual da interface será apresentada na seção B.2.3.

Administration of large Windows NT networks with DoIt4Me

Alessandro Augusto,
Célio Cardoso Guimarães, Paulo Lício de Geus

IC - UNICAMP
University of Campinas - Campinas, SP, Brazil
alaugusto@yahoo.com.br, {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 years has resulted in large and complex sites 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.

5.1 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 requi-

res a lot of manpower and be error-prone; operator errors can lead to machines being configured erroneously, improperly, or not at all.

5.2 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 [7]. 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 [24].

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.

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 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.

5.3 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 deficiency 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.

5.4 Perl

Perl has been used on UNIX platforms for administration purposes for many years. ActiveState [1] 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 [10].

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.

5.5 Previous Work

Harlan Carvey presents in [10] 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 [10] 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.

5.6 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 [10].

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 [10]. 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.

5.7 DoIt4Me

5.7.1 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, 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.

5.7.2 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 [10].

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 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 section 5.11 (appendix A).

5.7.3 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. Sections 5.12 and 5.13 (Appendix B and appendix C) present examples of DoIt4Me reports.

5.7.4 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.

5.7.5 Limitations

Some management tasks cannot be performed remotely because of Windows NT limitations, such as remotely accessing some parts of the Registry. No 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 [7].

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

5.8 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 [47].

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.

5.9 Availability

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

5.10 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
 -------------------------------------------------
```

5.11 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
```

5.12 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]
```

Capítulo 6

Conclusão

O primeiro adversário para administração de sistemas Windows NT são as afirmações de que este ambiente é incapaz de ser gerenciado remotamente. Administrar remotamente uma rede NT pode ser inicialmente uma tarefa complexa mas não impossível.

As motivações deste trabalho foram suprir a carência do sistema Windows NT em fornecer ferramentas para facilitar tarefas remotas de administração e segurança de redes e diminuir as chances de erros manuais nestas tarefas. Teve como objetivo automatizar essas duas tarefas de maneira escalável e eficiente.

O trabalho envolveu estudos de técnicas para excluir a interação e o componente humano nas tarefas de administração de redes além de focá-las para realização em tempo menor e na eliminação da possibilidade de erro manual, durante a instalação e/ou atualização de um conjunto de programas ou durante a implantação de uma política de segurança. O trabalho apresentou 3 técnicas para resolver essa deficiência do Windows NT.

Na área de segurança o problema é pior. Considerada sem retorno financeiro e difícil de ser mensurada, a área de segurança é erroneamente desvalorizada. A suposição de que o conceito de segurança seja um produto para os servidores de rede é duplamente incorreta, primeiramente segurança é um processo que deve estar sempre em monitoramento e, segundo, a segurança de uma rede é medida pelo seu computador mais vulnerável, ou seja, deve-se aplicar segurança em todos os computadores da rede e não somente nos servidores.

Da mesma forma que o sistema NT é carente de ferramentas de administração, ele também o é com relação à segurança. A maior parte dos trabalhos relacionados a segurança de NT apresenta apenas uma lista de configurações necessárias a ser aplicadas em cada computador, mas

nenhum descreve qual a melhor forma de implantar essas configurações de maneira automatizada e escalável.

Desta forma, este trabalho apresentou algumas recomendações para possiblilitar o Windows NT um sistema de nível de segurança semelhante a C2 (proteção por acesso controlado) do departamento de segurança dos EUA. Neste nível de segurança, o Windows NT é capaz de permitir ou negar uso e/ou acesso a recursos do sistema para certos usuários, entre outras restrições.

Além das técnicas de administração e das recomendações de segurança, outra contribuição do trabalho foi o desenvolvimento de uma ferramenta de gerenciamento de rede capaz de implantar remotamente em cada computador da rede um conjunto de recomendações, de maneira eficiente e independente do número de computadores. Esta ferramenta, DoIt4Me, foi desenvolvida em *Perl*, uma linguagem interpretada e de código aberto, permitindo que qualquer administrador customize ou adicione novas funções, dependendo de suas necessidades pessoais ou corporativas.

Com o DoIt4Me, tarefas de gerenciamento de rede como por exemplo melhorar o desempenho das máquinas da rede, assim como tarefas de segurança, auditar ou aplicar recomendações de segurança em cada máquina da rede podem ser realizadas de maneira simples a partir de um servidor da rede, independente do número de máquinas presente nessa rede.

Embora a versão atual do DoIt4Me já funcione em ambientes Windows 2000, sugere-se como pesquisa futura incluir novos módulos para solucionar problemas específicos das plataformas W2K. Outra sugestão é verificar as vantagens e barreiras trazidas pela ferramenta DoIt4Me sendo executada em ambientes com IPSec. Utilizando o DoIt4Me sobre o IPSec, as mensagens enviadas do servidor (onde a ferramenta DoIt4Me está instalada e sendo executada) para as máquinas clientes dessa rede serão enviadas cifradas, dificultando ataques do tipo *spoofing*.

Bibliografia

[1] ACTIVESTATE WebSite. 11/07/2001. http://www.activestate.com

[2] AUGUSTO, Alessandro and SENA, Jansen and DE GEUS, Paulo Lício. *Security Management of Windows 2000 Networks with DoIt4Me and IPSec.* Proceedings of SSI'2001: 3o. Simpósio de Segurança em Informática. São José dos Campos, SP, Brasil, outubro, 2001. *(em Inglês)*

[3] AUGUSTO, Alessandro. *Applying Security Configurations to a Large Number of Windows NT Computers Without Visiting Each Machine.* Proceedings of IEEE LANOMS'2001: 2nd Latin American Network Operations and Management Symposium. Belo Horizonte, MG, Brazil, setembro, 2001. Co-Sponsored by IEEE Communications Society. *(em Inglês)*

[4] AUGUSTO, Alessandro and GUIMARÃES, Celio and DE GEUS, Paulo Lício. *Administration of Large Windows NT with DoIt4Me.* Proceedings of SANS'2001: The 10th International Conference on System Administration, Networking and Security. Baltimore, MD, USA, maio, 2001. *(em Inglês)*

[5] AUGUSTO, Alessandro and GUIMARÃES, Celio and DE GEUS, Paulo Lício. *DoIt4Me.* Proceedings of SBRC'2001: 19o. Simpósio Brasileiro de Redes de Computadores. 1o. Salão de Ferramentas. Florianópolis, SC, Brasil, maio, 2001. *(em Inglês)*

[6] AUGUSTO, Alessandro and GUIMARÃES, Celio and DE GEUS, Paulo Lício. *DoIt4Me: a Tool for Automating Administrative Tasks on Windows NT Networks.* Proceedings of WSEG'2001: Workshop em Segurança de Sistemas Computacionais. Florianópolis, SC, Brasil, março, 2001. *(em Inglês)*

[7] AUGUSTO, Alessandro and GUIMARÃES, Celio and DE GEUS, Paulo Lício. *Administration Techniques for Implementing Security on Large Windows NT Networks.* Proceedings of SSI'2000: 2o. Simpósio de Segurança em Informática. São José dos Campos, SP, Brasil, outubro, 2000. *(em Inglês)*

[8] Briney, Andy. *Information Security. Infosecurity: A View From the Frontlines*, 1999. 11/07/2001. http://www.infosecuritymag.com/articles/1999/feboundtable.shtml

[9] CARTER, Gerald. *Patch32: A System for Automated Client OS Updates.* Proceedings of the Large Installation System Administration of Windows NT Conference. Seattle, Washington, USA. 1998.

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

[11] CERT. *Windows NT Configuration Guidelines.* April, 2000. 11/07/2001. http://www.cert.org/tech_tips/

[12] CIMA, Fernando. *Implementando Seguranca no Windows NT 4.0.* 11/07/2001. http://www.absoluta.org/seguranca/seg_nt_cima_01.htm

[13] COX, Phil. *Auditing: The Ugly duckling of Computers.* ;Login: The Magazine of USENIX & SAGE, 1998.

[14] CPAN. *Comprehensive Perl Archive Network.* 11/07/2001. http://www.cpan.org

[15] CUSTER. H. *Inside Windows NT.* Microsoft Press, 1993.

[16] DALY, Gregg et all. *NT Security in an Open Academy Environment.* Proceedings of USENIX LISA-NT: The 2nd Large Installation System Administration of Windows NT Conference, Seattle, Washington, USA, 1999.

[17] EVARD, Rémy & LESLIE, Robert. *Soft: A Software Environment Abstraction Mechanism.* Proceedings of USENIX LISA VIII: the Large Installation System Administration Conference, San Diego, CA, USA, 1994.

[18] FISK, Michael. *Automating the Administration of Heterogeneous LANs.* Proceedings of the 10th USENIX System Administration Conference. Chicago, IL, USA, 1996.

[19] FRISCH, Aeleen. *Introducing the NT Registry.* SunExpert Magazine. 1997.

[20] FRISCH, Aeleen. *Perl and Windows NT.* SunExpert Magazine. 1997.

[21] FULMER, Robert & LEVINE, Alex. *AutoInstall for NT: Complete NT Installation Over the Network.* In: Proceedings of the Large Installation System Administration of Windows NT Conference, USENIX LISA, Seattle, Washington, USA, 1998.

[22] GARFINKEL, Simson L; SPAFFORD, Gene. *Practical UNIX and Internet Security.* Second Edition. O'Reilly & Associates, Inc. 1996.

[23] GNU. *General Public License.* 11/07/2001. http://www.gnu.org

[24] GOMBERG, Michail; 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, Seattle, Washington, USA, 1998.

[25] GOMBERG, Michail; 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, Seattle, Washington, USA, 1999.

[26] GRANADO, Marcus Cunha. *Análise de Falhas de Segurança dos Protocólos de Comunicação do Windows NT.* Tese de Mestrado, IC-UNICAMP, Campinas, maio, 2001.

[27] HEDBOM, Hans; LINDSKOG, Stefan. *Analysis of the Security of Windows NT.* Department of Computer Engineering, Charmers University of Technology, Sweeden, 1999. 11/07/2001. http://secinf.net/info/nt/analysis/

[28] IC-UNICAMP. *Instituto de Computação.* Universidade Estadual de Campinas. Campinas. 11/07/2001. http://www.ic.unicamp.br

[29] International Organization for Standardization / International Eletrotechnical Committee. *Information Processing Systems - Open System Interconnection - Basic Reference Model - part 2: Security Architecture.* International Standart 7498-2, 1989.

[30] KIRCH, John. *The UNIX vs NT Organization.* 11/07/2001. http://www.unix-vs-nt.org

[31] KIRCH, John. *Troubleshooting and Configuring the Windows 95/NT Registry.* Macmillan Computer Publishing, 1999.

[32] KRANENBURG, Paul. *Monitoring Utilization in an NT Workstation Lab.* Proceedings of the Large Installation System Administration of Windows NT Conference. Seattle, Washington, USA, 1998.

[33] LUERKENS, Cameron D. & COLE, John & LEGG, Danielle. *Software Distribution to PC Clients in an Enterprise Network.* In: Proceedings of the Large Installation System Administration of Windows NT Conference, USENIX LISA, Seattle, Washington, USA, 1998.

[34] MICROSOFT. *Hiding the last logged on username in the logon dialog.* Microsoft Windows Knowledge Base. Article ID: Q114463. 11/7/2001. http://support.microsoft.com/support/kb/articles/q114/4/63.asp

[35] MICROSOFT. *How to protect Windows NT Desktops in public areas.* Windows Knowledge Base. Article ID: Q143164. 11/7/2001. http://support.microsoft.com/support/kb/articles/q143/1/64.asp

[36] MICROSOFT. *Resetting Default Access Controls on Selected Registry Keys.* Microsoft Windows Knowledge Base. Article ID: Q126713. 11/07/2001. http://support.microsoft.com/support/kb/articles/q126/7/13.asp

[37] MICROSOFT. *Security Issues that may occur due to the way Windows NT handles FPNWCLNT.DLL.* Microsoft Windows Knowledge Base. Article ID: Q99885. 11/07/2001. http://support.microsoft.com/support/kb/articles/q99/8/85.asp

[38] MICROSOFT. *Standard Security Practices fow Windows NT.* Microsoft Windows Knowledge Base. Article ID: q166992. 11/07/2001. http://support.microsoft.com/support/kb/articles/q166/9/92.asp

[39] MICROSOFT. *Srvany.exe. Running applications as services.* Microsoft Press.

[40] MICROSOFT *Technet. Windows NT C2 Configuration Checklist.* 11/01/2001. http://www.microsoft.com/technet/security

[41] MICROSOFT Web Site. 11/07/2001. http://www.microsoft.com/security/

[42] MICROSOFT *Windows NT Workstation Resource Kit: Comprehensive Resource Guide and Utilities for Windows NT Workstation Version 4.0.* Microsoft Press, 1996.

[43] MICROSOFT *Windows NT Server: Server Operating System. Securing Windows NT Installation.* Microsoft Corporation, Microsoft Press White Paper.

[44] NAKAMURA, Emilio. *Um Modelo de Segurança de Redes para Ambientes Cooperativos.* Tese de Mestrado, IC-UNICAMP, Campinas, Setembro, 2000.

[45] NSA - National Security Agency. *Microsoft Corporation Windows NT Workstation and Windows NT Server Version 4.0 with Service Pack 6a and C2 Update.* Trusted Computer System Evaluation. NSA Trusted Product Evaluation Program (TPEP). 11/07/2001. http://www.radium.ncsc.mil/tpep/epl/entries/TTAP-CSC-EPL-99-001.html

[46] PERL2EXE. *IndigoStar Software.* 11/07/2001. http://www.perl2exe.com

[47] ROTH, Dave. *A Networked Machine Management System.* Proceedings of USENIX LISA-NT, the 2nd Large Installation System Administration of Windows NT Conference, Seattle, Washington, USA, 1999.

[48] RUSSINOVICH, Mark. *Inside the Windows NT Registry.* 11/07/2001. http://www.shsu.edu/~ucs_kae/techdocs/windowsnt/insideregistry.htm

[49] SCHNEIER, Bruce. *Tradução de Secrets and Lies: Segurança.com: Mentiras Sobre a Proteção na Rede Digital.* Editora Campus, 2001.

[50] SJOLIN, Martin. *State-Driven Software Installation for Windows NT.* Proceedings of of USENIX LISA-NT, the 2nd Large Installation System Administration of Windows NT Conference, Seattle, Washington, USA, 1999.

[51] SMS. *Licenciamento do Systems Management Server.* 11/07/2001. http://www.microsoft.com/brasil/licenciamento/servidores/sms.stm

[52] SMS2. *Microsoft Systems Management Server 2.0*. 11/07/2001. http://www.microsoft.com/brasil/sms/

[53] SOARES, Luiz Fernando et al. *Redes de Computadores: das LANs, MANs e WANs às redes ATM*. Rio de Janeiro, Ed. Campus, 1995.

[54] SOLOMON, D.A. *Inside Windows NT Second Edition*. Microsoft Press, 1998.

[55] SOLOMON, D.A.; Russinovich, M.E. *Inside Microsoft Windows 2000 Third Edition*. Microsoft Press, 2000.

[56] SPITZNER, Lance. *Armoring NT*. 11/07/2001. http://www.enteract.com/~lspitz/nt.html

[57] SUTTON, Steve. *Windows NT Security Guidelines: Considerations and Guidelines for Securely Configuring Windows NT in Multiple Environments*. NSA Research, June, 1999. 11/07/2001. http://www.trustedsystems.com/tss_nsa_guide.htm

[58] SYMANTEC *PCAnywhere*. 18/09/2001. http://www.symantec.com/pcanywhere/

[59] TANENBAUM, Andew S. *Computer Networks*. Third Edition, Prentice Hall, 1998.

[60] VNC. *Virtual Network Computing*. 11/07/2001. http://www.uk.research.att.com/vnc/

[61] WALLI, Stephen. OpenNT: *UNIX Application Portability to Windows NT via an Alternative Environment SubSystem*. Proceedings of the USENIX Windows NT Workshop, Seattle, Washington, USA, 1997.

Apêndice A

Implementação da Ferramenta DoIt4Me

Este apêndice comenta e apresenta o código fonte da ferramenta DoIt4Me em partes separadas, tornando assim mais fácil a sua compreensão. Há uma seção para cada procedimento da versão atual da ferramenta.

A.1 Linguagem *Perl* e módulos

Com o objetivo de facilitar a alteração do código e permitir que qualquer administrador customize inclua funções extras, a ferramenta DoIt4Me foi desenvolvida em *perl* e seu código está disponibilizado livremente na Internet através da licença GNU GPL [23].

Perl é uma linguagem de programação utilizada em diversos sistemas operacionais. No ambiente Windows NT essa linguagem pode facilitar principalmente as tarefas de administração e segurança de redes [20]. Por ser uma linguagem interpretada, é necessário ter instalado o interpretador *perl* na máquina onde o programa será executado. Existem programas extras capazes de compilar códigos *perl* em aplicações executáveis, porém o tamanho do arquivo compilado fica muito maior se comparado com o código fonte [46].

No caso da ferramenta DoIt4Me, todo o controle e gerenciamento é centralizado no controlador primário de domínio da rede NT. Somente no PDC é necessário instalar o interpretador *perl*, as máquinas clientes não necessitam de nenhuma configuração.

Uma vantagem em utilizar a linguagem *perl* é a possibilidade de adição de módulos, isto é, bibliotecas extras capazes de facilitar algumas funções e tarefas. Devido a quantidade de módu-

los existentes e disponiveis na Internet, é recomendado que o desenvolvedor *perl* procure antes por módulos que possam ajudar na aplicação [14] [47].

A definição dos módulos a serem utilizados em aplicações *perl* acontece no início do código. O código abaixo apresenta a definição dos módulos utilizados pela ferramenta DoIt4Me.

```
use strict;
use Win32;
use Win32::Service;
use Win32::TieRegistry(Delimiter=>"/");
use Net::Ping;
```

Alguns módulos foram adicionados para tratarem de funções específicas, por exemplo o módulo `Win32` que tem como objetivo facilitar o tratamento de comandos do Windows NT, por exemplo reiniciar o computador. O módulo `Net::Ping` foi adicionado para a função de *ping*. O módulo `Win32::TieRegistry` torna o acesso ao *Registry* menos complexo, enquanto que o módulo `Win32::Service` facilita o tratamento dos serviços das máquinas NT. Uma característica com relação ao interpretador *perl* e aos módulos é que ambos são disponibilizados gratuitamente na Internet [1] [14].

A.2 Interface e Opções do DoIt4Me

Na atual versão o DoIt4Me possui 7 opções diferentes de tarefas. A escolha da opção é feita em linha de comando pelo administrador durante a execução do programa. A sintaxe para execução do DoIt4Me é: "`DoIt4Me.pl <opção>`", sendo o campo <opção> um dos atuais valores entre 1 e 8 descritos na interface. O código abaixo apresenta o procedimento da interface.

```
# ====================================================================
#                  Procedure Display DoIt4Me arguments
#
# This procedure displays the arguments of DoIt4Me
#
# ====================================================================

sub usage {

  print "_____\n\n";
  print "                        DoIt4Me\n";
  print "       Automate NT Administrative Tasks Remotely\n\n";
```

```
 print "\n";
 print " Usage: doit4me.pl <option>\n";
 print " Option: <1> Audit Registry keys\n";
 print "         <2> Configure the Registry\n";
 print "         <3> Check the status of ALL NT services\n";
 print "         <4> Check the status of a subset of NT services\n";
 print "         <5> Change NT services Status (Start/Stop)\n";
 print "         <6> Ping a subset of workstations\n";
 print "         <7> Reboot a subset of workstations\n";
 print "         <8> Help\n";
 print "\n\n";
 print " DoIt4Me, Copyright (C) 2001 Alessandro Augusto\n";
 print " DoIt4Me comes with ABSOLUTELY NO WARRANTY; for details\n";
 print " see option <8>.  This is free software, and you are wel come\n";
 print " to redistribute it under certain conditions; see DoIt4Me license.\n";
 print "_____________________________________________________________\n\n";
}
```

A.3 Auditoria do *Registry*

A opção número 1 da ferramenta DoIt4Me (*Audit Registry keys*) tem como objetivo realizar auditoria de um conjunto de chaves do *Registry* em um conjunto de máquinas remotas. Para realizar auditoria basta o administrador definir quais chaves e qual o conjunto de computadores, para que o DoIt4Me automatize de forma escalável toda a tarefa. Para maiores detalhes a respeito da sintaxe dos arquivos de configuração e de como definir o conjunto de chaves e computadores, o apêndice B apresenta o manual da ferramenta. Já o código do procedimento de auditoria remota é apresentado a seguir.

```
# ===========================================================================
#                          Procedure GetRegValues
#
# This procedure check the values of a subset of Registry keys of a subset
# of computers. The subset of Registry values are specified on the file
# <doit4me_folder>/cfg/regaudit.cfg and the subset of computers are
# specified on the file <doit4me_folder>/cfg/pclist.cfg
#
# Syntax of the file <doit4me_folder>/cfg/regaudit.cfg: key;Path
# (one key;Path for each line)
#
# ===========================================================================
```

```
sub getRegValues {
  my($value,$data);
  my %regkeys = ();
  my $datafile = "./cfg/regaudit.cfg";

  if (-e $datafile) {
  open(FL,$datafile) || die "There is a problem to open the file $datafile: $!\n";
    while(<FL>)
    {
      chomp;

      # Ignore comentaries and blank lines on the configuration files
      next if ($_ =~ m/^#/);
      next if ($_ =~ m/^\s+$/);
      my($key,$path) = split(/;/,$_);
      $regkeys{$key} = $path;
    }
    close(FL);

    # The subset of computers are specified on the
    #  <doit4me_folder>/cfg/pclist.cfg file

    my $pclistfile = "./cfg/pclist.cfg";

    if (-e $pclistfile)
    {
      open(FL,$pclistfile) || die "There is a problem to open the file $pclistfile:
$!\n";
     while(<FL>)
      {
        chomp;

        # Ignore comentaries and blank lines on the configuration files
        next if ($_ =~ m/^#/);
        next if ($_ =~ m/^\s+$/);
        my($pc) = split(/;/,$_);

        # First check if the computer is alive on the network
        $computer= $pc;
        &pc_exist($computer);
        if ($is_alive eq 1) { push(@computerlist, $pc); }
         else { push(@computer_off_list, $pc); }
      }
      close(FL);
    }

    $ultimo_indice = $#computerlist;
```

```
    print "_____\n\n";
    print "                DoIt4Me Registry Auditing\n";
    print "_____\n";
    print "\n\n";
    printf "%-15s %-27s %-5s\n","COMPUTER","KEY","VALUE";
    printf "%-15s %-27s %-5s\n","-" x 12, "-" x 7,"-" x 5;

    # if you remove the below sort command, the results wont be outputted alphabeticaly
    foreach my $key (sort keys %regkeys) {

    # in this loop, we verify the value of the current key ($key) in each computer
    # from the computerlist file.

    for ($i=0; $i <= $ultimo_indice; $i++)
    {
      $server= $computerlist[$i];
      # atribui o valor da posicao do vetor a variavel server

      if ($remote = $Registry->{"//$server"})
      {
        $value = $remote->{$regkeys{$key}};
        $data = $value->{$key};
        if (defined $data)
        {
          $data = hex($data) if ($data =~ m/^0x/);
          printf "%-15s %-27s %-5s\n","$server","$key","$data";
        }
         else
            { printf "%-15s %-27s %-5s\n","$server","$key","Not Found"; }
      }
      else {
        print "It was not possible to connect to the $server Registry\n";
      }

    }
    print "\n\n";
  }
  }

  # Print the Computers that are not alive on the network
  &print_not_alive;
}
```

A.4 Configurando o *Registry*

A opção número 2 (*Configure the Registry*) permite o administrador implantar e/ou modificar uma lista de configurações do *Registry* em um conjunto de máquinas remotas. O procedimento é similar ao de auditoria, porém nesse caso além de especificar o conjunto de máquinas e chaves, o administrador precisa especificar quais os novos valores a semem implantados nas máquinas remotas. O código abaixo apresenta o procedimento para configurar o *Registry*.

```
# ========================================================================
#                          Procedure ChangeRegValues
#
# This procedure changes a subset of Registry values specified on the
# the file <doit4me_folder>/cfg/regconfig.cfg on a subset of computers
#
# Syntax of the file <doit4me_folder>/cfg/regconfig.cfg: key; Path; New value
# (one key;Path;new value for each line)
#
# ========================================================================

sub changeRegValues{

my $novos_valores_file = "./cfg/regconfig.cfg";
my($value,$data,$path, $newvalue);
my %regkeys = ();

 # open the file and separe the values and keys in different variables
 if (-e $novos_valores_file)
 {

  # open the file with the subset of computers
  my $pclistfile = "./cfg/pclist.cfg";
  if (-e $pclistfile)
  {
   open(FL,$pclistfile) || die "There is a problem to open the file $pclistfile: $!\n";
   while(<FL>)
   {
    chomp;

    # Ignore comentaries and blank lines on the configuration files
    next if ($_ =~ m/^#/);
    next if ($_ =~ m/^\s+$/);
    my($pc) = split(/;/,$_);
```

```
    # First check if the computer is alive on the network

    $computer= $pc;
    &pc_exist($computer);
    if ($is_alive eq 1) { push(@computerlist, $pc); }
    else { push(@computer_off_list, $pc); }

   } # end while
   close(FL);
  } # end if

  $ultimo_indice = $#computerlist;

  print "_____\n\n";
  print "                DoIt4Me Registry Configure\n";
  print "_____\n";
  print "\n\n";
  printf "%-15s %-27s\n","COMPUTER","STATUS";
  printf "%-15s %-27s\n","-" x 12, "-" x 7;

  for ($i=0; $i <= $ultimo_indice; $i++)
  {
   $server= $computerlist[$i];
   open(FL,$novos_valores_file) || die "There is a problem to open the file
$novos_valores_file: $!\n";

   while(<FL>)
   {
    chomp;

    # Ignore comentaries and blank lines on the configuration files
    next if ($_ =~ m/^#/);
    next if ($_ =~ m/^\s+$/);

    my($key,$path,$newvalue) = split(/;/,$_);

    $regkeys{$path, $key} = $newvalue;
    $caminho="$path";
    $chave="$key";
    $valor="$newvalue";

    # set the new value on the Registry
    $Registry->{"//$server/$caminho//$chave"}="$valor";
   } #end while
   close(FL);
   printf "%-15s %-27s\n","$server","Configured";
  } #end for
```

```
 } #end if

 print "\n\n";
 # Print the Computers that are not alive on the network
 &print_not_alive;
}
```

A.5 Auditoria de serviços

A atual versão do DoIt4Me têm 2 procedimentos com objetivos de auditar os serviços de máquinas remotas na rede.

A.5.1 Auditoria de todos serviços

O primeiro procedimento (*Check the status of all NT services*), opção número 3 da atual versão, tem como objetivo auditar todos os serviços presentes em um conjunto de máquinas. Nesse procedimento o administrador necessita apenas especificar quais máquinas serão auditadas. O código abaixo apresenta o procedimento.

```
# ====================================================================
#                        Procedure Get_services_status
#
# This procedure checks the status a ALL NT services from a subset of
# computers
#
# ====================================================================


sub get_services_status {

  my($server) = @_;
  my %svchash = ();
  my %status;
  my(@state) = ("",
      "Stopped",
      "Start_Pending",
      "Stop_Pending",
      "Running",
      "Continue_Pending",
      "Pause_Pending",
      "Paused");
```

```
# Subset of Computers that will be scanned and ckecked
# file: <doit4me_folder>/cfg/pclist.cfg
my $pclistfile = "./cfg/pclist.cfg";

if (-e $pclistfile)
{
 open(FL,$pclistfile) || die "There is a problem to open the file $pclistfile: $!\n";
 while(<FL>)
 {
  chomp;

  # Ignore comentaries and blank lines on the configuration files
  next if ($_ =~ m/^#/);
  next if ($_ =~ m/^\s+$/);

  my($pc) = split(/;/,$_);

  # First check if the computer is alive on the network

  $computer= $pc;
  &pc_exist($computer);
  if ($is_alive eq 1) { push(@computerlist, $pc); }
  else { push(@computer_off_list, $pc); }
 } #end while
 close(FL);
}  #end if

$ultimo_indice = $#computerlist;

print "_____________________________________________________________________\n\n";
print "                DoIt4Me Services Status\n";
print "_____________________________________________________________________\n";
print "\n\n";

for ($i=0; $i <= $ultimo_indice; $i++)
{
 $server= $computerlist[$i];
 print "-------------------------\n";
 print "COMPUTER: $server\n";
 print "-------------------------\n";

 if (Win32::Service::GetServices("\\\\$server",\%svchash))
 {
  printf "%-53s %-20s\n","SERVICES","STATUS";
  printf "%-53s %-20s\n","-" x 48,"-" x 9;
  foreach my $svc (sort keys %svchash)
```

```
   {
    if (Win32::Service::GetStatus("\\\\$server",$svchash{$svc},\%status))
    {
     printf "%-53s %-20s\n","$svc","[$state[$status{CurrentState}]]";
    } #end if
   else
    {
     &lasterror;
    } #end foreach
  } #end if
 } #end for
 else
 {
  &lasterror;
 }

 print "\n\n";
 }

 # Print the Computers that are not alive on the network
 &print_not_alive;
}
```

A.5.2 Auditoria de alguns serviços

Existe uma segunda forma de realizar auditoria de serviços. Com a opção número 4 (*Check the status of a subset of NT services*) a ferramenta DoIt4Me permite ao administrador especificar apenas os serviços que ele deseja auditar. O código a seguir refere-se a este procedimento.

```
# ===========================================================================
#                          Procedure get_some_services_status
#
# This procedure checks the status of a subset of NT services
# Configure the subset of services on the file <doit4me_folder>/cfg/serviceaudit.cfg
# and the subset of computers on <doit4me_folder>/cfg/pclist.cfg
#
# ===========================================================================

sub get_some_services_status {

  my($server) = @_;
  my %svchash = ();
  my %status;
```

```
my ($ultimo_indice_servicos, $servico);
my @servicoslist;
my(@state) = ("",
   "Stopped",
   "Start_Pending",
   "Stop_Pending",
   "Running",
   "Continue_Pending",
   "Pause_Pending",
   "Paused");

my $pclistfile = "./cfg/pclist.cfg";

if (-e $pclistfile)
{
 open(FL,$pclistfile) || die "There is a problem to open the file $pclistfile: $!\n";
 while(<FL>)
 {
  chomp;

  # Ignore comentaries and blank lines on the configuration files
  next if ($_ =~ m/^#/);
  next if ($_ =~ m/^\s+$/);
  my($pc) = split(/;/,$_);

  # First check if the computer is alive on the network

  $computer= $pc;
  &pc_exist($computer);
  if ($is_alive eq 1) { push(@computerlist, $pc); }
  else { push(@computer_off_list, $pc); }
 } #end while
 close(FL);
} #end if

$ultimo_indice = $#computerlist;

print "_____\n\n";
print "                    DoIt4Me Services Audit\n";
print "_____\n";
print "\n\n";

my $srvlistfile = "./cfg/serviceaudit.cfg";
if (-e $srvlistfile)
{
  open(FIL,$srvlistfile) || die "There is a problem to open the file $srvlistfile:
$!\n";
```

```
 while(<FIL>)
 {
  chomp;

  # Ignore comentaries and blank lines on the configuration files
  next if ($_ =~ m/^#/);
  next if ($_ =~ m/^\s+$/);

  my($servico) = split(/;/,$_);
  push(@servicoslist, $servico);
 } #end while
 close(FIL);
} #end if

$ultimo_indice_servicos = $#servicoslist;

# Services Loop
# Port: Laco dos Servicos

for ($i=0; $i <= $ultimo_indice_servicos; $i++)
{
 $servico= $servicoslist[$i];
 print "SERVICE: $servicoslist[$i]\n";
 print "----------------------------------------\n";
 printf "%-23s %-20s\n","COMPUTER","STATUS";
 print "----------------------------------------\n";

 # Computer Loop
 # Port: laco dos computadores

 for ($j=0; $j <= $ultimo_indice; $j++)
 {
  $server= $computerlist[$i];

  # check the status of the current service

  if (Win32::Service::GetStatus("\\\\$server",$servicoslist[$i],\%status)) {
  printf "%-23s %-20s\n","$computerlist[$j]","[$state[$status{CurrentState}]]";
  } #end for
 else
 {
  printf "%-23s %-20s\n","$computerlist[$j]","[*ERROR*]";
 } #end else

 # Port: fim do laco dos computadores
} #end for
```

```
  print "\n\n";

  # Port: fim do for (laco dos servicos)
 }

 # Print the Computers that are not alive on the network
 &print_not_alive;
}
```

A.6 Configurando serviços remotos

Um alerta encontrado em várias recomendações de segurança sugere ao administrador desabilitar todos os serviços que não estão sendo utilizados [11] [35] [38] [40] [43] [45]. A título de exemplo, suponha que exista uma vulnerabilidade no serviço fictício "xyz", o qual está presente em algumas máquinas da rede alvo e não está sendo utilizado pelo administrador. Com o segundo procedimento de auditoria, opção número 5 (*Check the status of a subset of NT services*) apresentado na seção A.5.2 o administrador tem a possibilidade de descobrir em quais máquinas da rede esse serviço está ligado. Já com a opção número 6 do DoIt4Me (*Change NT services Status [Start/Stop]*), o administrador pode alterar o status desse serviço, ligando ou desligando um conjunto de serviços. Para isso basta especificar quais serviços, quais os novos status (ligar ou desligar) de cada serviço e em quais máquinas deseja-se alterar os status dos serviços. Esse procedimento é apresentado no código abaixo.

```
# ========================================================================
#                          Procedure change_services_status
#
# This procedure change the status of a subset of NT services
# Configure the file <doit4me_folder>/cfg/serviceconfig.cfg
#
# Syntax of <doit4me_folder>/cfg/serviceconfig.cfg: service; new status
# New Status = 1 starts the service
# New Status = 0 stops the service
#
# ========================================================================


sub change_services_status {
```

```
my %status;
my $status_atual;
my(@state) = ("",
    "Stopped",
    "Start_Pending",
    "Stop_Pending",
    "Running",
    "Continue_Pending",
    "Pause_Pending",
    "Paused");

my $srv_file = "./cfg/serviceconfig.cfg";
my($srv,$srv_bit);
my %regkeys = ();
my $pclistfile = "./cfg/pclist.cfg";

if (-e $pclistfile)
{
 open(FILE,$pclistfile) || die "There is a problem to open the file $pclistfile:
$!\n";
 while(<FILE>)
 {
  chomp;

  # Ignore comentaries and blank lines on the configuration files
  next if ($_ =~ m/^#/);
  next if ($_ =~ m/^\s+$/);
  my($pc) = split(/;/,$_);

  # First check if the computer is alive on the network

  $computer= $pc;
  &pc_exist($computer);
  if ($is_alive eq 1) { push(@computerlist, $pc); }
  else { push(@computer_off_list, $pc); }
 } # end while
 close(FILE);
} #end if

$ultimo_indice = $#computerlist;

print "_____\n\n";
print "              DoIt4Me Change Services Status\n";
print "_____\n";
print "\n\n";

# open the file and separe the values and keys in different variables
```

```
if (-e $srv_file)
{
  open(FL,$srv_file) || die "There is a problem to open the file $srv_file: $!\n";
  while(<FL>)
  {
   chomp;

   # Ignore comentaries and blank lines on the configuration files
   next if ($_ =~ m/^#/);
   next if ($_ =~ m/^\s+$/);

   my($srv,$srv_bit) = split(/;/,$_);
   $regkeys{$srv} = $srv_bit;

   print "SERVICE: $srv\n";
   print "----------------------------------------\n";
   printf "%-23s %-20s\n","COMPUTER","STATUS";
   print "----------------------------------------\n";


   for ($i=0; $i <= $ultimo_indice; $i++)
   {
    $server= $computerlist[$i];  # atribui o valor da posicao do vetor a variavel
server

    # Stop the Service
    # if the New Status = 0 -> then stop the service

    if ($srv_bit eq 0)
    {
     # first of all, check the service current status to see if needs to stop or not
     # Port: verifica o status currente do servico pra ver se precisa desligar ou nao

      if (Win32::Service::GetStatus("\\\\$server",$srv,\%status))
      {
       $status_atual=$state[$status{CurrentState}];
      } #end if

      # The service is not stopped yet
      # Port: se o servico nao estiver desligado

      if ($status_atual ne "Stopped")
      {
       Win32::Service::StopService("$server","$srv");
       printf "%-23s %-20s\n","$server","[stop]";
      }
      else
```

```
    {
     printf "%-23s %-20s\n","$server","[already stopped]";
    }
   } #end if

   # Ligar o Servico
   # if the New Status = 1 -> then start the service

   elsif ($srv_bit eq 1)
   {
     # first of all, check the service current status to see if needs to start or not
     # Port: verifica o status currente do servico pra ver se precisa ligar ou nao

     if (Win32::Service::GetStatus("\\\\$server",$srv,\%status))
     {
      $status_atual=$state[$status{CurrentState}];
     }

     # The service is not started yet
     # Port: se o servico nao estiver ligado

     if ($status_atual ne "Running")
     {
      Win32::Service::StartService("$server","$srv");
      printf "%-23s %-20s\n","$server","[start]";
     }
     else
     {
      printf "%-23s %-20s\n","$server","[already started]";
     }
    }

   # If the New Status is a value different from 0 or 1, then print an error message
   # Port: caso o valor do status seja diferente de 0 ou 1

   else
   {
    print "Wrong value >> $srv <<\n";
    print "Please fix the value on the serviceconfig.cfg file\n";
   }
  }

 print "\n\n";
}

close(FL);
```

```
  }

  # Print the Computers that are not alive on the network
  &print_not_alive;

}
```

A.7 Ping

Os códigos apresentados anteriormente utilizam um subprocedimento, uma função de *ping*, o qual tem serve para verificar se a máquina alvo está funcionando/ligada ou se está desligada naquele momento. Existe também uma opção na interface do DoIt4Me específica para esse procedimento, opção número 6 (*Ping a subset of workstations*). O código abaixo apresenta o procedimento *ping*.

```
# ==========================================================================
#                          Procedure ping and Pc_exist
#
# This procedure pings a subset of NT computers
# Specify the computers to be pinged on <doit4me_folder>/cfg/ping.cfg file
#
# The procedure pc_exist should be call with a host computer. The result will
# be saved on the variable "is_alive"
#
# $is_alive= 0 means the pc is down.
# $is_alive= 1 means the pc is up.
#
# ==========================================================================

sub ping {

 print "_____________________________________________________________________\n\n";
 print "                          DoIt4Me Ping\n";
 print "_____________________________________________________________________\n";
 print "\n\n";

 my $pinglistfile = "./cfg/ping.cfg";

 if (-e $pinglistfile)
 {
   open(PINGFILE,$pinglistfile) || die "There is a problem to open the file $pinglist-
file: $!\n";
```

```
   while(<PINGFILE>)
   {
    chomp;

    # Ignore comentaries and blank lines on the configuration files
    next if ($_ =~ m/^#/);
    next if ($_ =~ m/^\s+$/);

    my($pc) = split(/;/,$_);

    $computer= $pc;
    &pc_exist($computer);
    if ($is_alive eq 1) { push(@computerlist, $pc); }
    else { push(@computer_off_list, $pc); }
   }
   close(PINGFILE);
  }

  printf "%-23s %-20s\n","COMPUTER","STATUS";
  print "----------------------------------------\n";

  $ultimo_indice = $#computerlist;
  @computerlist= sort(@computerlist);
  for ($i=0; $i <= $ultimo_indice; $i++)
  {
   $server= $computerlist[$i];
   printf "%-23s %-20s\n","$server","[Host is Alive]";
  }

  print "\n";
  $ultimo_indice = $#computer_off_list;
  @computer_off_list= sort(@computer_off_list);
  for ($i=0; $i <= $ultimo_indice; $i++)
  {
   $server= $computer_off_list[$i];
   printf "%-23s %-20s\n","$server","[Host is Dead]";
  }

}


sub pc_exist($computer) {
 $p = Net::Ping->new("icmp");
 return $is_alive="0" unless $p->ping($computer, 2);
 return $is_alive="1";
 $p->close();
}
```

A.8 Procedimento para reiniciar

Algumas tarefas nos ambientes Windows requerem que o administrador reinicie os computadores após sua execução para que as novas configurações sejam carregadas, por exemplo, após implantar novas configurações no *Registry* . Para isso a opção número 7 (*Reboot a subset of workstations*) permite que o administrador reinicie remotamente um conjunto de máquinas simultaneamente. O código a seguir apresenta o procedimento para reiniciar um conjunto de máquinas NT.

```
# ====================================================================
#                          Procedure Reboot
#
# This procedure reboots a subset of NT computers
# Specify the computers on <doit4me_folder>/cfg/reboot.cfg file
#
# Configure the time to reboot (in seconds), the message and the set the 1
# to reboot or 0 to shutdown on the file<doit4me_folder>/cfg/rebootmsg.cfg
#
# Syntax: x_seconds; reboot_message; reboot(1) or shutdown(0)
#
# ====================================================================


sub reboot {

 print "_____\n\n";
 print "                    DoIt4Me Reboot Computers\n";
 print "_____\n";
 print "\n\n";

 my $rebootlistfile = "./cfg/reboot.cfg";
 my $rebootmsg = "./cfg/rebootmsg.cfg";

 # Split the time and the reboot message on different variables

 if (-e $rebootmsg)
 {
  open(MSG,$rebootmsg) || die "There is a problem to open the file $rebootmsg: $!\n";
   while(<MSG>)
   {
    chomp;
```

```
    # Ignore comentaries and blank lines on the configuration files
    next if ($_ =~ m/^#/);
    next if ($_ =~ m/^\s+$/);

    my($ttr,$rbtmsg,$rbt_or_stdwn) = split(/;/,$_);
    $timetoreboot= $ttr;
    $rebootmessage= $rbtmsg;
    $reboot_or_shutdown= $rbt_or_stdwn;
   }
   close(MSG);
  }

  # Reboot the computers

  if (-e $rebootlistfile)
  {
   open(FIL,$rebootlistfile) || die "There is a problem to open the file $rebootlist-
file: $!\n";
   while(<FIL>)
   {
    chomp;

    # Ignore comentaries and blank lines on the configuration files
    next if ($_ =~ m/^#/);
    next if ($_ =~ m/^\s+$/);

    my($pc) = split(/;/,$_);
    # First check if the computer is alive on the network

    $computer= $pc;
    &pc_exist($computer);
    if ($is_alive eq 1) { push(@computerlist, $pc); }
    else { push(@computer_off_list, $pc); }

   }
   close(FIL);
  }

  # Reboot the alive computers

  $ultimo_indice= $#computerlist;

  if ($ultimo_indice => 0)
  {
   print "-----------------------------------------\n";
   printf "%-23s %-20s\n","COMPUTER","REBOOT STATUS";
```

```
    print "----------------------------------------\n";
   }

   for ($i=0; $i <= $ultimo_indice; $i++)
   {
    $pc= $computerlist[$i];
    Win32::InitiateSystemShutdown("$pc","$rebootmessage","$timetore-
boot",0,"$reboot_or_shutdown");
    printf "%-20s %-16s\n","$pc","[Reboot in progress]";

    # syntax to Reboot
    #InitiateSystemShutdown($MachineName,$Message,$Timeout,$ForceAppsClosed,$Reboot-
AfterShutdown);
   }

   # Print the Computers that are not alive on the network
   print "\n\n";
   &print_not_alive;

}
```

Apêndice B

Descrição e Manual da Ferramenta DoIt4Me

Prólogo

Este apêndice é consituido por uma resumida descrição e pelo manual da ferramenta DoIt4Me. A descrição intitulada *"DoIt4Me: a Tool for Automating Administrative Tasks on Windows NT Networks"* e o manual eram requisitos para o salão de ferramentas do 19° Simpósio Brasileiro de Redes de Computadores (SBRC' 2001), que aconteceu em Florianópolis, Santa Catarina, em maio de 2001, na qual a ferramenta foi aceita para apresentação [5].

A descrição apresenta as funções da ferramenta enquanto o manual explica a parte de configuração.

Administration of large Windows NT networks with DoIt4Me

Alessandro Augusto,
Célio Cardoso Guimarães, Paulo Lício de Geus

IC - UNICAMP
University of Campinas - Campinas, SP, Brazil
alaugusto@yahoo.com.br, {celio, paulo}@ic.unicamp.br

The process to secure a Windows NT computer is simple when the system administrator knows the required configuration settings. However, even with this knowledge, to apply the same configuration to hundreds of NT-based computers can be frustrating and laborious. 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 years has resulted in large and complex sites but no significant new tools were created.

DoIt4Me is a tool designed to automate remote administrative tasks on Windows NT/2000 networks. It can manage small or large networks from a single console. Infrequent trips to distant machines will only be necessary in case of hardware failures.

Initially developed for administering and securing Windows NT 4.0 machines, it has recently been upgraded to work on Windows 2000. DoIt4Me is a must-have tool for Windows network administration.

DoIt4Me can automatically perform all the following options to a subset of NT/2000 machines:

1. Perform remote auditing of a subset of Registry settings. The administrator only needs to specify what Registry settings he or she wants to audit.
2. Remotely configure a subset of Registry settings. The administrator can modify the Registry settings specifying the new value of each Registry key.
3. Perform service status auditing. The administrator can configure DoIt4Me to audit the status of either all or a set of services. Auditing of specific services are also contemplated, such as "which machines are running the service "schedule"?"

4. Start or stop remote services. The administrator can start or stop any subset of services. For this purpose, he or she needs only to specify the service name and the action (to start or to stop it), and the subset of computers to apply these configurations on.
5. Reboot or Shutdown. There is an option where the administrator can reboot or shutdown a subset of workstations. In this option, the administrator can configure the grace period before rebooting, the message to send before rebooting, and the subset of machines to be rebooted.
6. Apply permissions on files, folders and Registry keys (ACLs); (module under construction)
7. And any other automated task. DoIt4Me is developed in Perl, and as such is open source. The administrator can add or modify DoIt4Me modules any time he or she wants. Please visit the DoIt4Me web site and read the DoIt4Me articles for extra information.

DoIt4Me Web Site: `http://www.ic.unicamp.br/~ra990866/doit4me/`

DoIt4Me Manual

Name

DoIt4Me - a tool for automating administrative tasks on Windows NT networks.

Synopsis

```
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> Ping a subset of workstations
        <7> Reboot a subset of workstations
        <8> Help
```

Examples

```
C:\>doit4me\DoIt4Me.pl 1
```

Scan a subset of computers for a Registry Auditing

```
C:\>doit4me\DoIt4Me.pl 7
```

Reboot a subset of workstations

Files

`Pclist.cfg`	Subset of computers
`Regaudit.cfg`	Subset of Registry values to Audit
`Regconfig.cfg`	Subset of Registry values to Configure
`Serviceconfig.cfg`	Subset of Services with the new status
`Serviceaudit.cfg`	Subset of Services to Audit
`Reboot.cfg`	Subset of computers to Reboot
`Rebootmsg.cfg`	Message sent before reboot
`Ping.cfg`	Subset of computers to Ping

Authors

Alessandro Augusto, Célio Guimarães, Paulo Lício de Geus
University of Campinas - UNICAMP
Computing Institute - IC
{alessandro.augusto, celio, paulo}@ic.unicamp.br

B.3.1 Overview

The process to secure a Windows NT computer is simple when the system administrator knows the required configuration settings. However, even with this knowledge, to apply the same configuration to hundreds of NT-based computers can be frustrating and laborious. 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 years has resulted in large and complex sites but no significant new tools were created.

DoIt4Me is an automated and remote administrative tool for Microsoft Windows NT operating systems. It can manage small or 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.

DoIt4Me can perform all the following options to a subset of computers:

- Perform remote Registry Auditing,
- configure remote Registry,
- perform services status auditing,
- start or stop remote NT services,
- ping workstations,
- reboot Workstations,
- apply ACLs (this module is under construction).

B.3.2 Installation

By installing DoIt4Me on the primary domain controller (PDC), the administrator can remotely control any subset of workstations served by the DC.

If you have the compiled version of DoIt4Me, on the domain controller, copy `"doit4me.exe"` to a folder, for example `"C:\DOIT4ME\"` and the configurations files to a subfolder called `cfg` (`"C:\DOIT4ME\CFG"`). If you have the source code of DoIt4Me, its required that you first install perl interpretor on the PDC and then copy the DoIt4Me source code and configuration files to folders such as the above explanation.

B.3.3 Usage and Interface

DoIt4Me needs to be execute by the command shell/prompt (cmd.exe) of the Windows NT. The interface should looks like as:

```
C:\> DoIt4Me.pl
```

```
                                DoIt4Me
                Automate NT Administrative Tasks Remotely

Usage: doit4me.pl <option>
Option: <1> Audit Registry keys
        <2> Configure the Registry
        <3> Check the status of ALL NT services
        <4> Check the status of a subset of NT services
        <5> Change NT services Status (Start/Stop)
        <6> Ping a subset of workstations
        <7> Reboot a subset of workstations
        <8> Help

DoIt4Me, Copyright (C) 2001 Alessandro Augusto
DoIt4Me comes with ABSOLUTELY NO WARRANTY; for details
see option <8>.  This is free software, and you are welcome
to redistribute it under certain conditions; see DoIt4Me license.
```

B.3.3.1 Option < 1 >: Auditing

The first feature of DoIt4Me, is the ability to scan any subset of network and to report the results for auditing. In this phase, also called Data Collection, the administrator specifies which configuration settings he or she wants to audit. It is only necessary to specify the subset of machines that will be scanned and the subset of Registry keys that will be collected.

Configure the file pclist.cfg with the computer's name that will be scanned, and the file regaudit.cfg with the subset of Registry keys that will be analyzed. The format of this file should be one computer per line followed by ";" . Each Registry key has to be in in line. Table shows an example of this files. In Regaudit.cfg, each Registry key should be followed by ";" and by the full path. Table 6.2 shows who to audit the keys `HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/ComputerName/ActiveComputerName/ComputerName` and

```
HKEY_LOCAL_MACHINE/Software/Microsoft/WindowsNT/CurrentVersion/Win-
logon/DontDisplayLastUserName
```

```
mustang;
porsch;
ferrari;
```

Table 6.1: Example of pclist.cfg configuration file

```
ComputerName;HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/ComputerName/
ActiveComputerName;

DontDisplayLastUserName;HKEY_LOCAL_MACHINE/Software/Microsoft/WindowsNT/Current-
Version/Winlogon;
```

Table 6.2: Example of regaudit.cfg configuration file

B.3.3.2 Option < 2 >: Registry Configuring

After the auditing, sometimes the system administrator needs to make some adjustments or configure some Registry values to make some computers compliance with the security policy.

The process is very similar to the Registry Auditing, the difference here is: in this option, besides the administrator specify the subset of computers and the subset of Registry keys, it is necessary to specify the new value that will receive this Registry key. The file that should be configured in this option is regconf.cfg.

Table 6.3 shows an example of how to configure the value 0 to the DontDisplayLastUserName Registry key.

```
DontDisplayLastUserName;HKEY_LOCAL_MACHINE/Software/Microsoft/Windows NT/Current-
Version/Winlogon; 0
```

Table 6.3: Example of regconfig.cfg configuration file

B.3.3.3 Option < 3 >: Services Status Auditing

DoIt4Me facilitates the process to get the status of NT services of remote computers. In this option, the administrator will audit the status of ALL services from a subset of computers. The

administrator needs just to configure the pclist.cfg file, specifying which computers he wants to collect the services status.

B.3.3.4 Option < 4 >: Some Services Status Auditing

This options differs from the last one by the subset of services that will be collect. In this option, the administrator is able to specify a subset of services that can be collect from a subset of computers.

To perform this option, the administrator needs to specify the subset of computers and the services. The serviceaudit.cfg file should be configured with the "name" of the services that will be collect.

```
Alerter;
NetDDE;
```

Table 6.4: Example of serviceaudit.cfg configuration file

B.3.3.5 Option < 5 >: Chance Service Status

It is also possible to change the status of any service. DoIt4Me permits the system administrator to start or stop any service in any subset of computers. Like the above configurations, it is only necessary to define the subset of computers (pclist.cfg), the subset of services and its news status, for example 1 to start or 0 to stop it.

The services and the values is configured on the file serviceconfig.cfg. Table 6.5 shows how to start the "alerter" service and how to stop the "NetDDE" service.

```
Alerter; 1
NetDDE; 0
```

Table 6.5: Example of serviceconfig.cfg configuration file

B.3.3.6 Option < 6 >: Ping

The administrator can ping a subset of computers. To do so, the administrator needs only to specify the subset of computers. The configuration is similar to the pclist.cfg, one computer name for each line.

File: ping.cfg

```
mustang;
ferrari;
```

Table 6.6: Example of ping.cfg configuration file

B.3.3.7 Option < 7 >: Reboot

This option permits the system administrator to reboot any subset of computer. To reboot a subset of computers it is necessary to configure the reboot.cfg file with the name of each computer (to be reboot) per line, and the rebootmsg.cfg file with the time (in seconds) to reboot, follow by the message to be send and the option 1 to reboot or 0 to shutdown.

```
porsch;
ferrari;
```

Table 6.7: Example of reboot.cfg configuration file

```
15;This Computer is Rebooting in 15 seconds; 1
```

Table 6.8: Example of reboot.cfg configuration file