Uma Proposta para Medição de Complexidade de Segurança em Procedimentos de Tecnologia da Informação

Giovane César Moreira Moura, Luciano Paschoal Gaspary

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{gcmmoura,paschoal}@inf.ufrgs.br

Resumo. *Segurança de TI se tornou nos últimos anos uma grande preocupação para empresas em geral. Entretanto, não é possível atingir níveis satisfatórios de segurança sem que estes venham acompanhados tanto de grandes investimentos para adquirir ferramentas que satisfaçam os requisitos de segurança quanto de procedimentos, em geral, complexos para instalar e manter a infra-estrutura protegida. A comunidade científica propôs, no passado recente, modelos e técnicas para medir a complexidade de procedimentos de configuração de TI. No entanto, apesar do papel central da segurança neste contexto, ela não foi objeto de investigação até então. Para abordar este problema, neste trabalho aplica-se um modelo de complexidade proposto na literatura para mensurar o impacto de segurança na complexidade de procedimentos de TI. A proposta deste trabalho foi materializada por meio da implementação de uma ferramenta para análise de complexidade denominada Security Complexity Analyzer (SCA), que foi utilizada para avaliar a complexidade de cenários reais de segurança.*

Abstract. *IT security has become over the recent years a major concern for organizations. However, it doesn't come without large investments on both the acquisition of tools to satisfy particular security requirements and, in general, complex procedures to deploy and maintain a protected infrastructure. The scientific community has proposed in the recent past models and techniques to estimate the complexity of configuration procedures, aware that they represent a significant operational cost, often dominating total cost of ownership. However, despite the central role played by security within this context, it has not been subject to any investigation so far. To address this issue, we apply a model of configuration complexity proposed in the literature in order to be able to estimate security impact on the complexity of IT procedures. Our proposal has been materialized through a prototypical implementation of a complexity scorer system called Security Complexity Analyzer (SCA), that was used to evaluate real-life security scenarios.*

1. Introdução

Segurança tem se tornado, ao longo dos anos, uma grande preocupação para as empresas, que dependem cada vez mais de infra-estruturas complexas de Tecnologia de Informação (TI) para realizar suas transações comerciais [Cazemier et al. 2000]. Neste contexto, tanto segurança *física* quanto *lógica* são cuidadosamente aplicadas nas operações diárias das empresas com o objetivo de fornecer diferentes níveis de proteção da informação. Confidencialidade, autenticidade, integridade, não-repúdio, controle de acesso e disponibilidade [Stallings 2006] são exemplos de serviços de segurança que diretores e gerentes de TI podem requisitar para preservar dados sensíveis, mantidos em *Configuration Itens* (CIs) presentes na infra-estrutura de TI [Cannon and Wheeldon 2007].

As demandas (ou requisitos) de segurança, expressas através de *Service Level Agreements* (SLAs) ou de políticas de segurança, são materializadas por meio de um conjunto de mecanismos tais como criptografia de sistemas de arquivos e de tráfego de rede, filtragem de pacotes e redundância de hardware/software. Estes mecanismos, por sua vez, podem ser implantados com ferramentas desenvolvidas por diferentes fornecedores. A instalação, a configuração e a manutenção destas ferramentas são realizadas tanto em procedimentos de TI mais gerais (por exemplo, junto à configuração de uma aplicação *web*) quanto em procedimentos voltados à implantação exclusiva de ferramentas de segurança.

Intuitivamente, quanto mais mecanismos de segurança a serem manuseados, mais complexos, longos e caros (em termos financeiros) os procedimentos correspondentes tendem a se tornar. Por exemplo, à medida que o número de mecanismos requisitados em uma instalação de sistema operacional aumenta, é de se esperar que o respectivo procedimento apresente um maior número de passos de execução e que demande mais parâmetros a serem fornecidos e lembrados. Além disso, há situações em que um mecanismo pode ser implantado com o uso de diferentes ferramentas (por exemplo, autenticação de usuários através de OpenLDAP ou Microsoft Active Directory), que podem também diferir em relação à complexidade de instalação.

Apesar de não haver dúvidas quanto à percepção de que atividades de segurança impactam negativamente a complexidade dos procedimentos de TI, ainda não se dispõe de uma abordagem sistemática e científica para caracterizar e, sobretudo, quantificar essa percepção. Neste contexto, determinar a *medida de complexidade* para estimar o custo da segurança de TI é fundamental por várias razões: primeiro, pode ser usada por diretores/gerentes para revisar os SLAs e as políticas de segurança levando em consideração os custos preditos associados a sua real implantação. Segundo, ela fornece diretrizes para a equipe de TI em relação a quais ferramentas utilizar (por exemplo, as de menor medida de complexidade). Terceiro, a avaliação de complexidade permite destacar os pontos de maior complexidade em um procedimento de segurança de TI que, por sua vez, são classificados como potenciais candidatos à automação [Brown et al. 2005].

A comunidade científica propôs no passado recente modelos e técnicas para estimar a complexidade de procedimentos de configuração e estabelecer sua relação com métricas de desempenho em nível de negócios, tais como custo financeiro e tempo [Diao et al. 2007]. Entretanto, apesar do papel central ocupado por segurança neste contexto, ela ainda não foi alvo de qualquer investigação.

Neste trabalho é aplicada a metodologia proposta por Brown *et al.* [Brown et al. 2005], com algumas extensões, para permitir tanto estimar o potencial impacto que tarefas de segurança podem causar em procedimentos de TI quanto mensurar/comparar medidas de complexidade associadas à manipulação de ferramentas que implementam mecanismos de segurança específicos. Esta proposta foi materializada em uma implementação de uma ferramenta para cálculo de complexidade denominada *Security Complexity Analyzer* (SCA). Como prova de conceito e viabilidade, a SCA foi utilizada para avaliar diversos cenários reais de segurança de TI.

O restante deste artigo está organizado da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados. Na Seção 3 é introduzido o modelo de segurança de TI, que abstrai a interação entre o administrador e a infra-estrutura de hardware e software a ser manipulada. Já na Seção 4 é detalhado o conjunto de métricas utilizadas para capturar a complexidade associada tanto a procedimentos quanto a ações. Na Seção 5 são apresentados e discutidos os resultados obtidos nos experimentos realizados com o apoio da SCA. Por fim, as conclusões e os trabalhos futuros são delineados na Seção 6.

2. Trabalhos relacionados

Nos últimos anos, vários esforços têm sido feitos para se obter uma metodologia que permita mensurar a complexidade tanto de procedimentos quanto de processos de TI. Apesar do reconhecido potencial que segurança pode ter sobre os indicadores de complexidade, este problema não foi investigado em trabalhos anteriores. Em paralelo, entretanto, diferentes perspectivas de segurança em TI têm sido analisadas. Refletindo estes esforços, esta seção primeiramente revisa os trabalhos mais proeminentes em relação à estimativa de complexidade procedimentos/processos de TI mais gerais e, então, discute outras contribuições que, indiretamente, se relacionam ao tema deste artigo.

Brown e Hellerstein [Brown and Hellerstein 2004] propuseram um arcabouço conceitual para obter um *benchmark* de complexidade de configuração, que foi melhor detalhado em estudos subseqüentes. No primeiro deles – de Brown *et al.* [Brown et al. 2005] – foi proposto um conjunto de métricas, inspirado na área de engenharia de software, para abstrair a complexidade de procedimentos de configuração. Os autores demonstraram que este conjunto de métricas foi capaz de capturar a redução de complexidade associada a um procedimento de configuração depois que algumas de suas ações foram automatizadas. Um passo adiante foi dado por Diao *et al.* [Diao et al. 2007], em que as métricas supracitadas foram utilizadas como entrada para criação de um modelo quantitativo que permitiu derivar métricas de nível de negócios, como tempo de execução e custo. Finalmente, Keller *et al.* descreveram [Keller et al. 2007] a ferramenta utilizada para capturar e analisar os dados de complexidade.

Abordando o problema a partir da perspectiva de negócios, Diao e Keller [Diao and Keller 2006] estenderam as métricas propostas por Brown *et al.* [Brown et al. 2005] para que fosse possível quantificar a complexidade associada a processos de gerenciamento de serviços de TI [Cannon and Wheeldon 2007]. Diferentemente dos procedimentos, que são diretamente relacionados ao nível de sistema (por exemplo, instalação, configuração e manutenção de CIs), processos se relacionam ao nível dos negócios. Neste contexto, as novas métricas capturam aspectos como a tomada de decisão entre múltiplos papéis (*roles*) e a interação entre dois ou mais deles.

Em se tratando de segurança de TI, não há trabalhos publicados, até onde sabemos, que objetivem determinar como a complexidade dos procedimentos é afetada pela manipulação de diferentes mecanismos de segurança. Ao contrário, por exemplo, os *benchmarks* de segurança CIS [CIS 2008] se concentram em determinar o nível de proteção de um sistema previamente instalado em relação a uma lista extensiva de ameaças. Em se tratando dos aspectos econômicos de segurança da informação, Cavusoglu *et al.* [Cavusoglu et al. 2004] apresentaram um modelo econômico que permite estimar a quantidade ótima a ser investida na segurança da informação, que utiliza como entradas, entre outras variáveis, os danos causados por falhas de segurança, bem como os parâmetros de qualidade e os custos de instalação/manutenção/configuração dos mecanismos de segurança disponíveis. Os princípios por trás da estimativa dos custos de instalação/manutenção/configuração – problema que poderia ser comparado ao abordado neste trabalho – não são descritos pelos autores.

Nas próximas seções é apresentada uma proposta para avaliar a complexidade de segurança em procedimentos de TI.

3. Modelo de segurança de TI

Para mensurar objetivamente complexidade de segurança em procedimentos de TI é fundamental ter acesso a uma visão precisa da infra-estrutura e, sobretudo, dos procedimen-

tos sob análise. Nesta seção é apresentado um modelo de segurança de TI que serve justamente a esse propósito.

Mais especificamente, o modelo de segurança de TI permite expressar uma abstração da infra-estrutura de TI de uma empresa, incluindo os mecanismos de segurança a serem instalados ou modificados, bem como os respectivos procedimentos. Baseado na proposta de Brown *et al.* [Brown et al. 2005], este modelo é composto de duas partes complementares: *infra-estrutura* e *atividade*. A primeira parte representa os componentes de hardware/software e o grau em que eles são relacionados com os serviços de segurança. Já a segunda, por sua vez, abstrai a seqüência de atividades que devem ser executadas para se alcançar um objetivo em particular.

Para ilustrar este modelo, apresenta-se um cenário exemplo que consiste na instalação da aplicação *web* Joomla [Joomla 2008] – que é um sistema de gerenciamento de portais desenvolvido em PHP – e todo o software necessário para que ela seja executada (por exemplo, o sistema de banco de dados MySQL e o servidor *web* Apache httpd). O cenário é composto por dois computadores (um hospeda o servidor de banco de dados e o outro, o servidor *web*) e um *firewall*, que filtra os pacotes destinados aos computadores. Além disso, o ambiente é incrementado com um conjunto de mecanismos de segurança, sumarizados na Tabela 1. Por exemplo, Dm-crypt [Dm-crypt 2008] e OpenSSL [OpenSSL 2008] são empregados para criptografar o conteúdo do sistema de arquivos do computador 1. As duas partes são apresentadas a seguir.

Tabela 1. Mecanismos de segurança utilizados no cenário de exemplo

Computador 1

Mecanismo de Segurança	*Ferramenta*
Criptografia do sistema de arquivos	dm-crypt e OpenSSL
Criptografia das conexões com o banco de dados	MySQL and OpenSSL

Computador 2

Mecanismo de Segurança	*Ferramenta*
Criptografia do sistema de arquivos	dm-crypt e OpenSSL
Criptografia das conexões com o servidor *web*	httpd and OpenSSL

Firewall

Mecanismo de Segurança	*Ferramenta*
Filtro de pacotes	netfilter/iptables

3.1. Modelo de infra-estrutura de TI

A infra-estrutura de TI pode ser modelada como um conjunto de contêineres, que representam recursos do sistema ou contêineres hospedeiros. A Figura 1 ilustra o modelo de infra-estrutura de TI para o cenário exemplo recém mencionado. O contêiner `Computador 1` é uma representação do computador onde o servidor de banco de dados é executado, enquanto o contêiner `Computador 2` representa o servidor que hospeda a aplicação *web*. Por sua vez, estes contêineres hospedam outros contêineres, como `Slackware_app` e `Slackware_bd` (sistema operacional de ambos os computadores). Pode-se notar que são utilizadas linhas pontilhadas e sólidas na figura, cuja finalidade é diferenciar os contêineres hospedeiros daqueles que representam recursos do sistema (como, por exemplo, `MySQL tables`).

Na mesma figura, a cor dos contêineres reflete como eles se relacionam com segurança. Na verdade, esta proposta é uma extensão do modelo de infra-estrutura de TI proposto em [Brown et al. 2005]. Contêineres em preto são aqueles cuja funcionalidade

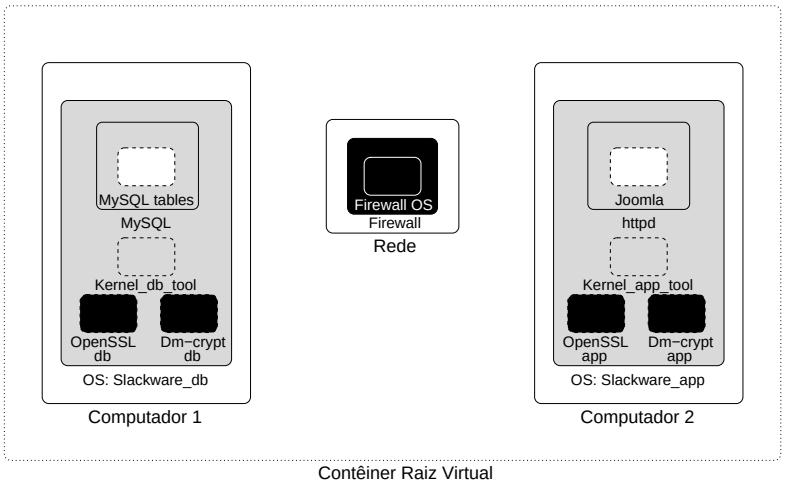

Figura 1. Modelo de infra-estrutura de TI

principal é diretamente relacionada com os mecanismos de segurança (por exemplo, o contêiner `Dm-crypt_db`, responsável por criptografar o sistema de arquivos do computador hospedeiro do banco de dados). Em contraste, contêineres representados em branco não são afetados quando mecanismos de segurança são empregados (por exemplo, `MySQL tables`). Por fim, contêineres em cinza caracterizam CIs que, apesar de não serem relacionados diretamente com segurança, podem necessitar de alterações para que os mecanismos de segurança demandados sejam suportados. Por exemplo, o servidor de banco de dados MySQL requer uma configuração especial para suportar o uso de criptografia em suas conexões (com o uso de OpenSSL).

3.2. Modelo de atividade

O modelo de atividade provê uma abstração para a interação entre o administrador e o sistema a ser manipulado, e é baseado em três pilares: *objetivos*, *procedimentos* e *ações* (ou *tarefas*). Objetivo pode ser definido como o estado final a ser atingido pela infra-estrutura de TI (como, por exemplo, a ativação do protocolo SSL para comunicações com o servidor *web*). Por sua vez, procedimento pode ser definido como uma seqüência de passos a serem seguidos para se atingir um objetivo específico. Por fim, uma ação denota um passo individual em um procedimento (como, por exemplo, a criação de uma chave privada).

A Figura 2 ilustra o procedimento para instalar a aplicação *web* Joomla com os mecanismos de segurança especificados na Tabela 1 . Ele é composto de 77 ações (representadas por retângulos). Cada ação é identificada por um título e o contêiner em que ela é executada, sendo que este último é indicado pelo uso de parênteses. Os parâmetros consumidos por uma ação são destacados por setas pontilhadas que se ligam às respectivas ações. Por exemplo, a ação de número 1 possui o título “Select kernel”, seu contêiner é o `Slackware_db` e ela consome o parâmetro `kernel_db`.

As ações são representadas em preto, branco e cinza, seguindo a mesma semântica utilizada no modelo de infra-estrutura de TI. Contêineres pretos e brancos são expandidos em ações pretas e brancas, respectivamente. Contêineres cinza são expandidos em ações cinzas e brancas, em que as primeiras representam ações que precisaram ser adicionadas ao procedimento para dar suporte aos mecanismos de segurança. Por exemplo, o sub-procedimento referente à instalação do contêiner `Slackware_db` (representado em cinza no modelo de infra-estrutura de TI, na Figura 1) é composto de ações relacionadas

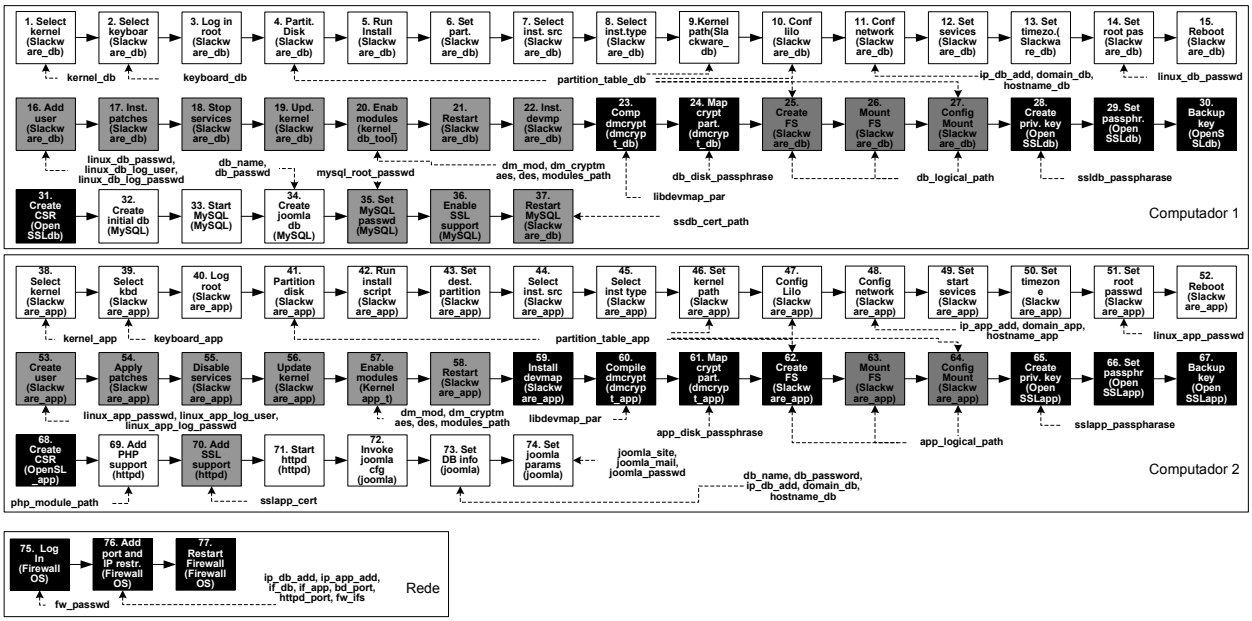

Figura 2. Procedimento completo para instalar Joomla com mecanismos de segurança

aos mecanismos de segurança (tais como as ações 16 a 22 e de 25 a 27) e ações ordinárias (como, por exemplo, as de 1 a 15). Em um primeiro momento, poderia-se questionar o porquê da ação de número 73 na Figura 2 não ser representada em preto, uma vez que é nela que são informadas as credenciais para autenticação (`db_name` e `db_passwd`). A razão para isso é que esta ação não é relacionada à implantação de quaisquer mecanismos de segurança especificados na Tabela 1. Além disso, ela não é uma ação opcional, uma vez que necessita ser obrigatoriamente executada para que se possa instalar o Joomla – com ou sem os mecanismos de segurança.

4. Métricas de complexidade

O modelo apresentado na seção anterior deve ser aplicado com um conjunto de métricas capaz de capturar a complexidade associada aos procedimentos abstraídos, relacionados à TI. As métricas utilizadas neste trabalho foram propostas por *Brown et al.* [Brown et al. 2005], que aplicaram as mesmas para capturar a complexidade de procedimentos gerais de configuração. Em uma primeira tentativa em se determinar a complexidade associada aos mecanismos de segurança, não foram propostas novas métricas. Ao invés disso, foram aplicadas aquelas utilizadas com sucesso no contexto dos procedimentos de configuração mais gerais.

As métricas de complexidade empregadas neste trabalho são divididas em três grupos: *execução*, *parâmetro* e *memória*. Essas métricas podem ser empregadas para realizar o cálculo de complexidade tanto no nível de *procedimentos* quanto no nível de *ações*. Mais detalhes sobre o uso das métricas de complexidade em cada nível são apresentados nas próximas sub-seções.

4.1. Complexidade em nível de procedimentos

Esta subseção apresenta as métricas de complexidade utilizadas para capturar a complexidade dos procedimentos. A avaliação de complexidade neste nível permite obter um panorama geral da complexidade associada aos respectivos procedimentos.

Complexidade de execução

As métricas pertencentes ao grupo de execução têm por função capturar a complexidade relacionada ao ato de executar a seqüência de ações definida nos procedimentos. Elas são descritas a seguir.

- *NumActions*: contabiliza o número de ações de um procedimento.
- *ContextSwitchSum*: efetua o somatório dos valores associados a todas as trocas de contextos *(ContextSwitch*) presentes em um procedimento. Uma troca de contexto ocorre quando um contêiner de uma ação difere do contêiner da ação anterior, assim como entre as ações 74 e 75 da Figura 2. O valor desta troca de contexto é calculado através da distância entre o contêiner da ação de origem e o contêiner raiz que o interliga com a ação de destino na *árvore hierárquica de coitêineres*. Essa árvore é derivada diretamente a partir do modelo de Infra-estrutura de TI ilustrado na Figura 1. A Figura 3 representa a árvore hierárquica referente ao cenário usado como exemplo. Pode-se observar, na mesma figura, que o valor da troca de contexto entre as ações 74 e 75 é 4.

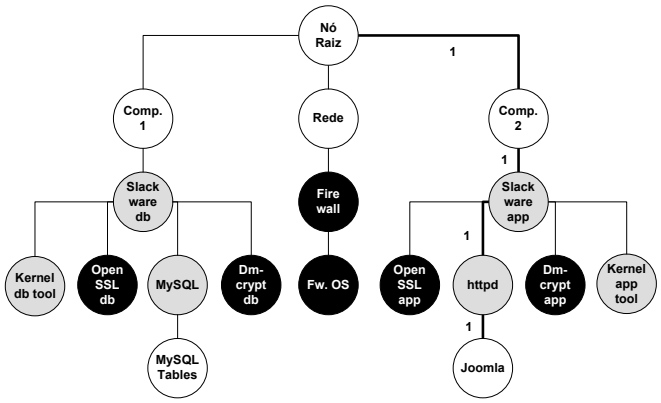

Figura 3. Árvore hierárquica de contêineres.

Complexidade de parâmetros

As métricas de complexidade definidas no grupo dos parâmetros são utilizadas para mensurar a complexidade associada ao fornecimento de parâmetros durante a execução do procedimento (seja pelo operador humano ou pelo sistema). As métricas são definidas a seguir.

- *ParamCount*: contabiliza o número de parâmetros distintos utilizados em um procedimento.
- *ParamUseCount*: contabiliza o número de parâmetros consumidos em um procedimento, levando em consideração as repetições.
- *ParamCrossContext*: totaliza a soma das ocorrências de trocas de contextos associadas a ações que utilizam um mesmo parâmetro em contêineres distintos.
- *ParamAdaptCount*: fornece o número de parâmetros utilizados de forma sintática diferente ao longo do procedimento (por exemplo, o caminho relativo e absoluto do *script* de inicialização do httpd).
- *ParamSourceScore*: efetua o somatório das medidas de *SourceScore* associadas a cada parâmetro distinto utilizado no procedimento. *SourceScore* reflete a dificuldade de se determinar os valores dos parâmetros, variando de 0 a 6. Neste trabalho

é proposta uma forma de quantificar esta métrica, de acordo com as regras apresentadas na Tabela 2. Por exemplo, o parâmetro da ação `php_module_path` possui um valor de 6, já que é obtido fora do contexto em questão. Para o procedimento da Figura 2, o valor calculado para a métrica *ParamSourceScore* é 110.

Tabela 2. Valores de *SourceScore* para avaliação dos parâmetros.

Valor	*Propriedade do parâmetro*
0	Sugerido - preenchido automaticamente
1	Seleção (por exemplo, combo-box)
2	Não sugerido - no mesmo contêiner - documentado
3	Não sugerido -no mesmo contêiner - não documentado
4	Não sugerido - em outro contêiner - documentado
5	Não sugerido - em outro contêiner - não documentado
6	Não sugerido -fora do ambiente

Complexidade de memória

O grupo de métricas de memória é utilizado para avaliar o número de parâmetros que devem ser lembrados durante a execução do procedimento, bem como o intervalo que eles necessitam permanecer armazenados em memória. Para realizar esta quantificação, a memória do administrador é modelada como uma pilha *Last-In-First-Out* (LIFO) com busca não-associativa. Seis métricas associadas à complexidade de memória são empregadas no contexto deste trabalho:

- *MemSizeAverage* e *MemSizeMax*: contabilizam os tamanhos médio e máximo da pilha durante a execução do procedimento.
- *MemDepthAverage* e *MemDepthMax*: contabilizam a profundidade média e máxima dos parâmetros acessados na pilha durante a execução do procedimento.
- *MemLatAverage* e *MemLatMax*: contabilizam a latência média e máxima dos parâmetros inseridos na pilha. Latência indica o intervalo (medido em número de ações) que um parâmetro deve permanecer na memória do administrador antes de ser utilizado numa ação subseqüente.

4.2. Complexidade em nível de ações

Para determinar a complexidade no nível das ações é necessário que o cálculo de algumas das métricas previamente apresentadas seja levemente modificado. Em relação às métricas que avaliam a complexidade de execução, *NumActions* sempre recebe 1 como valor, enquanto *ContextSwitchSum* recebe o valor de *ContextSwitch* da ação, em relação a sua predecessora.

As métricas relacionadas à complexidade de parâmetros também são ligeiramente diferenciadas. *ParamCount* não é aplicado no contexto de ações, enquanto *ParamUseCount* enumera o número de parâmetros que a ação em questão consume. *ParamAdaptCount* contabiliza o número de parâmetros que uma ação reutiliza de forma sintática diferente. Já *ParamSourceScore* contabiliza a soma dos valores de *SourceScore* relacionados aos parâmetros consumidos pela ação que aparecem no procedimento pela primeira vez. Por fim, as métricas do grupo de memória (*MemSize*, *MemDepth* e *MemLat*) já são calculadas em relação a ações e, desta forma, não requerem nenhuma mudança. Os valores médios e máximos destas métricas não são aplicados a ações.

5. Avaliação Experimental

Nesta seção é apresentada a abordagem empregada para mensurar a complexidade associada a procedimentos de segurança de TI, tendo como base as métricas descritas na Seção 4. Para automatizar o processo de análise de complexidade, foi empregada a SCA, uma ferramenta desenvolvida em ambiente Linux utilizando a linguagem de programação Java. Seu funcionamento básico consiste em analisar um arquivo XML como entrada – onde são descritos os procedimentos, incluindo ações, contêineres e parâmetros – e, a partir desses dados, calcular os valores das métricas de complexidade, como definido na Seção 4. Os resultados desta análise são posteriormente armazenados em uma base de dados de complexidade de segurança e exibidas ao administrador.

Com o apoio da ferramenta SCA, foi realizada uma série de experimentos para que fosse possível avaliar a complexidade relacionada à segurança em três dimensões: (i) complexidade adicional que mecanismos de segurança impõem a procedimentos de TI mais gerais; (ii) medida de complexidade de procedimentos que manipulam mecanismos de segurança de forma isolada; e (iii) comparação de medidas de complexidade associada a procedimentos demandados por diferentes ferramentas que satisfazem um mesmo mecanismo de segurança. Em seguida é apresentada em mais detalhes cada uma dessas dimensões, bem como a forma de uso das métricas de complexidade e uma discussão acerca dos resultados obtidos na avaliação dos cenários.

5.1. Complexidade adicional agregada por mecanismos de segurança em procedimentos de TI mais gerais

Segurança (e suas atividades) podem fazer parte de procedimentos mais gerais, como, por exemplo, na instalação e configuração de uma aplicação web segura. Para determinar a medida de complexidade que os mecanismos de segurança podem agregar a procedimentos mais gerais, primeiramente deve-se modelar um procedimento base, que não contém quaisquer mecanismos de segurança e, então, proceder com a avaliação de complexidade. Em seguida, um novo procedimento deve ser modelado – consistindo no procedimento base enriquecido com ações e parâmetros demandados pelos mecanismos de segurança especificados – e, então, deve-se realizar a sua avaliação de complexidade novamente. A diferença entre os valores observados para o novo procedimento e o procedimento base ($\Delta_{complexidade}$) reflete a complexidade adicional imposta pelos mecanismos de segurança no contexto específico em questão.

Para avaliar esta abordagem, foram analisados três cenários típicos, comumente encontrados em empresas. Eles compartilham um objetivo em comum – a instalação e configuração da aplicação web Joomla e todo o software que ela depende. Entretanto, os cenários diferem em relação aos mecanismos de segurança implantados. O primeiro, cenário A, é o cenário base e, portanto, não inclui qualquer mecanismo de segurança dentre os especificados na Tabela 1. Os cenários B e C podem ser vistos como versões, com o nível de segurança incrementado, do cenário A. O primeiro satisfaz parte dos mecanismos de segurança especificados na Tabela 1 (somente aqueles aplicados ao computador 1), enquanto o segundo implanta todos os mecanismos listados na mesma tabela.

Para prosseguir com a avaliação de complexidade, foram definidos procedimentos e árvores de hierarquias específicos para cada cenário, que então foram utilizados como entrada para a SCA. Como exemplo, a Figura 2 ilustra o procedimento do cenário C e a Figura 3 representa a respectiva árvore de hierarquia. Os resultados (por métrica) obtidos para os cenários A, B e C são sumarizados na Tabela 3. Os valores entre parênteses indicam a variação percentual da complexidade que os cenários B e C apresentam em relação ao cenário A.

Tabela 3. Medidas de complexidade para os cenários A, B e C

Métrica	*Cenário A*	*Cenário B*	*Cenário C*
NumActions	38	55 (44,7%)	77 (102,6%)
ContextSwitchSum	6	11 (83,3%)	21 (250,0%)
ParamCount	20	32 (60,0%)	46 (130,0%)
ParamUseCount	31	45 (48,3%)	72 (132,2%)
ParamAdaptCount	0	0 (0%)	0 (0%)
ParamCrossContext	21	21 (0%)	45 (114,2%)
ParamSourceScore	45	74 (64,4%)	110 (144,4%)
MemSizeAvg	3,42	4,23 (23,6%)	7,38 (115,7%)
MemSizeMax	6	7 (16,6%)	14 (133,3%)
MemLatAvg	3,42	4,23 (23,6%)	7,38 (115,7%)
MemLatMax	116	201 (73,2%)	264 (127,5%)
MemDepthAvg	0,52	0,41 (-21,1%)	1,21 (132,6%)
MemDepthMax	15	15 (0%)	40 (166,6%)

Examinando a tabela, pode-se notar que a implantação dos mecanismos de segurança impactou significativamente as medidas de complexidade dos cenários B e C em relação aos valores observados para o cenário base A (para quase todas as métricas). Por exemplo, com exceção da métrica *ParamAdaptCount*, os valores medidos para o cenário C foram, no mínimo, o dobro quando comparados ao cenário A. Em relação ao cenário B, as diferenças foram, no geral, também altas. Pode-se observar, entretanto, que o valor da métrica *MemDepthAvg* foi menor do que o valor calculado para o cenário A. Isto se deve ao fato de que a soma dos valores associados a *MemDepth* no cenário B não ter sido suficientemente alta, quando comparada ao cenário A, para acompanhar o valor maior do denominador (*NumActions*) utilizado para calcular as médias (53 para o cenário B e 38 para o cenário A).

Os resultados também destacam a grande quantidade de informação que um administrador deve lembrar enquanto executa estes procedimentos (*MemSizeMax*): no pior caso, 14 itens. Esse valor excede em pelo menos 5 o valor aceito comumente para a capacidade da memória de curto prazo dos humanos [Miller 1956]. Uma análise destes itens revela que 6 deles são diretamente relacionados com a instalação de mecanismos de segurança. Seguindo o aumento no número de itens a serem armazenados, o intervalo que eles devem ser mantidos na memória do administrador (*MemLatAvg*) também aumenta a medida que o cenário se torna mais sofisticado.

Uma visão mais detalhada dos pontos de maior complexidade em um procedimento pode ser obtida por meio da avaliação no nível das ações. A Figura 4 ilustra os valores mensurados para um subconjunto de ações do cenário C. As três barras representam os valores de complexidade para os grupos de métricas de execução, parâmetro e memória, respectivamente. Além disso, a cor de cada barra no gráfico denota a relação da ação com segurança (como indicado nos modelos de infra-estrutura de TI e de atividade). A ação 62, $Create\ filesystem$, é o passo final que necessita ser executado para criptografar o sistema de arquivos no computador 2 e, desta forma, é diretamente relacionado com segurança. O alto valor medido para a complexidade de memória desta ação é devido à necessidade do administrador se lembrar e reutilizar um parâmetro ($partition\_table\_app$) consumido 15 passos antes.

5.2. Medida de complexidade de procedimentos que manipulam mecanismos de segurança isolados

Ao contrário do problema formulado na subseção anterior, há situações em que mecanismos de segurança precisam ser instalados/desinstalados e/ou manutenidos em uma infra-

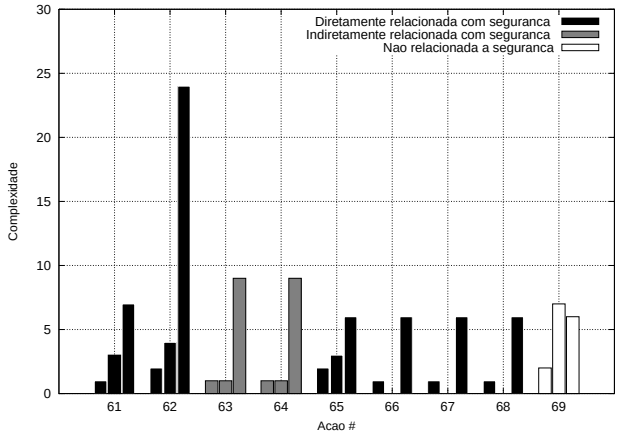

Figura 4. Complexidade de ações

estrutura de TI já existente e em operação. Por exemplo, considere o caso em que um gerente de TI determina o uso de criptografia no sistema de arquivos de um servidor (onde dados confidenciais são armazenados). Para que se possa mensurar a complexidade de procedimentos como esse, deve-se, primeiramente, especificar os mesmos isoladamente – isto é, ignorando as ações não relacionadas aos próprios mecanismos de segurança – e, então, efetuar a avaliação de complexidade.

Para esta dimensão em particular foi realizada a avaliação de complexidade de quatro cenários que implementam diferentes mecanismos de segurança em um servidor em produção, executando o sistema operacional Slackware Linux com os serviços básicos de sistema e rede (incluindo o servidor web httpd da Apache). No primeiro cenário, D, foi instalado e configurado o OpenSSL [OpenSSL 2008] para possibilitar comunicação criptografada com o servidor web. Já no segundo cenário, E, foi feita a criptografia do sistema de arquivos do servidor utilizando a ferramenta dm-crypt [Dm-crypt 2008], enquanto que no terceiro cenário, F, foi instalado e configurado um filtro de pacotes utilizando netfilter/iptables [Netfilter/iptables 2008]. Por fim, no cenário G os três mecanismos de segurança foram implantados em conjunto.

A Tabela 4 sumariza os resultados obtidos. Comparando os cenários D, E e F, pode-se notar que o cenário E – referente à criptografia do sistema de arquivos – apresenta as maiores medidas de complexidade, refletindo o tempo e o esforço que os administradores percebem ao executá-lo. Para instalar o dm-crypt, foi necessário, dentre outras ações, compilar o mesmo a partir do seu código fonte e ativar módulos específicos no kernel do Linux, o que resultou em um procedimento contendo mais ações, parâmetros e trocas de contextos (quando comparados com aqueles executados para os cenários D e F). Além disso, o mesmo procedimento demandou mais parâmetros (10) que foram, em geral, os mais difíceis de se obter (refletido pelo valor da métrica *ParamSourceScore*).

A partir da tabela, pode-se notar também que a medida de complexidade para o cenário G é, em geral, menor do que aquela representada pela soma dos valores computados para os cenários D, E e F. Por exemplo, *NumActions* resultou em 17 para G, enquanto que para D+E+F o valor foi 20. Num primeiro momento, pode causar supresa o fato de o resultado derivado da combinação dos três mecanismos de segurança não representar a soma dos resultados individuais . Isto é devido à existência de ações compartilhadas (e parâmetros) quando os três mecanismos são executados em um único procedimento, como a ação de se autenticar em um sistema operacional utilizando nome de usuário e senha como parâmetros.

Tabela 4. Medidas de complexidade para os procedimentos dos cenários D, E, F e G

Métrica/Cenário	*D*	*E*	*F*	*G*
NumActions	7	10	3	17
ContextSwitchSum	2	4	1	7
ParamCount	3	10	8	19
ParamUseCount	3	13	8	22
ParamAdaptCount	0	0	0	0
ParamCrossContext	0	0	0	0
ParamSourceScore	9	31	16	52
MemSizeAvg	0	0,4	0	0,23
MemSizeMax	0	2	0	2
MemLatAvg	0	0,4	0	0,23
MemLatMax	0	3	0	3
MemDepthAvg	0	0,5	0	0,29
MemDepthMax	0	3	0	3

5.3. Comparação de medidas de complexidade associadas a procedimentos demandados por diferentes ferramentas que implementam um mesmo mecanismos de segurança

Quando um gerente de TI avalia as possibilidades para satisfazer os requisitos definidos em um SLA ou em uma política de segurança, ele pode se deparar com situações em que mais de uma ferramenta de segurança pode ser utilizada para atingir resultados equivalentes ou similares. Neste contexto, a comparação das medidas de complexidade associadas aos procedimentos (por exemplo, de instalação, remoção, configuração e manutenção) demandados por diferentes ferramentas pode ajudar na escolha entre uma ferramenta em detrimento de outra.

Tabela 5. Medidas de complexidade para a instalação de OpenVPN e Openswan

Métrica	*OpenVPN*	*Openswan*
NumActions	13	20
ContextSwitchSum	5	14
ParamCount	24	38
ParamUseCount	40	69
ParamAdaptCount	0	0
ParamCrossContext	0	5
ParamSourceScore	66	94
MemSizeAvg	1,23	4,85
MemSizeMax	8	10
MemLatAvg	1,23	4,85
MemLatMax	8	43
MemDepthAvg	5,53	6,7
MemDepthMax	36	45

Para ilustrar com um exemplo concreto, foi realizada uma avaliação de complexidade dos procedimentos associados à instalação e configuração de diferentes ferramentas que implementam *Virtual Private Networks* (VPNs). As ferramentas utilizadas nos experimentos foram OpenVPN [OpenVPN 2008] e Openswan [Openswan 2008]. Para prover canais de comunicação seguros, a primeira emprega *Secure Sockets Layer* (SSL), enquanto a segunda utiliza *IP Security* (IPSec). Com o intuito de realizar uma comparação justa, as ferramentas foram avaliadas sob as mesmas condições. OpenVPN e Openswan foram instaladas em duas máquinas equivalentes, permitindo aos usuários remotos realizar conexões utilizando a Internet (modo *road warrior* de VPN). Ambas as ferramentas

foram configuradas para utilizar certificados X.509 para autenticação de sessões, processo que inclui, também, a criação de uma *Certificate Authority* (CA) e a emissão dos certificados.

A Tabela 5 sumariza os resultados obtidos neste experimento. Comparando as ferramentas, pode-se observar que a instalação e configuração da Openswan é significativamente mais complexa de se realizar do que a da OpenVPN (demandando 53% mais ações, 180% mais trocas de contexto e 58% mais parâmetros). Isto se deve ao fato da última prover *scripts* que automatizam a criação da CA e a emissão dos certificados, quanto que para a primeira é necessário realizar as operações manualmente. Além disso, a Openswan requer a instalação do gmp (GNU *Multiple Precision Arithmetic Library*) e a configuração explícita da interface de rede para aceitar redirecionamento de pacotes.

6. Conclusões e Trabalhos Futuros

Apesar do reconhecido papel ocupado por segurança na complexidade dos procedimentos de TI, esta tem sido caracterizada ao longo dos anos de maneira absolutamente intuitiva, sem o suporte de uma forma mais objetiva e mensurável (a exemplo do que se atingiu na área de Engenharia de Software para mensurar o "custo"associado ao desenvolvimento de sistemas). Em uma primeira interação para tratar este problema, foi aplicado neste trabalho um modelo de complexidade de configuração proposto por Brown *et al.* [Brown et al. 2005] para compreender a extensão da influência de segurança nos procedimentos de TI.

Os resultados obtidos, apesar de preliminares, são encorajadores. Foi possível realizar a avaliação em três dimensões: complexidade adicional agregada por mecanismos de segurança em procedimentos de TI, medidas de complexidade de procedimentos de segurança isolados e comparação de complexidade de procedimentos relativos a ferramentas que implementam mecanismos de segurança semelhantes.

Importantes lições foram aprendidas a partir dos experimentos. Primeiro, é difícil isolar e determinar a complexidade total que os mecanismos de segurança acarretam quando estes permeiam procedimentos de TI mais gerais. Isso se deve ao fato de (i) os valores calculados para as métricas (especialmente as do grupo de parâmetro e de memória) serem muito sensíveis à ordem em que as ações são executadas e (ii) existirem ações de segurança em contêineres que não possuem relação direta com segurança (como as ações em cinza presentes na Figura 2). Para superar tais empecilhos, foi utilizada uma abordagem sistemática e objetiva, descrita na Seção 5.1. Outra lição aprendida foi que as medidas de complexidade calculadas para os mecanismos de segurança são fortemente dependentes da infra-estrutura disponível e, desta forma, não são facilmente generalizadas para outros cenários.

A principal *contribuição* deste trabalho reside em uma proposta sistemática para isolar e mensurar a complexidade decorrente da implantação de mecanismos de segurança em procedimentos mais gerais de TI ou específicos, tendo por base o modelo de complexidade de configuração proposto por Brown *et al.* [Brown et al. 2005]. Esta proposta é fundamental para que se possa compreender o impacto ou sobrecarga imputados à segurança em procedimentos executados para manter a infra-estrutura de TI em pleno funcionamento. Além disso, os resultados obtidos representam um *importante* passo em direção à determinação de uma metodologia de *benchmarking* de complexidade de procedimentos associados a ferramentas de segurança. Esses *benchmarks* podem ser empregados com diferentes propósitos, como os três explorados neste artigo. Traduzir os valores observados para informações mais próximas de quem lida com operações de TI, como tempo e custo de execução estimados para executar um dado procedimento, é o tópico atualmente sob

investigação em nosso grupo. Para tal, estamos trabalhando na definição de um *modelo quantitativo* que, com base em valores calculados de métricas e medições de tempo realizadas sobre classes de procedimentos, nos permitirá realizar previsões, com certo grau de certeza, acerca do tempo de execução de procedimentos de segurança. Para realizar previsões do tempo necessário (e, indiretamente, custos), o modelo deverá receber, como entrada, o resultado da análise de complexidade de cenários segurança, introduzida neste trabalho.

Referências

(2008). Center for Internet Security. http://www.cisecurity.org/.

Brown, A. B. and Hellerstein, J. L. (2004). An approach to Benchmarking Configuration Complexity. In *Proceedings of the 11th ACM SIGOPS European Workshop*, page 18, Leuven, Belgium. ACM Press.

Brown, A. B., Keller, A., and Hellerstein, J. L. (2005). A Model of Configuration Complexity and its Application to a Change Management System. In IEEE, editor, *Proc. IFIP/IEEE International Symposium on Integrated Network Management*, IFIP/IEEE International Symposium on Integrated Network Management, pages 631–644, Nice, France.

Cannon, D. and Wheeldon, D. (2007). *Service Operation Itil, Version 3 (Itil)*. Stationery Office.

Cavusoglu, H., Mishra, B., and Raghunathan, S. (2004). A model for evaluating it security investments. *Commun. ACM*, 47(7):87–92.

Cazemier, J. A., Overbeek, P. L., and Peters, L. M. (2000). *Security Management (IT Infrastructure Library Series)*. Stationery Office, UK.

Diao, Y. and Keller, A. (2006). Quantifying the Complexity of IT Service Management Processes. In IEEE, editor, *Proc. of IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, Dublin, Ireland. IEEE.

Diao, Y., Keller, A., Parekh, S., and Marinov, V. V. (2007). Predicting Labor Cost throught IT Management Complexity Metrics. In IEEE, editor, *Proc. IFIP/IEEE International Symposium on Integrated Network Management*, IFIP/IEEE International Symposium on Integrated Network Management, Munich, Germany. IEEE.

Dm-crypt (2008). http://www.saout.de/misc/dm-crypt/.

Joomla (2008). http://www.joomla.org/.

Keller, A., Brown, A. B., and Hellerstein, J. L. (2007). A Configuration Complexity Model and Its Application to a Change Management System. *Network and Service Management, IEEE Transactions on*, 4(1):13–27.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97.

Netfilter/iptables (2008). http://www.netfilter.org/.

OpenSSL (2008). http://www.openssl.org.

Openswan (2008). http://www.openswan.org/.

OpenVPN (2008). http://openvpn.net/.

Stallings, W. (2006). *Network Security Essentials: Applications and Standards (3rd Edition)*. Prentice Hall.