

# Sistemas sobreviventes a intrusões utilizando replicação e auditoria de ações

F. R. Pellissari, *Mestrando, UFSC*, R. R. Righi, *Mestrando, UFSC*, e C. M. Westphall, *Doutora, UFSC*

**Abstract**—Current intrusion-detection systems empathize security violations identification and alarm triggering to the invaded system responsible entities. Consequently, the concerns with measures that can stop intrusions in the same moment they happen are left behind or are inexistent. This article specifies techniques which soften the damages caused by an invasion through replication and audit operations in intrusion detection systems. The developed prototype validates the proposed techniques and helps computer network managers to reinforce their security politics.

**Resumo**—Os atuais sistemas de detecção de intrusão enfatizam a identificação das violações de segurança e o envio de alarmes aos gerentes das redes. Conseqüentemente, as preocupações com medidas que parem as intrusões no mesmo momento em elas aconteçam são deixadas em segundo plano ou são inexistentes. Esse artigo especifica técnicas que amenizam os estragos causados por uma invasão através do uso de replicação e operações de auditoria em sistemas de detecção de intrusão. O protótipo desenvolvido valida as técnicas descritas neste documento e auxilia para que gerentes das redes de computadores reforcem suas políticas de segurança.

**Index Terms**—Security, Survivability.

## I. INTRODUÇÃO

AS preocupações do homem com a segurança existem desde antes do advento dos computadores. O homem sempre buscou proteger-se de ameaças, sejam elas naturais ou provenientes de outros seres humanos. O aprimoramento das técnicas de defesa foi constante na evolução da sociedade; porém a tecnologia e as ferramentas criadas nesse período também propiciaram o crescimento das ameaças e sua sofisticação. Essa idéia é facilmente percebida na segurança de computadores, onde vê-se um crescimento na mesma proporção de mecanismos de segurança e riscos às infraestruturas das organizações

Embora a segurança computacional aumente a cada dia com o surgimento de novas técnicas de criptografia, tecnologias de *firewall*, sistemas de detecção de intrusão<sup>1</sup>, ou outros mecanismos pró-ativos de proteção, não existem sistemas totalmente impenetráveis ou invulneráveis [11].

F.R.Pellissari é mestrando do curso CPGCC-UFSC (e-mail: rolim@lrg.ufsc.br)

R. R. Righi J. é mestrando do curso CPGCC-UFSC (e-mail: righi@lrg.ufsc.br)

C. M. Westphall é Professora Doutora do curso CPGCC-UFSC (e-mail: carla@lrg.ufsc.br)

1 Intrusion Detection Systems (IDS).

Algoritmos e arquiteturas tidas como seguras e bem projetadas na atualidade, podem se tornar obsoletas e fracas no futuro.

A partir dessa situação, a sobrevivência computacional<sup>2</sup> aceita que as invasões em sistemas são inevitáveis e utiliza técnicas para minimizar os estragos que sistemas podem sofrer diante de uma intrusão, dessa forma reforçando a segurança [2]. A sobrevivência computacional trabalha ainda com a recuperação de sistemas que sofreram acidentes ou falhas, além da recuperação na existência de uma invasão.

A recuperação de um sistema através do uso de técnicas de tolerância à faltas pode ser alcançada através da replicação e autoconfiguração do sistema na detecção de uma intrusão. Sendo assim, o sistema de detecção de intrusão deve ser eficiente para reconhecer o maior número possível de invasões. Com a replicação os serviços do sistema podem passar a operar em outro ponto da topologia sem a interrupção de suas funcionalidades. Além disso, se os dados do sistema forem corrompidos o sistema deve recuperar automaticamente os últimos dados válidos através de uma base de dados reserva.

O presente trabalho busca, através das idéias citadas, amenizar estragos causados por invasões utilizando uma base de dados auxiliar para recuperação de dados danificados e a utilização de um controle de ações. Através desse controle é possível identificar certos tipos de ações ou seqüências de ações realizadas por invasores e reconfigurar automaticamente o sistema para que ele opere normalmente mesmo diante de uma intrusão danosa.

Este artigo está organizado da seguinte forma. A seção 2 descreve as pesquisas que se relacionam com o tema discutido nesse artigo científico. A seção 3 apresenta os conceitos associados aos assuntos tolerância à faltas e sobrevivência computacional, ambos necessários para a compreensão desse documento. A especificação das técnicas para amenizar os estragos de uma invasão e a descrição do protótipo construído são os assuntos, respectivamente, das seções 4 e 5. O artigo encerra na seção 6, a qual reúne os principais resultados da pesquisa e mostra os possíveis complementos sobre ela, a cargo de trabalhos futuros.

## II. TRABALHOS RELACIONADOS

A área de sobrevivência computacional é nova e tem despertado muito interesse na comunidade científica internacional. Vários grupos de estudos em diversas partes do

2 Computer Survivability.

mundo estão pesquisando sobre o tema como, por exemplo, o grupo CERT<sup>3</sup> que é um dos pioneiros na pesquisa do assunto e possui diversos artigos já publicados sobre o tema. Vários grupos de estudos pesquisam sobre esse tema como, por exemplo, a equipe do CERT, a qual é pioneira em pesquisas nesse assunto, tendo já publicado vários artigos a respeito.

Um desses trabalhos é [2], o qual sugere os requerimentos e os passos necessários para o desenvolvimento de um sistema sobrevivente. Outro é [3], um estudo de caso sobre redes sobreviventes. Diferentes pesquisadores também têm alcançado resultados significativos no estudo de sistemas sobreviventes como [5], que define os requerimentos de um sistema sobrevivente através de um Processo de decisão de *Markov* restringido<sup>4</sup>.

O uso de tolerância a faltas para alcançar a disponibilidade de um sistema já foi tratado em [6], trabalho que expõe idéias para a construção de um sistema distribuído com uso de técnicas de tolerância a faltas para garantir a disponibilidade do sistema mesmo na ocorrência de uma falha ou acidente. O trabalho referenciado se diferencia deste artigo pois não trata conclusivamente os aspectos de segurança envolvidos no problema. Pode-se também citar o trabalho de [1], o qual enfatiza a segurança em aplicações tolerantes a faltas.

Além disso, este trabalho afasta-se dos demais por concentrar-se na utilização de ações de auditoria para o alcance da sobrevivência computacional. O trabalho de [15] é o único pesquisado que aponta a importância de sistemas de detecção de intrusão para a sobrevivência de sistemas computacionais, embora este não especifique como o uso de sistemas de detecção pode auxiliar a recuperação de sistemas.

### III. ASPECTOS TEÓRICOS

Os aspectos teóricos envolvidos no artigo são divididos em três partes: segurança computacional, sistemas sobreviventes e sistemas tolerantes a faltas. Em cada subseção pode-se encontrar referências das associações entre os tópicos. Além disso, os aspectos teóricos abordam especificamente os assuntos referentes ao artigo, deixando pontos não-relevantes ao trabalho de lado.

#### A. Segurança Computacional

A segurança de sistemas está entre as áreas da computação com maior proeminência, devida especialmente à importância dela no cotidiano das pessoas e negócios empresariais [11]. A segurança se importa com a proteção de ativos digitais armazenados em computadores e redes de processamento de dados [13].

Pode-se dizer que um computador é seguro se ele está livre de vulnerabilidades e preocupações a respeito de ameaças. A segurança computacional, neste sentido, é a disciplina que nos ajuda a ficar despreocupados com os computadores, sendo assim possível reconhecer a palavra “seguro” como um atributo de um sistema ou objeto [8].

Para indicar um sistema como sendo seguro ele deve manter três propriedades básicas: confidencialidade, integridade e disponibilidade [8]. A primeira é a propriedade de certas informações que não podem ser disponibilizadas ou divulgadas sem autorização. Analogamente, integridade pode ser descrita como a condição na qual recursos são protegidos contra modificações sem prévia autorização. Por fim, disponibilidade é responsável por garantir que a informação estará acessível para usuários legítimos quando estes requererem.

Para assegurar que sistemas implantem as propriedades citadas anteriormente e sejam ditos seguros, existe a necessidade de adoção de mecanismos de segurança. A tabela 1, conforme escrito por [10], mostra os mais notáveis mecanismos de proteção.

TABELA I  
DEFINIÇÃO DOS MECANISMOS DE SEGURANÇA

Mecanismo	Definição
Criptografia	Transforma dados em algo ininteligível para o inimigo, isto é, esconde o seu conteúdo semântico.
Autenticação	Utilizada para verificar se uma entidade é quem afirma ser.
Autorização	Processo de determinar que tipos de atividades são permitidos
Auditoria	Exame dos registros e das atividades do sistema para avaliar sobre sua confiabilidade

A política de segurança relaciona entidades, propriedades e mecanismos de segurança, além de definir o escopo e as características de cada serviço que se pretende proteger [12]. Ela determina regras que, quando seguidas corretamente, diminuem os riscos de incidentes de segurança à organização. Não existe como garantir a totalidade da segurança de um sistema; o que se busca é alcançar patamares admissíveis para o problema. Conforme [8], um sistema ou organização sem uma política de segurança pode ser comparado com uma sociedade sem leis.

#### B. Sistemas Sobreviventes

Segundo [9], a sobrevivência de sistemas é a capacidade que um sistema tem de manter seus serviços essenciais sob a ocorrência de um ataque, uma falha ou um acidente. A definição pode ser incorporada à questão da segurança computacional da seguinte maneira: “Sobrevivência de sistemas é a capacidade de um sistema manter suas propriedades de segurança, ou seja, confidencialidade, integridade e disponibilidade durante a ocorrência de um ataque, uma falha ou um acidente”. Outras propriedades importantes para os sistemas sobreviventes apontadas por [7] são: autonomia, eficiência, transparência, escalabilidade, manutenção, tempo de resposta e medição.

Os termos *ataque*, *falha* e *acidente* englobam todos os eventos potencialmente danosos a um sistema. O escopo deste documento destaca somente ataques, embora em muitos casos mais de um termo pode ser aplicado a um mesmo evento. Um ponto importante com relação aos eventos é como os serviços são afetados por tais eventos. Uma análise da sobrevivência

<sup>3</sup> CERT Coordination Center – <http://www.cert.org/ressearch/>

<sup>4</sup> Constrained Markov Decision Process.

deve ser feita no sistema e ser totalmente dependente dos serviços oferecidos [5].

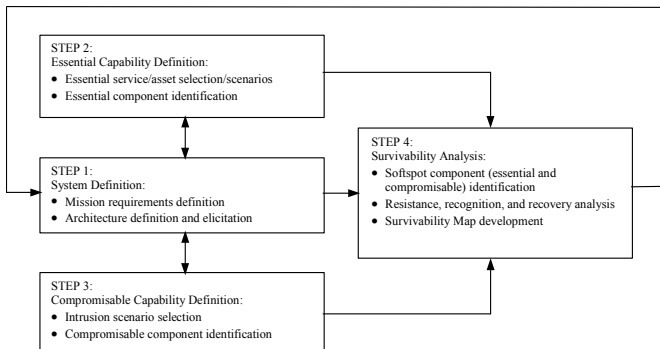


Figura 1. Método de Análise de um Sistema Sobrevivente [3].

Os serviços são divididos em essenciais e aqueles não-essenciais. Os serviços essenciais devem ser protegidos e recuperados em sistemas sobreviventes. Já os serviços não-essenciais podem sofrer danos críticos temporários sem prejudicar o funcionamento do sistema. O trabalho de [3] define quatro passos para identificar, proteger e recuperar os serviços essenciais de um sistema, como ilustrado na figura 1.

O primeiro passo consiste na definição do sistema e fazem parte dessa etapa as definições de requerimentos e da arquitetura do sistema. O segundo passo consiste na definição dos serviços essenciais que compõem o sistema e a identificação dos componentes essenciais. No terceiro passo é feita a definição da capacidade de comprometimento do sistema, além da criação de um cenário de intrusão e identificação de componentes passíveis de comprometimento. Finalmente, no quarto passo são feitas as análises de reconhecimento, recuperação e sobrevivência do sistema. É importante salientar que o processo é contínuo, ou seja, pode-se passar várias vezes pelas mesmas etapas.

### C. Tolerância a Falhas

A disciplina de tolerância a falhas tem duas preocupações básicas: a detecção de erros e a recuperação de erros [6]. O conceito de detecção de erros tem relação com o conceito de sistemas de detecção a intrusão, que é um tradicional mecanismo de segurança, embora que, no âmbito de tolerância a falhas, os erros sejam mais relacionados a falhas e acidentes.

O ponto-chave de tolerância a falhas utilizado neste trabalho é a recuperação de erros. Esta pode ser alcançada basicamente com um conceito: replicação. A replicação de sistemas sempre traz custos extras, por isso pode ser rejeitada por algumas organizações, mas devido à importância de alguns sistemas, o uso de desse artifício é essencial.

Uma técnica muito utilizada e conhecida de tolerância a falhas é o uso de registros de recuperação de dados. A idéia é o sistema manter duas bases de dados iguais, uma com os dados efetivamente usados e outra com uma cópia fidedigna das informações. Se a base principal falhar, a base de recuperação pode reparar o sistema provendo novamente a sua integridade.

## IV. ESPECIFICAÇÃO DE TÉCNICAS DE SOBREVIVÊNCIA UTILIZANDO REPLICAÇÃO E AUDITORIA DE AÇÕES

Para especificar um sistema sobrevivente é preciso primeiramente descrever o ambiente no qual o sistema será implantado. Em seguida, deve-se enumerar os serviços a serem providos por tal sistema e classificá-los em serviços essenciais ou não-essenciais. O foco da sobrevivência deve ser implantado sob os serviços essenciais. Por fim, uma análise do comportamento do sistema com e sem o uso das técnicas de sobrevivência é recomendada, para garantir a eficácia e validade desses processos.

A especificação de um sistema sobrevivente segue como a figura 1. Mas, para a utilização de replicação e auditoria de ações o modelo pode ser modificado conforme a figura 2, a qual diferencia-se da figura 1 em alguns pontos. Inicialmente, inclui-se no primeiro passo a especificação dos serviços essenciais, já que isso pode ser feito na própria definição do sistema. Na fase dois, a especificação de ações maliciosas deve ser feita para o uso na fase 3. Pode-se perceber que a replicação é definida na fase 1 e só é utilizada na fase 4, já que a replicação não é necessária para a detecção de uma intrusão, mas somente para a recuperação do sistema.

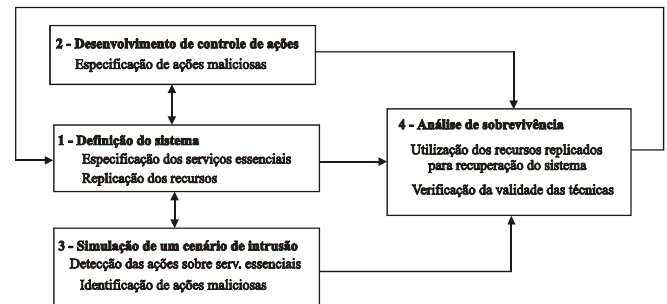


Figura 2. Especificação de um sistema sobrevivente utilizando replicação e auditoria de ações.

Os recursos referenciados na figura 2 podem ser de qualquer tipo, desde que possam ser replicados. Por exemplo, um servidor de páginas *web* pode recorrer a uma máquina substituta devidamente configurada em caso de funcionamento anormal. Por outro lado, se um dispositivo é caro o suficiente para invalidar sua replicação, esse modelo não deve ser aplicado.

## V. VALIDAÇÃO

No processo de validação foi adotado um serviço *web* que utiliza uma base de dados para armazenamento de informações de uso geral. A arquitetura é exposta na figura 3. Nela, pode-se ver pontos importantes. O primeiro é o banco de dados *Data*, que possui todos os dados relevantes aos usuários do sistema, incluindo suas próprias informações. O segundo banco é o *Backup*, que funciona como um banco de dados de replicação, porém, no modo tradicional os dados de replicação desse tipo de serviço só são requisitados manualmente, em vista de um problema nos dados originais da base principal. A Aplicação Servidor se comunica através da internet com outras Aplicações Clientes e, além disso, tem o controle sobre

as tabelas. Um cliente executa visualizações e atualizações no banco de dados Data através de uma Aplicação Cliente. Um invasor pode, com o uso de uma Aplicação Cliente através de falhas na aplicação ou descuido de usuários legítimos, ou até mesmo sem o uso dessa aplicação aproveitando-se de brechas no protocolo de comunicação, conseguir acesso ao banco de dados e danificar o sistema.

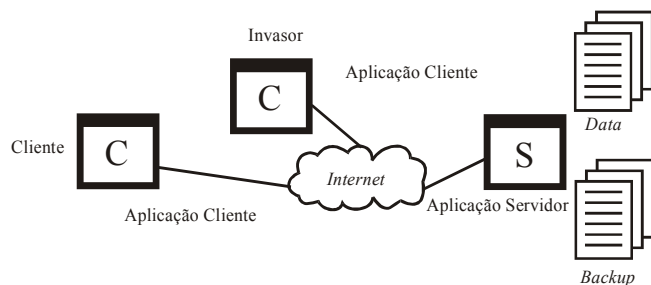


Figura 3. Exemplo de sistema vulnerável a ataques.

Nos sistemas convencionais, o uso do banco de dados Backup é reativo e só é acionado depois do comprometimento do sistema, seja por um ataque ou outro problema. Em um sistema sobrevivente, esse uso é pró-ativo, ou seja, o sistema deve ser capaz de detectar ataques e automaticamente acionar o banco de dados Backup e realizar os reparos necessários à base de dados Data. Além disso, o sistema sobrevivente utiliza-se de um controle de ações de usuários para análise, a fim de encontrar possíveis ataques.

Os passos realizados no aperfeiçoamento da Aplicação Servidor para um sistema sobrevivente seguem os mesmos da figura 2, apresentada na seção 4. No primeiro passo, o sistema é definido e sua arquitetura especificada. Portanto, deve-se identificar os serviços essenciais do sistema. O ponto mais importante para o funcionamento da aplicação é a corretude, ou seja, a integridade e a disponibilidade do banco de dados Data, como mostra a tabela 2. Nesse banco se concentram todas as informações necessárias para o bom uso do sistema. Portanto, é nele que serão concentrados os esforços para alcance da sobrevivência do sistema. Na tabela 3, ainda vê-se que o serviço de Backup é essencial com relação às propriedades de segurança. Porém, como a base somente é acessível diretamente pelas ações do sistema, ela é segura contra atacantes externos.

TABELA 2  
SERVIÇOS OFERECIDOS RELACIONADOS COM AS PROPRIEDADES DE SEGURANÇA

Serviços	Confidencialidade	Integridade	Disponibilidade
Banco de Dados Data	Não-Essencial	Essencial	Essencial
Banco de Dados Backup	Não-Essencial	Essencial	Essencial

No caso do exemplo, a replicação consiste no uso da base de dados Backup, que pode ser removida para diminuir gastos, embora isso aumente consideravelmente a ocorrência de danos

irreparáveis no sistema, já que além de mais vulnerável a ataques, o sistema torna-se vulnerável também a acidentes ou falhas. Além do uso da base Backup, a Aplicação Servidor utiliza uma auditoria de ações, ou seja, todas as ações tomadas por uma Aplicação Cliente são temporariamente armazenadas no servidor para futuras análises em busca de possíveis ataques.

No passo 2 é feita uma especificação de ações maliciosas, além do desenvolvimento do controle de auditoria. No modelo usado na validação foi tomado como exemplo o caso de um atacante obter uma senha de administrador do sistema. Embora esse fato possa ser extremamente difícil de ocorrer principalmente em sistemas seguros, sistemas tradicionais normalmente não se preocupam com o tipo de ações cometidas por um agente que se identificou como administrador do sistema, mesmo que exista a possibilidade desse ato ser uma invasão.

Deve-se criar um cenário de intrusão e a detecção pelo sistema sobrevivente à ações maliciosas no passo 3. O cenário de intrusão é criado através de possíveis ataques e ações danosas ao sistema, como sugere [4]. Como foram mencionadas para esse exemplo, as ações maliciosas do cenário de intrusão são provenientes de um atacante com senha de administrador. Este, para tentar danificar o sistema pode cometer uma série de abusos contra a base de dados. Foram definidos para a validação 8 tipos de ações que podem causar dano ao sistema. A tabela 3 mostra essas atitudes.

TABELA 3  
AÇÕES DO CENÁRIO DE INTRUSÃO

	Ação
1	Apaga todos os dados de uma tabela
2	Apaga todo o banco de dados
3	Remove relacionamento entre registros de tabelas
4	Altera senhas de usuários
5	Altera dados de usuários
6	Apaga alguns registros de uma ou mais tabelas
7	Remove um ou mais ligações entre tabelas
8	Altera senha de acesso ao banco

São desenvolvidas no passo 4 técnicas para amenizar ou restaurar danos definidos no cenário de intrusão. Depois disso, uma análise do comportamento do sistema com e sem as técnicas de sobrevivência desenvolvidas é feita em cima desse cenário para avaliar os custos e benefícios finais das técnicas de sobrevivência. Os resultados dessa análise estão demonstrados na tabela 4. É importante salientar que o processo é iterativo, ou seja, caso os resultados finais não sejam satisfatórios, pode-se retroceder alguns passos a fim de aprimorar o sistema sobrevivente. Além disso, o processo pode ser repetido quando surgirem novos métodos de intrusão e detecção de intrusão.

A tabela 4, como exposto anteriormente, mostra os resultados alcançados através do experimento realizado neste trabalho. Percebe-se que no sistema construído somente há um

tipo de ataque, o qual o atacante, por algum motivo qualquer, obtém a senha de um usuário administrador legítimo. Caso um atacante acesse o sistema tradicional com privilégios de administrador, ele é capaz de realizar qualquer ação que em nenhum momento será questionada sua legitimidade. Além disso, o invasor ataca a propriedade de confidencialidade do sistema, porém a confidencialidade do sistema não afeta os serviços essenciais, como se sugerido na tabela 2.

TABELA 4  
RESULTADOS DO EXPERIMENTO DE SOBREVIVÊNCIA

	<b>Ação</b>	<b>Danos sem método de sobrevivência</b>	<b>Danos com método de sobrevivência</b>
1	Apaga todos os dados de uma tabela	Tabela danificada	Tabela recuperada
2	Apaga todo o banco de dados	Banco de dados comprometido	Banco de dados recuperado
3	Remove relacionamento entre registros de tabelas	Banco de dados comprometido	Banco de dados recuperado
4	Altera senhas de usuários	Senhas danificadas	Senhas danificadas
5	Altera dados de usuários	Dados danificados	Dados danificados
6	Apaga alguns registros de uma ou mais tabelas	Banco de dados comprometido	Banco de dados Comprometido
7	Remove um ou mais ligações entre tabelas	Integridade dos dados comprometida	Integridade restaurada
8	Altera senha de acesso ao banco	Banco de dados comprometido	Senha restaurada

O sistema desenvolvido nessa pesquisa, diferentemente dos demais sistemas convencionais, realiza uma auditoria e replicação para verificar se o usuário é realmente quem diz ser. Se esse usuário realizar alguma ação danosa ao sistema, como, por exemplo, apagar as tabelas do banco de dados, o sistema reconhece como uma ação danosa e repara o ambiente. Entretanto o sistema desenvolvido não é capaz de envolver todo tipo de ação, já que, em alguns casos, as ações cometidas por intrusos confundem-se com ações legítimas. Mas, pode-se solucionar essas ações não-resolvidas voltando alguns passos no desenvolvimento do sistema e aumentando, por exemplo, a abrangência do controle de auditoria para que detecte essas ações. Como o processo é contínuo, essa tática é possível no desenvolvimento da aplicação.

#### VI. DIFICULDADES DE IMPLANTAÇÃO

O sistema, apesar de contribuir com a segurança do ponto

de vista da recuperação de um sistema atacado, possui algumas dificuldades de implementação. Vê-se na tabela 4 (seção 5), que nem todas as ações são possíveis de serem detectadas. Isto se deve à possibilidade da ação ser realmente legítima. Segundo [11], ações de invasores podem ser confundidas com ações legítimas por causa da liberdade dada aos usuários. É possível aumentar a efetividade da detecção se fossem diminuídas as ações permitidas pelos sistemas a usuários legítimos, mas essa atitude gera problemas operacionais ao sistema.

Outro problema já citado é a real necessidade da replicação de recursos. Nem sempre é válido realizar a replicação dos recursos, já que replicação de recursos necessariamente é acompanhada de gastos extras, sejam estes financeiros ou computacionais, como custos de processamento gerado pela funcionalidade de manutenção dos dados replicados.

Outra dificuldade encontrada no desenvolvimento desse tipo de sistema é a proteção dos dados replicados, já que estes não podem ser passíveis de ataques. No caso do exemplo da validação, esse problema foi sanado, pois o banco de dados replicado só é acessível internamente pelo sistema, não podendo ser acessado por um atacante externo, mesmo que este possua as credenciais de administrador.

#### VII. CONCLUSÃO

Nesta sessão são apresentados os resultados obtidos a partir dos objetivos iniciais do trabalho, mais especificamente a aplicação de técnicas de replicação e operações de auditoria para o aprimoramento da sobrevivência de um sistema.

Além disso, o artigo apresenta o enfoque dado à área de segurança dentro da disciplina de sobrevivência computacional. Sistemas sobreviventes abrangem a segurança computacional sobre um ponto de vista diferente dos demais. A sobrevivência computacional se preocupa em recuperar sistemas, ao invés de simplesmente bloquear ataques como trata a segurança tradicional [14]. Sendo assim, esse enfoque gera novos rumos ao avanço da segurança e esse trabalho é uma contribuição nesse sentido.

Este trabalho especifica como deve-se utilizar um controle de auditoria e replicação de recursos para obtenção da sobrevivência computacional e contribui para que desenvolvedores de sistemas possam reforçar os mecanismos de segurança existentes em seus projetos. O controle de auditoria traz para o sistema uma maior segurança que o controle de acesso sozinho, detectando possíveis ações maliciosas realizadas por usuários supostamente legítimos. Já a replicação de recursos protege o sistema contra acidentes, além de prover uma garantia de recuperação do sistema em caso de um ataque detectado.

O experimento constituído validou as técnicas de sobrevivência que utilizam replicação e operações de auditoria. O cenário de intrusão é fundamental, já que é essencial para simular o comportamento do sistema face a diversas ações danosas. Percebe-se, baseado nos resultados da Tabela 4 (seção 5) que estes são positivos, já que solucionam alguns problemas de segurança do sistema. Apesar disso,

sistemas que utilizam replicação de dados requerem custos maiores e nem sempre é vantajoso um gasto extra com segurança.

Diversos trabalhos futuros podem ser propostos a partir deste artigo. Outras técnicas podem ser encontradas a fim de otimizar a sobrevivência de sistemas, com ou sem o uso de replicação ou operações de auditoria. Um ponto importante, e promissor citado em [2] é a aplicação de técnicas de inteligência artificial na detecção de intrusões e na evolução do sistema sobrevivente.

#### VIII. REFERÊNCIAS

- [1] Chothia, T. and Duggan, D., "An architecture for Secure Fault-Tolerant Global Applications," *In: Workshop on Principles of Dependable Systems (PoDSy'2003)*, June 24, 2003, San Francisco, CA, USA.
- [2] Ellinson, R. J., Fisher, D. A., Linger, R., Lipson, H. F., Longstaff, T. and Mead, N., "An Approach to Survivable Systems," <http://www.cert.org/ressearch/>, 2000.
- [3] Ellinson, R. J., Linger, R., Longstaff, T. and Mead, N., "Survivable Network System Analysis," *IEEE Software*, 16(4) p.70–77, 2002.
- [4] Grosh, A. and Voas, J. "Inoculating Software for Survivability," *Communications of the ACM*, 42(7) p.38-44, 1999.
- [5] Jha, S. and Wing, J. M., "Survivability Analysis of Networked Systems," *In: Proceedings of the 23<sup>rd</sup> International Conference on Software Engineering*, 2001, Toronto, Canada, p.307–317.
- [6] Knight, J. C. and Elder, M. C., "Fault-Tolerant Distributed Information Systems," *In: 12<sup>th</sup> International Symposium on Software Reliability Engineering (ISSRE'01)*, November 27–30, 2001, Hong Kong, China.
- [7] Krings, A. W. and Oman, P., "Secure and Survivable Software Systems," *In: 36<sup>th</sup> Hawaiian International Conference on System Sciences (HICSS'36)*, January 6–9, 2003, Big Island, Hawaii, USA.
- [8] Landwehr, C. E., "Computer security," *International Journal of Information Security*, 1(1) p.3–13, 2001.
- [9] Levitin, G. and Lisnianski, A., "Optimizing Survivability of Vulnerable Series – Parallel Multi-State Systems," *Reliability Engineering & System Safety*, 79(2003) p.319–331, 2002.
- [10] Pernul, G., "Information Systems Security: Scope, State-of-the-art, and Evaluation of Techniques," *International Journal of Information Management*, 15(3) p.239–255, 1995.
- [11] Stallings, W., "Cryptography and Network Security," page 44. Prentice Hall, New Jersey, United States, 3th edition, 2003.
- [12] Uchoa, J. Q., "Políticas de Segurança e Políticas de Uso," *In: Simpósio de Segurança em Informática (SSI)*, São José dos Campos, Brasil, 2001.
- [13] Venter, H. and Eloff, J., "A Taxonomy for Information Security Technologies," *Computers and Security*, 22(4) p.299–307, 2003.
- [14] Yue, C., "Cyber Security," *Technology in Society*, 25(4) p.565–569, 2003.
- [15] Zang, Y., Vin, H., Alvizi, L., Lee, W. and Dao, S. K., "Heterogeneous Networking: A New Survivability Paradigm," *In: Network Security Paradigms Workshop (NSPW'01)*, September 10-13<sup>th</sup>, 2002, Cloudcroft, New Mexico, USA.

#### IX. BIOGRAFIAS

**Felipe Rolim Pellissari** nasceu em 1981, é bacharel em ciência da computação pela Universidade Estadual de Londrina (2003) e atualmente é membro do programa de Pós-Graduação da Universidade Federal de Santa Catarina (PPGCC-UFSC). Suas áreas de interesse são segurança em sistemas distribuídos, ambientes colaborativos Peer-to-Peer (P2P) e acordos de níveis de serviço (SLA).

**Rafael Righi** nasceu em 1979, é bacharel em ciência da computação pela Universidade Federal de Santa Maria (2003) e atualmente é membro do programa de Pós-Graduação da Universidade Federal de Santa Catarina (PPGCC-UFSC). Rafael também é bolsista do PoP-SC (Ponto de Presença da RNP em SC). Suas áreas de interesse são segurança em sistemas distribuídos, ambientes colaborativos Peer-to-Peer e acordos de níveis de serviço (SLA).