Uma Arquitetura Simplificada para Sistemas Tolerantes a Intrusão

Giani Petri
PPGI — UFSM
gianipetri@gmail.com

Raul Ceretta Nunes
PPGI — UFSM
ceretta@inf.ufsm.br

***Resumo*—Juntamente com o avanço da utilização da Internet o número de invasões em sistemas informatizados também tem crescido. Uma das formas para mitigar os efeitos dessas invasões é com a implantação de uma arquitetura que consiga tolerar as falhas ocasionadas por um intruso. As arquiteturas tolerantes a intrusão, existentes atualmente, são formadas por vários componentes de *hardware* replicados e possuem um alto custo, o que muitas vezes, inviabiliza a implantação. Este trabalho apresenta uma revisão das arquiteturas existentes e propõe uma arquitetura simplificada, com base nos componentes similares identificados nas arquiteturas estudadas. Empresas podem adotar esta arquitetura simplificada e com isso reduzir custos de implantação e evitar desperdícios de recursos replicados, comparado com a complexidade e infraestrutura das arquiteturas existentes.**

I. INTRODUÇÃO

O aumento gradativo na utilização de serviços informatizados juntamente com a expansão do uso da Internet, faz com que as vulnerabilidades existentes nos sistemas sejam mais facilmente exploradas. Partindo de um pressuposto que nenhum sistema é totalmente invulnerável, os atacantes aproveitam esses espaços para invadir e, muitas vezes, fraudar informações estratégicas às organizações.

Para reduzir os efeitos causados por uma invasão, várias pesquisas estão sendo realizadas na área de Tolerância a Intrusão. Alguns trabalhos [1][2][3] mostram que é possível criar um sistema que tolere uma intrusão e continue funcionando normalmente, de forma transparente ao usuário, protegendo assim a integridade das informações e do sistema.

As arquiteturas tolerantes a intrusão, existentes atualmente, são formadas por um grande número de componentes e barreiras a fim de impedir que um atacante comprometa todo o sistema. As arquiteturas [1][4][5] demonstram que a utilização de redundância e diversidade garante maior segurança contra invasores. Por outro lado, a utilização de uma arquitetura com inúmeros componentes de *hardware* é inviabilizada, muitas vezes, pelo seu alto custo de implantação e manutenção.

Este trabalho apresenta a proposta de uma arquitetura simplificada para um sistema tolerante a intrusão, a arquitetura é composta por barreiras de proteção formadas pelos componentes identificados como imprescindíveis para tolerar uma intrusão. Além disso, mostra as principais características e aspectos, juntamente com os componentes arquiteturais apresentados em [1][4][5], objetivando identificar as similaridades entre as arquiteturas analisadas e destacar os componentes essenciais para tolerar uma intrusão.

O restante do trabalho está organizado da seguinte forma. A seção II identifica as similaridades entre as arquiteturas existentes na literatura. A seção III apresenta a proposta de uma arquitetura simplificada, baseada nas propriedades de um sistema tolerante a intrusão e na análise das soluções existentes. Por fim, a seção IV apresenta as conclusões do trabalho.

II. ARQUITETURAS EXISTENTES, CARACTERÍSTICAS EM COMUM

Esta seção descreve as propriedades e características das arquiteturas tolerantes a intrusão encontradas na literatura, destacando aspectos de seus componentes, seu funcionamento interno e suas características em comum.

A. *Arquiteturas tolerantes a intrusão*

1) Encaminhamento centralizado – Gestão centralizada [1]: Essa arquitetura é uma das variações da arquitetura geral apresentada em [1]. Seu objetivo é a utilização de um baixo número de componentes confiáveis (*trusted*) para garantir a sua contínua operação. As requisições realizadas pelos clientes são filtradas por um *firewall* e posteriormente enviadas para um *gateway*, determinado como confiável, que então, encaminha os pedidos para um dos servidores ativos. Cada servidor possui internamente um CMDaemon (*configuration management daemon*) que é responsável por detectar intrusões em cada servidor. A arquitetura proposta possui também um gestor de configuração, que é definido como confiável.

2) Encaminhamento por difusão – Gestão centralizada [1]: Essa arquitetura também é uma das variações arquiteturais em [1]. Ela é composta por praticamente o mesmo fluxo da arquitetura por Encaminhamento centralizado - Gestão centralizada, porém, o que as difere, é a inexistência de um *gateway* e de um *firewall* à entrada das requisições. Então, os pedidos dos clientes chegam direto aos servidores que estes, com a utilização de seu *firewall* individual, proposto nesta arquitetura, decide quais pedidos processarem, de acordo com uma política de balanceamento de carga. Em caso de detecção de alguma intrusão os CMDaemon realizam a mesma função de notificar o gestor de configuração para suas devidas correções.

3) Encaminhamento centralizado – Gestão descentralizada [1]: A última variação da arquitetura geral citada em [1] difere das anteriores pela não utilização de um gestor de configuração. Esta arquitetura procura combinar o bom desempenho das arquiteturas

citadas em nas seções 1 e 2 com a redundância de todos os componentes. Cada requisição feita pelo cliente é processada em apenas um servidor. A função dos CMDaemons nessa arquitetura é a identificação de intrusão e também a decisão em comum acordo sobre as reconfigurações necessárias para a recuperação de um servidor comprometido.

4) Arquitetura com firewall de privacidade [1]: Na arquitetura com *firewall* de privacidade, as requisições dos clientes são feitas para os servidores de acordo e estes distribuem para os servidores de execução processar o pedido. Entre esses grupos de servidores, há um *firewall* que é responsável por filtrar as informações que comprometam algum servidor. No caso de ocorrer alguma intrusão em algum servidor de execução, o *firewall* é responsável por filtrar essas informações, garantindo a confiabilidade no serviço. Só respostas bem formadas são liberadas pelo *firewall*.

5) SITAR (Scalable Intrusion-Tolerant Architecture for Distributed Services) [4]: SITAR é uma arquitetura criada a partir do programa OASIS (*Organically Assured and Survivable Information Systems*) financiado pelo DARPA (*Defense Advanced Research Projects Agency*) [6]. A arquitetura SITAR engloba fundamentos básicos de tolerância a falhas, dentre eles: redundância, votação e reconfiguração adaptativa, juntamente com testes de aceitação. O principal objetivo da arquitetura SITAR é a capacidade de aumentar a tolerância a falhas, mais especificamente em servidores COTS (*commercial off-the-shelf*) distribuídos. A arquitetura SITAR objetiva ainda, a mitigação de efeitos de ataques conhecidos e desconhecidos a fim de manter a correta operação do sistema, mesmo na presença de um intruso ou algum código malicioso. SITAR é composta por um conjunto de cinco componentes que estendem a capacidade de tolerância a intrusões em servidores COTS. Os componentes de processamento são: servidor *proxy*, monitor de votação, monitor de aceitação e os componentes de monitoramento são: módulo de controle de auditoria e módulo de reconfiguração adaptativa. O fluxo dos dados na arquitetura SITAR segue a seguinte sequência. Os servidores *proxy* representam o ponto de acesso externo, eles atuam quando há requisições de clientes, compartilham entre si endereços de IP (*Internet Protocol*) virtuais, facilitando a substituição de máquinas comprometidas. E encaminham as requisições dos clientes para os monitores de aceitação, que após validá-las as direciona para os servidores COTS processá-las. As respostas após serem processadas pelos servidores COTS são encaminhadas para o monitor de aceitação novamente, onde é verificada a validade das respostas certas com testes de aceitação e só então as direciona para o monitor de votação com a indicação do resultado da seleção. Por sua vez, os monitores de votação servem como representantes para os servidores COTS e decidem sobre uma resposta final. Essa decisão é tomada dependendo do nível atual da ameaça à segurança detectada. Então, a resposta final é devolvida aos servidores *proxy* onde esses entregam aos clientes. O componente de reconfiguração adaptativa recebe informações de *triggers* de intrusões de todos os outros componentes, avalia ameaças de intrusão, o custo do impacto e o desempenho e, se necessário, gera novas configurações para os componentes comprometidos. O módulo de controle de auditoria fornece meios para auditar o comportamento de todos componentes através de logs armazenados.

6) DPASA (Designing Protection and Adaptation into a Survivability Architecture) [5]: A arquitetura DPASA foi desenvolvida no projeto OASIS Dem/Val, projeto criado pelo DARPA após o término do projeto OASIS, objetivando demonstrar a tolerância a intrusão em um sistema real. Conforme [1] essa arquitetura é a mais complexa e interessante entre as arquiteturas tolerantes a intrusão. DPASA é utilizada no sistema da força aérea americana *Join Battlespace Infosphere* (JBI). Sua proposta é a criação de múltiplas camadas de proteção, aumentando assim o nível de proteção do sistema, exigindo mais força dos possíveis atacantes e a utilização de mais recursos para concretizar um ataque. A arquitetura DPASA, consiste em um núcleo central responsável por fornecer serviços de comunicação a um grupo de clientes. DPASA estende a noção de zonas desmilitarizadas (DMZ - *Demilitarized Zones*), dividindo o núcleo em três zonas: zona de embate, zona de operação e zona executiva. Além das três zonas, a arquitetura replica o núcleo em quatro quadrantes e as máquinas dentro desses quadrantes utilizam sistemas operacionais diferentes (SELinux, Solaris e Windows), garantindo a diversidade. A zona de embate atua como a primeira zona de impacto, amortecendo um potencial ataque. A zona de operação é onde se encontra a realização dos serviços. A zona executiva é responsável pela gestão e controle de serviços. A comunicação entre os elementos das zonas é feita através de protocolos, controlando rigorosamente os limites de progressão do ataque a partir de um componente comprometido. Cada *host* possui um conjunto de controladores locais em execução, para monitorar os *host* e realizar ações proativas e reativas contra atividades suspeitas no *host*. Um *proxy* de acesso, existente na zona de embate, trabalha como uma interface entre os protocolos e *middleware* de comunicação utilizados pelos clientes e o JBI. A zona executiva contém um componente que correlaciona os dados sobre intrusões obtidas através de sensores. Essa informação é filtrada para descartar alarmes falsos. Os alarmes importantes são enviados para outro componente na mesma zona, chamados de gestor de sistemas. Este, por sua vez, gera respostas às intrusões.

B. Componentes em comum entre as arquiteturas estudadas

Após o estudo das arquiteturas tolerantes a intrusão descritas na seção II é possível destacar alguns aspectos que são comuns entre as arquiteturas analisadas e que são essenciais para a implementação de uma arquitetura tolerante a intrusão.

Todas as arquiteturas descritas são compostas por vários componentes, sendo esses componentes redundantes. Essa estrutura contribui para elevar o nível de segurança contra invasores, evitando que o atacante ao conseguir acesso a um componente do sistema tenha o seu total controle.

Como ponto de entrada às requisições dos clientes três arquiteturas utilizam um *firewall*, seja ele único ou individual em cada servidor, esse *firewall* é responsável pelo filtro das informações. Em outra arquitetura estudada são usados os servidores de acordo para executarem essa função e as outras duas arquiteturas utilizam servidores definidos como *proxy* para atuarem como esse ponto de acesso externo do sistema. A utilização de mecanismos de segurança nos componentes que recebem as requisições externas auxilia na identificação de tentativas de ataques, podendo prevenir uma intrusão imediata.

A maioria das arquiteturas pesquisadas possui um módulo responsável pelas reconfigurações. São componentes que são notificados, em tempo de execução, de alguma anormalidade ou comprometimento em um servidor do sistema e então gera algumas diretivas de configuração para recuperar os serviços ou então retirar o servidor de operação. De modo em que nenhum sistema é totalmente invulnerável, invasões possivelmente podem ocorrer. Desta forma, ter um componente responsável por recuperar máquinas comprometidas garante maior confiabilidade na continuidade da prestação dos serviços do sistema.

No geral, as arquiteturas utilizam algum mecanismo para a identificação de intrusões, algumas arquiteturas possuem um IDS (*Intrusion Detection System*) responsável pela análise do tráfego da rede a fim de detectar intrusos, outras utilizam CMDaemons, cujo objetivo é o mesmo do IDS.

A partir da identificação das semelhanças entre as arquiteturas estudadas e os componentes essenciais para a implantação de uma arquitetura, pode-se destacar um padrão de nível arquitetural. Com base nisso a proposta de uma arquitetura simplificada é apresentada na seção III.

III. Uma Arquitetura Simplificada para Sistemas Tolerantes a Intrusão

A importância da segurança da informação é conhecida pelas organizações, porém o alto custo e a complexidade para manter uma proteção muitas vezes inviabiliza a implantação de mecanismos de segurança.

Tendo em vista esse cenário organizacional, esta seção apresentada a proposta de uma arquitetura simplificada para sistemas que necessitam uma maior segurança de uma forma tolerante a intrusão. A arquitetura proposta é baseada nas características de um sistema tolerante a intrusão, nas propriedades e similaridades entre as arquiteturas estudadas e nos componentes identificados como essenciais para a segurança.

A abordagem de características de um sistema tolerante a intrusão vai ao encontro dos conceitos e princípios de tolerância a falhas, já que uma intrusão pode ser considerada como uma falha bizantina [1]. Independente das propriedades da arquitetura tolerante a intrusão adotada, ela deve sempre atingir o seu principal objetivo que é garantir a integridade, disponibilidade e confidencialidade dos serviços, mesmo que alguns servidores sejam comprometidos e controlados com sucesso por um invasor [1].

Para o desenvolvimento de um sistema tolerante a intrusão é necessário a implementação de uma arquitetura com várias barreiras que previnam o avanço do atacante. A utilização prática das propriedades de tolerância a falhas devem ser unificadas com mecanismos de segurança a fim de criar um considerável número de barreiras de segurança, dificultando a invasão de um intruso. E caso ela venha a ocorrer, é necessário uma rápida recuperação e mitigação dos efeitos para então continuar a operar normalmente, sem percepção por parte dos usuários finais.

Conforme mostra a Figura 1, a arquitetura proposta é inserida entre a comunicação de um cliente e servidor, podendo ser adotada por qualquer empresa.

A arquitetura proposta (Figura 1) é formada por quatro componentes principais, são eles: *proxy* de acesso, testes de aceitação, área contaminada e um reconfigurador definido como confiável.

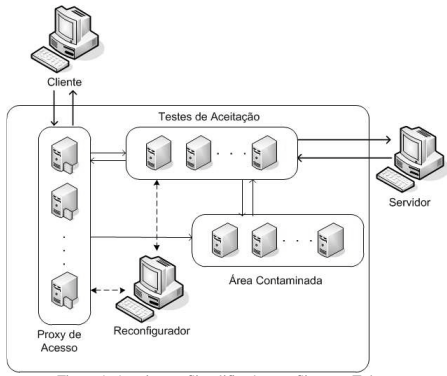

Figura 1. Arquitetura Simplificada para Sistemas Tolerantes a Intrusão.

Os *proxies* de acesso são responsáveis por receber as requisições oriundas dos clientes e então enviar para os servidores aplicarem testes de aceitação para validar a requisição. Os servidores *proxy* são equipados com sistemas de detecção de intrusão responsáveis pelo monitoramento do tráfego da rede e geração de alarmes em caso de encontrar alguma anomalia, caso encontre, a requisição do cliente é redirecionada para a área contaminada, para sondagem ao atacante e registro das atividades para posteriores medidas de prevenção. Os IDS são responsáveis também por identificar se há algum servidor comprometido, caso tenha, ele notifica o reconfigurador para então aplicar medidas de correção na máquina com falha. Da mesma forma que a arquitetura SITAR, propõe-se que os servidores *proxy* trabalham com compartilhamento de IPs virtuais onde, em caso de comprometimento de alguma máquina ela seja migrada facilmente. As requisições dos clientes são realizadas através de um IP virtual, os endereços dos servidores da arquitetura, bem como dos servidores de execução não são conhecidos publicamente.

Os servidores que aplicam testes de aceitação são responsáveis pela validação das requisições enviadas pelos *proxies* de acesso, caso as solicitações possuam alguma anormalidade, as mesmas são direcionadas para a área contaminada, caso a validação ocorra, a requisição é então encaminhada para o servidor executar o processamento. Após a execução da requisição, o servidor encaminha a solicitação já processada novamente para o processo de testes de aceitação, para validar e verificar se a resposta enviada está correta, mecanismo utilizado para garantia que não exista algum comportamento anormal que esteja gerando retornos errôneos. Muitas vezes, uma intrusão não possui uma requisição, ou seja, o atacante invadiu um determinado servidor. Uma notificação ao reconfigurador o faz avaliar o estado da máquina invadida e decidir se a mesma será recuperada ou excluída. Em caso de encontrar alguma falha nos testes de aceitação, o fluxo é então redirecionado para a área contaminada para sondagem de informações do atacante e o reconfigurador é notificado para que refaça a configuração da máquina que está gerando respostas erradas.

A área contaminada é utilizada para sondagem de informações dos atacantes, ou seja, quando identificado algum comportamento duvidoso no sistema, o mesmo é direcionado para essa área onde será monitorado e todas suas atividades são registradas para serem posteriormente analisadas. Caso o comportamento duvidoso seja confirmado como um ataque, as alterações realizadas pelo atacante são removidas. Caso as atividades realizadas na área contaminada são consideradas normais, as alterações são recuperadas e enviadas novamente para os servidores de testes de aceitação e então são removidas da área contaminada. O comportamento malicioso de um atacante nessa área pode ser usado para novas medidas preventivas para a arquitetura.

A arquitetura Encaminhamento centralizado – Gestão centralizada, apresentada na seção 1, possui um gestor de configuração que é responsável pela reconfiguração, limpeza e recuperação, dos servidores no qual ocorre alguma intrusão. Os gestores de configuração são informados sobre o estado dos componentes a partir de informações enviadas pelos CMDaemon existente em cada servidor.

A reconfiguração em um sistema tolerante a intrusão deve ser de forma automática. Onde imediatamente após a identificação de algum componente defeituoso, o mesmo seja reiniciado, reconfigurado ou então excluído, objetivando a continuidade no fornecimento dos serviços corretos.

Da mesma forma que na arquitetura Encaminhamento centralizado – Gestão centralizada, o reconfigurador proposto na arquitetura simplificada possui vital importância, sua responsabilidade é de gerar diretivas de reconfiguração objetivando recuperar componentes comprometidos por um ataque e, além disso, possui a tarefa de retirar um componente faltoso de funcionamento, evitando com isso uma parada do sistema. O reconfigurador é notificado em tempo de execução pelos servidores de testes de aceitação e pelos *proxies* de acesso, caso identificam alguma anormalidade, para então analisar se é necessária a reconfiguração de algum componente.

IV. Conclusões

A utilização de sistemas tolerantes a intrusão vem aumentando em decorrência da expansão de ataques as vulnerabilidades existentes nos sistemas informatizados. Para a garantia da continuidade do funcionamento de sistemas críticos é de vital importância a implantação de uma arquitetura que tolere uma intrusão.

Este trabalho apresentou uma revisão das arquiteturas existentes na literatura, destacando suas principais características e propriedades. Com a realização de uma análise dos conceitos e identificação das equivalências entre as arquiteturas estudadas, foi proposta uma arquitetura simplificada tolerante a intrusão composta pelos componentes destacados como essenciais para a garantia da segurança, com isso reduzindo o custo da implantação de uma arquitetura muito complexa, evitando o desperdício de recursos na replicação de vários componentes de *hardware*.

Como resultado do trabalho, tem-se que para a criação de uma arquitetura tolerante a intrusão é necessária: *i*) a implementação de barreiras de proteção, evitando que ao ocorrer um ataque o mesmo comprometa todo o sistema; *ii*) o uso de réplicas em cada componente arquitetural; *iii*) a utilização de mecanismos que identificam potenciais ataques e disparam alertas a outros componentes e o uso de ferramentas de reconfiguração, que recuperam componentes comprometidos garantindo o correto funcionamento do sistema.

Uma análise dos trabalhos apresentados considerando a proposta da arquitetura simplificada permitiu avaliar e realizar a prova de conceito, ao verificar a compatibilidade entre os componentes que compõem as arquiteturas.

Referências

[1] M. Correia, "Serviços distribuídos tolerantes a intrusões: resultados recentes e problemas abertos". In *V Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, 2005, pages 113–162. Sociedade Brasileira de Computação.

[2] R. R. Obelheiro, A. N. Bessani and L.C. Lung, "Analisando a viabilidade da implementação prática de sistemas tolerantes a intrusões". In *V Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*. Sociedade Brasileira de Computação, 2005.

[3] M. Atighetchi, P. Rubel, P. Pal, J. Chong and L. Sudin, "Case study: The intrusion tolerant JBI". BBN Technologies, 2005.

[4] F. Wang, F. Jou, F. Gong, C. Sargor, K. Goseva-Popstojanova and K. Trivedi, "Sitar: A scalable intrusion-tolerant architecture for distributed services". *Foundations of Intrusion Tolerant Systems*, 2003, page 359.

[5] J. Chong, P. Pal, M. Atigetchi, P. Rubel and F. Webber, "Survivability architecture of a mission critical system: the DPASA example". *In Computer Security Applications Conference, 21st Annual*, 2005, pages 10 pp. –504.

[6] S. Bryant and F. Wang, "Aspects of adaptive reconfiguration in a scalable intrusion tolerant system". *Complex*, 2003, 9:74–83.