UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

O PAPEL DAS DIRETRIZES DE QUALIDADE EM SURVIVABILITY – OS NOVOS LIMITES DA SEGURANÇA

CLAUDIANA PEREIRA BATISTA

Recife, Janeiro de 2009.

SUMÁRIO

Resumo

RESUMO

ESTE ARTIGO TEM COMO OBJETIVO PROCURAR MOSTRAR A IMPORTÂNCIA QUE OS CONHECIMENTOS EM QUALIDADE DOS SERVIÇOS ESTÃO INTRINSECAMENTE LIGADOS AO BOM DESEMPENHO DAS POLÍTICAS DE SURVIVABILITY. E O BOM DESEMPENHO DA OPERAÇÃO DE UM DADO SISTEMA É MEDIDO POR VÁRIAS VARIÁVEIS, DENTRE ELAS: DISPONIBILIDADE, QUALIDADE DE SERVIÇO E CUSTOS ASSOCIADOS, EXIGIDOS E APRESENTADOS PELOS USUÁRIOS DO SISTEMA, DURANTE A CONCEPÇÃO DO PROJETO. ENFIM, PODEMOS CONCLUIR QUE O DESEMPENHO, NO CONTEXTO, BUSCA OBTER UM ALTO NÍVEL DE QUALIDADE DE SERVIÇO E DE DISPONIBILIDADE A UM CUSTO OTIMIZADO.

CAPÍTULO 1 – GARANTIA DA INFORMAÇÃO E SURVIVABILITY

Na sociedade da informação, ao mesmo tempo em que as informações são consideradas o principal patrimônio de uma organização, estão também sob constante risco, como nunca estiveram antes. Com isso a segurança da informação tornou-se ponto crucial para sobrevivência das instituições [4].

Segurança é a proteção de informações, sistemas, recursos e serviços contra desastres, erros e manipulação não autorizada, de forma a reduzir a probabilidade e o impacto de incidentes de segurança. Estas expectativas correspondem aos objetivos da segurança, que são: confidencialidade, integridade dos dados, disponibilidade, consistência, isolamento ou uso legítimo, auditoria e confiabilidade. Analisando a evolução dos conceitos de segurança, o primeiro foi o de *Communications Security* (medidas e controles tomados para negar que pessoas não autorizadas obtenham informação resultante de comunicações e garantir a autenticidade das mesmas.), seguido pelo *Computer Security* (medidas e controles que assegurem a confidencialidade, integridade, e disponibilidade dos sistemas de informação incluindo hardware, software, firmware e a informação em processamento, guardada e comunicada. O DoD - United States Department of Defense, define a garantia da informação como as medidas que protegem e defendem a informação e sistemas de informação assegurando a sua disponibilidade, integridade, autenticação, confidencialidade e não-repúdio**[12]**. Isto inclui restauro dos sistemas de informação através da incorporação das capacidades de proteção, detecção e reação.

1.1 Introdução

A sobrevivência dos sistemas de TI é uma preocupação significativa, particularmente entre os provedores de infra-estrutura crítica. As análises de disponibilidade e confiabilidade supõem que as falhas são um tanto aleatórias e o trabalho do engenheiro é projetar um sistema que seja robusto face à falha aleatória. Há assim uma suposição implícita que a falha de sistema é largamente evitável.

A análise da sobrevivência faz implicitamente a suposição conservadora que a falha ocorrerá e que o resultado da falha poderia impactar negativamente um grande segmento da infra-estrutura de TI combinada. Não obstante a causa da falha, a análise de sobrevivência supõe que tais eventos podem e ocorrerão e o impacto para a infra-estrutura de TI e àqueles que dependem dela será significativo.

Survivability têm sido definido como a capacidade de um sistema para cumprir sua missão, em tempo oportuno, na presença de ataques, falhas, ou dos acidentes [9]. O termo sistema é usado no sentido mais amplo, incluindo redes e sistemas em larga escala.

O termo missão se refer a um conjunto de muitas exigências ou metas de alto-nível. Qualquer organização bem-sucedida de ter uma visão de seus objetivos se expressada implicitamente ou como uma indicação formal da missão. Os julgamentos se uma missão esteve ou não cumprida são feitos tipicamente no contexto das circunstâncias externas que podem afetar a realização daquela missão.

Os termos ataque, falha, e acidente são significados para incluir todos os eventos potencialmente prejudiciais. Os ataques são eventos potencialmente prejudiciais orquestrados por um adversário inteligente. Os ataques incluem intrusões, espionagem/investigação, e negações de serviço. As falhas e os acidentes são incluídos como parte do survivability. As falhas são eventos potencialmente prejudiciais causados por deficiências no sistema ou em um elemento externo de que o sistema depende. As falhas podem ser devido aos erros de projeto de software, à degradação do hardware, às falhas humanas, ou aos dados corrompidos. Os acidentes descrevem uma vasta ocorrência de eventos aleatórios e potencialmente prejudiciais, tais como desastres naturais. Os acidentes são eventos frequentemente gerados externamente (por exemplo, fora do sistema) e as falhas são eventos gerados internamente.

No que diz respeito à sobrevivência do sistema, uma distinção entre um ataque e uma falha ou acidente é menos importante do que o impacto do evento. Nossa abordagem concentra-se no efeito de um evento potencial prejudicial. Para que um sistema sobreviva, deve reagir (ou recuperar-se) a um efeito prejudicial muito antes que a causa subjacente seja identificada. De fato, a reação e a recuperação devem ser bem sucedidas mesmo se a causa nunca for determinada.

Finalmente, é importante reconhecer que é a realização da missão que deve sobreviver, não algum subsistema particular ou componente do sistema. A idéia central do survivability é a capacidade de um sistema para cumprir sua missão, mesmo se as partes significativas do sistema são danificadas ou destruídas [10].

A análise de sobrevivência é influenciada por vários princípios importantes:

- Retenção: os sistemas devem ser projetados para minimizar o impacto da missão retendo a falha geograficamente ou logicamente;
- Reconstituição: os projetistas de sistemas devem considerar o tempo, o esforço, e as habilidades exigidas para restaurar a missão crítica essencial da infra-estrutura de TI após o evento catastrófico;
- Diversidade: os sistemas que são baseados em múltiplas tecnologias, vendedores, posições ou modos de operação podem fornecer um grau de imunidade para ataques, especialmente aqueles alvejados em apenas um aspecto do sistema;
- Continuidade: é o negócio das funções de missão crítica que devem continuar no caso de um evento catastrófico, e não algum aspecto específico da infra-estrutura de TI.

Se as funções críticas são compostas de ambos infra-estrutura de TI (network) e componentes de função específicos da tecnologia (servers), então ambos devem ser projetados para serem sobreviventes (survivable).

1.2 Domínio do survivability: ilimitado

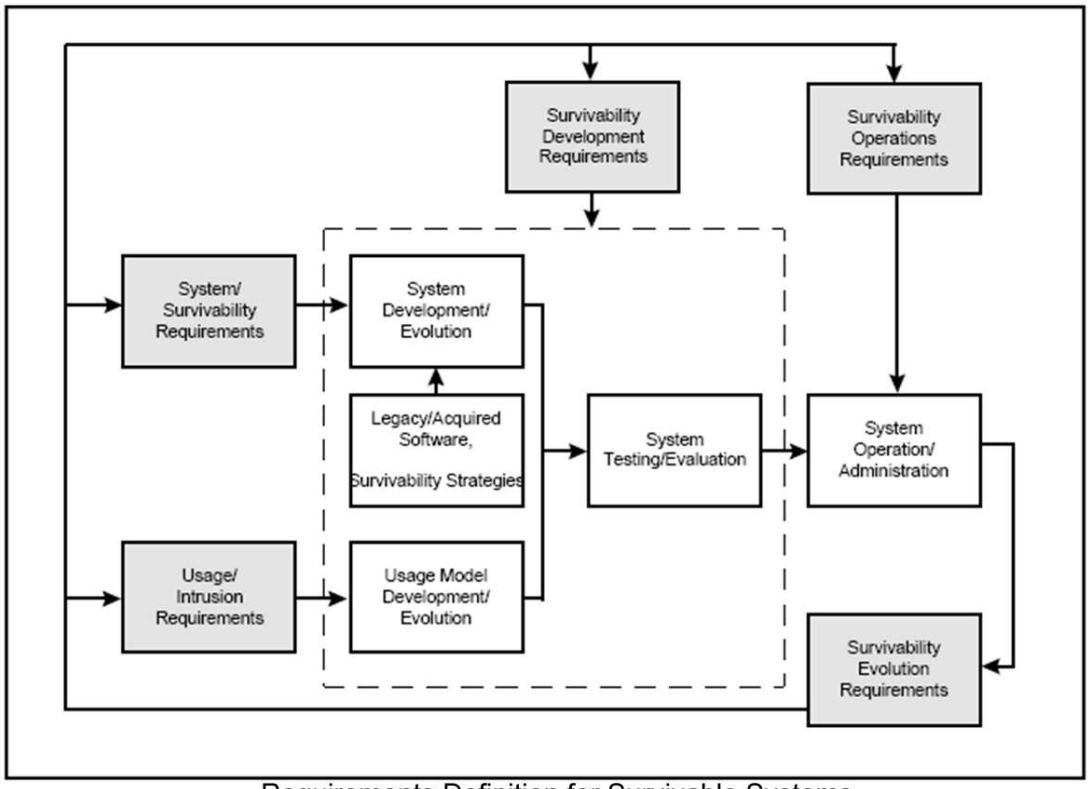
O sucesso de um sistema survivable depende do ambiente de computação em que o sistema sobrevivente opera [9]. A tendência em ambientes de computação conectados é para infra-estrutura de rede na maior parte ilimitados. Um sistema limitado é um em que as peças de todo o sistema são controladas por uma administração unificada e podem ser completamente caracterizadas e controladas. Em um sistema ilimitado, cada participante tem uma idéia incompleta do todo, depende da informação confiável fornecida por seus vizinhos, e não pode exercitar o controle fora do seu domínio local. Um ambiente ilimitado exibe as seguintes propriedades:

- Domínios administrativos múltiplos sem autoridade central;
- Ausência de visibilidade global;
- A interoperabilidade entre domínios administrativos é determinada pela convenção;
- Sistemas extensamente distribuídos e interoperáveis;
- Os usuários e os atacantes podem ser iguais no ambiente;
- Não pode ser dividido em um número finito de ambientes limitados.

A maioria das tecnologias de segurança depende de suposições subjacentes sobre a natureza e a estrutura dos sistemas. Geralmente, estes incluem suposições que os sistemas são fechados com controle administrativo central, e que é possível observar toda a atividade desejada dentro do sistema. Estas suposições podem ter sido apropriadas quando os sistemas eram consoles isolados com relações altamente controladas ao descaso do mundo. Hoje, entretanto, os sistemas são abertos, sem uma pessoa ou organização que detêm o controle administrativo, e com todo o observador, dentro ou fora do sistema, com visibilidade limitada na estrutura, na extensão e na topologia do sistema. A falta do controle administrativo central e uma ausência de visibilidade global são propriedades da Internet e das aplicações distribuídas.

1.3 Exigências do survivability

A definição e a análise das exigências do survivability são uma etapa crítica para conseguir o sistema sobrevivente. A **figura** abaixo descreve um modelo interativo que define essas exigências. O survivability abarca não somente as exigências de funcionalidade do software, mas igualmente exigências para o uso do software, desenvolvimento, operação e evolução do mesmo. Assim os cinco tipos de exigências são relevantes aos sistemas survivable no modelo, que estão denominadas: *systems, usage, development, operations* e *evolution.*



Requirements Definition for Survivable Systems

Fonte: Ellison et al, 1997

1.4 Princípios e garantia da informação

Segundo a CERT [3] existem 10 princípios que tratam do survivability e da garantia de informação simultaneamente, são estes: Princípio 1 – a survivability/sobrevivência é uma preocupação transversal à organização; Princípio 2 – tudo que existe na organização deve ser considerado dado; Princípio 3 – os dados não tem igual valor para a empresa, os riscos devem ser geridos; Princípio 4 – a política de garantia da informação governa ações; Princípio 5 – a identificação dos usuários, dos sistemas informáticos e dos componentes da infra-estrutura de rede é uma atividade critica; Princípio 6 – as unidades funcionais de sobrevivência(SFUs) são uma maneira útil de pensar sobre as redes da empresa; Princípio 7 – o conhecimento da segurança na prática(SKiP) fornece uma abordagem estruturada; Princípio 8 – o plano (road map) guia escolhas de implementação, já que a tecnologia não é toda igual; Princípio 9 – desafio supõe compreender o risco; Princípio 10 – a habilidade de comunicação é crítica para alcançar todos os componentes.

CAPÍTULO 2 - ARQUITETURAS SURVIVABILITY

As arquiteturas do survivability realçam a sobrevivência de sistemas críticos de informação fornecendo um mecanismo que permite a detecção e o tratamento de vários tipos de falhas.

Para Knight et al [5], a abordagem para as arquiteturas survivability é baseada no uso do controle explícito para gerenciar os sistemas de informação usando dados das infra-estruturas, seus sistemas de informação, e seus ambientes operacionais. Na essencia, este sistema de controle é responsável por escolher uma configuração para um sistema crítico de informação baseado nas condições atuais para minimizar a perda de serviço a qualquer tempo, com garantias que sob as circunstâncias definidas, os serviços cumprirão as exigências do survivability. Na prática, implementar arquiteturas de sobrevivência levanta muitas observações que devem ser tratadas antes que as técnicas possam ser aplicadas. As observações particulares *a priori* incluem a reconfiguração da aplicação e a segurança dos mecanismos de survivability.

O quadro abaixo mostra algumas arquiteturas survivability para sistemas de TI existentes.

Caracterist.	SABER	ITDOS	C4I
Mecanismo segurança	Integrates several security and survivability mechanisms	Symmetric Encryption Session Keys	"Plan-Based Survivability", Mobile IP and Ad Hoc network protocols for military use
Mecanismo reação/ proteção	Process Migration System, Network Denial-of-Service (DoS), Secure Overlay Services (SOS), An automated software-patching system	Distributed Object Middleware, CORBA	"Plan-Based Survivability"
Detecção intrusão	Network- and host-based intrusion detection, Anomaly-, registry- and file-based detection, Surveillance detection	Fault Tolerant Multicast Protocol + CORBA	"Plan-Based Survivability"
Dominio	Network Systems	Heterogeneous Distributed Middleware Systems	Mobile Military Tactical Systems
Maturidade	Prototype	Prototype	Prototype
Ano de publicação	2003	2002	1999

E a seguir um quadro comparativo entre métodos, modelos e frameworks de survivability usados para sistemas de TI.

Métodos	Tipo de Método	Abordagem	Dominio	Maturidade
ATAM	Design and Analysis Method	Intrusion Scenarios, Quality attributes	Not limited	High, widely used
SNA	Design and Analysis Method	Scenarios	Large-Scale Distributed-Network Systems	High, based on ATAM
Willow	Modeling tool	Fault Avoidance, Fault Elimination, Fault Tolerance	Critical Distributed Applications	Medium
QuO	Modeling tool	Quality Objects, Adaptation, Protection	Middleware Applications	Medium
Framework WANs	Modeling tool	Metrics, Restoration Techniques	Wireless Access Networks	Low
Framework arquitetural	Modeling tool	System Description Language, Runtime Monitoring, Failure Detection, Dynamic Recourse Allocation	Military C2 Systems	High
Easel	Modeling and Simulation Language	Discrete Event Simulations Models	Unbounded Systems, Ad Hoc Networks	Medium

As arquiteturas seguras atentam limitando o impacto das vulnerabilidades através da aplicação de uma abstração arquitetural [**8**].

CAPÍTULO 3 – EXIGÊNCIAS DE DESENVOLVIMENTO PARA APLICATIVOS SOBREVIVENTES

Os sistemas práticos disponíveis quase nunca são 100% construídos customizados, mas são construídos com um pouco dos componentes geralmente disponíveis fora com estruturas internas que são conhecidas. A tendência para os sistemas é a busca pela integração, e o reuso da codificação com os esforços direcionados a personalização do projeto, que são fundamentais na tecnologia de programação moderna. Infelizmente, a complexidade intelectual se associou com o projeto e a codificação do software, e o teste assegura-se virtualmente de que os erros exploráveis possam e estejam descobertos nos produtos de domínio público cujas estruturas internas são extensamente disponíveis para análise. Quando estes produtos são incorporados como componentes de sistemas maiores, aqueles sistemas tornam-se vulneráveis às estratégias de ataque baseadas nos erros exploráveis. Os componentes populares da web e do domínio público oferecem a atacantes um conjunto dos alvos com estruturas internas conhecidas e geralmente inalteradas. Esta falta de variabilidade entre componentes traduz em falta de variabilidade entre sistemas, e cria vulnerabilidades comuns a todos. Esses sistemas permitem um ataque potencial em que uma única estratégia tenha um impacto amplo e devastador.

O survivability coloca exigências estritas no sistema, práticas de desenvolvimento e teste. Funcionalidade inadequada e erros de software podem ter um efeito devastador na sobrevivência do sistema e forneça oportunidades para exploração da intrusão [9]. As práticas seguras da engenharia são requeridas para criar softwares sobreviventes. Segundo Ellison et al, cinco princípios são exigências para o desenvolvimento de sistemas survivable e práticas de testes:

- Especificar precisamente as funções requeridas do sistema em todas as possíveis circunstâncias do uso do sistema;

- Verificar a exatidão de implementações de sistema no que diz respeito às especificações formais;
- Especificar a *function usage(),* que mostra as informações do usuário, em todas as circunstâncias possíveis do uso do sistema, incluindo o uso do intruso;
- Teste e certifique o sistema baseado na *function usage* e métodos estáticos;
- Estabelecer equipes permanentes de prontidão para a monitoração do sistema, a adaptaçãoe e evolução.

Componentes prontos, ditos de prateleira ou COTS (Commercial Off-The-Shelf), são parte integrante da maioria dos sistemas atuais. Tais componentes geralmente são usados como parte de um sistema maior, seja na construção do sistema, como compiladores e bibliotecas de rotinas, seja para implementar grandes subsistemas, como um sistema gerenciador de banco de dados (SGBD) ou um middleware [11]. As práticas da engenharia são requeridas para lidar com os softwares legados como também os componentes COTS.

3.1 Aspectos das estratégias de solução do survivability

Há quatro aspectos da solução do survivability que podem servir como base para estratégias de sobrevivência. Estes quatro aspectos são: resistência, reconhecimento, recuperação, e adaptação e evolução do sistema.

Existem inúmeras técnicas para tratar estes quatro aspectos. Algumas ou todas as técnicas podem aplicar-se aos sistemas survivable. Ellison [9] não lista todas as técnicas, mas as categoriza dentro de aspectos bem amplos. A tabela abaixo contém os 4 aspectos das soluções survivability e as taxonomias representativas das respectivas estratégias.

Survivability Aspect	Taxonomies of Strategies
Resistance	• traditional security, including encryption and covert channels • diversity and maximized differences in individual nodes • analytic redundancy and voting • specialization, division of labor, trust, and information • continuous validation of trust • exhibited stochastic properties and random behavior
Recognition	• analytic redundancy and testing (including failures in software, encryption, and trust) • intrusion monitoring and suspicious activities • system behavior and integrity monitoring
Recovery	• physical and information redundancy • non-local copies of information resources • preparation, readiness, contingency planning, and response teams
Adaptation and Evolution	• general or specific changes to resist, recognize, or recover from new vulnerabilities that are discovered • broadcast of warnings to other nodes • broadcast of adaptation and evolution strategies • deterrence through retaliation or punishment

Fonte: Ellison et al, 1997

A Taxonomy of Strategies Related to Survivability

3.2 Técnicas automatizadas para sobreviver aos erros fatais de software

Muitos erros em sistemas de software não se manifestam até que o sistema esteja disposto e em uso na produção. Os erros fatais podem ter consequencias severas em tais situações, e a abordagem padrão para lidar com os erros é notificar a organização que produziu o sistema causador do problema. Uma publicação do host faz esta abordagem sub-ótima: Error Notification, Error Correction Delays, Distribution Difficulties, Error Reprodution e New Errors or Anomalies.

Diversas técnicas foram desenvolvidas, e juntas eliminam a possibilidade de determinados tipos de erros fatais [6]:

- Forced Loop Termination: esta técnica de eliminação de laço infinito aprende simplesmente limites razoáveis para o número de iterações que cada laço pode executar, observando as execuções bem sucedidas, a seguir retira cada laço se ele atenta para exceder seu limite razoável da iteração;

- Forced Recursion Termination: a técnica simplesmente limita o tamanho da pilha, então imediatamente retorna a todas chamadas de procedimento que tentem exceder este limite;
- Deadlock Elimination: simplesmente libera os locks até que todo deadlock é eliminado;
- Memory Leak Elimination: o algoritmo simplesmente aloca um buffer de tamanho fixo para os locais que podem exceder a memória alocada. Ele então aloca dados ciclicamente fora deste buffer.
- Resource Leak Elimination: alocar um número conceitualmente ilimitado de recursos fora dos pools de tamanho fixo aplicando alguma política para realocar recursos fora dos pools quando o sistema esgotar seus recursos;
- Invalid Addressing Elimination: a técnica de eliminação de endereçamento inválido (também conhecida como computação failure-oblivious) executa verificações dos limites para rejeitar escritas fora da fronteira e valores arbitrários produzidos fora dos limites lidos.

Os resultados obtidos indicam que os sistemas de software geralmente toleram a perturbação que estas técnicas podem introduzir.

CAPÍTULO 4 – ASPECTOS RELACIONADOS À QUALIDADE

A entrega de serviços essenciais é a capacidade de um sistema para manter as propriedades essenciais, isto é, níveis específicos de integridade, segurança, desempenho, e atribuitos de qualidade definidos na missão organizacional. Assim, é importante definir níveis mínimos de atributos da qualidade que devem ser associados com os serviços essenciais.

O survivability deve ser integrado e tratado em uma paridade com outras propriedades de sistema para desenvolver sistemas com funcionalidade e desempenho exigidos que podem igualmente suportar falhas e acordos. Um padrão do survivability precisa ser estabelecido razoavelmente cedo, por exemplo, durante o desenvolvimento do conceito das operações, e de ser revisitado em milestones principais do desenvolvimento tais como as exigências, a arquitetura, etc. Isto soa como se sugere um ciclo de vida do tipo cascata, mas trabalha de fato com modelos do ciclo de vida mais moderno tais como o modelo espiral.

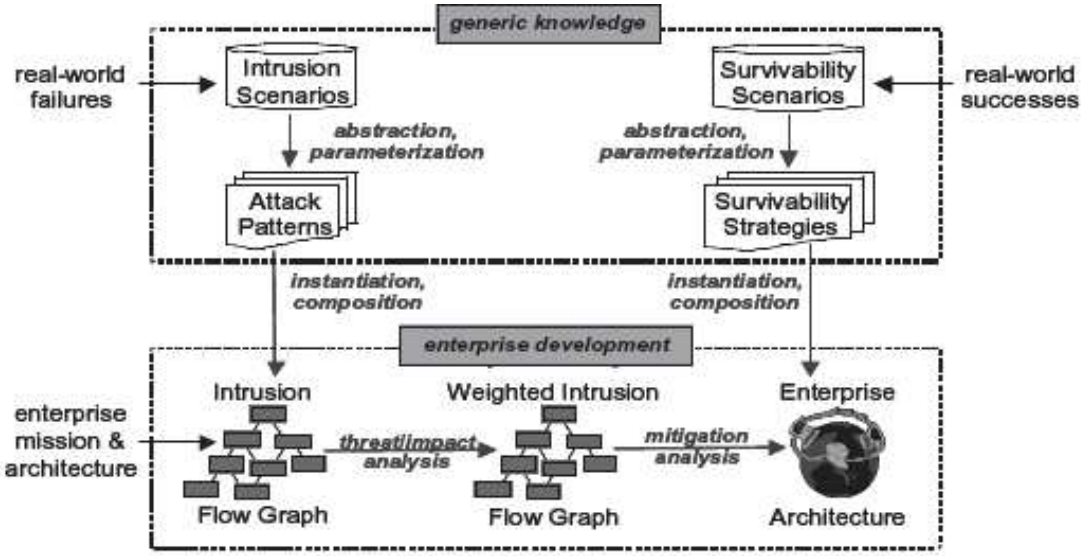
As complexidades de sistema em grande escala da rede podem ser reduzidas e controladas por uma disciplina unificada da engenharia para a análise e o projeto que inclui o survivability em um framework detalhado. Definimos fundamentos da engenharia para a tecnologia da Flow-Service-Quality (FSQ) [1] baseada nas estruturas do fluxo da tarefa do usuário e nos traços da sua arquitetura, uma abordagem computacional para atributos da qualidade, incluindo o survivability, e um framework da arquitetura para a gerência dinâmica dos fluxos e dos seus atributos da qualidade. Este processo pode ser aplicado à especificação, ao projeto, e à operação de sistemas novos, assim como à análise de sistemas existentes para as dependências e os riscos do survivability que podem impactar o desempenho da missão.

O objetivo desta abordagem é fornecer métodos de engenharia para representar e raciocinar sobre fluxos de sistema como artefatos essenciais da análise e do desenvolvimento de sistema complexo. Os fluxos de sistema são compostos de serviços de sistema e devem satisfazer atributos da qualidade tais como a confiabilidade, o desempenho, e o survivability. Conseqüentemente, são estes três conceitos, fluxo, serviço, e qualidade que dão forma à base da estrutura de FSQ para sistemas de rede em grande escala da engenharia. Na engenharia de FSQ, os atributos da qualidade tais como a segurança, o survivability, a confiabilidade, e a disponibilidade são definidos como funções computacionais e associados com ambos os fluxos e serviços [1]. Um aspecto chave desta abordagem computacional é a habilidade de associar atributos da qualidade com os fluxos específicos menos do que com os sistemas inteiros, permitindo desse modo a diferenciação entre as capacidades do atributo baseadas na criticidade da missão na engenharia do survivability.

Em um mundo da engenharia centrada no fluxo e de atributos computacionais da qualidade, é natural considerar moldes da arquitetura de sistema baseados na gerência dinâmica do atributo do fluxo e da qualidade.

As estruturas do fluxo têm o potencial para reduzir a complexidade e melhorar a viabilidade na aquisição de sistema da rede, no desenvolvimento, na gerência, e na operação e pode contribuir à integração de sistemas diversos para cumprir exigências novas da missão. Ademais, as estruturas do fluxo podem ser usadas para extrair e documentar operações de missão crítica em sistemas existentes para compreender melhor dependências componentes para a análise do survivability.

Para **Ellison** et al, 2002 [10] a informação do ataque e do survivability precisa ser estruturada e reusável assim como eles podem ser aplicados no refinamento iterativo de arquiteturas do survivability. Construindo a base de conhecimento de modo que seja independente da empresa, é fornecido um conhecimento para construção de fluxos gráficos específicos da intrusão em uma maneira disponível, assim fazendo o refinamento iterativo e a análise da arquitetura de custo efetivo da empresa. A **figura** a seguir mostra a construção dos cenários de intrusão das falhas do mundo real, documentadas.



Fonte: Ellison et al, 2002

Moore et al [2] interpretam as intrusões amplamente para incluir os ataques que alvejam pessoas e fluxos de tarefa assim como aquelas que alvejam a tecnologia. Desenvolveram um conhecimento para derivar padrões geralmente de retorno do ataque de cenários de intrusão. Estes padrões do ataque são parameterizados de modo que possam instanciá-los para variados ambientes da empresa. Os gráficos de fluxo específicos da empresa de intrusão são gerados destes padrões do ataque através de um processo de instanciação e da composição.

Explorar a viabilidade desta abordagem com sua aplicação para melhorar a segurança e o survivability de uma arquitetura particular de empresa para uma classe particular de ataques é a idéia proposta.

CAPÍTULO 5 – ESTIMANDO O SURVIVABILITY

Das perspectivas de designers e compradores, comparando vários projetos baseados em seu survivability é crítico elaborar trade-offs de custo e benefício. Em seguida discutiremos diversos tipos de análise que podem ser executados em um projeto de rede que possa fornecer uma avaliação mais quantitativa do survivability.

Medidas residuais para uma infra-estrutura de TI são as mesmas medidas usadas para descrever a infra-estrutura antes de um evento catastrófico, mas são aplicadas para prever o estado da infra-estrutura depois que os efeitos do evento são considerados [**7**]. Abaixo são discutidas quatro medidas residuais que são geralmente importantes:

- Únicos pontos residuais de falha: em comparando 2 projetos de infra-estrutura candidatos, o projeto com menores pontos únicos de falha é considerado geralmente o mais robusto do que a alternativa. Ao examinar o survivability de uma infra-estrutura no que diz respeito a um evento catastrófico particular, a infra-estrutura com menos pontos residuais da falha é intuitivamente mais survivable. Esta medida é uma contagem simples.
- Disponibilidade residual: a mesma análise da disponibilidade feita em uma infra-estrutura não danificada pode ser aplicada a uma infra-estrutura depois que foi danificada por um evento qualquer. Geralmente, a mais elevada disponibilidade residual de uma infra-estrutura é mais survivable.
- Desempenho residual: uma infra-estrutura residual que não tenha nenhum ponto de falha e tenha alta disponibilidade residual não pode ser útil da perspectiva da sobrevivência dos usuários. Consequentemente, o desempenho recebido pela comunidade de usuários sobreviventes precisa ser analisado. A análise deve levar em consideração todo aumento ou decréscimo na atividade da infra-estrutura resultante da resposta da organização ao evento que está sendo estudado.

- Tempo de reconstrução: uma vez que o evento ocorreu, o tempo exigido para recomeçar as atividades críticas de missão é uma das medidas residuais mais importantes para descrever a sobrevivência de uma infra-estrutura de TI. Os cálculos de tempo da reconstrução devem levar em consideração ambos os planos de recuperação de emergência e prejuízo das capacidades de recuperação causadas pelo evento.

Em avaliar arquiteturas alternativas algumas medidas compostas através de um conjunto de eventos potenciais são frequentemente úteis. A seguir teremos dois métodos para comparar a sobrevivência de arquiteturas alternativas.

É importante o desenvolvimento de um conjunto de eventos cônicos, ou seja, um conjunto que inclue todos os tipos importantes de eventos que a infra-estrutura de TI deve ser projetada para sobreviver. Então o conjunto de eventos cônicos inclui pelo menos um exemplo de cada tipo observado. Idealmente, inclui o pior caso.

Depois do cálculo das métricas residuais para cada um dos eventos no conjunto cônico e arquiteturas alternativas, é desejável fazer uma comparação objetiva dos survivabilities [7], e os métodos estão abaixo:

- Métodos de múltiplos critérios: dentro da área de especialidade da otimização dos multi-critérios, o conceito das melhores alternativas foi formalizado. Este conceito indica o estado que uma alternativa tem de scores que são maiores ou iguais aos scores correspondentes de todas as alternativas restantes e tem um único melhor score para pelo menos um critério. Consequentemente se uma das arquiteturas avaliadas tem alta disponibilidade residual para todos os eventos, o tempo de reconstrução mais baixo para todos os eventos e poucos pontos de falha, é claramente a melhor arquitetura entre aqueles que estão sendo comparados.
- Métodos de ponderação: nas situações onde há muitos critérios que podem ser pesados por alguns meios subjetivos e não existe melhor alternativa baseada nos métodos multi-critérios existentes, uma abordagem razoável para selecionar a alternativa mais survivable é criar um survivability index. O índice seria uma soma ponderada dos valores do critério Cij para cada evento multiplicado pelos seus

respectivos pesos, Wij. Um valor de índice é computado para cada alternativa e a alternativa com o índice mais elevado é selecionado como a melhor alternativa, e caso contrário, pior alternativa. A fórmula é:

$$\text{Survivability index} = \sum i \sum j W_{ij} C_{ij}$$

Na equação anterior, o index *i* é o conjunto de eventos prejudiciais, e o index *j* é o critério de sobrevivência.

CAPÍTULO 6 – COMENTÁRIOS FINAIS

As soluções do survivability são melhor compreendidas como estratégias da gerência de riscos, que dependem primeiramente de um conhecimento íntimo da missão que está sendo protegida. O foco da missão expande soluções do survivability independente das soluções puramente técnicas. Para os sistemas novos, o survivability impõe restrições em todas as fazes do processo de programação do software e para os sistemas existentes fornece uma perspectiva nova na evolução e melhoramento.

As estruturas de fluxos podem ser usadas para extrair e documentar operações de missão crítica em sistemas existentes, para compreender melhor as dependências componentes para análise do survivability.

Novos métodos e ferramentas estão em desenvolvimento para suportar soluções de survivability e merecem destaque, a exemplo de: sistema de controle de comportamento, adaptativo; exigências de infra-estrutura do software e a identificação de um trajeto da migração para sistemas legados; e os simuladores de sistemas sobreviventes.

REFERÊNCIAS BIBLIOGRAFICAS

[1]A. Hevner, R. Linger, G. Walton and A. Sobel. The Flow-Service-Quality Framework: Unified Engineering for Large-Scale Adaptive Systems. In *IEEE Computer Society Press*, 2002.

[2]A. Moore, R. Ellison e R. Linger. Attack Modeling for Information Security and Survivability. Technical Report CMU/SEI-2001-TN-001,ADA388771, 2001.

[3]CERT Coordination Center. Principles of Survivability and Information Assurance. http://www.cert.org/info_assurance/principles.html, 2005.

[4]L. Figueirêdo. Segurança da Tecnologia da Informação. UFMG/DCC Fev 2002.

[5]J. Knight, K. Sullivan, M. Elder and C. Wang. Survivability Architectures: Issues and Approaches.

[6]M. Rinard. Automated Techniques for Surviving (otherwise) Fatal Software Errors. In *Electronic Notes in Theoretical Computer Science 174*, pages 113-116, Massachusetts, 2007.

[7]J. Murphy and T. Morgan. Availability, Reliability, and Survivability: An Introduction and Some Contractual Implications. In *Journal of Defense Software Engineering.* http://www.stsc.hill.af.mil/CrossTalk/2006/03/0603MurphyMorgan.html, Mar 2006.

[8]P. Tarvainen. On survivability of IT Systems. In *Seminar of Embedded Systems Development.* Finland, Nov 2004.

[9]R. Ellison, D. Fisher, R. Linger, H. Lipson, T. Longstaff, and N. Mead. Survivable Network Systems: An emerging discipline. Technical Report CMU/SEI-97-TR-013, Nov 1997.

[10]R. Ellison, R. Linger, H. Lipson, N. Mead e A. Moore. Foundations for Survivable Systems Engineering. In *The Journal of Defense Software Engineering*, pages 10-15, Pittsburg, Jul 2002.

[11]R. Obelheiro, A. Bessani e L. Lung. Analisando a Viabilidade da Implementação Prática de Sistemas Tolerantes a Intrusões.

[12]T. Bellocci, C. Ang, P. Ray and S. Nof. Information Assurance in Networked Enterprises: Definition, Requirements, and Experimental Results. In *School of Industrial Engineering*, Jan 2001.