5
Dificuldades na Geração de Métricas de Segurança Quantitativas

Até hoje a ciência da computação não obteve sucesso em definir métricas de segurança de software quantitativas que sejam simples, confiáveis e aceitas por todos para a avaliação da segurança de sistemas de informação. Muitos pesquisadores da comunidade de garantia da segurança de um software e da Engenharia de Software se opõem à introdução de uma métrica para quantificar a segurança e a qualidade de software respectivamente (Ortalo, 1999).

Se não podemos medir a segurança do software, então como podemos melhorá-la? Bellovin (2006) afirma que quando podemos medir e expressar em números algo sobre o objeto medido; mas quando não conseguimos expressar em números, o nosso conhecimento sobre segurança é pobre e insatisfatório. Estaria Bellovin certo em relação à segurança de software? Seria realmente impossível avaliarmos a segurança de um software antes que ele seja colocado em produção? Os métodos apresentados no capítulo anterior ainda se encontram em fase de amadurecimento e não são conclusivos em relação ao assunto, apesar de algumas vantagens por eles oferecidas.

De fato, temos dificuldades para definir métricas de software quantitativas, isto porque a segurança ao conter sutilezas e desafios diferentes, principalmente relacionadas a intenção de alguém em explorar vulnerabilidades dos sistemas, os problemas de segurança da informação tornam-se mais difíceis de serem entendidos do que em outras disciplinas da garantia da qualidade, tais como *safe software* e tolerância à falhas. Por exemplo, é óbvio que uma série de eventos não seguros certamente ocorreram antes de um desastre de avião, mas a maioria dos ataques à segurança de computador são muito menos observáveis e são quase sempre não detectados, ou só detectados muito tempo após a ocorrência do evento.

Uma abordagem interessante sobre as sutilezas que estão por trás da segurança de software está em Voas (1996), que faz uma comparação entre invasões militares em guerras tradicionais com invasões de software, a chamada guerra cibernética. Voas (1996) diz*: "uma ofensiva não bem sucedida ao software quase sempre fortalece o atacante ao dar a ele o conhecimento sobre o alvo de ataque e não fortalece o alvo atacado por meio do enfraquecimento do atacante. Nas tradicionais invasões militares, muitas vezes, uma ofensiva não bem sucedida enfraquece o atacante pelo menos tanto quanto o site atacado. Além disso, em tradicionais estratégias de guerra, o potencial para retaliação fornece uma intimidação ao atacante antes dele atacar. Na batalha da informação, contudo, o medo de retaliação é mínimo, e não afeta a balança do poder. A maior parte das técnicas de segurança da informação usadas hoje são ou baseadas em táticas ultrapassadas de 20 anos atrás, ou são baseadas em táticas aplicáveis somente em guerras convencionais"*. Como muitas vezes não conseguimos nem detectar a ocorrência de um evento inseguro ou um padrão de ataque, isto torna mais difícil a medição de certas características de segurança do sistema, como por exemplo, confidencialidade.

Além do modo intencional como o software é atacado, há outras complicações adicionais que precisamos considerar na busca por métricas de software significativas. Neste ponto, o valor dos ativos, as ameaças e as vulnerabilidades são elementos críticos para a análise de riscos de segurança como um todo e são (ou deveriam ser) considerados na maioria das decisões que têm a ver com segurança. Cada um destes elementos traz dificuldades quando tentamos incorporá-los em uma métrica de segurança útil. O valor do ativo é o mais fácil de medir destes três elementos; porém, certos aspectos do valor, como a boa reputação de uma companhia ou de uma pessoa é difícil, se não impossível, de se quantificar.

Alguns acreditam que ameaças não podem ser sempre medidas, isto devido a impossibilidade de realmente calcularmos o potencial dos danos (prejuízos), embora os resultados de pesquisas e de outras fontes externas de informação possam ser úteis para quantificar a ameaça em um nível mais alto, no nível gerencial. Hoje até existem alguns benchmarks e ferramentas (Chess, 2007)

automatizadas desenvolvidas para descobrir níveis de vulnerabilidades de sistemas de computador, porém medições de outros aspectos de vulnerabilidades, como o grau de conhecimento sobre assuntos de segurança entre usuários de computador, permanecem um pouco subjetivas.

Medir os riscos de segurança, a probabilidade de uma agressão intencional ou não ser bem sucedida, junto com uma avaliação do tamanho do dano – impacto – pode nos dar uma boa idéia de qual a nossa situação face à segurança.

5.1 Evolução de Nossa Capacidade de Tratar a Insegurança

Um aspecto interessante para compreendermos as dificuldades de medirmos segurança é avaliarmos como está evoluindo a nossa capacidade de tratar a insegurança. O gráfico (figura 21) a seguir (Williams et al., 2006) denominado *Hype Cycle* nos dá uma idéia que tipos de ameaças existem hoje em relação à segurança e quais conseguimos tratar de forma eficiente ou não. Para compreender a curva, é importante entender alguns conceitos sobre ela:

Technology trigger - período que vai do conhecimento da existência de uma determinada classe de ameaça até se tornar umas das maiores preocupações de segurança (ponto mais alto da curva). Neste período ainda não existe uma forma razoável para o tratamento da ameaça, como por exemplo *Insecure Application Development.*

Peak of Inflated hyperbole – neste período as organizações devem começar a pesquisar a ameaça e seu impacto na organização, bem como entender as possíveis soluções, pois o impacto sobre os negócios pode ser significativo.

Trough of complacency – a partir do pico em direção ao ponto mais baixo da curva temos o período de desilusão em relação às contramedidas usadas pelas organizações para minimizar os efeitos finais das ameaças. Normalmente isto ocorre porque não temos soluções maduras para tratar uma determinada classe de ameaça.

Slope of Enlightenment - Período em que ocorre o início do amadurecimento

das tecnologias e organizações no tratamento das classes de ameaças. As organizações já começam a ser capazes de operacionalizar estas tecnologias.

Plateau of Permanent Annoyance – Período em que podemos constatar a presença na maioria das organizações de programas, tecnologias e processos maduros para tratar classes de ameaças.

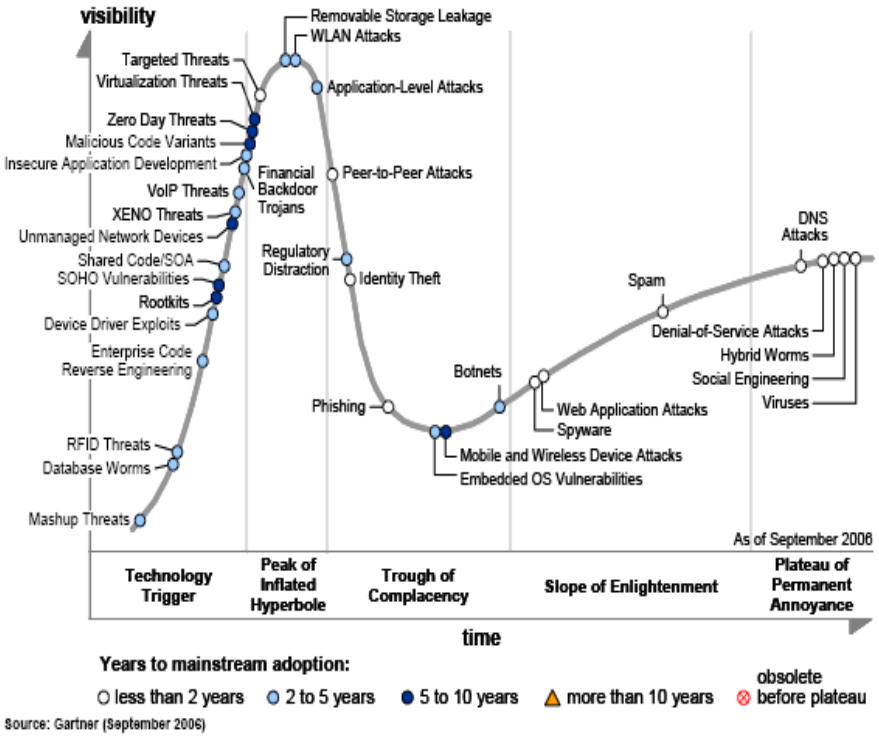

Figura 21: Hype Cycle for Cyberthreats, (extraída de Williams et al, 2006)

A cada classe de ameaças temos associado um tempo que significa o tempo estimado para que esta classe de ameaças atinja o *Plateau of Permanent Annoyance.*

Uma análise da curva acima, particularmente o item *Insecure Application Development*, nos mostra que a maioria das empresas ainda não tem adotado práticas relacionadas com o desenvolvimento seguro, conseqüentemente, métricas de segurança. Além disso, o desenvolvimento de aplicações carece de metodologias, processos e ferramentas que possam ser aplicadas na Engenharia de Software seguro, principalmente nos estágios iniciais do desenvolvimento e ao longo do ciclo de vida da aplicação (Gartner, 2006). Vulnerabilidades podem ser

introduzidas nas fases de análise, projeto e construção dos programas. Hoje, boa parte dos esforços de segurança é aplicada na fase de operação e, com menor grau de dedicação, nas fases de garantia da qualidade e testes. Como resultado disso, software é construído com vulnerabilidades e colocado em operação. Ainda desenvolvemos sistemas não confiáveis que resultam em muitas reclamações dos usuários. A questão é se não somos capazes de provar corretude, usabilidade, confiabilidade e outros requisitos não quantificáveis, como podemos reivindicar por métricas que quantifiquem segurança?

A curva acima também nos permite concluir que ainda levaremos um tempo para estabelecermos métricas de segurança confiáveis. Tais métricas são difíceis porque a própria disciplina ainda está em um estágio inicial de desenvolvimento, basta verificar (antes do pico da curva) a quantidade de classes de vulnerabilidades ainda não adequadamente tratadas.

A seguir são detalhadas algumas dificuldades encontradas na busca por métricas de avaliação quantitativa de segurança.

5.2 Dificuldades que encontramos na busca por métricas quantitativas de segurança

5.2.1 Outros campos têm números para expressar. Qual é a equivalência para segurança do software?

O software não obedece às leis da física (Bellovin, 2006). Na maioria dos casos, não podemos aplicar matemática no código para provar corretude da mesma forma que um engenheiro civil pode aplicar fórmulas para provar a características da resistência estrutural de uma ponte.

De fato, as métricas de segurança são mais qualitativas do que quantitativas. Como segurança é freqüentemente não quantificada, o mínimo de segurança necessário é quase sempre um sentimento. A natureza humana e a natureza da segurança estão em conflito neste ponto, pois as pessoas e as organizações tendem

a se acomodar com o passar do tempo com a situação atual, mas, de fato, a segurança pode degradar com o passar de tempo. Tipos novos de ataques e novas aplicações para tipos velhos de ataques podem prejudicar a segurança de um programa – Mesmo quando uma organização torna-se mais e mais complacente porque segurança "Não tem sido ainda um problema!".

5.2.2 Composição pode Introduzir Falhas

Não temos conhecimento sobre as conseqüências da composição de vários mecanismos de segurança (Bellovin, 2006). A agregação de várias medidas para reduzir vulnerabilidades pode, paradoxalmente, resultar em um sistema menos seguro. Simplesmente não sabemos o que nós temos quando montamos um perímetro de segurança em um sistema de informação. Não temos nenhuma certeza de que implementamos adequadamente a composição ou que o resultado será um sistema mais resistente se nós desenvolvemos recursos adicionais. Qualquer um que tem tentado corretamente configurar um *firewall* confirmará um falso sentimento de segurança que pode ocorrer devido à alta probabilidade de uma só má aplicação de uma regra ou a omissão de uma simples regra, emparelhada com a propagação de dados de configuração através da empresa, e nós compomos a possibilidade de um compromisso seguro. Nós mantemos a confiança na experiência de nossos administradores de sistemas ou engenheiros de segurança e seus conhecimentos específicos para garantir a corretude do sistema.

Medir segurança a partir da composição de sistemas de segurança é hoje algo impossível. Precisamos estudar como podemos avaliar a segurança a partir da composição de elementos, principalmente considerando a interação entre estes elementos. Exemplo: *firewall* e tecnologia Java. Como um elemento, tecnologia Java, pode influenciar a segurança estabelecida pelo outro, firewall?

5.2.3 Pessoas e processos podem diminuir a segurança

Pessoas, que são por natureza propensas ao erro, constroem software. Nós podemos avaliar certas características do processo de construção do software e das

pessoas que trabalham nele, mas no fim – qualquer um deles pode, intencionalmente ou não, corromper o sistema e diminuir sua segurança. Isto torna questionável se abrir ou não o desenvolvimento de sistemas é uma medida útil ou um pesadelo de controle de versão.

Hoje boa parte dos problemas ocorrem dentro do próprio ambiente de trabalho (engenharia social) . Interessante observar é que estudos do Gartner e do CERT® comprovam que os invasores mais comuns, que realizaram os crimes mais significativos envolvendo software, são aqueles promovidos por pessoas que hoje têm o acesso legitimo à informação, ou que tiveram recentemente o acesso.

Mesmo depois de um ou mais processos comprovarem que são ótimos para produzir software seguro, estes processos e práticas devem permanecer eficientes quando utilizados por diferentes pessoas e em diferentes organizações e ambientes de desenvolvimento de software. A comprovação desta eficiência é relativamente difícil.

5.2.4 Falta de embasamento teórico

Muitas vezes as métricas são definidas sem um modelo formal embasado. Requisitos de segurança são definidos baseados em modelos de segurança. Métricas de segurança são obtidas ou por análise estatística ou testes dinâmicos baseados nos requisitos específicos de segurança e evidência da garantia da qualidade. Contudo, apesar de existirem inúmeros modelos de avaliação de segurança (demonstrados no capítulo 4), tais modelos ainda não estão bem consolidados na prática e em pesquisas acadêmicas.

A literatura carece de uma clara definição de quais propriedades devem ser consideradas ao fazer a avaliação da segurança do software. Então, para preencher este *gap*, o primeiro ponto na agenda de pesquisa deve ser a elicitação de tais propriedades.

5.2.5 O aspecto tempo

O aspecto tempo não vem sendo associado com as definições atuais de métricas de segurança. Um sistema medido hoje e considerado seguro, não significa que seja seguro amanhã. Isto porque novas vulnerabilidades são descobertas com o passar do tempo. Além disso, devidos as mudanças na tecnologia, práticas de desenvolvimento, políticas de segurança e leis, uma métrica que é significativa e útil hoje pode ser menos relevante amanhã.

Este é um aspecto importante que dificulta o estabelecimento de uma métrica quantitativa de segurança de software.

5.2.6 Valores Lógicos Tradicionais

Os valores lógicos tradicionais, verdadeiro ou falso, parecem não ser adequados para analisar segurança. Isto torna difícil afirmar, sem considerar uma margem de segurança, que um sistema é totalmente seguro.

5.2.7 Terminologia de segurança

A terminologia associada à segurança tem se demostrado inconsistente, o que tem complicado o desenvolvimento de métricas de segurança. Como podemos medir segurança se não conseguimos definir que propriedades devem ser consideradas para esta medição? (Vaughn et al, 2003)

5.2.8 As facilidades e recursos hoje disponíveis

É mais fácil e rápido atacar um sistema hoje do que era há 5 anos atrás devido aos recursos de comunicação existentes e ao conhecimento compartilhado da Internet. Esta tendência é provável que vá continuar à medida que ferramentas de ataque são desenvolvidas mais adiante, compartilhadas, e exploradas em uma base global (Bellovin, 2006).

Um atacante pode adquirir facilmente uma cópia destes aplicativos e praticar ataques em casa. Mesmo que um sistema de detecção de intrusão puder detectá-lo, ele não poderá reagir rapidamente contra um ataque automatizado.

5.2.9 Dificuldades dos Sistemas de Detecção de Invasão

Sistemas de detecção de invasão podem ser usados para avaliação de segurança. Porém, sistemas de detecção de invasão são muito famosos por falsos negativos (Bellovin, 2006). Isto compromete o uso de suas informações para a geração de resultados de segurança. Além disso, um atacante pode comprar uma cópia de teu sistema e praticar ataques em casa. Mesmo se um IDS puder detectá-lo, ele não poderá reagir rapidamente contra um ataque automatizado. Neste caso, a métrica pode não servir para muita coisa.

5.2.10 A garantia da segurança após a evolução do software

Mesmo que tenha sido desenvolvido um software seguro, sua evolução, suas correções não devem comprometer sua segurança. Geralmente, não existe uma maneira aceitável de se verificar que suas propriedades de segurança permaneceram após a realização destas atividades. Testes de regressão podem minimizar este problema.

5.2.11 A complexidade de verificar segurança em grandes sistemas

Verificar em grandes sistemas se a quantidade de defeitos de segurança está em um nível aceitável é extremamente difícil. Além disso, não existe uma taxonomia dos principais *bugs* de segurança existentes. (Vaughn et al, 2003)

5.2.12 A limitação de métodos baseados em contagem de Bugs de segurança

A contagem de bugs pode ser usada como uma métrica de segurança. Contudo, temos as seguintes desvantagens:

- O processo de detecção de bugs pode perder alguns bugs e pode leventar falsos positivos,
- Igual importância é dada para todos os bugs, mesmo que alguns bugs sejam mais fáceis de ser explorados e produzam mais prejuízos do que outros.

5.2.13 Outras propriedades desejáveis podem afetar a segurança como um todo

Propriedades desejáveis como *safety*, performance, usabilidade, etc podem comprometer a segurança. Encontrar o ponto de equilíbrio da qualidade desejada do software é um desafio para a Engenharia de Software.

5.2.14 Nem sempre podemos só considerar uma análise quantitativa

De fato, a segurança de sistemas baseados em software deve ser ponderada em relação ao ambiente humano em que o sistema é utilizado. Segurança está relacionada com ameaças, que dependem de fatores humanos. Por exemplo, considere a aplicação do princípio de menor privilégio para reduzir os riscos de exploração de um código executável. Para avaliar a efetividade de tal propriedade, o comportamento humano deve ser considerado e, portanto, tal propriedade é difícil de ser avaliada por números.

Avaliar quantativamente por meio de métricas de software pode não ser suficiente para compreender todas as facetas da segurança.

5.3 Recomendações para a Avaliação de Segurança

Apesar de todos os problemas citados acima, existem algumas métricas e práticas usadas hoje no desenvolvimento que podem ser de forma proveitosa entendidas para endereçar requisitos de segurança. Métricas baseadas em densidade de defeitos hoje já são utilizadas por alguns autores (Alhazmi et al, 2006)

Nenhuma métrica de segurança de software consegue quantificar com precisão a garantia de segurança em um sistema. Uma prova disso é que hoje não temos ainda qualquer métrica para medir quanto esforço um inimigo esperto gastou para explorar uma vulnerabilidade de um sistema (Bellovin, 2006). Ainda que não tenha sido encontrada tal métrica, segundo o relatório final produzido no ***1th Workshop on Information-Security-System Rating and Ranking***, tais métricas podem ser em resumo supridas, a partir de uma combinação de métricas (Henning, 2001). As métricas podem ser combinadas com outras métricas em um contexto específico, mesmo que pareçam não ser muito úteis em seus próprios contextos. Por exemplo, métricas baseadas em análise de riscos podem ser combinadas com métricas baseadas em densidade de vulnerabilidades (Alhazmi et al, 2006). Obviamente que esta combinação não pode ser aplicada de forma genérica em todos os domínios de interesse.

Muitas métricas supostamente consideradas subjetivas e/ou qualitativas mostraram-se mais úteis do que métricas quantitativas (Henning, 2001). Embora as técnicas de penetração não sejam consistentes e repetíveis, seus resultados são úteis e significativos, sendo uma das medições mais úteis que encontramos hoje para a garantia da segurança de sistemas.

Avaliação de riscos pode ser empregada na avaliação de segurança de software Trabalhos como System Development Life Cycle, SDLC, (Redwine, 2004) utilizam a análise de riscos ao longo do processo de desenvolvimento. Uma boa vantagem ao utilizarmos a técnica de avaliação de riscos como métrica de segurança é que ela faz uma boa análise da segurança ao endereçar possibilidades e danos. Outra boa propriedade é que ela causa defensores para combater com

como adversários realmente atacam sistemas.