6
Conclusões e Trabalhos Futuros

Até hoje, a ciência da computação não obteve sucesso em sua tentativa de definir métricas de segurança de software, por não conseguir fornecer métricas confiáveis e que sejam aceitas por todos para a avaliação da segurança de sistemas de informação. Talvez o problema esteja em estabelecer um objetivo inadequado. O objetivo baseado em risco poderia ser mais adequado.

Enquanto estamos buscando desenvolver um software capaz de resistir a ataques, tornando-o confiável sob o aspecto da segurança, ainda não temos hoje com os métodos atuais como provar que o software é seguro. Hoje só podemos provar que o software não é seguro. Provavelmente, será muito difícil podermos um dia provar que seja seguro. Aqui ocorre o mesmo problema que em teste: nunca saberemos se a última vulnerabilidade foi adequadamente considerada. Segundo Kaner e co-autores (Kaner et al., 1993), o *"planejamento de testes é governado pela necessidade de selecionar alguns poucos casos de teste de um gigantesco conjunto de possíveis casos. Independentemente de quão cuidadoso você seja, você deixará de incluir alguns casos relevantes. Independentemente da perfeição e esmero do seu trabalho, você nunca encontrará a última falta, e se encontrar, jamais o saberá."*

Nós não conseguiremos sem um maior avanço da tecnologia. Precisamos encontrar uma maneira de fazer o software menos frágil e que seja capaz de detectar vulnerabilidades e fechá-las, mesmo sob um ataque a segurança.

Precisamos também de um estudo mais aprofundado sobre composições de mecanismos de segurança. Estudo este que nos dê mais do que um aumento linear em resistência a ataques contra a segurança.

Não existe dúvida de que muito trabalho ainda precisa ser feito pela comunidade de pesquisa e a parte restante desta seção tenta destacar algumas

destas prioridades.

6.1 Trabalhos Futuros

Com relação ao tema segurança de software este trabalho sugere alguns trabalhos futuros nas seguintes linhas de pesquisa: linguagens de programação orientadas a segurança, técnicas de programação que facilitem a implementação de mecanismos de segurança, prova de corretude na construção do software, evolução do software, taxonomias para a área de segurança (bugs, classes de ameaças, vulnerabilidades, etc), composição de elementos de segurança. Esta dissertação também sugere os seguintes trabalhos futuros de dissertação relacionados ao tema segurança no desenvolvimento:

1. Como extrair políticas de segurança a partir dos requisitos?
2. Projeto de interface segura
3. Práticas de testes para segurança e privacidade
4. Como identificar usuários da Internet?
5. Como verificar propriedades de segurança de artefatos de software?
6. Arquitetura segura (otimizar, melhorar; fazer melhor ou tornar mais bem definida)
7. Como descobrir *bugs* de segurança em um projeto (design) de software?
8. Estudos empíricos de padrões de ataques
9. Recuperabilidade do software. Como voltar a execução de um programa, comando após comando, por causa de vulnerabilidades de segurança?
10. Análise de modo de falha aplicada a segurança
11. Como analizar a composição de especificações de sistemas para encontrar falhas de segurança?
12. Avaliação econômica de ataques e contramedidas.
13. Como métricas de segurança diferentes podem ser interpretadas e correlacionadas?
14. Metodologias para identificar métricas de interesse, como o “Goal Question Metric” (GQM) de Basili, deve ser considerada e possivelmente adaptada para segurança. Por exemplo, *“security patterns”* empregados durante a fase de design (projeto) podem conter informações sobre métricas adequadas a serem monitoradas.