MODELOS DE RESUMOS PARA TRABALHOS CORRELATOS

Desenvolvido e destinado aos meus orientandos e alunos de disciplinas afim.

Profª Ana Maria Martins Carvalho – 1º Sem/2024.

A seguir, como exemplo, apresento um texto com resumos de trabalhos correlatos (bibliografia correlata) reais. Este exemplo pode ser usado para escrever o capítulo de uma monografia (capítulo referente trabalhos correlatos), auxiliar a direcionar as citações no decorrer do texto científico, relembrar o conteúdo de um trabalho científico, entre outros.

EXEMPLO:

Neste capítulo é apresentado um levantamento da bibliografia correlata, utilizado para o desenvolvimento deste trabalho, analisando as contribuições e as limitações de cada trabalho. Essas informações auxiliaram a definir de forma literal a contribuição que este trabalho irá proporcionar ao meio acadêmico.

Qual a colaboração científica do autor em sua pesquisa? Qual o objetivo do autor nesse texto científico lido?
Metodologia?
Limitação?
Resultado e/ou contribuição?

No trabalho de Basso (2010) o autor faz uma abordagem da eficácia dos scanners de Vulnerabilidades em Aplicações Web. A proposta principal do trabalho é avaliar os scanners de vulnerabilidades permitindo que diretrizes sejam fornecidas para a utilização das ferramentas. O trabalho é baseando em técnicas de injeção, falhas e modelagem de árvores de ataque. As contribuições que esse trabalho traz são as técnicas para testar as aplicações contra vulnerabilidades de segurança abordando a eficácia dos *scanners* e a sua aplicação para um estudo de casos. Também verifica-se neste estudo, a relação das falhas por meio do software e as vulnerabilidades de segurança.

Duas limitações são encontradas e apresentadas pelo autor, a primeira é que os experimentos foram feitos manualmente, limitando a quantidade de aplicações e vulnerabilidades analisadas, não permitindo generalizar os resultados obtidos. E a segunda limitação é a falta de cobertura da aplicação sob um teste. Diz-se que o scanner detectou uma vulnerabilidade de injeção na aplicação e por sua vez essa aplicação já existia as falhas de injeção. Mas na versão original não existe uma falha de injeção e o scanner detecta uma falha não existente, isso é definido como falha de cobertura na aplicação.

Qual a colaboração científica do autor em sua pesquisa? Qual o objetivo do autor nesse texto científico lido?

Metodologia?

Limitação?

Resultado e/ou contribuição?

No trabalho de Luz (2011) é feito uma análise de vulnerabilidades em *Java Web Applications*. Em seu projeto de pesquisa, o objetivo é criar um *software* para analisar o código fonte de sistemas projetados para *Web* ou que foram migrados para a *Web*, em busca de falhas de segurança. Primeiramente, analisando sistemas desenvolvidos na linguagem Java. Esse analisador foi denominado de *Open Web Vulnerabilities Hunter* ou OWVH. O autor busca um estudo nas principais vulnerabilidades associadas ao documento OWASP *Top Ten* (ENDRES, 2013) e anomalias relacionadas a linguagem *Java* entre outros, com o objetivo de criar o *software* para que seja uma distribuição *Open Source*. Suas contribuições foram bastante eficientes, buscando várias ferramentas de *scanner* e teste de vulnerabilidades em aplicações *Web*, como o *Websecurify*, *Acunetix Web* e o *HP Scrawlr*, fazendo testes dessas ferramentas em aplicações locais e relacionando os resultados umas com as outras. Suas limitações foram os testes feitos em aplicações locais, sem trazer uma abordagem de exemplos em aplicações reais, ou seja, limitando o leque de possíveis ataques diferenciados.

Qual a colaboração científica do autor em sua pesquisa? Qual o objetivo do autor nesse texto científico lido?

Metodologia?

Limitação?

Resultado e/ou contribuição?

Oliveira (2012) aborda em seu trabalho testes de segurança em aplicações *Web* segundo a metodologia tratada na OWASP. Seu principal objetivo é avaliar a segurança em aplicações *Web* na categoria de *e-commerce* por se tratar de operações sensíveis executadas na *Web*. No seu trabalho três aplicações *e-commerce* são testadas: a primeira em um *e-commerce* real, a segunda em uma aplicação de código aberto e a terceira desenvolvida segundo as recomendações do Guia de testes da OWASP (MEUCCI, 2008). Nos resultados obtidos ele conclui que as três aplicações testadas possuem fraquezas similares. Por exemplo a enumeração de usuários acontece em todas elas podendo ser feitas de maneira automatizadas sem que haja alguma ação humana ou autorização como a utilização de uma CAPTCHA. Outra falha comum observada pelo autor é que nenhuma das três aplicações tem algum mecanismo para avaliar a força das senhas dos usuários. São fatores que levam um atacante a tentar um ataque através de força bruta.

A SEGUIR, DÊ ENFASE A IMPORTANTE COLABORAÇÃO DOS TRABALHOS CORRELATOS RESUMIDOS ACIMA E FAÇA UMA BREVE DESCRIÇÃO DA CONTRIBUIÇÃO DO SEU TRABALHO REFERENTE A ÁREA QUE ESTÁ SENDO PESQUISADA:

Embora todos os trabalhos acima relatam vários esforços para testar e tratar determinadas vulnerabilidades em aplicações *Web* nenhum deles contempla a utilização de *Content Management System* (CMS) ou sistema de gerenciamento de conteúdo. No caso desse trabalho abordarei o CMS mais utilizado no mercado, conhecido como *WordPress*. Por ser fácil e simples de utilizar, muitos desenvolvedores buscam a praticidade para um retorno mais rápido. Mas se esquecem da segurança da aplicação, fornecendo ao cliente um serviço em que, na maioria das vezes, não há nenhum tipo de segurança existente. O motivo de abordar esse CMS é foi adquirida pelas análises estatísticas. Segundo Pingdown (2013), 59,4% das aplicações desenvolvidas no mundo são de sites feitos em *wordpress*. 3,5 bilhões é o número de sites visitados por mês, feitos em *wordpress.* Dentre os 100 blogs famosos e mais acessados, 48% são feitos em *wordpress.* A vulnerabilidade principal que será abordada nesse trabalho é *Cross-Site Scrinpting* conhecida como *XSS*. O motivo principal de escolher essa vulnerabilidade, é pelo fato de que grande parte dos trabalhos e documentos acadêmicos, abordam a principal e mais conhecida vulnerabilidade, sendo ela, Injeção SQL, que de acordo com o documento do *Top Ten* da OWASP, é a primeira no *ranking*. *XSS* fica em terceiro na colocação do *ranking* de vulnerabilidades. Mas levando em consideração os seus impactos, e a maneira como está sendo difundida, merece uma atenção especial.

REFERÊNCIAS

BASSO, Tania. **Uma abordagem para avaliação da eficácia de scanners de vulnerabilidades em aplicações web.** Universidade estadual de Campinas – Faculdade de Engenharia Elétrica e de Computação. Campinas: 2010.

LUZ, H. J. F. **Análise de Vulnerabilidades em Java Web Applications.** Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha". Marília: 2011.

OLIVEIRA, T. S. T. **Testes de Segurança em Aplicações Web Segundo a Metodologia OWASP.** Departamento de Ciência da Computação. Universidade Federal de Lavras. Lavras: 2012.