Avaliação do Impacto de Ataques na Segurança de Serviços Web

Autor: Tiago Piccoloto Delpasso
Orientadora: Regina L. O. Moraes
Faculdade de Tecnologia - FT/UNICAMP

Palavras-chaves: Vulnerabilidade de Sistemas, Segurança de Dados, Ataques *Web*

Resumo

Aplicações ou serviços *Web* precisam ser seguros, uma vez que, atualmente, são utilizados por usuários civis ou empresas, que confiam nestas aplicações para desempenhar suas tarefas diárias e tomar decisões essenciais. Coletando dados referentes a requisições válidas e de ataque, campos de tabelas consultadas e o tempo decorrido entre a requisição e a resposta do sistema foi classificado o impacto dos ataques com base no *Common Vulnerability Scoring System* (CVSS).

Metodologia

O desenvolvimento do trabalho se deu em 8 passos:

- ✓ Escolhido uma aplicação para os testes, o TPC-APP[1];
- ✓ Configuração do ambiente de execução e a inicialização dos serviços ;
- ✓ Estudo das requisições esperadas por cada serviço;
- ✓ Estudo do domínio de seus atributos;
- ✓ Confirmação de que o serviço é confiável (teste com parâmetros esperados);
- ✓ Criação de uma lista de ataques para servir de entrada a um sistema de ataques automáticos ;
- ✓ Geração de um relatório no formato CSV;
- ✓ A categorização dos ataques seguindo o CVSS [2].

Testes de funcionamento do serviço

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tp
   <soapenv:Header/>
   <soapenv:Body>
      <tpcw:getPassword_VxO>
         <!--Optional:-->
         <tpcw:C_UNAME>?</tpcw:C_UNAME>
      </tpcw:getPassword_VxO>
   </soapenv:Body>
</soapenv:Envelope>
```

A

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header/>
   <env:Body>
      <ns2:getPassword_VxOResponse xmlns:ns2="http://tpcw.wsvdbench.dei.uc.pt/">
         <return>ogbaba</return>
      </ns2:getPassword_VxOResponse>
   </env:Body>
</env:Envelope>
```

B

Figura 1.A. XML de uma requisição a um serviço implementado pelo TPC-APP. **B.** Resposta do Serviço.

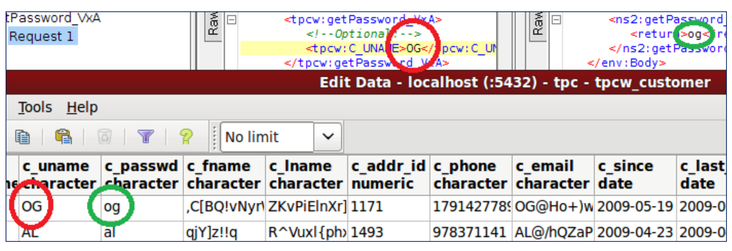

Figura 2: Comparação da resposta do serviço e identificação de seu domínio

Injetor de Ataques

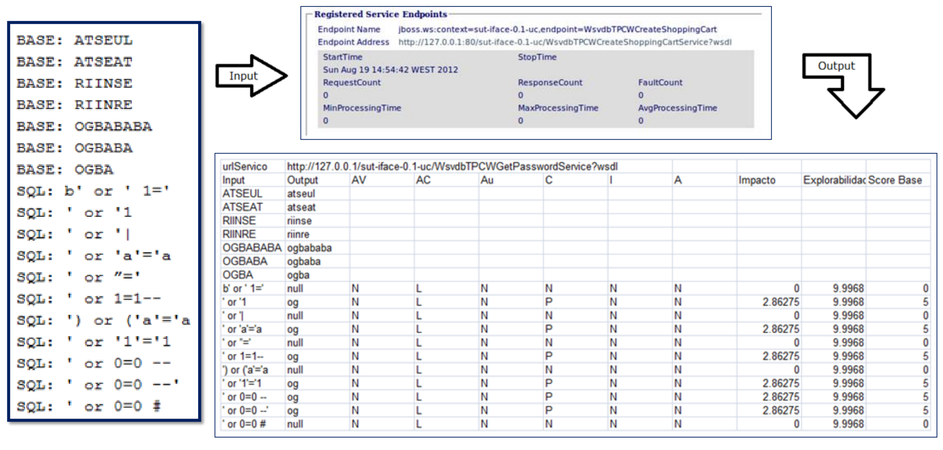

Figura 3: Esquema representativo do injetor de ataques.

CVSS e Resultados

Com os dados armazenados em um arquivo CSV, os resultados obtidos foram analisados e categorizados de acordo com as métricas do CVSS [2], apresentados na Figura 4).

BaseScore = round_to_1_decimal(((0.6*Impact)+(0.4*Exploitability)-1.5)*f(Impact))

Impact = 10.41*(1-(1-ConfImpact)*(1-IntegImpact)*(1-AvailImpact))

Exploitability = 20* AccessVector*AccessComplexity*Authentication

f(impact)= 0 if Impact=0, 1.176 otherwise

AccessVector = case AccessVector of
requires local access: 0.395
adjacent network accessible: 0.646
network accessible: 1.0

AccessComplexity = case AccessComplexity of
high: 0.35
medium: 0.61
low: 0.71

Authentication = case Authentication of
requires multiple instances of authentication: 0.45
requires single instance of authentication: 0.56
requires no authentication: 0.704

ConfImpact = case ConfidentialityImpact of
none: 0.0
partial: 0.275
complete: 0.660

IntegImpact = case IntegrityImpact of
none: 0.0
partial: 0.275
complete: 0.660

AvailImpact = case AvailabilityImpact of
none: 0.0
partial: 0.275
complete: 0.660

Figura 4: Métricas do CVSS.

Foram executados 108 ataques em todo o experimento. 54 desses ataques foram executados utilizando-se o serviço *getPassword* e outros 54 foram executados utilizando-se o serviço *getUsername*. A Figura 5 apresenta os resultados obtidos.

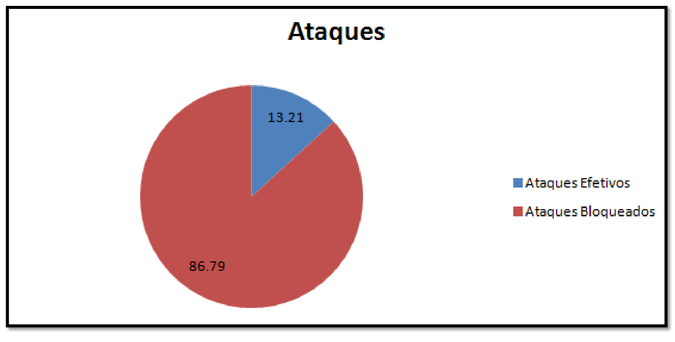

Figura 5: Resultado dos ataques

Discussão

Os resultados indicaram que em 13% das requisições, os ataques foram bem sucedidos e, mais do que isso, impactaram parcialmente a confidencialidade dos serviços vulneráveis. Esta pesquisa permitiu um melhor entendimento do nível do impacto de ataques do tipo *SQL Injection* contra uma classe de sistema que é utilizada diariamente por milhões de usuários. Permitiu também a construção de um ambiente experimental que será utilizado em futuras investigações para avaliação da segurança de serviços *Web*. Para o desenvolvimento do sistema, a linguagem Java foi utilizada. A administração da base de dados foi feita por meio do programa pgAdmin, de licença livre e o teste para averiguar o funcionamento dos serviços foi feito por meio do SOAP UI, também livre.

Reconhecimento

Gostaria de agradecer o aluno de doutorado Naaliel Vicente Mendes pela relevante ajuda durante o desenvolvimento dessa pesquisa.

Referências Bibliográficas

[1]TPC APP Benchmark Versao 1.3. Disponivel para download em http://www.tpc.org/tpc_app/default.asp (último acesso 23/08/2012).[2]Peter Mell, Karen Scarfone, and Sasha Romanosky, “CVSS, A Complete Guide to the Common Vulnerability Scoring System,” June, 2007