Instituto Superior de Engenharia de Lisboa

Licenciatura/Mestrado em Engenharia Informática e de Computadores

Segurança Informática

Segunda série de exercícios, Semestre de Inverno de 08/09

Data de entrega: 10 de Janeiro de 2012

---

1. Considere o protocolo de autorização OAuth 2.0 [1].

   1.1. Qual a diferença entre *dono de recursos* e *cliente*.

   1.2. Os clientes são caraterizados em função do seu perfil. Descreva os perfis *web application* e *user-agent-based application*.

   1.3. Qual a diferença entre *authorization token* e *access token*?

   1.4. De que forma o *redirection URI* pode ser usado para montar um ataque de *Cross Site Request Forgery* (CSRF)? O que está previsto para impedir este tipo de ataque?

2. Considere o modelo de certificados definido pela SDSI (Simple Distributed Security Infrastructure) e as seguintes entidades e nomes locais:

   a) K_{MAI} *Eleitor* $\rightarrow K_{MAI}$ *Freguesia Eleitor*

   b) K_{MAI} *Concelho* $\rightarrow K_{Loures}$

   c) K_{MAI} *Concelho* $\rightarrow K_{Lisboa}$

   d) K_{Lisboa} *Freguesia* $\rightarrow K_{Lapa}$

   e) K_{Lisboa} *Freguesia* $\rightarrow K_{Ajuda}$

   f) K_{Ajuda} *Eleitor* $\rightarrow K_{Carol}$

   g) K_{Lapa} *Eleitor* $\rightarrow K_{Alice}$

   h) K_{Lapa} *Eleitor* $\rightarrow K_{Bob}$

   K_{MAI} representa a chave pública do Ministério da Administração Interna.

   2.1. Qual o certificado que K_{MAI} tem de emitir para delegar nos concelhos a enumeração de freguesias (K_{MAI} *Freguesia*)?

   2.2. Qual o certificado que K_{MAI} pode emitir para substituir o certificado a) e o da alínea anterior?

   2.3. Considere a existência de uma aplicação *web* controlada por K_{MAI} a qual apresenta o número de eleitor a quem provar pertencer a K_{MAI} *Eleitor*. Apresente o processo de inferência que permite provar que o *principal* K_{Alice} tem acesso ao serviço.

3. Considere o artigo [2].

   3.1. Quais os alvos dos ataques de *buffer overflow*? Descreva sucintamente o ataque a ponteiros de funções.

   3.2. Descreva a forma como o alvo "old base pointer" pode ser usado para alterar a sequência de instruções executadas?

   3.3. Numa das propostas apresentadas, o valor da *guarda* ("canário") depende do valor do alvo a proteger. Apresente a razão para esta dependência e diga como ele é implementada.

   3.4. Descreva a abordagem da biblioteca `Libsafe` e qual a sua limitação.

   3.5. Para além de extensões a compiladores, os autores consideraram outras técnicas de protecção. Quais as limitação da técnica que impede a execução de código no *stack*?

4. O objectivo deste exercícios é configurar uma política para controlo de acessos a informação presente numa base de dados SQL. A base de dados que será usada no exercício guarda informação sobre alunos, professores, disciplinas e as notas que os alunos obtiveram nas disciplinas em que se inscreveram.

   4.1. Operações preparatórias:

   - Execute o *script* `DBEscola.sql` o qual cria a base de dados e respectivas tabelas, insere dados e cria os utilizadores correspondentes a alunos e professores.
   - Verifique que o dono da base de dados tem acesso a todas as tabelas e que os restantes utilizadores não têm qualquer acesso.

   4.2. Realize cada um dos seguintes passos e verifique se as permissões são aplicadas pelo SGDB:

   i. Atribua o direito para ler a tabela `Alunos` ao utilizador `Ana`.
   ii. Revogue o direito para ler a tabela `Alunos` ao utilizador `Ana`.
   iii. Crie o *database role* `Aluno` e adicione os utilizadores que são alunos. Atribua a este *role* direitos de ler a tabela `Alunos`.
   iv. Negue explicitamente o acesso ao utilizador `Ana`.
   v. Revogue esta permissão.

   4.3. Realize os comandos necessários para que os alunos (ou seja, todos os utilizadores excepto `Joao` e `Maria`) apenas consigam ver informação pessoal, ou seja, número, nome, as disciplinas a que se inscreveu, a respectiva nota e qual o professor.

   4.4. Crie uma vista com toda a informação sobre os professores, ou seja, nome do professor, nome da disciplina, e o número e nota dos alunos inscritos.

   - Defina o *role* Professor, adicionando os utilizadores `Joao` e `Maria`. Altere as permissões do *role* `Professor` para que os seus utilizadores possam ler a vista criada anteriormente.
   - Atribua a permissão de `UPDATE` sobre a coluna `Nota` ao *role* `Professor`. Verifique a sua correcta aplicação introduzindo notas usando utilizadores do *role* `Aluno` e `Professor`.

5. O objectivo deste exercício é explorar uma vulnerabilidade no procedimento armazenado presente no ficheiro `DBEscola_create_sp_notas.sql`. Adicione este procedimento à base de dados `Escola` e atribua permissões de execução ao *role* `Aluno`.

   5.1. Execute o procedimento com um utilizador do *role* `Aluno` e, usando *SQL Injection*, dê controlo sobre a base de dados a todos os alunos (`GRANT CONTROL TO Aluno`).

   5.2. Protega o procedimento contra este ataque usando, correctamente, o procedimento armazenado `sp_executesql`. Mais informações em *http://msdn.microsoft.com/en-us/library/ms188001.aspx*.

6. Realize um documento, não excedendo as 8 páginas de dimensão (excluindo referências), sobre um dos seguintes temas:

   - Ataques de *Cross Site Scripting* e *Cross Site Request forgery* em aplicações *web*.
   - Controlo de acesso mandatório em sistemas operativos.
   - Modelo de controlo de acessos do sistema de gestão de bases de dados *SQL Server 2008*.

Referências

[1] The OAuth 2.0 Authorization Protocol draft-ietf-oauth-v2-22, http://tools.ietf.org/html/draft-ietf-oauth-v2-22, visitado em 28/11/2011.

[2] John Wilander and Mariam Kamlar. 2003. A Comparasion of Publicly Available Tools for Dynamic Buffer Overflow Prevention. In *Proceedings of the 10th Network and Distributed System Security Symposium (NDSS)*

ISEL, 29 de Novembro de 2011.